UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

BRUNO ZATT

Energy-Efficient Algorithms and Architectures for Multiview Video Coding

Tese apresentada como requisito parcial para a obtenção do grau de Doutor em Microeletrônica

Prof. Dr. Sergio Bampi
Orientador

Porto Alegre, Outubro de 2012.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Zatt, Bruno

Energy-Efficient Algorithms and Architectures for Multiview Video Coding [manuscrito] / Bruno Zatt. – 2012.

236 f.:il.

Orientador: Sergio Bampi.

Tese (doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Microeletrônica . Porto Alegre, BR – RS, 2012.

1.Multiview Video Coding. 2.Energy-Efficiency. 3.Hardware Architectures. 4. Low-Complexity Algorithms. I. Bampi, Sergio. II. Energy-Efficient Algorithms and Architectures for Multiview Video Coding.

This thesis is dedicated to my beloved mother
Amantina Cardoso Ribeiro Zatt (in memoriam).

Dedico esta tese à minha amada mãe
Amantina Cardoso Ribeiro Zatt (in memoriam).

AGRADECIMENTO

"On this team we fight for that inch. On this team we tear ourselves and everyone else around us to pieces for that inch. We claw with our fingernails for that inch. Because we know when add up all those inches, that's gonna make the freaking difference between winning and losing! Between living and dying! I'll tell you this, in any fight it's the guy whose willing to die whose gonna win that inch... You've got to look at the guy next to you, look into his eyes. Now I think ya going to see a guy who will go that inch with you. Your gonna see a guy who will sacrifice himself for this team, because he knows when it comes down to it your gonna do the same for him. That's a team, gentlemen, and either, we heal, now, as a team, or we will die as individuals."
(film: Any Given Sunday)

Tem sido uma longa jornada, cheia de aprendizado e diversão . Mas ainda assim, uma longa jornada, um grande desafio, um esforço que exigiu “sangue suor e lágrimas” (Winston Churchill) de mim e daqueles ao meu redor. Acredito que a vida não se resume a alcançar objetivos. Seu maior valor está na caminhada que nos leva aos objetivos e nas pessoas que encontramos pelo caminho. É por isso que os obstáculos e decepções que enfrentamos tem tamanha importância É por isso que as pessoas que estão ao nosso lado nesses momentos devem ser lembradas. Portanto, dedico essa tese a essas pessoas.

Começo agradecendo àqueles que me deram mais do que uma formação técnica, àqueles que me deram formação como ser humano. Agradeço aos meus pais João Francisco Zatt e Amantina Cardoso Ribeiro Zatt, bem como aos meus avôs (Rozimbo Zatt e Getúlio Ribeiro) e avós (Maria Zatt e Romilda Ribeiro), pelo amor, cuidado, ensinamentos e apoio incondicional em minhas decisões. Também agradeço a todos aqueles que compuseram meu ambiente familiar (tios, tias, primos e primas das famílias Zatt e Ribeiro) e social (colegas e amigos dos velhos tempos de Santo Ângelo, do chão colorado no “ Garrão do Brasil”). Foram esses que trouxeram alegria, carinho, segurança e conforto ao longo da minha formação. Em especial ao meu irmão Gustavo Zatt, irmão no verdadeiro e mais profundo sentido da palavra, e a Marli “Tita” de Almeida, minha “segunda mãe”. Também aos amigos que me acompanharam durante muito tempo como irmãos, Eduardo Sbabo Flores, Rafael Kerber, José Mauro Zimmermann Júnior e Felipe Guimarães.

Muitos dizem que não fazemos grandes amigos depois de adultos. Me considero a prova do contrário. Durante a faculdade fiz grandes amigos como Fábio Ramos, Osvaldo Martinello, Leonardo Kunz, Giancarlo Franciscato, Jonas Bragagnolo e Rafael Nondillo. Sou fã de cada um deles. Sinto falta da companhia para beber algumas cervejas ou comer um assado na praia de Imbé.

Em minha jovem vida como pesquisador, começada em 2004, sempre tive o prazer de conviver com pessoas incríveis técnica e pessoalmente. Gostaria de agradecer ao Prof. Sergio Bampi que me deu a oportunidade de ingressar em seu grupo de pesquisa, sem nenhum motivo aparente, e que até hoje fornece grande suporte como orientador. Fui inserido em um grupo repleto de grandes exemplos como Arnaldo Azevedo, Luciano Agostini e Vagner Rosa. Por ter ficado ligado ao mesmo grupo na graduação, mestrado e doutorado (2004-2012) pude conviver com outras gerações de pessoas brilhantes como meus colegas de mestrado Claudio Diniz, Dieison Deprá, Guilherme Mauch, Marcelo Porto, Roger Porto, Debora Matos, Thaisa Leal e Leandro Max. Além dos mais jovens cujo crescimento pude acompanhar: Leandro Zanetti, Guilherme Corrêa, Daniel Palomino, Eduarda Monteiro, Bruno Vizzotto e Felipe Sampaio (em especial aos dois últimos que tiveram participação direta neste trabalho).

Acredito ter crescido bastante, do ponto de vista técnico e pessoal, ao sair do Brasil para aventuras em terras germânicas, mais exatamente em Karlsruhe, Alemanha. Lá aprendi outra forma de encarar a pesquisa enquanto conhecia novas culturas e partes do mundo que antes só conhecia através de filmes ou livros. Isso aconteceu através da oportunidade concedida pelo Prof. Jörg Henkel e através do financiamento da CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior). Devo agradecer de forma especial ao Muhammad Shafique que viabilizou minha ida e me impulsionou, com sua vocação para pesquisa, ao longo dos últimos 2 anos e meio. Ali pude conhecer outros colegas incríveis como Fazal Hameed, Semeen Rehman, Waheed Ahmed, Usman Khan e demais colegas do Chair for Embedded Systems.

Certamente a distância traz saudades de casa. Nessas horas é bom ter ao nosso redor pessoas que compartilham de nosso jeito de pensar e viver. Durante o período em Karlsruhe fizemos amigos aos quais agradeço pela companhia. Entre eles Monica Pereira, José Azambuja, Paola Tarouco, Família Marchesan, Bruno Vizzotto, Mateus Grellert, Paulo Roberto Aguiar, Jill Ewert e Andreas Schultheiss. Também agradeço àqueles que ajudaram a matar um pouco dessa saudade com suas tão esperadas visitas (meus pais, meu irmão, minha sogra Neiva Sallet, Eduardo Flores, Franciele Bruinsmann, Cláudio Diniz e Família Machado).

Talvez alguém possa estranhar que boa parte dos agradecimentos tenham sido escritos na primeira pessoa do plural. Pois bem, foi intencional. Não estive sozinho nesse período, tive sempre ao meu lado uma grande companheira, minha namorada Arielli Sallet Nacimento, a quem tanto amo. Nosso companheirismo e cumplicidade, que se intensificaram muito quando viemos morar juntos na Alemanha, foi o combustível que nos fez percorrer todo esse caminho. Nos momentos nebulosos, ela foi minha luz e meu apoio. Ari, espero ter sido um companheiro tão bom quanto tu foste para mim. Te amo.

Pretensiosamente, encerro dando minha contribuição aos que pretendem trilhar caminhos semelhantes ou semelhantemente desafiadores: – Vá em busca daquilo que quer, mas não esqueça que tudo tem um custo. Esse custo tipicamente é pago com “sangue, suor e lágrimas” (Winston Churchill). – Saiba abrir mão de coisas superficiais para focar naquilo que realmente importa. Mas não faça isso sempre pois equilíbrio é fundamental na vida. – Trabalhe sério, seja honesto e coloque todo seu esforço naquilo que fizer; os resultados são meras consequências. A propósito, não se deixe deprimir por resultados ruins. Frequentemente eles apenas ofuscam os bons resultados que vem no futuro. – Escute os mais experientes, mas não deixe de colocar seu toque pessoal naquilo que faz. – Trabalhe em grupo. Saber lidar com outras pessoas será, muito provavelmente, bastante útil no futuro. – Primeiro faça, depois busque o reconhecimento. Nunca o contrário. “Você não pode construir uma reputação baseado no que ainda vai fazer” (Henry Ford) – Faca aquilo que gosta, pois como diria um homem relativamente inteligente, “Todo mundo é um gênio. Mas, se você julgar um peixe por sua capacidade de subir em uma árvore, vai gastar toda a sua vida acreditando que ele é estúpido.” (Albert Einstein).

ACKNOWLEDGEMENTS

"On this team we fight for that inch. On this team we tear ourselves and everyone else around us to pieces for that inch. We claw with our fingernails for that inch. Because we know when add up all those inches, that's gonna make the freaking difference between winning and losing! Between living and dying! I'll tell you this, in any fight it's the guy whose willing to die whose gonna win that inch... You've got to look at the guy next to you, look into his eyes. Now I think ya going to see a guy who will go that inch with you. Your gonna see a guy who will sacrifice himself for this team, because he knows when it comes down to it your gonna do the same for him. That's a team, gentlemen, and either, we heal, now, as a team, or we will die as individuals."
(film: Any Given Sunday)

It has been a long journey, full of learning and fun. Still, it was a big journey, a big challenge, an effort that took “blood, toil and tears” (Winston Churchill) from me and from those close to me. I believe the life is not about reaching goals. Life is the way you walk to reach these goals and those people you meet in the way. That is why the challenges and the disappointments you face in this way are so important. That is why the people you find is this way must be acknowledge. Therefore, I dedicate this thesis to these people.

I start from those that gave me no technical formation but a formation as human being. I would like to thank my parents João Francisco Zatt and Amantina Cardoso Ribeiro Zatt along with my grandparents (Rozimbo Zatt, Maria Zatt, Getúlio Ribeiro, and Romilda Ribeiro) for their love, care, teachings, and unconditional support in all my decisions. Also, I thank all those that composed my familiar (Zatt and Ribeiro families) and social (colleagues and friends from the old times in Santo Ângelo) environments bringing happiness, affection, security, and comfort during my development. In special to my brother Gustavo Zatt, a true brother present in every single moment, and to Marli “Tita” de Almeida, a “second mother” for me. Also, to the friends that were good partners along many years: Eduardo Sbabo Flores, Rafael Kerber, José Mauro Zimmermann Júnior, and Felipe Guimarães.

Some people believe it is not possible to make real good friends after growning up. I consider myself a proof to the contrary. Along my Engineering studies I made great friends such as Fabio Ramos, Osvaldo Martinello, Leonardo Kunz, Giancarlo Franciscato, Jonas Bragagnolo, and Rafael Nondillo. I am a big fan of each one of them. Frequently I miss their company to drink some beers and eat our traditional barbecue in Imbé Beach.

In my young life as researcher, started in 2004, I always had the pleasure to work side-by-side with incredible people at both technical and personal perspectives. I would like to thank Prof. Sergio Bampi for the opportunity to join his research group and for his support as advisor along these years. I was inserted in a team full of great examples for me such as Arnaldo Azevedo, Luciano Agostini, and Vagner Rosa. As I remain linked to this group after bachelor, master and Ph.D. studies, I was able to share this experience with other generations of brilliant people. I would like to thank my master studies colleagues Cláudio Diniz, Dieison Deprá,

Guilherme Mauch, Marcelo Porto, Roger Porto, Debora Matos, Thaisa Leal, and Leandro Max and to other colleagues whose growth I could observe: Leandro Zanetti, Guilherme Corrêa, Daniel Palomino, Eduarda Monteiro, Bruno Vizzotto e Felipe Sampaio (special thanks to the last two that directly contributed to this work).

I believe that moving to Karlsruhe, Germany, gave me the opportunity to grow technically and personally. I learned a new way to face research while knowing new cultures and places. This was possible due to the opportunity given to me by Prof. Jörg Henkel and the funding provided by CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*). Special thanks to Muhammad Shafique that made my internship possible and stimulated me with his vocation for research along the last two and a half years. Also, during my time in the Chair for Embedded Systems I came to know amazing colleagues such as Fazal Hameed, Semeen Rehman, Waheed Ahmed, and Usman Khan.

For sure the distance makes us miss home. In this moments it is great to have around us people that think and leave like us. In Karlsruhe we made good friends. Thanks for the company Monica Pereira, José Azambuja, Paola Tarouco, Family Marchesan, Bruno Vizzotto, Mateus Grellert, Paulo Roberto Aguiar, Jill Ewert, and Andreas Schultheiss. Additionally, I would like to thank those that brought I little of home with their so expected visits (my parents, my brother, my mother-in-law Neiva Sallet, Eduardo Flores, Franciele Bruinsmann, Cláudio Diniz, and Family Machado).

Perhaps someone may think it was a mistake writing some parts of this acknowledgment in plural. Well, it was intentional. I have never been alone in this period, I always had at my side a great company, my girlfriend, my love Arielli Sallet Nacimento. Our complicity and affinity became stronger and stronger during this time living together in Germany. She was my light and support in dark times. Ari, I hope to be a companion just as good as you are for me. Love you.

Pretentiously, I close giving my contribution to those planning to follow similar ways: – Fight for what you want, but remember, it has a cost. This cost is typically in “blood, toil and tears” (Winston Churchill). – Give up superficial things to focus on the real important ones. However, do not do it always. Balance is essential. – Work hard, be honest and put your best effort in everything you do regardless the results, they are mere consequences. By the way, do not get depressed because of bad results. Frequently they hide the good ones. – Listen to the most experienced but put your own touch in everything you do. – Work in group. Dealing with people will probably be very helpful in the future. – First do your work and after look for the recognition. Never the opposite. “You can't build a reputation on what you are going to do” (Henry Ford). – Work in something you like because, according to a “relatively smart man”, “Everybody is a genius. But if you judge a fish by its ability to climb a tree, it will live its whole life believing that it is stupid.” (Albert Einstein).

SUMMARY

ABBREVIATIONS

3D	*Three-Dimensional*
3DV	*Three-Dimensional Video (future video standard)*
3DTV	*Three-Dimensional Television*
ASIP	*Application-Specific Instruction-Set Processor*
AVC	*Advanced Video Coding*
BR	*Bitrate*
BU	*Basic Unit*
CABAC	*Context-Based Adaptive Binary Arithmetic Coding*
CAVLC	*Context-Based Adaptive Variable Length Coding*
CIF	*Common Intermediate Format*
CODEC	Coder/Decoder
DC	*Direct Current*
DCT	*Discrete Cosine Transform*
DDR	*Double Data Rate*
DE	*Disparity Estimation*
DF	*Deblocking Filter*
DMV	*Differential Motion Vector*
DPB	*Decoded Picture Buffer*
DPM	*Dynamic Power Management*
DSP	*Digital Signal Processing*
DV	*Disparity Vector*
DVS	*Dynamic Voltage Scaling*
EPTZ	*Early Prediction Termination Zone*
FIR	*Finite Impulse Response*
FPGA	*Field Programmable Gate Array*
FPS	*Frames Per Second*
FRExt	*Fidelity Range Extensions*
FSM	*Finite State Machine*
FTV	*Free-Viewpoint Television*

GB	*Giga Bytes*
GDV	*Global Disparity Vector*
GGOP	Group of *Group of Pictures*
GIPS	*Giga Instructions per Second*
GOP	*Group of Pictures*
HBP	*Hierarchical Bi-Prediction*
HD1080p	*High Definition 1920x1080 Progressive*
HDTV	*High Definition Digital Television*
HEVC	*High Efficiency Video Coding*
HRC	*Hierarchical Rate Control*
HVS	*Human Visual System*
IC	*Integrated Circuit*
IEC	*International Electrotechnical Commission*
IEEE	*Institute of Electric and Electronics Engineers*
IQ	*Inverse Quantization*
ISO	*International Organization for Standardization*
ITU-T	*International Telecommunication Union – Telecommunication*
IT	*Inverse Transform*
JM	*Joint Model for H.264*
JMVC	*Joint Model for MVC*
JVT	*Joint Video Team*
KIT	*Karlsruhe Institute of Technology*
MB	Macroblock
MD	*Mode Decision*
MDP	*Markov Decision Process*
MC	*Motion Compensation*
ME	*Motion Estimation*
MPC	*Model Predictive Controller*
MPEG	*Moving Picture Experts Group*
MSE	*Mean of Square Errors*
MV	*Motion Vector*
MVC	*Multiview Video Coding*
MVP	*Motion Vector Predictor*
PC	*Personal Computer*
PDF	*Probability Density Function*

PID	*Proportional-Integral-Differential Controller*
PMV	*Predictive Motion Vector*
PSM	*Power-State Machine*
PSNR	*Perceptible Signal-to-Noise Ratio*
POC	*Picture Order Counter*
Q	*Quantization*
QCC	*Quality-Complexity Class*
QCIF	*Quarter Common Intermediate Format*
QHD	*Quad HDTV*
QP	*Quantization Parameter*
QS	*Quality State*
RC	*Rate Control*
RD	*Rate-Distortion*
RDO	*Rate-Distortion Optimization*
RDO-MD	*Rate-Distortion Optimized Mode Decision*
RGB	*Red, Green, Blue*
RL	*Reinforcement Learning*
RoI	*Region of Interest*
RTL	*Register-Transfer Level*
SAD	*Sum of Absolute Distances*
SATD	*Sum of Absolute Transformed Distances*
SI	*Switching I*
SIMD	*Single Instruction Multiple Data*
SoC	*System on Chip*
SP	*Switching P*
SRAM	*Static Random Access Memory*
SSE	*Sum of Square Errors*
T	*Transform*
UFRGS	*Universidade Federal do Rio Grande do Sul*
UVLC	*Universal Variable Length Code*
VCEG	*Video Coding Experts Group*
VGA	*Video Graphics Array*
VHDL	*VHSIC Hardware Description Language*
VHSIC	*Very High Speed Integrated Circuit*
VLIW	*Very Large Instruction Word*

VP	*Viewpoint*
YUV	*Luminance, Chrominance Component 1, Chrominance Component 2*

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The robust popularization of 3D videos noticed along the last decade, allied to the omnipresence of smart mobile devices handling multimedia-capable features, has led to intense development and research focusing on efficient 3D-video encoding techniques, display technologies, and 3D-video capable mobile devices. In this scenario, the Multiview Video Coding (MVC) standard is key enabler of the current 3D-video systems by leading to meaningful data reduction through advanced encoding techniques. However, real-time MVC encoding for high definition videos demands high processing performance and, consequently, high energy consumption. These requirements are attended neither by the performance budget nor by the energy envelope available in the state-of-the-art mobile devices. As a result, the realization of MVC targeting mobile systems has been posing serious challenges to industry and academia.

The main goal of this thesis is to propose and demonstrate energy-efficient MVC solutions to enable high-definition 3D-video encoding on mobile battery-powered embedded systems. To expedite high performance under severe energy constraints, this thesis proposes jointly considering energy-efficient optimizations at algorithmic and architectural levels. On the one hand, extensive application knowledge and data analysis was employed to reduce and control the MVC complexity and energy consumption at algorithmic level. On the other hand, hardware architectures specifically designed targeting the proposed algorithms were implemented applying low-power design techniques, dynamic voltage scaling, and application-aware dynamic power management.

The algorithmic contribution lies in the MVC energy reduction by shorten the computational complexity of the energy-hungriest encoder blocks, the Mode Decision and the Motion and Disparity Estimation. The proposed energy-efficient algorithms take advantage of the video properties along with the strong correlation available within the 3D-Neighborhood (spatial, temporal and disparity) space in order to efficiently reduce energy consumption. Our Multi-Level Fast Mode Decision defines two complexity reduction operation modes able to provide, on average, 63% and 71% of complexity reduction, respectively. Additionally, the proposed Fast ME/DE algorithm reduces the complexity in about 83%, for the average case. Considering the run-time variations posed by changing coding parameters and video content, an Energy-Aware Complexity Adaptation algorithm is proposed to handle the energy versus coding efficiency tradeoff while providing graceful quality degradation under severe battery draining scenarios by employing asymmetric video coding. Finally, to cope with eventual video quality losses posed by the energy-efficient algorithms, we define a video quality management technique based on our Hierarchical Rate Control. The Hierarchical Rate Control implements a frame-level rate control based on a Model Predictive Controller able to increase in 0.8dB (Bjøntegaard) the overall video quality. The video quality is increased in 1.9dB (Bjøntegaard) with the integration of the basic unit-level rate control designed using Markov Decision Process and Reinforcement Learning.

Even though the energy-efficient algorithms drive to meaningful energy reduction, hardware acceleration is mandatory to reach the energy-efficiency demanded by the MVC. Aware of this requirement, this thesis brings architectural solutions for the Motion and Disparity Estimation unit focusing on energy reduction while attending real-time throughput requirements. To achieve the desired results, as shown along this volume, there is a need to reduce the energy related to the ME/DE computation and related to the intense memory communication. Therefore, the ME/DE architectures incorporate the Fast ME/DE algorithm in order to reduce the computational complexity while the memory hierarchy was carefully designed to find the optimal energy tradeoff between external memory accesses and on-chip video memory size. Statistical analysis where used to define the size and organization of the on-chip cache memory while avoiding increased memory misses and the consequent data retransmission. A prefetching technique based on search window prediction also supports the reduction of external memory access. Moreover, a memory power gating technique based on dynamic search window formation and an application aware power management were proposed to reduce the static energy consumption related to on-chip video memory. To implement these techniques a SRAM memory featuring multiple power states was used. The architectural contribution contained in this thesis extends the state-of-the-art by achieving real-time ME/DE processing for 4-views HD1080p running at 300MHz and consuming 57mW.

1 INTRODUCTION

The consumers' thirst for new and more immersive multimedia technologies allied to the industry interest to boost the entertainment market have driven the fast popularization of 3D video content generation, 3D-capable devices, and 3D applications. Although the first 3D video device was developed in 1833 and the first 3D film demonstration dates from 1915 (ZONE, 2007), this format only became worldwide known in the 1980s through IMAX (IMAX, 2012) technology. The real 3D video hype, however, was noticed in the late 2000s through the massive popularization and availability of 3D movies followed by the 3D-capable televisions dedicated to home cinema. For a better perspective of this popularization, more than 10% of the televisions sold in USA in 2011 were 3D capable (RESEARCH AND MARKETS, 2010). The latest field to be affected by the 3D video popularization is exactly the field responsible for the biggest IC (integrated circuits) industry growth after the popularization of personal computers: the mobile embedded systems. Smartphones, tablets, personal camcorders, and other mobile devices shipments already surpassed PC shipments (KAY, 2011) (IC INSIGHTS, 2012). For instance, more than 650 million smartphones are expected to be shipped in 2013 compared to 430 million PCs (GASSÉE, 2010) in the same year. Jointly, the popularization of 3D videos and mobile devices is leading to a scenario where a large amount of such 3D-capable smart devices is reaching the users every day, resulting in a large amount of 3D video content being generated, encoded, stored, transmitted, and displayed. According to CISCO (CISCO, 2012), video content already represents 51% of the current Internet traffic and is envisaged to touch the 90% mark due 2014 (SOCIAL TIMES, 2011). It is also important to consider that the 0.6 Exabytes/month mobile traffic in 2011 is expected to reach 10.8 Exabytes/month in 2016 (CISCO, 2012).

To cover the gap between 3D video content generation and network and storage capabilities there is a need to efficiently encode 3D videos and reduce the amount of data required for their representation. The Multiview Video Coding (MVC), an extension to the H.264/AVC, is the state-of-the-art on 3D video coding. Based on the multiple views paradigm, as the majority of current 3D video technology, the MVC reduces the 3D videos representation in 20%-50% compared to H.264/AVC simulcast. The cost of this efficiency improve comes from an increased coding complexity and increased energy consumption, mainly at the encoder side. The energy consumption incurs form multiple processing units working in parallel to attend throughput constraints (processors, DSPs, GPUs, ASICs) and intense memory access. In a scenario dominated by mobile devices, the increase in energy consumption goes against the battery restrictions posed by these mobile embedded systems. This conflict of interests between coding efficiency and energy constraints brings the main challenge related to 3D-video realization on embedded systems: jointly *design algorithmic and architectural* *energy-efficient solutions to enable real-time high-definition 3D video coding, while maintaining high video quality under severe energy constraints*. The main goal of this thesis is to address this challenge by presenting novel algorithms and hardware architectures designed to show the feasibility of 3D video encoding on embedded battery-powered devices.

In the next sections, after this introduction, an overview of 3D video applications that make the 3D video field so promising is presented. After that, a brief introduction on the trends for 3D video coding and multimedia embedded systems is presented, followed by the related issues and research challenges. This section is finalized by a summary containing the contributions of this work.

1.1 3D Video Applications

The adoption of 3D videos is directly associated to the existence of new applications requiring the deepness sensation in order to improve the users' immersion experience. From here onwards an overview of the main 3D video applications is presented. These applications share the same concept of capturing multiple views in the same 3D scene. To give the depth illusion, distinct views are displayed to each eye with displays that employ technologies based on parallax barriers, lenticular sheets, color polarization, directional polarization, or time interleaving (DODGSON, 2005), more details on this phenomenon are provided in Chapter 2.

- Three-Dimensional Video Personal Recording: Popularized by the 3D-capable mobile devices and the 3D video sharing services (YOUTUBE 3D, 2011) (VIMEO, 2012) the 3D video personal recording is the most massive 3D video service in terms of video content availability. With a 3D video recorder device the users are free to create and publish their own video content.

- Three-Dimensional Television (3DTV): 3DTV is an extension of the traditional 2D with the depth perception (SMOLIC, MUELLER, *et al.*, 2007). In this kind of application two or more views are decoded and displayed simultaneously where each viewer sees two views, one for the right eye, and other for the left eye. The simplest 3D displays, which are the stereoscopic displays that show two simultaneous views requiring the use of special glasses (polarized or active shutter glasses) to provide 3D sensation. The evolution of stereoscopic displays is the auto-stereoscopic display, which eliminates the need for glasses. In this case, parallax barriers and lenticular sheets are the most common solutions. Multiview displays are able to display higher number of views at the same time increasing the observer freedom by supporting head parallax, i.e. the viewpoint changes when the observer changes its position.

- Free-Viewpoint Television (FTV): In this application, the user is able to select the desired viewpoint in a 3D scene (POURAZAD, NASIOPOULOS e WARD, 2009). It provides realism and interactivity to the user, i.e., the focus of attention can be controlled. The display technology used may vary from 2D televisions to multiview displays.

- Three-Dimensional Telepresence: Allows the user to communicate and interact to interlocutors as if they were in the same location. Telepresence has been widely used for video teleconferencing, mainly in corporative environments, and for the implementation of the so called virtual offices. The evolution towards 3D

(BLANCHE, BABLUMIAN, *et al.*, 2010) represents a meaningful step in order to improve the perception and interaction level between the conference attendees.

- Three-Dimensional Telemedicine: Telemedicine (WELCH, SONNENWALD, *et al.*, 2005) was defined to surpass physical limitations and make it possible for a doctor to attend patients or perform surgeries while in a distinct location by using telecommunications methods. The 3D video capability brings the telemedicine to a whole new level where the specialist can precisely perceive the 3D space and proceed accurately through robotic actuators. This technology enables a better health care quality in remote places that do not count on qualified specialists.

- Three-Dimensional Surveillance: Traditional video surveillance systems rely in 2D videos and pose difficulties to authorities if precise depth information is required. Employing 3D-videos for surveillance (KRÜGERA, NICKOLAYB, *et al.*, 2005) provides a much richer information once it is possible to accurately extract depth and angulations data for all objects in the 3D scene. Therefore, a better description on the interaction between objects, such as possible criminals and victims, is obtained.

Among these applications, some are not designed for mobile use (e.g., 3D Surveillance and 3D Telemedicine) or require only decoding at the mobile device (e.g., 3DTV, FTV). For other applications, however, the capability to encode 3D videos is mandatory. For instance, 3D video personal recording requires real-time and energy-efficient 3D video encoding. 3D Telepresence, when running on embedded devices, demands real-time, energy-efficient and low-delay 3D video encoding. Aware of the challenges posed by the presented set of applications, this work focuses on the MVC video encoder.

1.2 Requirements and Trends of 3D Multimedia

Although the processing power of computational systems, mainly for embedded systems, has increased meaningfully (as detailed in Section 1.3), the multimedia applications performance and energy requirements are increasing in a significantly higher pace due to increased video resolutions, frame rates, sampling accuracy, and number of views in case of 3D videos. In other words, the amount of data to be processed in a video sequence has been increasing in multiple axes simultaneously.

Figure 1.1 relates the number of macroblocks (MB – 16x16 image block used as basic coding unit in MVC - for details refer to Chapter 2) to be processed per second considering the different video resolutions, frame rates and number of views. Previous coding standards, for instance MPEG-2, were designed and typically used in videos with low-medium resolutions and low-medium frame-rates such as CIF (352x288), VGA (640x480) and SDTV (768x576) at 15-30 fps (frames per second) (note that these numbers refer to the typical use and main target operation profiles, the standards define a very high operation range). The H.264 additionally targets high resolutions and high frame-rates such as 720p (1240x720) and HD1080p at 30-60fps. The next generation of coding standards, represented by H.265/HEVC (High Efficiency Video Coding), will also target on high and ultra-high resolutions and frame-rates including QHD (3840x2160) and UHDTV (7680x4320) videos at 60-120 fps (MCCANN, MATTEI, *et al.*, 2012)(LING, 2010). To quantify this growth, the relation between the corner cases shown in Figure 1.1a, CIF@15fps and QHD@60fps, is equivalent to a 327x factor. Also, targeting improved quality, the samples bit-depth is increasing from 8 bits up to

14-bit samples, requiring wider data operators. At the complexity and energy consumption perspective, the scenario is even worse once there is a non-linear relation with the data amount. The increase in resolution, for instance, leads to higher processing effort per MB, higher memory traffic and larger on-chip memory related to the Motion Estimation (ME, see Chapter 2), resulting in energy consumption increase. Moreover, the video coding standards evolution severely contributes to the increase of complexity and energy requirements. For example, the H.264 encoder is approximately 10x more complex than the MPEG-4 (OSTERMANN, BORMANS, *et al.*, 2004) encoder, while the HEVC is expected to bring additional 2-10x (DÍAZ-HONRUBIA, MARTÍNEZ e CUENCA, 2012) complexity increase factor in relation to H.264.

Considering 3D videos, the scaling scenario becomes more dramatic, as shown in Figure 1.1b. Besides the resolution and frame-rate increase, it is necessary to deal with the linear data growth in relation to the number of views. As MVC includes new coding tools the complexity and energy consumption increase in a non-linear (above-linear) fashion, as quantified in Section 3.1. The impacts of the fast 3D multimedia requirements scaling on embedded systems are discussed in the next section.

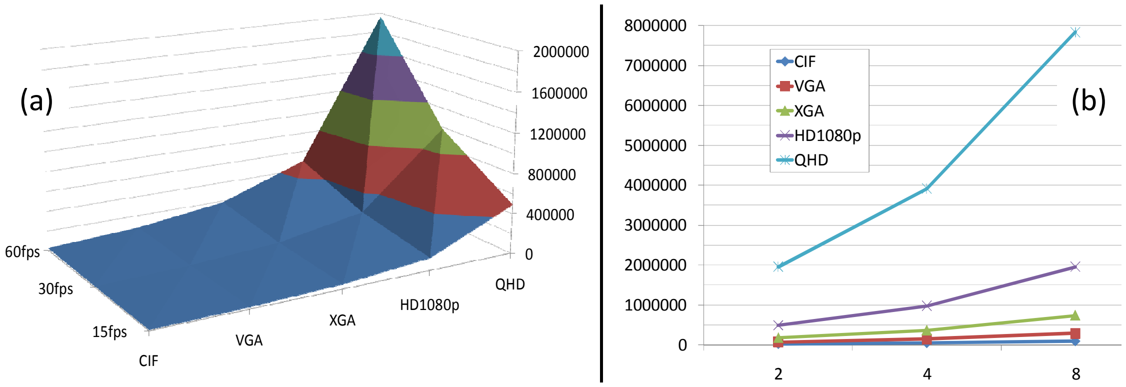


Figure 1.1: Video scaling trend

1.3 Overview on Multimedia Embedded Systems

The fast evolution of multimedia embedded systems has been driven by the so called smart devices (smartphones, tablets, and other mobile devices capable of data, audio, and video communication) popularization. Meaningful progress has been done by the major players in the field, (ARM LTD., 2012)(NVIDIA, 2012)(QUALCOMM INC., 2011)(TEXAS INSTRUMENTS INC., 2012) (SAMSUNG ELECTRONICS CO. LTDA., 2012), in terms of performance boost and energy efficiency. The progress, however, is not enough to fill the gap between multimedia application requirements and technology evolution. The ARM SoCs (System-on-Chip), whose processors equip about 90% of the current embedded devices (SOFTPEDIA, 2010), predicts a performance increase in the order of 10x when comparing the state-of-the-art in 2009 to the predicted one for 2016, as shown in Figure 1.2a. Energy restrictions related to slow battery evolution is the major factor limiting the performance of embedded systems. According to Panasonic (KUME, 2010), the capacity of Li-Ion batteries has been increasing, on average, 11% annually since 1994, as shown in Figure 1.2b.

The high performance and energy efficiency required by the current 3D-video applications are not met by generic embedded solutions such as embedded processors, GPUs and DSPs. There is a need to implement application-specific hardware accelerators to deliver the required throughput while minimizing energy consumption at the cost of a flexibility drawback. The latest high-end embedded SoCs already implement this approach for multimedia processing, e.g. H.264 video encoding and decoding, as detailed in Section 2.5. Some examples are Qualcomm Snapdragon S4 (QUALCOMM INC., 2011), Nvidia Tegra 3 (NVIDIA CORP., 2012), Samsung Exynos 4 (SAMSUNG ELECTRONICS CO. LTDA., 2012) and Texas Instruments OMAP 5 (TEXAS INSTRUMENTS INC., 2012). The hardware support, however, needs to be extended in order to efficiently handle 3D videos.

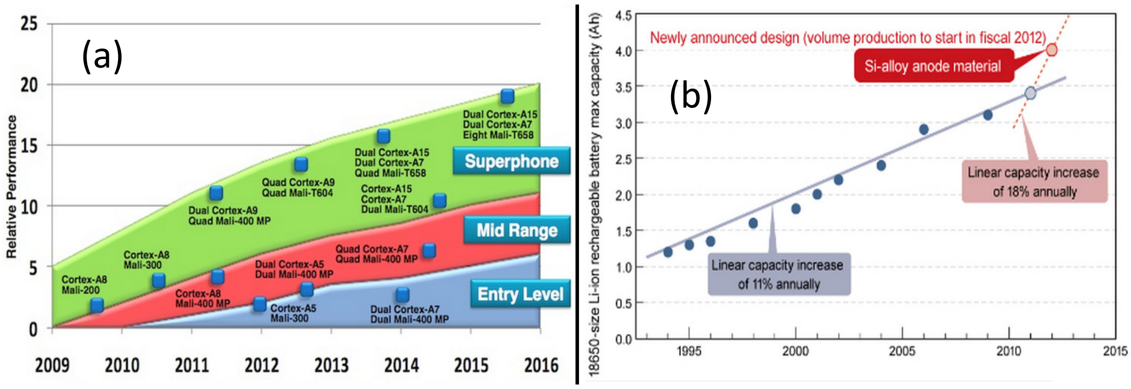


Figure 1.2: (a) Mobile systems performance trend (SHIMPI, 2011) and (b) Li-Ion battery capacity growth (KUME, 2010)

1.4 Issues and Challenges

The demand for mobile 3D multimedia processing allied to high performance demands and severe embedded devices energy constraints pose serious challenges to the researchers and developers actuating in the embedded multimedia systems field. In this scenario, employing hardware accelerators optimized for specific Multiview Video Coding applications is mandatory. Given the gap between 3D multimedia processing and the embedded processing reality, there is a need to further reduce the complexity and energy consumption at algorithmic and architectural levels. Such optimizations are only possible by employing deep application knowledge to perform a coupled and integrated optimization of the algorithms employed and the underlying hardware architecture.

In addition to the varying coding settings and battery state, multimedia applications are susceptible to input content variations that significantly change the system behavior and requirements. For instance, videos with higher motion intensity require more processing and memory accesses resulting in more processing units and larger on-chip memory finally leading to increased energy consumption. Such variations are only detected at run-time. Therefore, energy-efficient MVC encoding systems require algorithmic and hardware run-time adaptivity that employ application and video content characteristics knowledge. The adaptation schemes must be able to handle the energy-efficiency vs. video quality tradeoff in order to find the optimal operation point for each given system state and video input.

Energy reduction algorithms and energy-oriented optimizations might lead to rate-distortion (RD) performance losses, i.e., video quality reduction for the same bitrate. To avoid or minimize this drawback, there are mechanisms able to control the losses through the optimization of the bit distribution among different views, frames and image regions.

The in-depth study of the issues and challenges related to MVC encoding are presented in Chapter 3. In the following section, the contribution is summarized.

1.5 Thesis Contribution

The goal of this thesis is to understand the run-time behavior of the MVC encoder at the energy consumption perspective and propose algorithms and hardware architectures able to jointly attend the performance constraints and respect the energy envelope restrictions for state-of-the-art embedded devices. In this section, a summary of the contributions of this thesis is presented, highlighting the main innovations proposed. A deeper description of these contributions is found in Chapter 3, while the technical details are presented in Chapter 4 and Chapter 5, and results in Chapter 6.

1.5.1 3D-Neighborhood Correlation Analysis

The novel energy-efficient algorithms and hardware architectures proposed in this work are designed upon a strong MVC application knowledge including all MVC encoder algorithms and their run-time response to distinct input data. Along this work, the application knowledge, for many cases, is studied in terms of the correlation within the 3D-Neighborhood. The 3D-Neighborhood concept is a space domain defined in this thesis that contains the MBs belonging to the neighboring regions in the spatial, temporal, and disparity axes. Due to the redundancies existing within this neighborhood (see discussion in Section 2.2), the 3D-Neighborhood provides valuable information to predict video encoding side information, algorithms behavior, memory access pattern, etc. Therefore, the offline and online 3D-Neighborhood data are used to define and control the energy-efficient algorithms, hardware design, memory architecture and sizing, and adaptation schemes.

1.5.2 Energy-efficient MVC Algorithms

The energy-efficient algorithms for MVC are concentrated in three MVC encoding blocks: mode decision, motion and disparity estimation (ME/DE) and rate control. Mode decision (MD) and ME/DE units are responsible for the dominant energy consumption in the MVC encoder, as discussed along Chapter 3. The proposed fast MD and ME/DE target energy reduction through complexity reduction. These algorithms interact with the novel energy-aware complexity adaptation algorithm that controls the energy consumption by changing the coding efficiency considering battery state. The drawback posed by the energy-efficient algorithms comes in terms of quality drop under certain coding conditions. To minimize this negative impact a hierarchical rate control solution to optimize the bit utilization while maximizing and smoothen video quality in spatial, temporal, and disparity domains, is proposed.

- *Early SKIP Prediction*: Exploits the high occurrence of SKIP coded macroblocks and the image properties to correlate the MBs within 3D-Neighborhood. Quantization Parameter (QP)-based thresholding is employed to react to QP changing scenarios.

- *Multi-Level Fast Mode Decision*: Incorporates the Early SKIP prediction to a sophisticated mode decision scheme composed of six decision steps and bad prediction protection. This fast MD employs multiple MD aggressiveness strengths (to control energy vs. quality losses), 3D-Neighborhood data, coding modes ranking, video properties-based prediction, and Rate-Distortion cost (RDCost) prediction.

- *Energy-Aware Complexity Adaptation*: Defines four quality states employing distinct MD that are interchanged at run-time according to the actual battery state. The complexity adaptation algorithm employs asymmetric view coding to maximize the video quality in face of battery discharging and provides graceful quality degradation along the time.

- *Fast Motion and Disparity Estimation*: The proposed Fast ME/DE widely exploits the motion and disparity vectors correlation within the 3D-Neighborhood in order to avoid the search for non-key frames in the MVC prediction structure. According to the confidence in the neighboring MBs, the algorithm selects the Fast or Ultra-Fast prediction mode.

- *Hierarchical Rate Control (HRC)*: This innovative solution for the MVC rate controller employs two actuation levels, frame-level and basic unit-level rate control, with coupled feedback loop. The frame-level RC uses the Model Predictive Controller (MPC) to estimate the bitrate for future frames and decide the best QP. Markov Decision Process (MDP) with Reinforcement Learning (RL) and Regions of Interest (RoI) weighting is employed at BU-level to further optimize the QP selection within the frames.

1.5.3 Energy-efficient Hardware Architectures

The energy-efficient hardware architectures target the motion and disparity processing, which represents the most complex and energy-intense coding block of the MVC encoder. Three ME/DE architectures are proposed, all of them aiming to reduce the energy consumption for 4-views real-time HD1080p encoding through implementing, in hardware, the fast ME/DE algorithms proposed in this thesis. By doing so the on-chip memory size is reduced, the external memory accesses are reduced, and an efficient dynamic power management to the processing path and memory architecture is employed. The architectural innovations are introduced in the following and detailed in Chapter 5.

- *Motion and Disparity Estimation Architectural Template:* Defines the main building blocks used to design all hardware architectures and the interaction between them. It is designed to provide support to multiple search algorithms, throughputs and memory hierarchy.

- *Motion and Disparity Estimation Hardware Architectures*: Along this thesis are proposed three architectural solutions for the ME/DE block in the MVC encoder. These architectures feature distinct techniques to improve the performance and reduce the overall energy consumption. Additionally, by providing multiple solutions, this work enables the selection of the architecture that better adapts to a specific MVC encoding system.

- *Multi-Bank On-Chip Video Memory:* This proposal enables a reduced on-chip video memory and sector-level power gating in order to reduce the energy consumption through leakage current lowering. The on-chip memory works in a cache fashion and employs multiple banks for high throughput. Distinct Dynamic

Power Management (DPM) techniques are proposed based on the memory prediction using the 3D-Neighborhood information.

- *Memory Design Methodology:* A study of the memory requirements under different coding scenarios and video contents is presented to provide the basis for defining the optimal memory size and organization. Based on this study, an offline statistical analysis is used to define the memory hierarchy considering on-chip memory size and number of external memory access.

- *Dynamic Search Window Formation-Based Date Reuse:* Macroblocks previously encoded in the 3D-Neighborhood are used to create a search map that tracks the search pattern behavior. From the search map, a prefetch scheme named Dynamic Search Window formation is employed. This technique focuses on the reduction of external memory accesses and the reduction of active memory sectors in the on-chip video memory.

- *Application-Aware Dynamic Power Management*: This proposal implements a sophisticated multi-level memory requirements prediction scheme to accurately control power states of the on-chip video memory sectors. Each sector is associated, at frame level and refined down to MB level, to one of the multiple power states according to its usage probability. Once again, the MBs within the 3D-Neighborhood are used as source of information for decision making.

1.6 Thesis Outline

This thesis is organized as follows:

Chapter 2 presents an overview of the background knowledge required to understand this work along with the related works published in academic channels and state-of-the-art industrial solutions. The basics of 2D and 3D digital videos concepts, 3D video systems, multimedia architectural options, and Multiview Video Coding are provided. Afterwards, a state-of-the-art revision is presented including the latest reduced-complexity and energy-efficient solutions for the MVC encoding.

Chapter 3 brings a deep study and discussions on the requirements and challenges related to the realization of MVC real-time encoding on embedded devices. The discussions are centered on the energy consumption and encoded video quality. Chapter 3 also presents the overview of contributions presented along this thesis. For simplicity, the thesis contribution is also summarized using a high-level diagram.

In **Chapter 4** all the novel energy-efficient algorithms proposed in this work are thoroughly explained. They are classified and described in three sections: coding mode decision, motion and disparity estimation, and video quality management. Technical details ranging from case studies down to implementation level are followed by algorithm specific results.

The architectural contribution for motion and disparity estimation is presented in **Chapter 5**. Firstly, an architectural template is presented to avoid description redundancies between the three proposed hardware architectures. At this point onwards, Chapter 5 is organized in three sections that describe each proposed architecture separately. Inside each section the architecture-specific contributions, such as memory architectures and control schemes, are presented. Additionally, the architectural specific results are presented in this section.

Chapter 6 brings the overall results for the proposed novel algorithms and architectures compared to state-of-the-art related works. **Chapter 7** depicts the conclusions of this work and points to future research opportunities and challenges related to the next generations of 3D multimedia processing and 3D video coding. The works referred along this volume are presented in **Chapter 8**.

Additional tools and simulation environments are presented in the appendixes. **Appendix A** presents the MVC reference software, the JMVC, and details the modifications applied to the JMVC in order to enable software experimentation. The in-house developed Memory Access Analyzer tool and its graphic interface is presented **Appendix B**. **Appendix C** presents the CES Video Analyzer tool highlighting the extensions implemented to support multiview videos. Finaly, **Appendix D** brings an extended abstract of this thesis written in Portuguese.

2 BACKGROUND AND RELATED WORKS

In this chapter the basic notions on digital videos, multiview video systems and the Multiview Video Coding (MVC) standard are presented. The mode decision, motion and disparity estimation and, rate control modules are detailed since they are the main foci of this thesis. Detailed state-of-the-art review is presented considering 3D-video systems, multimedia architectures, energy-efficient algorithms and architectures for video coding.

2.1 2D/3D Digital Videos

A video is formed by a sequence of frames (or pictures) of a scene captured in a given frame rate providing to the spectator the sense of motion. Usually, the frame rate goes from 15 to 60 frames per second (fps) depending on the application requirements. Each frame is formed by a number of points named picture elements, i.e. pixels. The number of pixels in each frame is called resolution, i.e. the number of horizontal and vertical pixel lines. The typical resolutions also depend on the target application. For instance, mobile devices use to handle relatively lower resolution and lower frame rate sequences if compared to home cinema that targets high resolution and high frame rates.

Different color spaces are used to represent raw and decoded videos, the most usual ones are RGB (Red, Green, Blue) and YUV. Most monitors operate at the RGB space while most of video coding standards work over the YUV space. The YUV space is composed by three color channels: one luminance (Y) and two chrominance channels (U and V). The main reason for using YUV space for video coding is related to its smaller correlation between color channels, making easier to independently encode each channels. Since the Human Visual System (HVS) is less sensible to chrominance when compared to luminance, it is possible to reduce the amount of chroma information without affecting the overall perception. The reduction of chroma information is made using color sub-sampling (also known as pixel decimation). The most used color sub-sampling pattern is the YUV 4:2:0 that stores one U and one V sample for each four luminance samples reducing in 50% the total amount of raw video data (RICHARDSON, 2010).

All current widely used video coding standards are based on block coding. In other words, they divide each frame in pixel blocks to encode the video. These blocks are named macroblocks (MB). In the H.264, the latest video coding standard (JVT, 2003), the MBs are blocks of 16x16 luma pixels and its associated chroma samples (see Figure 2.1). A group of MBs is called slice. The slice can be formed by one or more MBs that may be contiguous or not. One frame is formed by one or more slices. In turn, each slice is classified in one of three different types (here the SI and SP slices are not considered): Intra (I), Predictive (P) and Bi-predictive (B) slices. The example in Figure 2.1 is composed of three slices, one contiguous (Slice 0) and two non-contiguous slices

(Slices 1 and 2). Note, the terminology used here is based on the H.264 standard and is directly applicable to the MVC standard (RICHARDSON, 2010)(JVT, 2009).

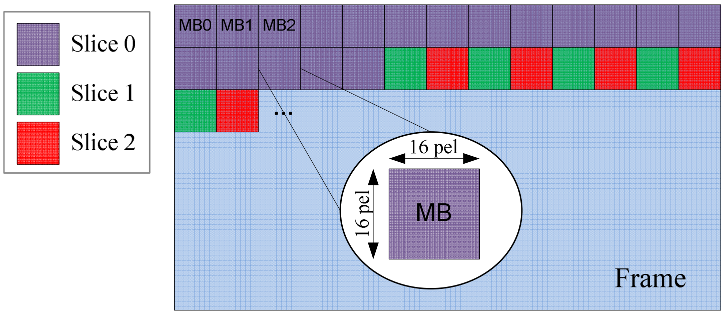


Figure 2.1: Macroblocks and slices organization

For a better comprehension on the different slice types it is necessary to understand the two basic prediction modes used by the state-of-the-art video encoders: intra-frame and inter-frame prediction. The intra-frame prediction only exploits the spatial redundancy by using surrounding pixels to predict the current MB. The inter-frame prediction exploits the temporal redundancy (similarity between different frames) by using areas from other frames, called reference frames, in order to better predict the current MB. Intra (I) macroblocks use the intra-frame prediction while predictive (P) and bi-predictive (B) macroblocks use the inter-frame prediction. While P macroblocks only use past frames as reference (in coding order) the B macroblocks can use reference frames from past, future or a combination of both. Intra slices are formed only by I MBs. Predictive (P) slices support I and P macroblocks and Bi-predictive (B) slices support I and B macroblocks (RICHARDSON, 2010)(JVT, 2009).

Multiview video sequences are composed of a finite number of single view video sequences captured from independent cameras in the same 3D scene (MERKLE, SMOLIC, *et al.*, 2007). Usually these cameras are carefully calibrated, synchronized and positioned. They are typically aligned in a parallel 1D-array or 2D-array, however, there are systems where the cameras are positioned in arch or cross shapes (KAUFF, ATZPADIN, *et al.*, 2007). The typical spacing between cameras is 5cm, 10cm or 20cm for most of the available test sequences (SU, VETRO e SMOLIC, 2006). In Figure 2.2 a multiview video with four views and the captured frames along the time axis are presented. At the video encoding perspective, the MVC, as detailed in Section 2.3, extends the concept of inter-frame prediction to inter-view prediction where the correlation between different views is exploited. A deeper discussion regarding the spatial, temporal and view/disparity correlations is provided in Section 2.2.

Figure 2.3 depicts the complete system required to capture, encode, transmit, decode and display multiview videos (CHEN, WANG, *et al.*, 2009). The captured sequence is encoded by an MVC encoder in order to reduce the amount of data to be transmitted. The generated bitstream may be transmitted using broadcast, internet or stored in media servers or local storage. At the decoder side the bitstream, or part of it, is decoded and displayed according to the displaying technology available at the receiver end. In a simple single-view display the decoder considers only the base view that is decodable with a regular (H.264/AVC) video decoder. In the case of stereoscopic displays (two views) only two views are decoded and displayed. In FTV (Free Viewpoint Television) systems the user selects the desired viewpoint within the 3D scene and the video decoder selects which views to decode. For multiview displays all views displayed must be decoded along with the reference views used to reconstruct them.

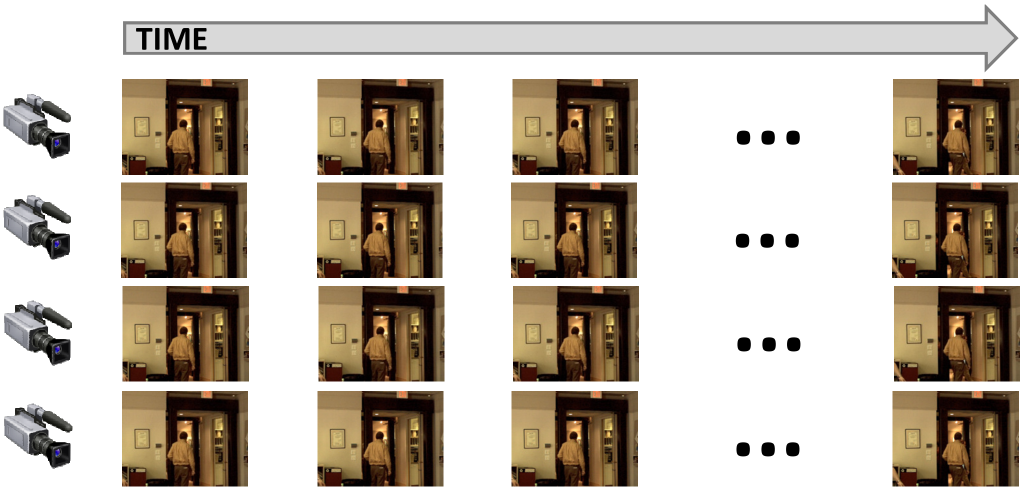


Figure 2.2: Multiview video sequence

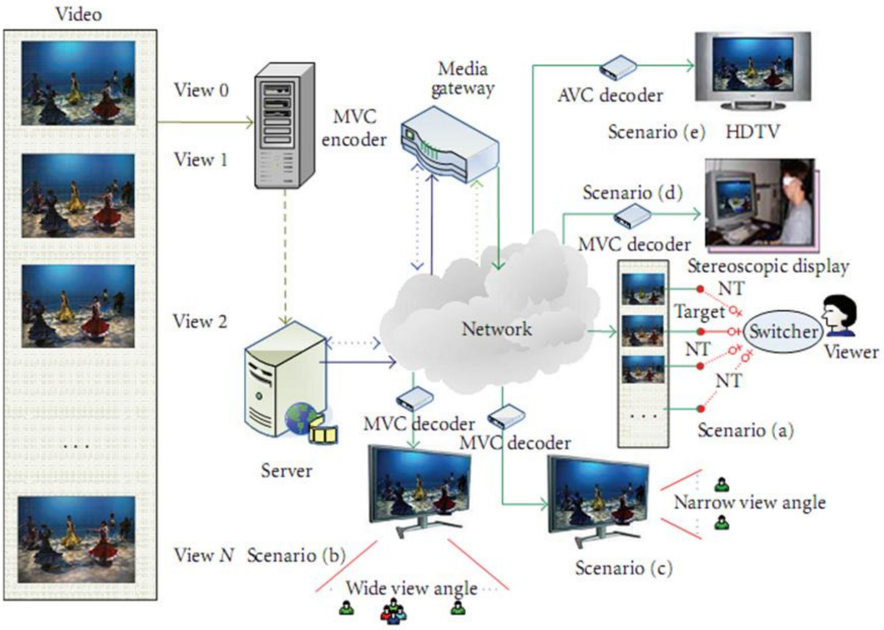


Figure 2.3: Multiview video capture, (de)coding, transmission and display system
Source: (CHEN, WANG, *et al.*, 2009)

2.2 Multiview Correlation Domains

This section defines the three types of redundancies or correlations present in multiview video sequences in order to provide the background required for a better understanding of the MVC coding tools, detailed in Section 2.3, and for the 3D-Neighborhood concept presented in Section 3.5.1. Here we discuss the correlation at pixel level, i.e., the similarities used to predict the image pixels, and at coding information level, i.e., how neighboring blocks share coding properties such as coding modes, vectors, etc. To have a more general description we present independently the three correlation dimensions: (i) spatial correlation, (ii) temporal correlation and, (iii) view/disparity correlation. Single-view video coding standards are able to exploit (i) and (iii) while MVC incorporates (iii) to provide improved prediction for multiview videos.

2.2.1 Spatial Domain Correlation

The spatial correlation is the similarity within regions in the same frame. Previous image and video coding standards, such as JPEG2000 and H.263, were already able to exploit this similarity through MB prediction based on neighboring pixels (see Section 2.3). Neighboring MBs tend to belong to the same image region and share similar image properties. For this reason, the surrounding pixels typically are good block predictors for the intra-frame prediction process. Exception cases happen in object borders where the image properties may change abruptly. Consider the example in Figure 2.4, all the MBs in the white background share similar image properties. The same happens for the MBs within one of the objects. The discontinuity occurs when an object border is found leading to increased prediction error. Note that, for simplicity, the spatial correlation is referred as one dimension but it is actually composed by two dimensions, the width and height of a frame.

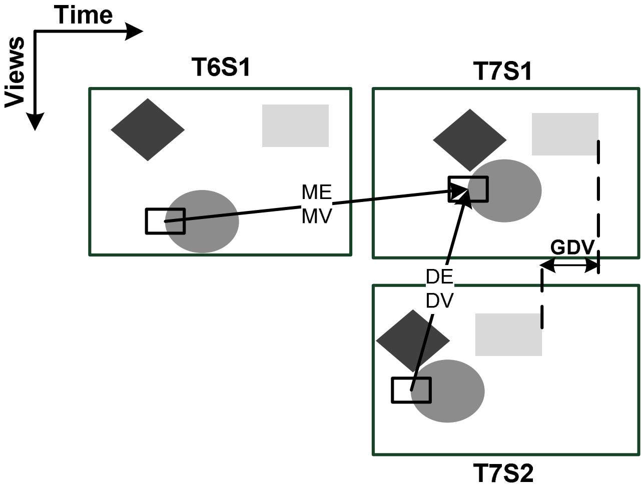


Figure 2.4: Neighborhood correlation example

On average, the current coding standards are able to efficiently employ the intra-frame prediction for pixel data. However, the correlation of coding side information (coding mode, motion vectors, disparity vectors, etc) is just superficially exploited. In H.264, a few simple techniques exploit this kind of correlation. The differential coding of intra prediction modes inside a macroblock exploits spatial correlation of coding information. In this technique, the intra coding mode is coded considering the coding mode of the previous block. Another example is the motion vector prediction process that uses the neighboring vectors to predict the current vector. By employing the motion vector prediction, only the differential motion vector need to be coded and transmitted. These examples show that there is also significant correlation at coding side information level.

2.2.2 Temporal Domain Correlation

The temporal correlation represents the similarities between different frames in the same view of a video sequence. That is, the objects of a given frame are usually present in neighboring temporal frames with a displacement that depends on its motion. Consider the frames T6S1 (view 1, time 6) and T7S1 (view 1, time 7) in Figure 2.4, the same objects are seen in both frames with a small displacement. Thus, frame T7S1 may be accurately predicted from the reference frame T6S1. The displacement between the two frames is found using the motion estimation (see Section 2.3.2). Besides the pixel-level prediction, the coding data is also similar for the same object along the time. In other words, for the same object in distinct time instants the same set of coding modes and motion behavior tend to be employed. The correlation is lost when there is an occlusion or the object moves out of the captured scene.

Analogous to the spatial correlation, there are tools able to exploit the temporal correlation at pixel level, i.e. the motion estimation (Section 2.3.2). At coding side information level, an attempt to exploit this correlation was proposed in the H.264 standard by using the temporal direct prediction for motion vectors. This prediction uses the collocated MB (MB sharing the same relative position in the frame) motion vector in order to predict the current one.

2.2.3 Disparity Domain Correlation

The disparity is a complete new domain introduced by multiview videos. It refers to the similarities between frames in different views. The similarities or redundancies at pixel level are exploited by the disparity estimation tool (Section 2.3.2). However, no tool is able to exploit this correlation at the coding information level.

As depicted in Figure 2.4, frames T7S1 (view 1, time 7) and T7S2 (view 2, time 7), the same objects are present in the neighboring views displaced by the so-called disparity vector. Since they are the same objects, the same image properties are shared and similar coding information tends to be used in different views. The disparity neighborhood correlation is lost when a given object is out of the area captured by a given camera or there is an object occlusion for a given camera point of view.

In order to obtain an accurate evaluation of the available correlation, we have carried out an extensive analysis of multiview videos. For this analysis we have used different multiview video sequences following the MVC test recommendation by JVT (SU, VETRO e SMOLIC, 2006). These sequences have coding structures similar to the one presented in Figure 2.7. Our analysis, discussed in Section 3.5.1, constitutes an in-depth exploration of coding mode distribution, video statistics, motion and disparity vectors, coding mode and RDCost correlation in the so called 3D-Neighborhood (spatial, temporal, and view neighborhood).

2.3 Multiview Video Coding

Encoding Multiview video sequences can be performed using different techniques. The most primitive one is the simulcast approach, where a single-view video coding standard (usually H.264/AVC) is used to encode independently each view. As presented in Figure 2.5, the simulcast approach considers the intra-frame prediction and inter-frame prediction (a.k.a. motion estimation) exploiting the spatial and temporal redundancy. However, the disparity or inter-view redundancy (i.e. the redundancy between frames of different views) is not considered. The Multiview Video Coding (MVC) standard uses the inter-view prediction (a.k.a. disparity estimation) to take advantage of the similarities between views from the same scene. The inter-view prediction represented by the red arrows in Figure 2.5 is responsible for a bitstream reduction of 20-50% for the same video quality (MERKLE, SMOLIC, *et al.*, 2007). Details on the MVC new tools, coding efficiency and complexity are discussed along this section.

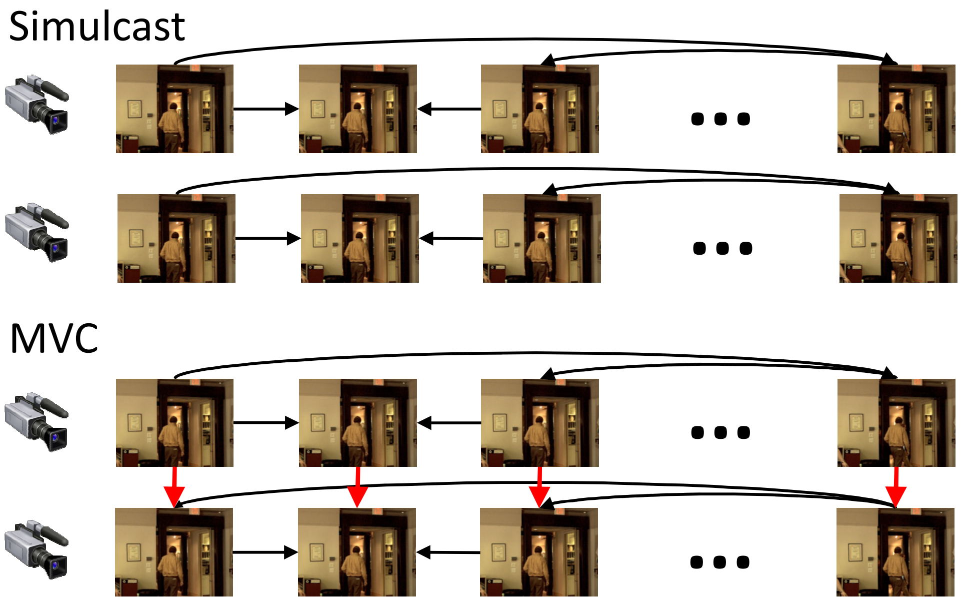


Figure 2.5: Prediction comparison between simulcast and MVC

In a strict definition, the Multiview Video Coding (MVC) is not a coding standard but an extension of the H.264/AVC or MPEG-4 Part 10 (JVT, 2003). The MVC was defined by the Joint Video Team (JVT) in March 2009 (JVT, 2009). The JVT is the group of experts formed by the Motion Picture Experts Group (MPEG) from ISO/IEC and the Video Coding Experts Group (VCEG) from ITU-T.

The standard usually works over the YUV (or YCbCr) (MIANO, 1999) color space that is composed by one luminance channel and two chrominance channels (red and blue chrominance) but other color spaces are supported, such as RGB and YCgCo (orange and green chrominance). The MVC also supports different subsampling patterns including 4:2:0 (four luminance samples for one sample of each chrominance channel), 4:2:2 (two luminance samples for one sample per chrominance channel) and 4:4:4 (one luminance channel for one sample in each chrominance channel). The supported color space/subsampling and coding tools depend on the profile of video coding operation (JVT, 2009).

Originally three profiles were defined in the H.264 standard: Baseline, Main and Extended. The Baseline profile focuses on video calls and videoconferencing. It supports only I and P slice and the CAVLC entropy coding method. The Main profile was designed for high definition displaying and video broadcasting. Besides the tools defined by the Baseline profile, it also includes the support to B slices, interlaced videos and CABAC entropy coding. The Extended profile targets video streaming on channels with high package loss and defines the SI (Switching I) and SP (Switching P) slices (RICHARDSON, 2010). In 2005 the Fidelity Range Extension (FRExt) defined the High profiles: High, High 10 (in which 10 bits per Y, Cb or Cr sample is used), High 4:2:2 and High 4:4:4 targeting high fidelity videos (JVT, 2009).

The MVC extension introduced to the standard a new set of CABAC contexts and new SEI (Supplemental Enhancement Information) messages to simplify parallel decoding and the transmission of sequence parameters (JVT, 2009). Additionally, the disparity estimation or inter-view prediction was proposed (MERKLE, SMOLIC, *et al.*, 2007). This is the most important innovation in the MVC that allows the exploration of similarities between different views. Its function is to find the best matching for the current macroblock in a reference frame within the reference view. The possible search criteria, search patterns and objective are similar to the motion estimation. However, the dynamic behavior of the disparity estimation differs significantly with respect to the ME. In the following section, details of the MVC encoding process are presented.

2.3.1 MVC Encoding Process

In Figure 2.6 the high-level block diagram of the MVC encoding process is presented. As a hybrid coding standard it is composed of three phases: prediction, transforms and entropy coding. The transform and entropy phases are similar to H.264/AVC, except for the new syntax elements to be encoded by the entropy encoder. The main innovation is in the prediction phase, which incorporates the inter-view prediction tool, the disparity estimation (DE).

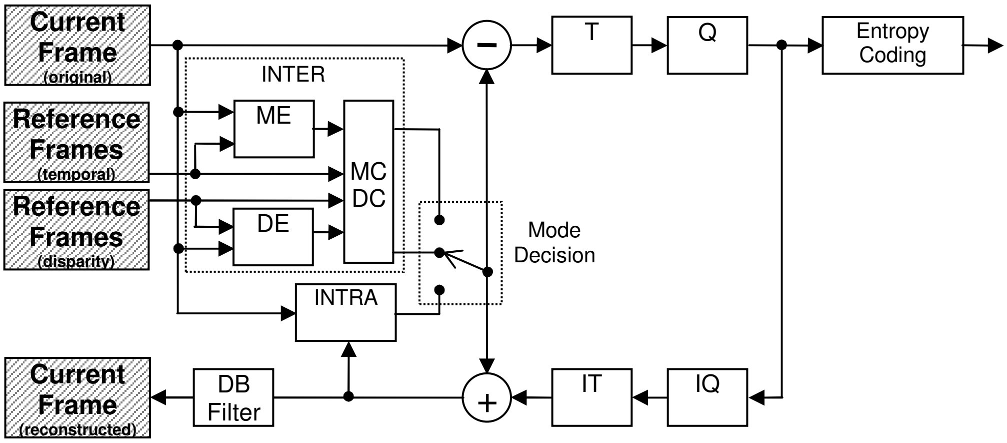


Figure 2.6: MVC encoder block diagram

The base view, the first one to be encoded, is encoded in compliance to the H.264 standard. Then, the prediction has two options, the intra-frame or the inter-frame prediction. Other views are named dependent views and also employ inter-view prediction. The complete encoding process is described in this section, considering the Main profile tools in YUV color space with 4:2:0 sub-sampling, while further extensions available in the High profiles are omitted for simplicity.

The MVC prediction structure inherits all the possibilities for temporal references and coding orders defined by the H.264. In addition, distinct possibilities of view coding order may be employed. The most used view coding orders are IPP and IBP (MERKLE, SMOLIC, *et al.*, 2007). The prediction structure depicted in Figure 2.7 employs IBP view coding order using Hierarchical Bi-Prediction (HBP) structure in temporal domain for 8 views and GOP (Group of Pictures) size equals to 8. The set of GOPs for all views are referred in MVC as GGOP (Group of Groups of Pictures). The frames located in the GGOP borders are called anchor frames while all others are the non-anchor frames.

The intra-frame prediction uses the neighboring pixels within the frame to predict the samples in the current MB. The MVC supports two MBs partitioning sizes for intra-frame prediction. The size 4x4 has nine prediction directions, as presented in Figure 2.8, where modes 0 and 1 apply a simple copy of the neighboring blocks and modes 3-8 perform a weighted interpolation according to the prediction direction. Mode 2 (DC) replicates the average of the neighboring samples to the entire block. Each one of the sixteen blocks inside the MB may use different prediction directions in order to find the best prediction.

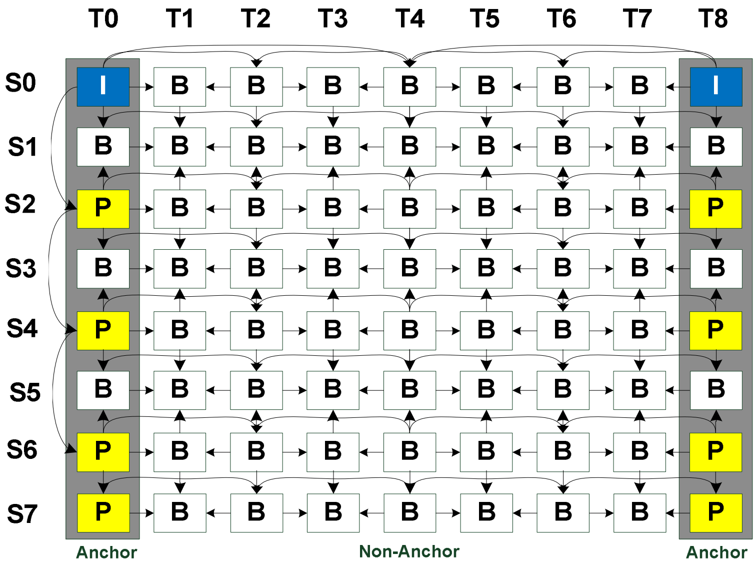


Figure 2.7: MVC prediction structure example

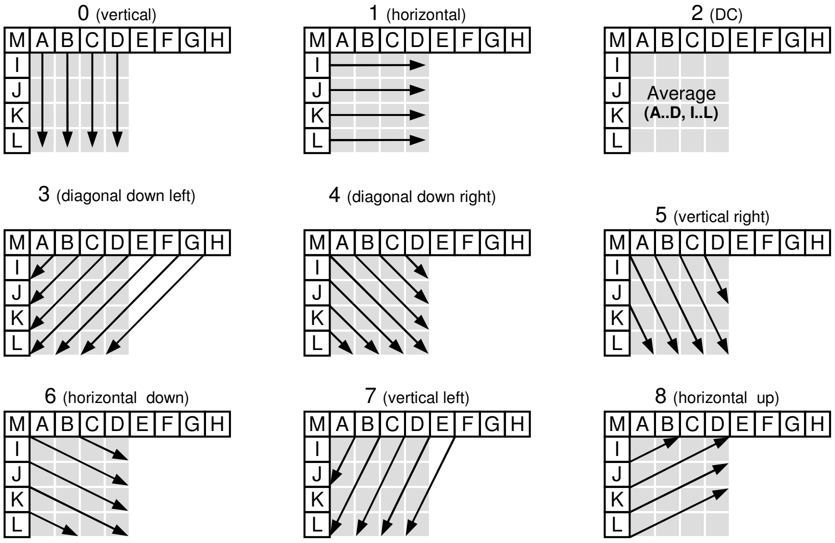


Figure 2.8: Nine prediction directions for intra-prediction 4x4

The intra-prediction can also be performed using the 16x16 block size. However, in this mode the number of prediction directions is restricted. Figure 2.9 presents the four prediction directions. Modes 0-2 are analogous to the modes 0-2 of the 4x4 block size. The plane mode (3) applies one linear filtering (RICHARDSON, 2010) to the neighboring samples resulting in a gradient texture. The 4x4 and 16x16 presented predictions are used for luminance samples. The chrominance prediction uses the same four directions present in 16x16 intra-prediction. The block size depends on the color sub-sampling; for the 4:2:0 color sub-sampling, 8x8 chroma blocks are used.

The inter-frame prediction or motion estimation (ME) provides other possibility of prediction. Its function is to perform a search in the past or future previously encoded frames to find the best matching candidate in order to provide a good prediction. The

ME features bi-prediction, multiple block-size, motion vector prediction, ¼ sample motion vector accuracy, weighted prediction and other tools that help to improve the prediction quality (RICHARDSON, 2010), as defined in the H.264/MVC video coding standard and detailed in Section 2.3.2.

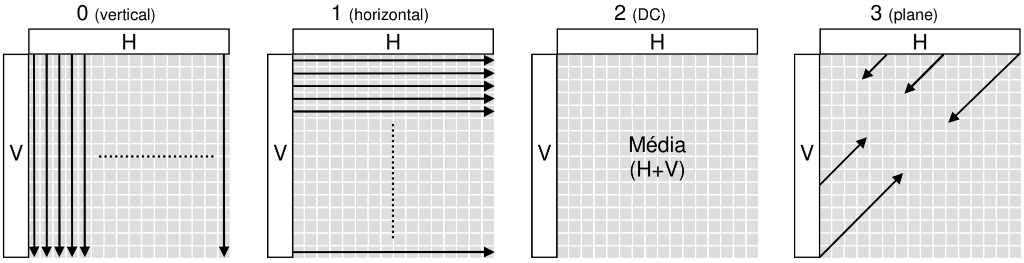


Figure 2.9: Four prediction directions for intra-prediction 16x16

For the dependent views (all views except the base one), the inter-view prediction or disparity estimation (DE) is also available (MERKLE, SMOLIC, *et al.*, 2007). This MVC extension searches for the best matching candidate in the frames belonging to previous encoded views (left, right, up or down, depending on the cameras arrangement and view prediction structure). All features from ME are supported in DE, more details about these features and how they influence the encoder efficiency and complexity will be discussed in Section 2.3.2 and Section 3.1.1.

The output of the prediction phase is a large set of prediction candidates. Among all different block sizes for intra-prediction, inter-frame prediction and inter-view prediction, the best prediction mode must be selected by the mode decision (MD) in order to provide the optimal rate-distortion (RD) tradeoff (RICHARDSON, 2010). The rate is the number of bits required to encode the MB and distortion is the objective video quality measured in Peak Signal-to-Noise Ratio (PSNR). To have the optimal solution all modes must be completely encoded, reconstructed and evaluated according to an RD optimization equation. Therefore, the MD (represented by the selection key in Figure 2.6) is of key importance since it controls the quality versus efficiency tradeoff and the computational complexity of the encoder. The MD optimization process is discussed in Section 2.3.3.

After the prediction phase is completed, the predicted macroblock and the original macroblock are subtracted to generate the image residues. To reduce the energy in a few coefficients the residues are transformed from the space domain to the frequency domain using an integer approximation of the 4x4 2D-DCT transform. If the intra-prediction 16x16 is selected, an additional Hadamard transform is applied after the DCT. In this case, the DC coefficients of each 4x4 block (left upper corner of each block as depicted in Figure 2.10) compose another 4x4 coefficient block and are submitted to a 4x4 Hadamard transform. The values inside each block in Figure 2.10 represent the double-Z processing order of the blocks in the transform.

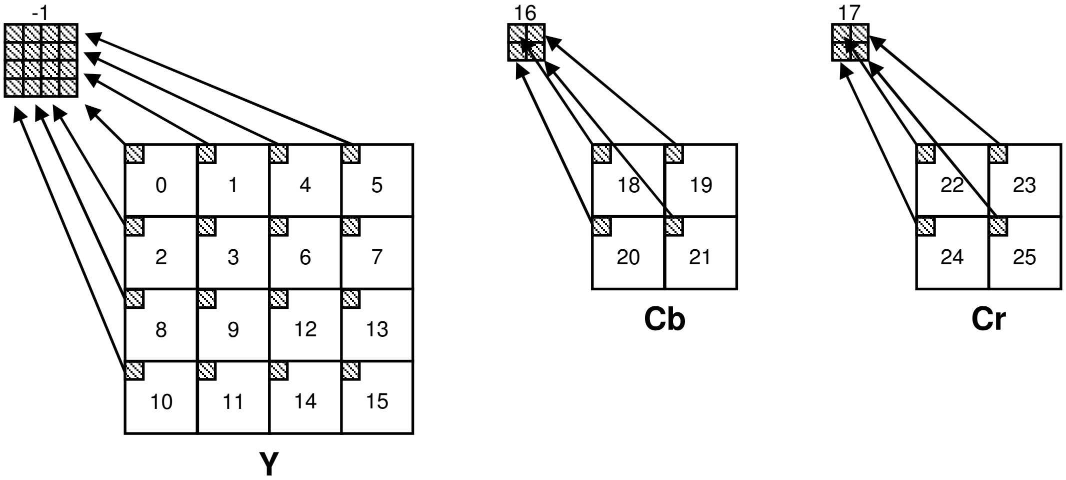


Figure 2.10: Block processing order in the transform module

Once the transforms are concluded, each block is quantized to reduce the dynamic range of the coefficients for the entropy coding. In MVC a linear quantization is used. The quantization step is defined by the H.264/MVC standard (RICHARDSON, 2010).

Finally, the quantized coefficients are sent to the entropy encoder. Each block is scanned in zigzag order, according to Figure 2.11, and encoded by one of the two standard entropy encoders: CABAC or CAVLC. The Context-Adaptive Variable Length Coding (CAVLC) use predefined tables depending on the syntax element being encoded. The coding method is an evolution of the Huffman coding to better adapt to multiple contexts. The Context-Adaptive Binary Arithmetic Coding (CABAC) is a new tool defined by the H.264/AVC standard and implements a novel coding technique able to reduce the bitstream size by about 5-15% (WIEGAND, SULLIVAN, *et al.*, 2003) in comparison to the CAVLC encoder. The tables of probability used in CABAC are updated at bit level and present strong data dependencies. For further information please refer to (JVT, 2009)(RICHARDSON, 2010).

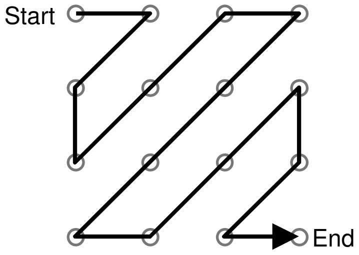


Figure 2.11: Zigzag scan order for a 4x4 block

After the entropy coding, the bitstream is assembled and the encoding is complete. However, every macroblock has to be reconstructed to work as reference for further MBs. For that, the inverse quantization and inverse transforms are applied to the quantized coefficients (the same data previously sent to the entropy).

Once the residues are inversely quantized, they are added to the predicted block in order to reconstruct the decoded MB. The reconstruction loop guarantees the consistency between encoder and decoder sides avoiding drifting between encoder and decoder. To reduce the blocky effect (due to different prediction modes) in the reconstructed frames, the standard defines an in-loop deblocking filter (DF). The filtered MBs are used for displaying and to generate the reference for inter-frame and inter-view references. Intra-prediction uses unfiltered macroblocks inside a frame. The DF has five filtering strengths and filters the borders of each 4x4 block of the image following the order presented in Figure 2.12 (RICHARDSON, 2010).

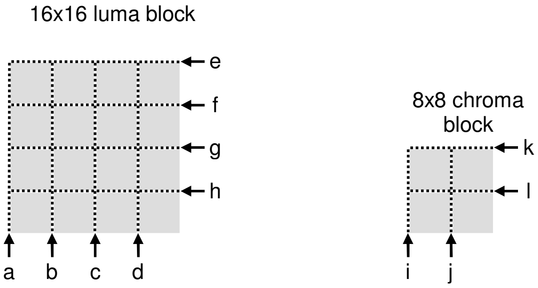


Figure 2.12: Order of macroblock borders filtering

2.3.2 Motion and Disparity Estimation

Multiview video sequences are usually captured using a high sample rate, over 30 frames per second, to improve the motion flow and give the observer a sense of smoother motion. This high frame rate implies in a high redundancy or similarity between neighboring frames in the time axis. As noticed in Figure 2.13, frames S0T0 and S0T1 are very similar, hence only the differences between them have to be transmitted. The algorithm that exploits these inter-frame similarities is the motion estimation (ME). It searches in the temporal neighboring frames, known as reference frames (see Figure 2.14), the region that represents the best match for the current block or macroblock. Once the best matching block is found, a vector pointing to that position, the motion vector (MV) in Figure 2.14, is generated. Consider, for example, a background region (one of the yellow boxes in Figure 2.13), there is no motion between T0 and T1, so the motion vector m_2 is probably zero. The dancers moving (woman's face in the yellow box) present a displacement along the time; this displacement is represented by m_1. The set of motion vectors of a given frame is called motion field and represent valuable information to understand the motion of an object as time progresses.

The cameras that capture the different sequences in a given 3D scene are located near each other (typically, about 5cm to 20cm apart) (SU, VETRO e SMOLIC, 2006), thus there are many regions that are shared between neighboring cameras. A very high similarity is perceived between neighboring cameras as exemplified in frames S0T0 and S1T0 of Figure 2.13. The MVC defines the disparity estimation (DE) to exploit the redundancy between different views and minimize the transmission of replicated information multiple times. The approach of the DE is similar to the ME. It searches for the best matching candidate block within frames of the neighboring views. The frame used for search is called reference frame while the view is called reference view, as shown in Figure 2.14. Once the matching block is found the position is pointed by the so-called disparity vector (DV), see Figure 2.14. The set of DVs in a frame are referred as disparity field and represent the disparity of the objects between views. While the length of motion vectors (MV) represents the speed an object is moving (or the camera is moving) the disparity vectors denote the displacement of a given object between two views. The disparity depends on the distance between cameras, and the distance between the camera and the object (KAUFF, ATZPADIN, *et al.*, 2007). The closer the object is the larger the displacement or disparity. For instance, in Figure 2.13, the background presents almost no disparity between S0 and S1 (d_2) while the dancers have a much larger disparity vector (d_1). The average disparity vector between two views considering all objects and background is named Global Disparity Vector (GDV) (KAUFF, ATZPADIN, *et al.*, 2007) (HAN e LEE, 2008), see Figure 2.7.

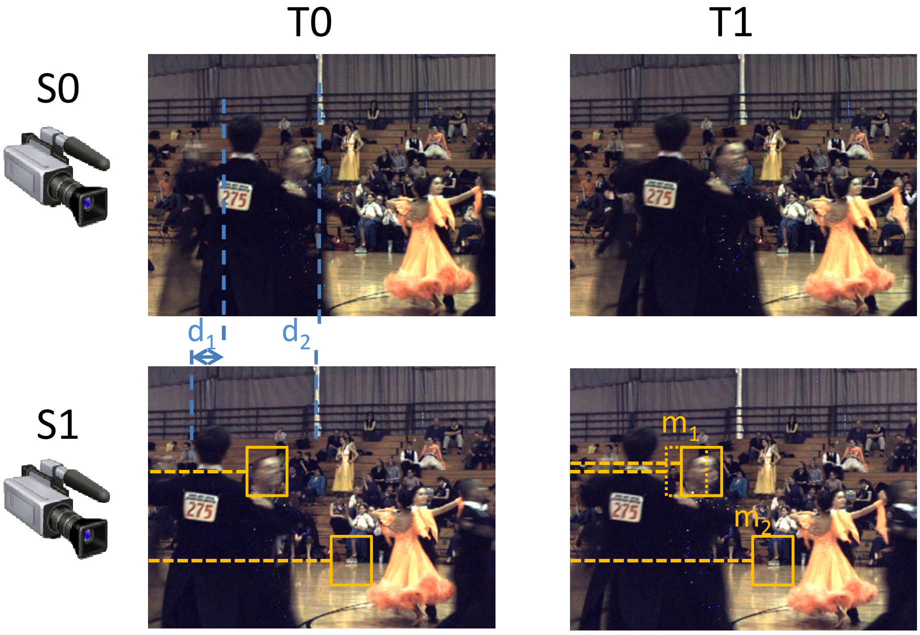


Figure 2.13: Temporal and disparity similarities

The ME/DE search is not performed over the complete reference frame but in a region called search window (SW) defined by a search range (SR), as shown in Figure 2.14, for instance a SR [±16,±16] covers a SW of 33x33 samples. Many search schemes for ME were proposed along the last two decades and their characteristics are well known. The exhaustive search algorithm, the *Full Search* (FS) (YANG, 2009), provides the optimal results at the cost of a very high computational effort. Many fast algorithms focusing on complexity reduction with small quality loss are found such as *Log Search* (JVT, 2009), *Diamond Search* (DS) (KUHN, 1999), *Three Step Search* (TSS) (JING e CHAU, 2004), *UMHexagon Search* (CHEN, ZHOU e HE, 2002) and *Enhanced Predictive Zonal Search* (EPZS) (TOURAPIS, 2002), to list a few. These algorithms are based on multiple search steps oriented by geometric shapes. The most recent schemes also consider the neighboring MBs as predictors to define the search starting point. Using predicted starting point is an evolution compared to the previous search schemes that use the collocated MB as starting point. Recalling, the collocated MB is the macroblock in the reference frame that belongs to the same relative position of the current MB.

Despite the similarity between ME and DE there are behavioral differences that make solutions defined for ME inefficient when applied to DE. For instance, most of the traditional ME fast search patterns perform badly for DE. The reason for that is the motion vectors are usually located in a relative small length range while disparity vectors usually are much longer. The disparity vectors frequently have 50-100 samples length. For this reason, the recommended search range is at least [±96,±96] for SD resolutions (XU e HE, 2008). In this scenario most of the fast algorithms tend to fall in local minima and do not find the optimal candidate. For this reason the JMVC, the reference software for MVC (JVT, 2009), implements the *TZ Search* that is more complex in comparison to DS and EPZS, for example, but is still 23x times faster than

FS (YANG, 2009). The TZ employs predictor centered search start and a larger geometric shape search pattern. It performs well for both ME/DE with negligible or no quality loss in comparison to FS.

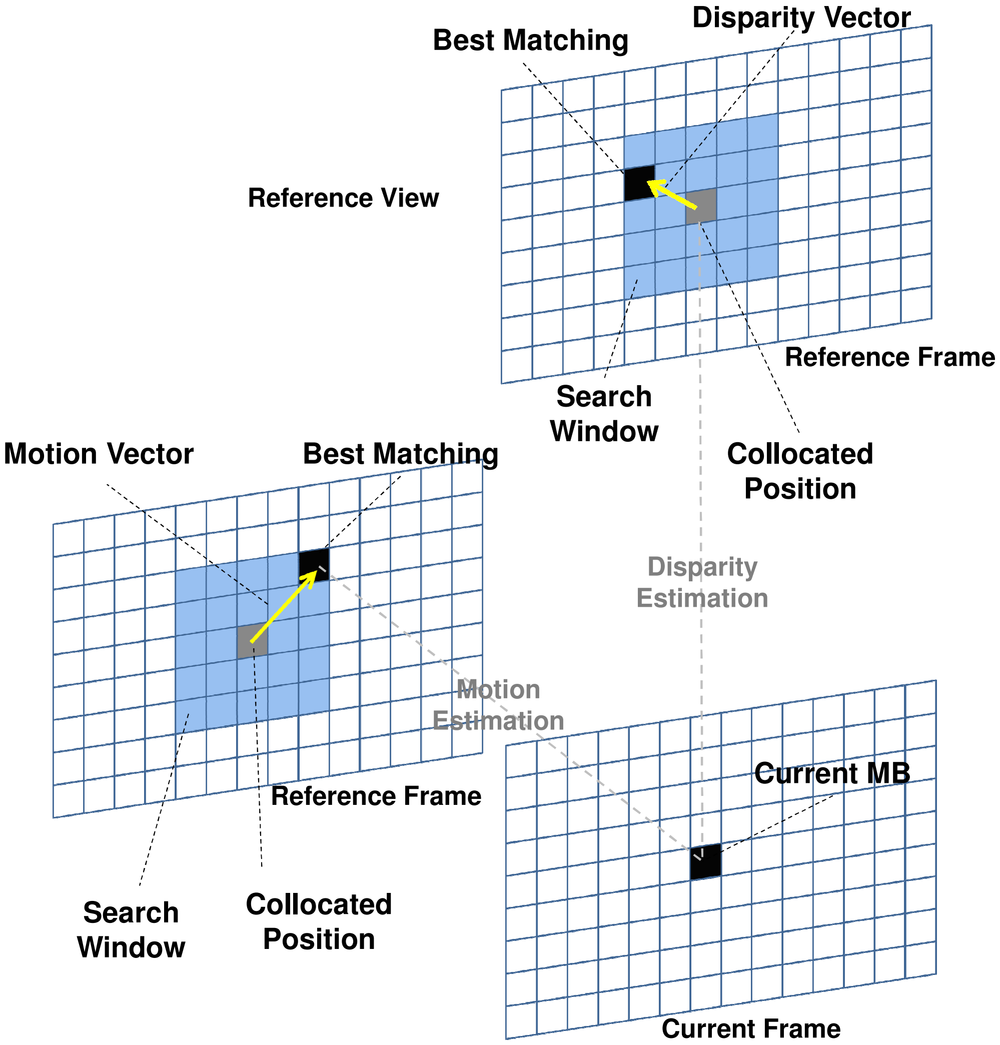


Figure 2.14: Motion and disparity estimation

However, once the conceptual tasks of ME and DE are similar, the available features are the same and together they represent the most computational and memory intensive tasks in the video encoder, see discussion in Section 3.1. For this reason, ME/DE have to be jointly considered in order to propose smart fast algorithms and efficient *architectural solutions for real-time MVC encoding.*

In the following, the motion and disparity estimation features are detailed. Note that all these tools are mandatory at the decoder side depending upon the operation profile but are optional for the encoder.

Bi-Prediction: In MVC, two types of MBs employ the ME/DE: Predictive (P), which is coded using inter-frame prediction referencing only past frames and backward views, in display order; or Bi-predictive (B), which is coded using both, reference frames from past/backward and from future/forward (this is possible due to the out-of-order coding and decoding allowed by the standard). In a B macroblock, each partition can be predicted from one or two reference frames (SULLIVAN e WIEGAND, 2005).

In case of bi-prediction the final prediction is generated by calculating the average of the prediction from past/backward and future/forward.

The reference frames are stored in two lists: List 0 and List 1. List 0 orders the frames from the past and backward views and List 1 orders the frames from the future forward views (JVT, 2003). Both lists can be ordered using temporal references first or disparity references first. For temporal references first, in List 0 the reference index 0 is the closest past encoded frame. For disparity references first, the index 0 in List 0 is the closest backward view reference frame. Analogous organization is observer in List 1.

Multiple Block-Sizes: MVC allows ME/DE blocks of several sizes. The 16x16 MB can be segmented in two 16x8, two 8x16 or four 8x8 partitions (JVT, 2009). Each 8x8 partition can be segmented in other two 8x4, two 4x8 or four 4x4 sub-partitions. Each partition may point to one reference frame per list (List 0 and List 1) while each sub-partition may use only the frames referenced by the partition that it belongs. Each partitions or sub-partitions may have a single MV or DV.

Multiple Reference Frames and Reference Views: Differently from earlier standards, in MVC the past and future reference frames are not only fixed to the immediate ones. Therefore, to reconstruct one given macroblock, temporally distant frames can be used in the prediction process and this distance is limited only by the size of the Decoded Picture Buffer (DPB) (SULLIVAN e WIEGAND, 2005). The reference frames are managed in List 0 and List 1 as previously cited. Analogously, the reference views are not restricted to the closest backward or forward views, any previously encoded views may be used as referenced depending if obeying the coding settings.

Quarter-Sample Motion Vector Accuracy: In general, the motion of blocks does not match exactly in the integer grid of pixels in a frame, then fractional-sample motion vector accuracy is used to reach a better match. The MVC (JVT, 2003) defines the use of a quarter-sample motion compensation for the reference frame blocks. For luma samples, a six-tap FIR filter is used to interpolate half-samples, and then a simple average of integer and generated half-samples is used to generate the quarter-sample interpolation (JVT, 2003). When working with 4:2:0 sub-sampling, the chroma samples interpolation applies 1/8 sample accuracy.

Weighted Prediction: The MVC defines a weighted prediction in the inter-frame coding process to apply a multiplicative weighting factor and an additive offset to each interpolated sample of a given reference frame (JVT, 2003). For single directional prediction from List 0 or List 1 this tool is defined as presented in Eq. (2.1), where ‘x’ is replaced by the list number (0 or 1), ‘w’ is the weighting factor, ‘logWD’ is a scaling factor and ‘o’ is the additive offset. P represents the interpolated pixels and P’ the weighted sample. For bi-predictive prediction the weighted prediction is defined as presented in Eq. (2.2).

$$P'(i,j) = ((P_x(i,j) * w_x + 2^{\log WD-1}) >> \log WD) + o_x \tag{2.1}$$

$$\begin{aligned} P'(i,j) = {} & ((P_0(i,j) * w_0 + P_1(i,j) * w_1 + 2^{\log WD-1}) \\ & >> (\log WD + 1)) + ((o_0 + o_1 + 1) >> 1) \end{aligned} \tag{2.2}$$

Motion/Disparity Vector Prediction: Exploiting the neighboring blocks correlation, the MVC standard defines that motion vectors and reference indexes (pointer to the reference frame in List 0 or List 1) have to be inferred from the reference index and motion/disparity vectors of neighboring blocks. The inferred vectors are called predicted motion vectors (PMV). Differential motion vectors (MVD) are coded in the bitstream and summed up to the PMVs, obtaining the current motion vector (MV) or disparity vector (DV). The PMVs are normally obtained applying the median to the spatial neighbor blocks vectors. However, SKIP macroblocks (those that transmit no vectors or residuel) and direct predicted macroblocks are differently processed using the direct spatial or direct temporal predictions. The motion/disparity vector prediction is one example of using the video correlation to predict coding side information, as previously mentioned in Section 2.2.

2.3.3 MVC Mode Decision

The MVC provides a big number of options for the macroblocks prediction. Intra-prediction defines two prediction sizes (three in case FRExt is considered), 16x16 and 4x4, with four and nine prediction modes, respectively. ME evaluates multiple candidate blocks for seven different block sizes. Additionally, the new disparity estimation adds a set of coding possibilities as large as the motion estimation possibilities.

The mode decision (MD) module is the responsible to deal with this large optimization space. For that it implements an optimization algorithm and defines a cost function called RDCost, the Rate-Distortion cost (a.k.a. *J* cost). The objective is to evaluate the coding modes and to find the one that minimizes the RDCost to obtain the best coding relation between rate and distortion. Eq. (2.3) presents the *J* function where *c* and *r* represent the current original MB and the reconstructed one, *MODE* is the prediction mode used and *QP* is the quantization parameter. *D* represents the distortion measured after the complete MB reconstruction according to a distortion metric and *R* is the number of bits used to encode the current MB, this number is available once the entropy encoding is completed. λ is the Lagrange Multiplier used to control the rate-distortion tradeoff. The Lagrange Multiplier value is not defined by the standard; however, typically it is defined by the Eq. (2.4) and depends upon the QP. To quantify the distortion, different metrics may be used; some examples are SAD (Sum of Absolute Differences), SATD (Sum of Absolute Transformed Differences) and SSE (Sum of Square Errors). The SSE is mostly used in the mode decision step since it provides better PSNR results. The reason is that PSNR is calculated using MSE (Mean Square Errors) which is only a division of SSE value, so the SSE is directly related to PSNR. It is important to understand that PSNR is currently the most accepted objective video quality metric. However, SAD is widely used in real-time systems due to its light-weight computation.

$$J(c,r,Mode|QP)=D(c,r,Mode|QP)+\lambda_{Mode}*R(c,r,Mode|QP) \qquad (2.3)$$

$$\lambda = 0.85*2^{(QP-12)/3} \qquad (2.4)$$

Although the algorithm to find the mode that minimizes the RDCost is not defined by the standard, the MVC reference software JMVC (JVT, 2009) implements an exhaustive search by completely encoding all possible coding modes and selecting the best mode. It is known as Rate-Distortion Optimized mode decision (RDO-MD) also referred as Full RDO or Exhaustive RDO. The RDO-MD guarantees the optimal MB encoding but drastically increases the encoder computational effort and makes the same approach for real-time MVC encoding unfeasible for the current technology.

2.3.4 MVC Rate Control

According to (LI, PAN, *et al.*, 2003), the rate control (RC) is a block of the video encoder that aims to regulate the output coded bitstream to produce high video quality at a given target bitrate. In the MVC scenario, an efficient RC scheme must be able to provide increased video quality for a given target bitrate with smooth visual quality variation along the time, for different views and within the frames. Most important, the RC should keep the bitrate as close as possible to the target bitrate (optimizing the bandwidth usage) while avoiding sudden bitrate variations.

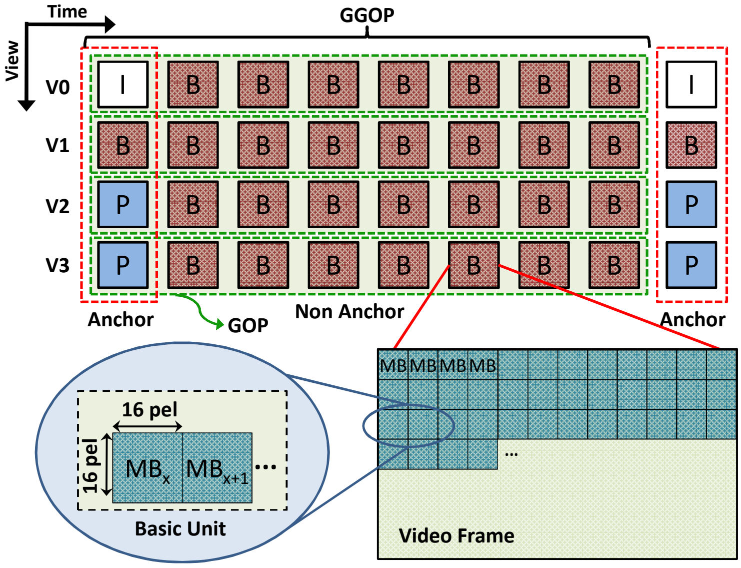


Figure 2.15: MVC rate control actuation levels

The rate control unit typically controls the quality vs. bitrate through QP adaptation. The bitrate and/or the video distortion metric are predicted using a prediction model. According to the prediction and the target bitrate (amount of bits per second used to encode the video), an adequate QP is selected. As QP grows higher, more residual data are quantized (video details lost) and more quality losses are inserted. The actual generated bitrate and the video quality may be used as feedback for the RC unit in order to update the prediction and QP definition. The QP adaptation may be performed in distinct actuation levels. In general, the RC for MVC can be classified in at least three actuation levels: (i) GOP level (Group of Pictures – set of frames); (ii) frame level, and (iii) Basic Unit (BU – set of one or more macroblocks *MB*) level, as shown in Figure 2.15. It is possible to combine GOP-level and frame-level and based on this observation, for simplicity, they are jointly discussed in this thesis.

In the following sections we present the state-of-the-art related to 3D video systems, MVC encoders and multimedia processing. Also, a literature overview on the latest low-complexity and energy-efficient solutions focusing mode decision, ME/DE and rate control for the MVC standard is presented. An overview on low-power techniques is also provided to give the technical background required for our energy-efficient architectural solutions.

2.4 3D Video Systems

The advances in video coding techniques targeting multiview videos have been driven by the increasing set of commercial systems employing 3D video capabilities.

These systems range from high-end cinemas and 3DTVs to mobile devices including content suppliers. Wider adoption is expected for the upcoming years with the increase in the available video content through 3D-capable television broadcasters, optical media (BLU-RAY DISC ASSOCIATION, 2010), popularization of personal 3D camcorders, 3D video stream services (YOUTUBE 3D, 2011) (VIMEO, 2012), etc. All these commercial systems are, currently, based on stereoscopic videos (two views). An increase in the number of views is expected for the near future (FUJII, 2010) to improve the observer freedom and provide a more immersive experience. Some experimental and academic multiview systems are already available or under development to support the next generations of 3D video systems. In this section we start presenting the most prominent commercial 3D-video systems.

3D-cinema systems are based on three market-leader technologies based on stereoscopic videos (IMAX, 2012), (REALD, 2012), and (DOLBY, 2012). The technology employed in (IMAX, 2012) requires the use of linear polarized glasses to block the light for one eye at a time allowing each eye to see only the frames intended for that eye. Two projectors are used to display 48 fps where each eye is able to effectively see 24fps in a time-sharing strategy. (REALD, 2012) is also based on time-sharing between the two eyes, however, the glasses are circular polarized glasses where each glass is polarized in an opposite direction. Also, (RealD, 2012) requires a single projector able to display 144 fps. Each effective frame for each eye is displayed thrice resulting in effective 24fps per eye. Finally, (DOLBY, 2012) employs passive glasses with dichroic filters where each view is displayed with a distinct chromatic filter and perceived by a single eye. With this strategy both views are simultaneously displayed allowing the use of regular 24fps projectors. At the video coding perspective, all these high-end applications support the stereoscopic MVC (THE DIGITAL ENTERTAINMENT GROUP, 2009).

Stimulated by the content available in the 3D Blu-Ray (BLU-RAY DISC ASSOCIATION, 2010) - optical media that supports the MVC coding standard - the 3D televisions already exceed 10% of the televisions sold in the United States, in 2011, and this number is expected to reach 37% of the market share in 2014 (RESEARCH AND MARKETS, 2010). Other countries are expected to follow this trend. The majority of those 3DTVs is based on stereoscopic displaying and requires active shutter or passive polarized glasses to provide the 3D sensation. Many devices employ built-in decoders supporting the MVC standard. Along with the cinema solutions, the 3D televisions are not energy-critical and typically implement only the video decoder side, less complex in relation to the encoder.

Currently, portable devices capable of handling digital video are available everywhere for a reasonable cost. The omnipresence of these gadgets implies a very large amount of data being produced. In this scenario the coding efficiency is a key issue in order to reduce the storage and transmission costs for digital video. Various devices are also capable of real-time 3D videos recording, such as (PANASONIC, 2011), (FUJIFILM, 2011), (SHARP, 2011), and (SONY, 2011). Most of them feature 2 cameras and encode the video sequences independently (simulcast). However, the increase in number of views from 2 up to 4-8 views (FUJII, 2010) in order to provide enhanced 3D experience freedom is envisaged for the next 3-5 years. In this scenario it is simple to conclude that the large amount of data generated requires the use of the state-of-the-art MVC standard. The first personal camcorder to fully support stereo MVC was released by Sony in 2011 (SONY, 2011).

Although 3D capable mobile devices are already available, attending MVC performance and energy constraints remains a big challenge for industry and academia, as discussed in Section 3.2. The current multimedia processing systems based on processors, DSPs and non MVC-optimized ASIC implementations are not efficient to provide the required throughput with the required energy efficiency while sustaining video quality and coding efficiency. In the following section we present an overview of the main multimedia architectural approaches and solutions in the current state-of-the-art.

2.5 Multimedia Architectures Overview

In this section we present the multimedia processing architectures classified in four main classes: Multimedia Processors/DSPs, Reconfigurable Multimedia Processors, ASIC Multimedia cores and, Heterogeneous Multicore SoCs. On the one hand, ASIC solutions provide the highest performance and energy efficiency at the cost of reduced flexibility limiting the applicability to upcoming video standards. Still, the current lack of MVC-oriented ASIC optimizations prohibits further increase in both performance and energy efficiency. On the other hand, multimedia processors/DSPs allow high flexibility to multiple standards while providing reduced performance and poor energy efficiency if compared to ASICs. Additionally, reconfigurable processors may allow significant increase in performance and flexibility through ISA (Instruction Set Architecture) extensions. The reconfigurable processors, however, present reconfiguration energy issues and are unable to reach the ASIC-like performance and energy-efficiency required by the 3D multimedia applications.

2.5.1 Multimedia Processors/DSPs

Aware of multimedia processing characteristics, the Multimedia Processors/DSPs are designed to exploit the parallelism inherent to these applications. Massive multicore architectures are proposed to target task parallelism by supporting multiple parallel threads. Data-level and instruction-level parallelisms are exploited by employing SIMD (Single Instruction Multiple Data) and VLIW (Very Large Instruction Word) architectures, respectively. Some proposals are able to implement hybrid parallelism by handling multiple cores with SIMD and/or VLIW instruction sets.

In (ABBO, KLEIHORST, *et al.*, 2008), the Xetal-II employs 320 SIMD processing elements with a dedicated 10Mb on-chip frame memory. It is able to provide 107 GOPS with a 60W power consumption with instructions designed targeting video analysis applications. A multicore system for video decoding is proposed in (FINCHELSTEIN, SZE e CHANDRAKASAN, 2009) employing a caching mechanism to reduce the memory reads. The work presented in (KHAILANY, WILLIAMS, *et al.*, 2008) describes a processor with 16 parallel lanes where each lane is a 5-ALU VLIW core. At 800MHz, this solution delivers 512 GOPS (82pJ/MAC) and guarantee baseline HD1080p H.264 encoding at 30fps. The multi-streaming SIMD multimedia engine proposed in (CHIU e CHOU, 2010) claims a 3.3-5.5x performance increase compared to MMX architecture (Intel Multimedia Extension) by employing 12 multimedia kernels. These parallel architectures provide a relative high performance but are still far below MVC requirements and the power envelope is out of embedded devices boundaries.

A 2-issue VLIW stream processor is presented in (CHIEN, TSAO, *et al.*, 2008) with throughput for CIF encoding at 30fps. Stereo processing-oriented optimizations for

VLIW processors are presented in (PAYÁ-VAYÁ, MARTÍ-LANGERWERF, *et al.*, 2010). The authors claim performance improvements by implementing a new register file access mechanism and disparity functional unit to calculate disparity map. Also, an ASIP (Application Specific Instruction Processor) based on a VLIW DSP architecture is described in (ZHANG, HE, *et al.*, 2009) and delivers increased performance if compared to traditional DSP and SIMD.

2.5.2 Reconfigurable Processors

In (OTERO, DE LA TORRE, *et al.*, 2010) an architectural template for run-time scalable systolic coprocessors is presented. It focuses on run-time adaptation to time-variable tasks or changing system conditions. It exploits replacing and relocation of basic processing elements of the array using FPGAs dynamic reconfiguration. In (BECK, RUTZIG, *et al.*, 2008) is employed a coarse-grained reconfigurable array with a run-time mechanism designed to translate MIPS instruction to be executed in the reconfigurable array. (BEREKOVIC, KANSTEIN, *et al.*, 2009) presents the mapping of MPEG-2 and H.264/AVC to the ADRES (coarse-grain reconfigurable processor) delivering throughput for real-time CIF decoding at 50MHz with a 4x4-core array. CRISP, a coarse grain reconfigurable stream processor (CHEN e CHIEN, 2008), implements an image processing pipeline reaching 55fps for HD1080p resolution. Aggressive performance losses are expected for video coding due to increased complexity compared to the implemented image processing algorithms.

In (BAUER, SHAFIQUE, *et al.*, 2007), the RISPP (Rotating Instruction Set Processing Platform) is presented bringing more flexibility to extensible processors. It features a special instruction forecasting algorithm able to predict the hotspots and allows to adapt at run time the different *Molecules* (implementation of the special instructions). This architecture was evaluated using some H.264 processing hotspots (SATD, DCT, etc) and demonstrated high flexibility to deal with the performance versus hardware tradeoff. This concept was extended in (BAUER, SHAFIQUE, *et al.*, 2008) by integrating a special instruction run-time scheduler able to outperform the state-of-the-art in 2.38x for the H.264 application. When integrated to a transmutable embedded processor (BAUER, SHAFIQUE e HENKEL, 2008), the RISPP concept was able to present up to 7.19x speedup in relation to related works for H.264.

Compared to regular processors, reconfigurable processors target to increase the overall performance by adapting, at run time, to distinct applications properties. Also, the adaptivity can be efficiently exploited within the same application. Considering multimedia applications, the performance/energy requirements may vary with the video content, user settings, battery-level, etc. It brings a big optimization potential at the system perspective. However, when considering a single application for a given description, in this case real-time encoding for MVC HD1080p, the profit of this adaptive behavior is not be perceived. Moreover, in this scenario, the energy and time costs for reconfiguration pose additional difficulties in terms of throughput and energy efficiency if compared to processors, DSPs and ASIPs.

2.5.3 ASIC

Multiple ASIC hardware architectures were proposed targeting real-time high definition (de)coding in accordance to the latest video coding standards trying to reduce the total energy consumption for embedded devices. The architecture presented in

(CHEN, CHEN, *et al.*, 2009) delivers H.264 encoding for D1 (720x480) resolution with 43.5-67.3 mW consumption. In (LIN, LI, *et al.*, 2008) an H.264 video encoder able to process HD1080p sequences at 242mW is presented. (CHANG, CHEN, *et al.*, 2009) proposes a real-time 720p H.264 encoder at 183mW consumption. It implements a 3-stage pipeline, 8-pixel intra prediction parallelism and a parallelized subsampling algorithm. A 59.5 mW AVC/SVC/MVC decoder for up to 3-view HD1080p videos is presented in (CHUANG, TSUNG, *et al.*, 2010). The first complete video encoding solution for MVC encoding was presented in (DING, CHEN, *et al.*, 2010). The AVC/MVC 3D/Quad Full HDTV supports 3-views HD1080p and consumes 522 mW.

Compared to other approaches, ASIC provides high throughput and energy efficiency. Considering the state-of-the-art IC manufacturing technology, ASIC implementation is the only solution able to encode high-definition MVC for an increased number of views at real-time, as shown in the related work overview presented. Still, further optimizations are possible in relation to the presented ASIC related works. Having (DING, CHEN, *et al.*, 2010) as comparison basis, the future MVC encoding systems will require increased number of views. Moreover, in (DING, CHEN, *et al.*, 2010) single-view-based optimization techniques (such as search window reduction that seriously affects the disparity estimation) are employed leaving a high potential for multiview-aware optimizations.

2.5.4 Heterogeneous Multicore SoCs

Heterogeneous multicore architectures are also proposed targeting multimedia applications. In (KOLLIG, OSBORNE e HENRIKSSON, 2009), the proposed systems handles an ASIC HW video codec, audio codecs, VLIW processors, MIPS host CPU, DSP and other HW accelerators. The system proposed by (KONDO, OTANI, *et al.*, 2009) is composed of 2 specific accelerators (video decoder and descriptor), 1 general accelerator (MX), 3 RISC CPUs and caches. (WOO, SOHN, *et al.*, 2008) describes a 195mW mobile multimedia SoC with ARM 9, AVC/MPEG decoder, JPEG codec, fully programmable 3D engine, and multiple peripheral interfaces. The heterogeneous multicores approach is the most used in the current set-top-boxes, digital TV decoders, and smart mobile diveices. It takes advantage of some degree of flexibility along with the performance of specific accelerators.

The SoCs in current commercial mobile devices such as smartphones and tablets implement heterogeneous multicore SoCs employing processors with SIMD extensions, DSPs, ASIC codecs or hardware accelerators, and programmable embedded GPUs. Qualcomm Snapdragon S4 (QUALCOMM INC., 2011) is composed of up to 4 ARM cores, Hexagon DSPs, video coding hardware accelerators, and the Adreno embedded GPU. Nvidia Tegra 3 (NVIDIA CORP., 2012) is based on up to 4 ARM cores and employs dedicated video encoder/decoder and ULP GeForce GPU. Samsung Exynos 4 (SAMSUNG ELECTRONICS CO. LTDA., 2012) is composed of quad-core ARM processor, video/image/audio ASIC codecs and the ARM Mali GPU. Texas Instruments OMAP 5 (TEXAS INSTRUMENTS INC., 2012) employs 2 ARM Cortex-A15, 2 ARM Cortex-M4, DSPs, video/audio accelerators and, the PowerVR GPU. Note, even with efficient ARM processors, SIMD extensions, DSPs and programmable massively parallel GPUs, the embedded SoCs require ASIC codecs/acceleration units to deliver the throughput and energy efficiency for real-time high definition video encoding. To deal with multiview videos and attend performance/energy requirements on embedded battery-powered devices, these SoCs will require MVC-oriented optimizations at algorithmic and architectural (including datapath and application-aware units/memory management optimizations) levels.

2.6 Energy/Power Consumption Background

Before moving to the discussion related to energy-efficient algorithms and architectures it is necessary to understand the sources of energy consumption and how they might be reduced. Moreover, the energy consumption is directly related to the hardware implementation and only indirectly related to the algorithms. However, it is possible to design algorithms able to result in energy reduction at the hardware level by reducing computational complexity, processing time, memory access, etc.

In Figure 2.16 are represented the three main power dissipation sources for CMOS circuits using an inverter as example: leakage current (static), switching power (dynamic), and short circuit current (dynamic). Eq. (2.5) shows the total power in terms of these three components. The static power dissipation is a result of the leakage currents. Consider Figure 2.16a where the input voltage (V_I) is lower than the NMOS transistor threshold voltage (V_{TN}). In this case, an ideal inverter NMOS transistor do not conduce any current. However, real MOSFET transistors cannot completely block this current, the so called leakage current. The closer V_I is to V_{TN}, the stronger the leakage. The same happens to PMOS transistors when a $V_I > V_{TP}$ is applied to the gate (Figure 2.16b). The leakage power for the case represented in Figure 2.16b is calculated by Eq. (2.6).

The dynamic power is composed of two components the switching power (Figure 2.16c) and the short circuit power (Figure 2.16d). Eq. (2.7) defines the switching power that linearly depends on the load capacitance (C_L, that depends on the fanout of the device), the source voltage V_{DD}, the frequency of operation (f), and the frequency of switching of that device (α). It represents the energy that is charged in the load capacitance and later drained to the ground. Note that only after two switches the energy is actually drained, in the first time instant (shown in Figure 2.16c) the capacitance CL is charged and in the second time instant (after another switch) the energy is drained from C_L to ground. It justify the ½ factor in Eq. (2.7). The short circuit current happens while the input signal changes V_{DD}-GND or GND-V_{DD}. There is a given input voltage where both PMOS and NMOS transistors are conducing and a current is drained directly from V_{DD} to the ground. It is depicted by the current in Figure 2.16d and the short circuit power is defined by Eq. (2.8). The total energy drained is the total power along the time (t) as represented in Eq. (2.9). Other power dissipation sources (such as gate leakage) exist in the CMOS devices but they are omitted in this short overview for simplicity reasons.

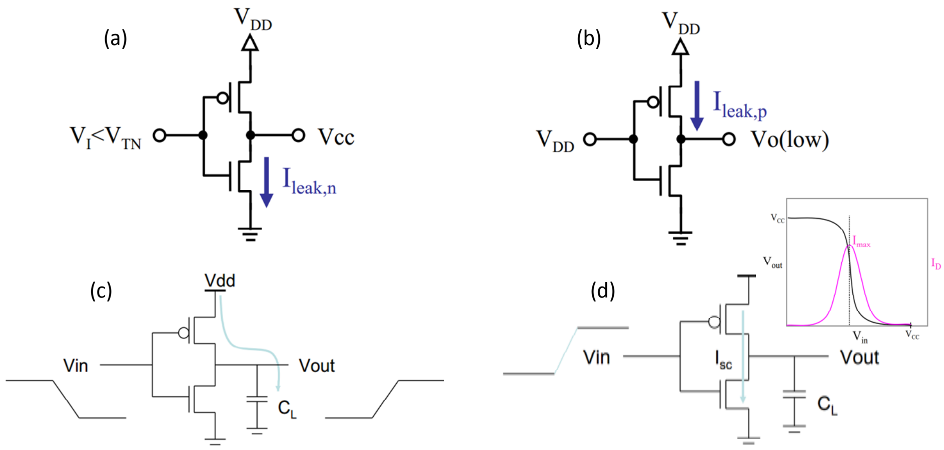


Figure 2.16: Energy/Power Dissipation Sources

As can be seen from this overview it is possible to reduce both static and dynamic power. For instance, reducing the computation reduces the dynamic power once α is reduced. If frequency scaling is used, *f* is also reduced. Moreover, if voltage scaling the dynamic energy is reduced in a quadratic order because V_{DD} is reduced. For leakage reduction, circuits featuring multiple thresholds are used. Hardware support is required, however, application knowledge and energy-aware control algorithms are required to accurately control thresholds, frequency, and voltage.

$$P_{Total} = P_{Leak} + P_{Switch} + P_{Short} \tag{2.5}$$

$$P_{Leak} = I_{Leak} * V_{DD} \tag{2.6}$$

$$P_{Switch} = \frac{1}{2}\alpha * f * C_L * V_{DD}^2 \tag{2.7}$$

$$P_{Short} = I_{Short} * V_{DD} \tag{2.8}$$

$$E_{Total} = P_{Total} * t \tag{2.9}$$

2.7 Energy-Efficient Algorithms for Multiview Video Coding

2.7.1 Energy-Efficient Mode Decision

The mode decision is one of the main contributors for the MVC high complexity and consequent energy consumption. The optimal solution using the exhaustive RDO-MD requires the evaluation of all possible inter-prediction and intra-prediction modes defined by the standard. Such solution is not feasible for real-world implementations. Thus, there is a need to reduce the number of evaluated modes during the coding process. Statically defining modes to be tested does not perform well due to changing coding parameters and video input characteristics. For this reason, it is necessary to dynamically define the most probable coding modes using the run-time available data. Figure 2.17 shows a hypothetical fast MD scheme which selects a few candidate modes out of all possible modes. Current solutions, as detailed along this section, use information extracted from the video content (texture, edges, brightness), coding mode history, video geometry, etc.

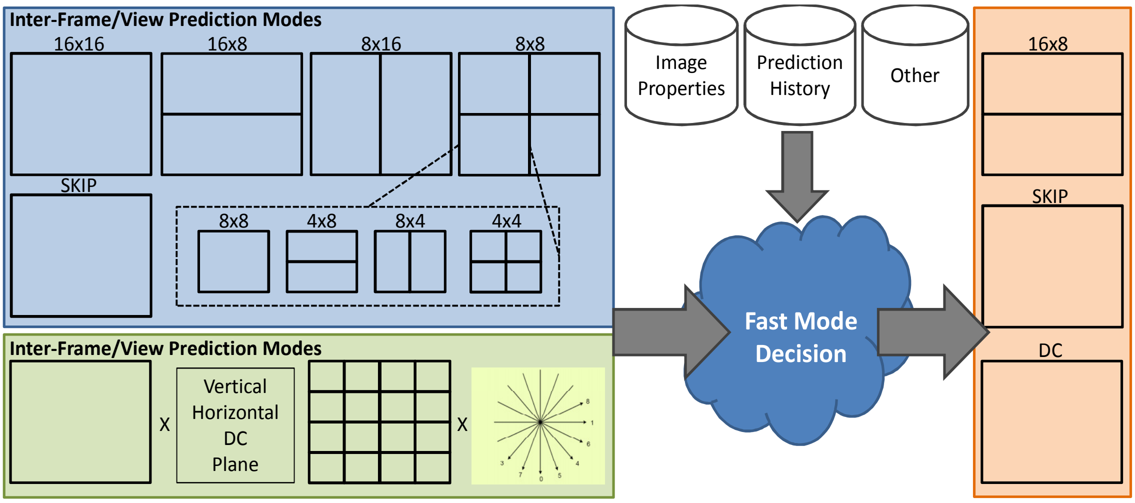


Figure 2.17: Fast mode decision example

Several fast MD schemes have been proposed to reduce single-view H.264 computational complexity, such as fast I-MB MD (PARK e SONG, 2006)(PAN, LIN, *et al.*, 2005)(MENG, AU, *et al.*, 2003)(KIM e LEE, 2011)(DE-FRUTOS-LÓPEZ, M., *et al.*, 2010)(WEI, NGAN e LI, 2008), fast SKIP MD (JEON e LEE, 2003), fast P-MB MD (JING e CHAU, 2004)(GRECOS e YANG, 2005) (LIM, 2003)(ARSURA, DEL VECCHIO, *et al.*, 2005)(LU, TOURAPIS, *et al.*, 2005)(KO, YOO e SOHN, 2009)(SALGADO e NIETO, 2006)(KIM e CHO, 2007)(WANG, SUN, *et al.*, 2007)(YU, 2004)(PARK e CAPSON, 2008), and the combination of the above (JEON e LEE, 2003) (ARSURA, DEL VECCHIO, *et al.*, 2005)(LU, TOURAPIS, *et al.*, 2005)(KO, YOO e SOHN, 2009). These fast mode decisions of H.264 may be deployed for MVC. However, they will perform inefficiently for the non-anchor frames as they do not exploit the inter-view correlation and the knowledge of GDV.

Recently, multiple fast MD schemes have been proposed for MVC (PENG, JIANG, *et al.*, 2008)(LEE, SHIN e CHUNG, 2008)(HAN e LEE, 2008)(SHEN, YAN, *et al.*, 2009)(SHEN, LIU, *et al.*, 2009) (DING, TSUNG, *et al.*, 2008) (SHEN, LIUA, *et al.*, 2010)(ZENG, MA e CAI, 2011)(CHAN e TANG, 2012) considering the GDV, camera geometrical properties, inter-view correlation and early SKIP prediction.

The authors in (LEE, SHIN e CHUNG, 2008) proposed an object-based mode decision that uses image segmentation to evaluate different prediction modes for foreground and background regions. The image is segmented using a motion based approach, considering the vectors size and the SAD in relation to the collocated block (in the same relative position). In case the motion vector significantly defers from the vector average (respecting a threshold) and the SAD exceeds a given value the regions is considered as a foreground object; otherwise it is a background. A regions growing is used to merge the foreground objects. The foreground regions are coded using DE, while the background are coded using ME. The boundary MBs are coded using the exhaustive RDO-MD.

A fast mode decision based on GDV is presented in (HAN e LEE, 2008). In this scheme the base view – encoded using exhaustive RDO-MD – is used to segment other views in foreground and background regions. The coding modes of the base view are used to classify the image regions. SKIP and Inter 16x16 MBs are defined as background while the remaining modes are considered foreground objects. AS the objects present displacement between views, the GDV is used to displace the classified regions as well. Finally, the foreground regions are encoded using exhaustive RDO-MD and the background regions are encoded using big block sizes.

The fast mode decision scheme of (SHEN, YAN, *et al.*, 2009) considers the information of reference view to classify the current MB in three complexity classes. For that, the authors propose a mode complexity metric (MDC) defined as the sum of each mode complexity in a 3x3 MBs window. SKIP and Inter 16x16 have "0" complexity, Inter 16x8 and 8x16 have "1" complexity, Inter 8x8 (or smaller) and Intra have "2" and "3" complexity values, respectively. If the MDC is smaller than a given threshold (T0), that regions is classified as *simple*. In the opposite, if MDC exceeds another threshold (T1>T0) it is classified as *complex*. Regions presenting MDC between these thresholds are defined as *medium* complexity. The *simple* regions test only Inter 16x16 mode. *Medium* regions evaluate Inter 16x16, 16x8 and 8x16 modes. *Complex* MBs are encoded using the exhaustive RDO-MD.

In (ZENG, MA e CAI, 2011) a fast mode decision approach is proposed based on the classification of the current MB according to its motion activity based on the coding modes of the base view. Firstly, the five motion-activity classes are defined in relation to the coding modes. SKIP belong to the motionless class (1). Slow motion class (2) is defined for SKIP and ME 16x16. ME 16x8 and 8x16 are considered Moderate Motion (3). Fast motion regions (4) are defined by ME 8x8 or smaller. Finally, DE and Intra define High-texture with fast motion or scene cuts (5). The Mode Correlation-Based Mode Decision (MCMD) metric is defined and calculated using the 3x3 collocated MB window. Within this 3x3 neighboring MBs, each neighbor MB has an offline defined weight. This MCMD metric is used to classify the current MB motion activity in one of the classes described above. Independent of the motion-activity, the SKIP mode is firstly evaluated and an early termination test is employed. If the SKIP prediction was not effective other modes are evaluated according to the motion class. The same classification described above is used here. For instance, A slow motion MB evaluates only the ME 16x16.

The work proposed in (CHAN e TANG, 2012) exploits the statistical behavior of the RDCost for the different coding modes along with the motion vectors difference in order to speed up the MVC encoding. In this solution, an interactive mode decision is employed. Based on statistical knowledge showing the ME is used more frequently than DE, the first interaction evaluates only the ME modes (all sizes). If the ME-based prediction is not satisfactory, a second interaction is used to evaluate the ME modes. However, only the block sizes that presented better coding performance for ME are evaluated for DE in the second interaction.

State-of-the-art schemes mainly achieve the complexity reduction via fast MD. However, they do not exploit the full space of neighborhood correlation in all spatial, temporal, and view domains. These schemes deploy fixed-thresholding (HAN e LEE, 2008)(SHEN, YAN, *et al.*, 2009) and, consequently, are unable to react to the changing QPs (i.e. changing bitrates). Moreover, in their worst case, state-of-the-art schemes - like (HAN e LEE, 2008)(SHEN, YAN, *et al.*, 2009) - check all prediction modes, thus falling back to the exhaustive RDO-MD. As a result, these schemes provide limited complexity reduction.

In general, state-of-the-art schemes consider reference view encoded using the exhaustive RDO-MD and employ their fast MD scheme on the other views. These schemes prioritize the frames from the base view and the encoded quality of other views rely on the one encoded using exhaustive RDO-MD. This might lead to meaningful prediction error increase for the last views.

2.7.2 Energy-Efficient Motion and Disparity Estimation

To find a single optimal/good matching block the ME/DE performs several block-matching operations in multiple reference frames. Additionally, this search is replicated for multiple block sizes defined by the MVC standard. However, there are search directions (ME or DE), reference frames and, reference regions that are highly unlikely to provide a good matching. Also, there are sub-optimal points that provide similar results at the cost of much reduced searching effort. See the example in Figure 2.18a. A good matching for the diamond object is available in just one of the four reference frames, the past temporal reference. In the future temporal reference the diamond is partially occluded by a second object (rectangle). In the disparity references the diamond is either occluded or out of the captured scene. In this scenario there is no need to perform searches in all reference frames, resulting in complexity/energy reduction. Other example is depicted in Figure 2.18b. Note that the previously encoded neighboring MBs share a similar motion/disparity vector since they belong to the same object. Therefore, the current MB, which also belongs to the same object, is very likely to share a similar vector. This knowledge may be used to reduce the number of search operations by reducing the number of candidate blocks. A wide range of techniques to reduce the ME/DE complexity is available as presented in the following.

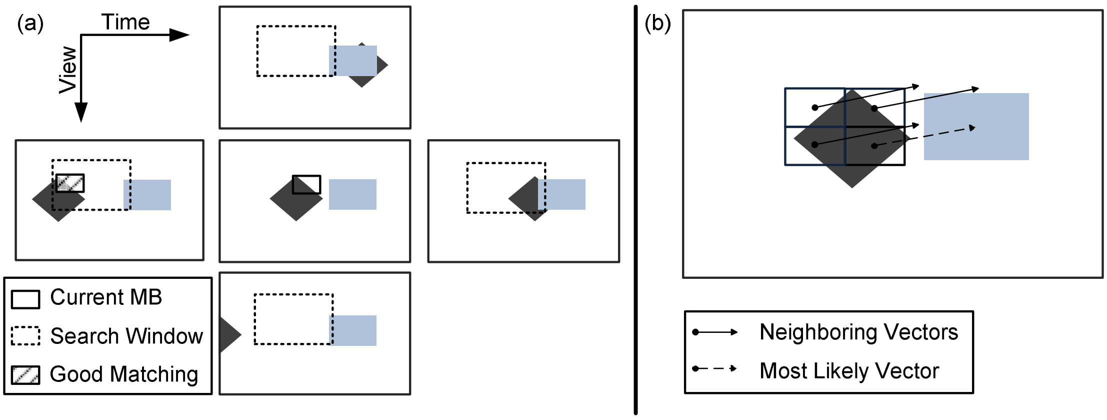


Figure 2.18: ME/DE search conceptual example

State-of-the-art fast ME/DE algorithms employ variable search range based on disparity maps (XU e HE, 2008) taking into account the distinct behavior between ME and DE. The work presents an study on how the search window size impacts in the coding efficiency showing the importance of big search windows. However, disparity maps show that it is possible reduce the effective search window by monitoring the disparity maps. From the disparity maps two parameters named vertical and horizontal scales (VS, HS) are defined. From the parameters the search window is reduced or increased in an asymmetric way, i.e., the search window may assume rectangular shapes. The increase and reduction are done in a factor 2.

In (KIM, KIM e SOHN, 2007) two strategies are used to predict motion and disparity vectors. One vector predictor the traditional spatial median predictor from upper, left and upper right neighboring MBs. The other predictor used the camera geometry and vectors from previously encoded frames to estimate the current vectors. The difference between the two predicted vector is used to calculate the search window size. A small difference means accurate predictors and a small search window is required. Otherwise, for big differences a larger window is needed.

A fast direction prediction (ME or DE) based on the blocks motion intensity is proposed in (LIN e TANG, 2009). It exploits the inter-view correlation to predict a search direction for reducing the ME/DE complexity. The base view is encoded using the ME and the frame regions are classified as slow motion if the SAD is smaller than a threshold. Similarly, the anchor frames of all views are classified according to this strategy. For each MB to be encoded, the collocated MBs from base view and anchor frame are tested. In case both are slow motion, the current MB probably is also a slow motion MB and will be encoded using ME only. If only the base view collocated MB is not slow motion, DE is employed. Other cases require ME and DE processing.

The schemes in (HAN e LEE, 2008)(DING, TSUNG, *et al.*, 2008)(DENG, JIA, *et al.*, 2009) exploit the information from the base view and classify MBs into foreground and background regions. In (DING, TSUNG, *et al.*, 2008) a fast ME based on complete DE is proposed. The DE is used to locate the correlated MB in the base view. After that, the coding information extracted from the base view is used to predict the motion vectors and partition sizes for the current MB.

The view-temporal correlation is exploited in (DENG, JIA, *et al.*, 2009) by using the motion information of the first encoded view in order to reduce the computational complexity of further views. Additionally, disparity vectors from anchor frames are also taken into consideration. Using the geometric relation between the vector from base view and anchor frames, the authors predict the motion and disparity vectors that are used as search start point. A 2x2 refinement is applied around the predicted point. This process is repeated for each search direction.

The inter-view correlation is also evaluated in (SHEN, LIU, *et al.*, 2009) (SHEN, LIU, *et al.*, 2010) to reduce ME/DE search window. The so called motion homogeneity is calculated using the collocated motion field from previous frames. If the MB presents a complex motion (homogeneity higher than a threshold) the complete search window is used for searching. Homogeneous motion MBs use a search window reduced in a factor of 4, 1/4 of vertical size and 1/4 of horizontal size. For the intermediate case, the search window is reduced in a factor of 2. Simultaneously, this solution employs a search direction selection. Homogeneous regions employ only ME search while complex motion regions employ both ME and DE. Moderate motion regions use the RDCost information to enable DE search.

Algorithm and architecture for disparity estimation with mini-census adaptive support is proposed in (CHANG, TSAI, *et al.*, 2010). A minicensus transform is applied over a pair of frames in two neighboring views to define a matching cost at pixel level. Weights are additionally generated using color distance. The cost and weights are aggregated to find the best disparity between the pair of frames. According to the authors, the two-pass strategy reduces the complexity if compared to a direct approach. The architecture proposed is discussed in Section 2.9.

The main drawback of these fast ME/DE algorithms resides in the fact that they do not exploit the full potential of the 3D-Neighborhood correlation available in spatial, temporal and disparity domains. Moreover, even the more sophisticated techniques are dependent on the complete first view encoding. However, it does not scale well for a large number of views as the prediction quality degrades in a hierarchical prediction structure. By encoding one view, the motion field information can be extracted but not the disparity field information (as no inter-view prediction is performed in this case). Therefore, it potentially limits the speedup of disparity estimation. Additionally, most of the techniques use fixed thresholding, thus perform inefficient under varying Quantization Parameters (QPs).

2.8 Video Quality on Energy-Efficient Multiview Video Coding

Techniques to reduce the complexity and energy consumption of the video encoder (such as fast mode decision and motion/disparity estimation) typically lead to video quality losses. To control the quality losses rate control methods may be employed through QP adaptation. Several rate control schemes are found in the current literature. Mostly they are developed targeting single-view encoders such as H.264. Recently, a few works specific to the MVC standard have been proposed focusing on frame- and BU-level RC. In this section we present an overview of the state-of-the-art on rate control.

In the single-view domain the majority of recent proposals are extensions of the RC implemented in the H.264 reference software that employs a quadratic model for MAD (Mean Absolute Differences) distortion prediction (LI, PAN, *et al.*, 2003). However, the quadratic model leads to limited control performance, as discussed in (TIAN, SUN, *et al.*, 2010). Aware of this limitation, the authors in (JIANG, YI e LING, 2004) and (MERRITT e VANAM, 2007) propose improved MAD prediction techniques. The scheme presented in (KWON, SHEN e KUO, 2007) implements both distortion and rate prediction models while in (MA, GAO e LU, 2005) the RC exploits rate-distortion optimization models. A RC based on a PID (proportional–integral–derivative control) feedback controller is presented in (ZHOU, SUN, *et al.*, 2011). In (WU e SU, 2009), a RC scheme for encoding H.264 traffic surveillance videos using regions of interest (RoI) to highlight regions that contain significant information is proposed. In (AGRAFIOTIS, BULL, *et al.*, 2006), RoI is used to highlight preset regions of interests using priority levels. However, single-view approaches do not fully consider the correlation available in the spatial, temporal and view domains and, consequently, cannot efficiently predict the bit allocation or distortion resulting in inefficient RC performance.

The early RC proposals targeting the MVC encoder are based on simple extension of single-view approaches (LI, PAN, *et al.*, 2003) and are still unable to fully exploit multiview properties. Novel solutions, however, have been proposed and most of them are limited to frame-level actuation. The solution in (YAN, AN, *et al.*, 2009) uses an improved MAD prediction that differentiates the frame types. Intra frames, P and B frames with only temporal prediction, P and B frames with only disparity prediction, and B frames with both temporal and disparity prediction, features distinct MAD prediction equations. Once the MAD is predicted, the target bitrate is predicted for the GGOP, refined to the GOP and finally defined for each frame. An appropriate QP for each frame is defined base on the target bitrate. This work is extended in (YAN, SHEN, *et al.*, 2009) by employing a technique to define the first QP in the GGOP; it is used to encode the I-frame. But these solutions are unable to properly handle the complex Hierarchical Bi-Prediction (HBP) structure of MVC limiting the number of input samples and the rate control learning.

The authors of (XU, KWONG, *et al.*, 2011) define an pyramid-based priority structure extracted from the MVC HBP. The higher pyramid levels are used as reference to encode lower pyramid level, e.g., I and P frames belong to the highest level, B frames that refer to I and P frames belong to the second highest level and so on. The higher levels are prioritized and are encoded using lower QPs (high quality) in order to reduce error propagation. This solution, however, considers a fixed HBP structure and do not exploit the inter-GOP correlation.

To deal with distinct image regions within a frame there is a need for a BU-level RC. Moreover, in order to find a global optimal solution, a joint frame- and BU-level rate control scheme must be designed. Recent works have proposed solutions for the BU-level RC in MVC. In (PARK e SIM, 2009) is presented a solution that deals with the frame-level and Macroblock(or BU)-level rate control. Firstly, the rate for each view is calculated based on weight parameters defined by the user. After that, the QP for each GOp is defined using the traditional H.264-based approach (LI, PAN, *et al.*, 2003) followed by a QP refinement for each frame. The frame-level QP definition considers the HBP coding structure to prioritize frames in higher hierarchical levels. A MAD-based strategy is used to calculate the target bitrate at MB level and a rate-distortion model (not described in the paper) is employed to define the QP for each MB.

The authors in (LEE e LAI, 2011) consider the Human Visual System (HVS) properties to propose a BU-level rate control solution that prioritizes the regions that are visually more important to the observer. For that, they define regions of interest using the Just-noticeable difference (LIU, LIN, *et al.*, 2010) metric along with luminance difference and edge information. Depending on the relation between these metrics, the QP is increased or decreased in relation to the initial QP (maximum QP in the neighborhood). However, this solution does not employ feedback-based control and just considers the coding information from one reference frame.

Generally, the available rate control techniques cannot fully exploit the correlation potential available in the spatial, temporal and view domains of MVC. In addition, they are unable to adapt to multiple HBP structure and cannot employ the inter-GOP periodic behavior for RC optimization. Moreover, at the best of our knowledge, no work has proposed a Rate Control scheme for MVC able to jointly consider frame- and BU-level in a hierarchical and integrated fashion.

2.8.1 Control Techniques Background

In this section are presented the background concepts required to understand the rate control solution proposed in this thesis. Firstly, are presented the control theory basics behind the Model Predictive Control (MPC) used for the frame-level RC. On the following, we present the statistical foundation supporting the Markov Decision Process (MDP) that is implemented in our BU-level RC. Finally, the concepts related to Reinforcement Learning (RL) are introduced.

2.8.1.1 Model Predictive Control (MPC)

The control theory is a subfield of mathematics originated in engineering to deal with influences in the behavior of dynamic systems (TATJEWSKI, 2010). Several control methods have been proposed ranging from very general to application-specific solutions to cope with a wide range of applications. Control problems specifications may significantly vary and the selected control method must ensure the stability of the given system. Thus, the selection of a control method for a given dynamic system may be very challenging. In case the controller does not fit the system it may compromise the stability of the entire system.

Among state-of-the-art control methods, the MPC has gained prominence by being able to accurately predict and actuate on a dynamic multivariable systems. It represents not a single control algorithm but a controller design scheme applicable to distinct systems including: continuous or discrete in time, linear or nonlinear, integrated or distributed systems. MPC outperforms conventional feedback controllers (like PID) by keeping explicit integration of input and state constraints while considering state space constrains. Also, MPC can dynamically adapt to new contexts by employing rolling input and output horizons (see more details below).

The main goal of the MPC is to define the optimal sequence of actions to lead the system to a desired and safe state by considering the system feedback to previous states and previously taken actions (see conceptual MPC behavior in Figure 2.19). To define this sequence of actions the MPC minimizes the performance function presented in Eq. (2.10). It minimizes the cost by defining a set of outputs *y* based upon a set of inputs *u*. Where *u[k + i – 1|k], i = {1,... , m}* denotes the set of process inputs with respect to which the optimization is performed; *u* is known as the control horizon or input horizon in the MPC theory. Analogous, *y[k + 1|k], i = {1,... , p}* is the set of outputs, named prediction horizon or output horizon (see Figure 2.19). The control horizon determines the number of actions to find. The prediction horizon determines how far the behavior of the system is predicted. *m* and *p* are the size of control/input and prediction/output horizons, respectively. *m* is the number of measured outputs (history size) used for the optimization process while *p* defines how many outputs are predicted; that is, how many future actions are considered in the optimization processes. *k* is the horizons index and represents the *k-th* input/output horizon. y^{SP} defines the output set point that limits the prediction horizon.

$$\min_{u[k|k]\ldots u[k+p-1|k]} \sum_{i=1}^{p} w_i (y[k+1|k] - y^{sp})^2 + \sum_{i=1}^{m} r_i \Delta u[k+i-1|k]^2 \tag{2.10}$$

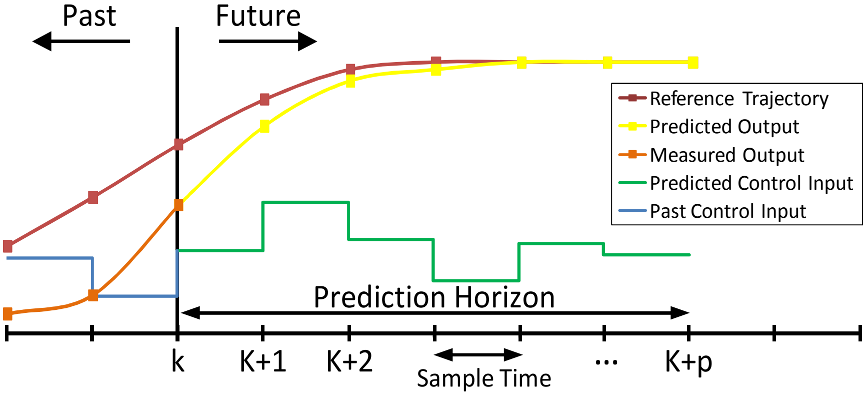


Figure 2.19: Model Predictive Control (MPC) conceptual behavior

2.8.1.2 *Markov Decision Process (MDP)*

The Markov Decision Process (MDP) is a mathematical decision-maker framework for systems that outcome partly random and partly controlled by a decision maker (ARAPOSTATHIS, KUMAR e TANGIRALA, 2003). MDP is a time discrete stochastic control process based on the extension of Markov Chains that adds the concepts of actions and rewards.

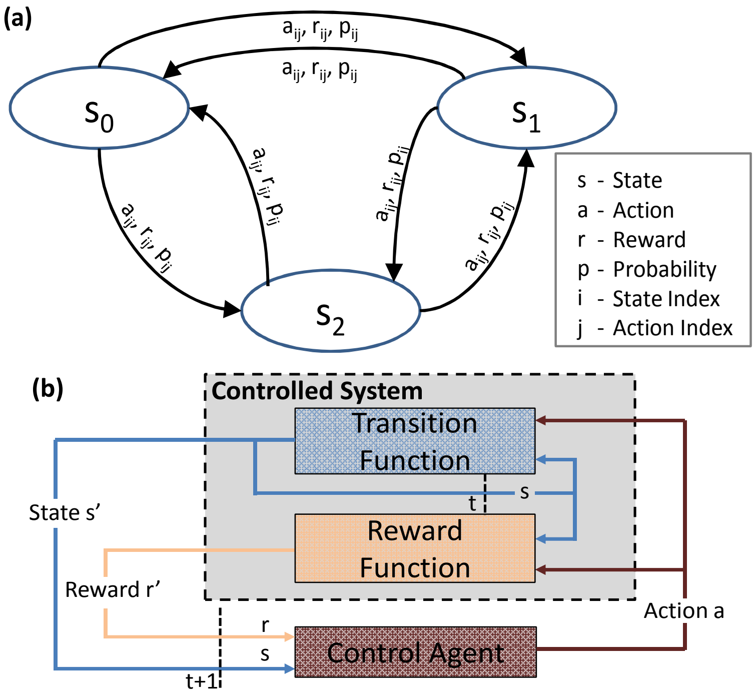


Figure 2.20: Markov Decision Process (MDP)

The symbolic representation of the MDP is a state machine or an automaton, as depicted in Figure 2.20a, which evolves in response to the occurrence of events. It is formally defined by 4-tuples *(S,A,P(.,.),R(.,.))* composed by a finite set of states $S=\{s_0, s_1,\ldots\}$, actions $A=\{a_0, a_1,\ldots\}$, rewards $R=\{r_0, r_1,\ldots\}$ and transition probabilities $\mathrm{P}=\{p_0, p_1,\ldots\}$. The *S* includes all possible states assumed by the controlled system, actions *A* are the possible acts to be taken by the decision-maker in face of a given system state. *P(S)* is the probability distribution of transitions between system states and, finally, *R(S)* is the reward related to a given action for a given state. At each discrete time step *t* the process lays in a state $s \in S$ and the decision maker may choose any action $a \in A$ that will lead the process to a new state $s' \in S$ providing a shared reward $R_{at}(s,s')$, as shown in Figure 2.20b. The rewards are used by the decision maker in order to find an action that maximizes, for a given policy, the total accumulated reward, as shown in Eq. (2.11) (where $0 \leq \gamma \geq 1$ denotes the discount factor).

$$\sum_{t=0}^{\infty} \gamma^{t} R_{at}(s_{t}, s_{t+1}) \tag{2.11}$$

By definition, the Markov process is considered a controlled Markov process if the transition probabilities *P(S)* can be affected by an action. Eq. (2.12) defines the probability P_a that an action *a* in the state *s* at time *t* will lead to state *s'* at time *t+1*.

$$\mathrm{P_a} = \mathrm{R_{at}}(\mathrm{s}_{t+1} = \mathrm{s}' | \mathrm{s} = \mathrm{s}_{t,}, \mathrm{a_t} = \mathrm{a}\,) \tag{2.12}$$

Multiple extensions have been proposed to the MDP in order to best fit to distinct problem classes. For systems where the transitions probabilities or the rewards are unknown *a priori*, the Reinforcement Learning method may be applied to solve the MDP, as detailed in the following section.

2.8.1.3 Reinforcement Learning (RL)

Reinforcement learning model is an agent to improve autonomous systems performance through trial and error by learning from previous experiences instead from specialists (BARTO, 1994), that is, the agent learns from the consequences of actions. In reinforcement learning model the agent is linked to the system to observe its behavior and take actions. RL theory is based on the Law of Effect, that is, if an action leads to a satisfactory state the tendency to produce this action increases. For each discrete time step *t* the RL agent receives the system state $s \in S$ and rewards *R(S)* to take an action $a \in A$ that maximizes the reward $R_{at}(s,s')$. This action may lead the system to a new state $s' \in S$ and produce a system output, in terms of a scalar reinforcement value, used to define the new reward $R_{a(t+1)}(s,s')$ according to Eq. (2.13). The general representation of reinforcement learning value given by *RL* in Eq. (2.14), where *U* denotes the function that changes the system state from *s* to *s'* and h_R denotes the history of reinforcement learning.

$$RL_{\mathrm{a}(t+1)}\left(\mathrm{s},\mathrm{s}'\right) = RL_{\mathrm{at}}\left(\mathrm{s},\mathrm{s}'\right) + RL \qquad (2.13)$$

$$\mathrm{RL} = \mathrm{U}\left(\mathrm{s},\mathrm{s}'\right) + \mathrm{h}_{\mathrm{R}} \qquad (2.14)$$

2.8.1.4 Region of Interest (RoI)

Within a video frame there may exist multiple regions or objects with distinct image properties and distinct importance for the observer. The image regions that are considered, for some reason, more important are called Regions of Interest. In this thesis, we consider all regions of semantically equal importance leaving space for application specific optimizations such as for 3D-surveillance, 3D-telemedicine, etc. However, at the encoding perspective, textured regions tend to have different coding properties at the mode decision and bit allocation perspectives if compared to homogeneous regions. To classify the image regions we use the variance map (Figure 2.21) to characterize the texture complexity. Variance depicts the degree of dissipation of a given population (see definition in Eq. (4.1)). In this case, how the pixels values of an image region are distributed. High variance define textured regions (represented by brighter points in Figure 2.21) while low variance define homogeneous regions (dark regions in Figure 2.21).

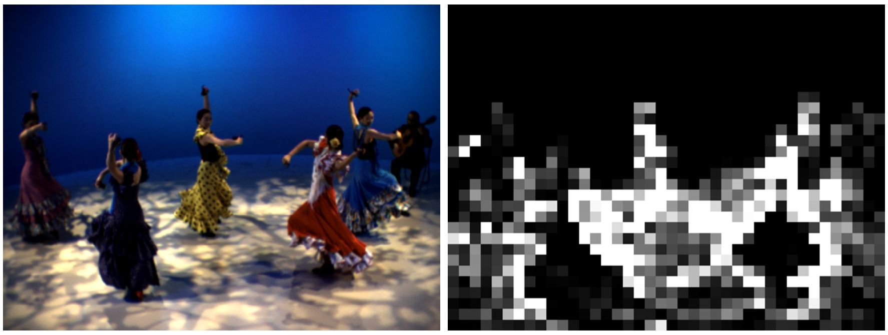


Figure 2.21: Variance-based Region of Interest map (*Flamenco2*)

2.9 Energy-Efficient Architectures for Multimedia Processing

In this section we introduce the state-of-the-art on energy management along with an overview on energy-efficient techniques and architectures for multimedia processing.

Before that, the infrastructure to support dynamic voltage scaling on SRAM memories is presented. This technique is extensively used in the literature and in some solutions proposed along this thesis.

2.9.1 SRAM Dynamic Voltage Scaling Infrastructure

The static energy due to leakage current has become a significant source of the total energy consumption in deep submicron technologies. Also, current integrated circuit footprints are dominated by embedded memories which are typically implemented as SRAM (Static Random Access Memory). Therefore, reducing SRAM static consumption is a key challenge to reach overall energy reduction.

The fabrication technology evolution has provided meaningful contribution to leakage reduction by employing high-K oxides (HUFF e GILMER, 2004), FinFET transistors (PEI, KEDZIERSKI, *et al.*, 2002), etc. Still, there is a need to further reduce the leakage at architecture and system levels through techniques such as power-gating, dynamic voltage scaling (DVS) and, dynamic power management (DPM). In the following paragraphs, we present the state-of-the-art infrastructure that enables the application of these techniques.

In (FUKANO, KUSHIDA, *et al.*, 2008) a DVS using dual power supply is used to implement a 65nm SRAM memory employing three operation modes: (i) high speed, (ii) low power and, (iii) sleep mode. In this work the low power and sleep modes are data retentive avoiding data refetching but it does not support partial DVS for specific sectors of the SRAM. (YAMAOKA, KATO, *et al.*, 2004) presents a similar solution employing three operation modes while supporting bank-level DVS. It achieves leakage reduction through adapting the virtual supply voltage using PMOS transistors. Finally, the 65nm SRAM design presented in (ZHANG, BHATTACHARYA, *et al.*, 2005) provides more flexibility through adopting multiple power states and fine grain power control. The DVS is controlled at sector level using a custom NMOS sleep transistor to control the virtual ground voltage.

Along this thesis we assume the use of an on-chip SRAM memory featuring multiple power states with data retention capabilities. The high-level memory organization is presented in Figure 2.22 along with the picture of the silicon die implementing this memory organization (ZHANG, BHATTACHARYA, *et al.*, 2005). However, to extract the potential benefit of the SRAM DVS an efficient dynamic power management is required. The overview of power/energy management techniques is presented in Section 2.9.3.

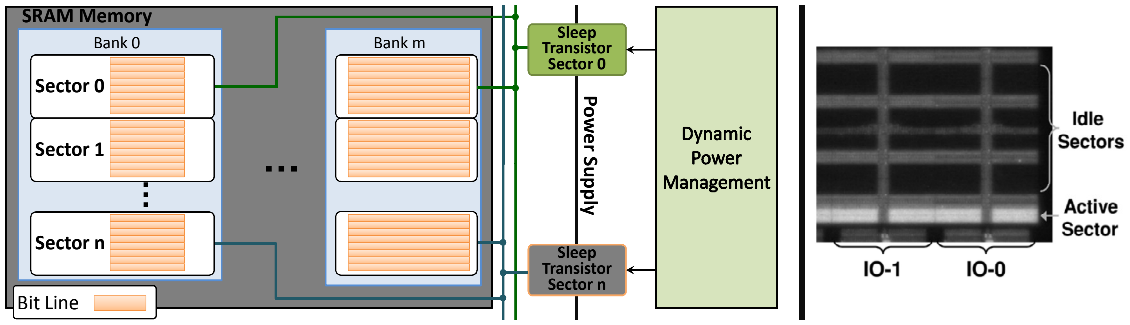


Figure 2.22: (a) SRAM voltage-scaling infrastructure and (b) silicon die picture from (ZHANG, BHATTACHARYA, *et al.*, 2005)

2.9.2 Energy Management for Multimedia Systems

Energy and power management for multimedia systems has been studied in many research works mostly targeting embedded applications. The authors in (CAO, FOO, *et al.*, 2010) employ DVS with five distinct voltage levels. It is controlled using the application-specific knowledge through workload modeling for a wavelet video encoder. In (KAMAT, 2009) a battery level-aware MPEG-4 video encoder with a notification manager and an application-specific controller is presented. Some solutions exploit the energy vs. video quality tradeoff at run time to adapt to the system scenario. (JI, CHEN, *et al.*, 2010) partitions the input data in distinct profiles used for energy budgeting generating scalable video quality according to the energy scenario. Similar work is presented in (JI, LI, *et al.*, 2009) applying game-theory algorithms to control the video encoder. (LIANG e AHMAD, 2009) proposes a rate-complexity-distortion model to progressively adjust the H.263+ encoder behavior considering the video content. It employs DVS providing and reaches up to 75% energy reduction. A power-rate-distortion model (HE, CHENG e CHEN, 2008) is used for energy minimization in video communication devices by exploring energy tradeoff between video encoding and wireless communication providing up to 50% energy reduction. A dynamic quality adjustable H.264 encoder is proposed in (CHANG, CHEN, *et al.*, 2009). It defines quality states to change the number of coding modes considering the power vs. quality tradeoff. The implemented ASIC provides real-time 720p encoding at 183mW consumption. The proposals summarized in this section are useful at the MVC scenario but lack the MVC-specific knowledge such as workload model, quality states, rate-distortion behavior, etc. Thus, the simple application of these approaches lead to inefficient energy management performance.

2.9.3 Energy-Aware Architectural Techniques

Generic techniques for reducing the on-chip SRAM leakage (like (SINGH, AGARWAL, *et al.*, 2007), (AGARWAL, NOWKA, *et al.*, 2006)) propose memories with multiple sleep modes in order to better exploit the leakage vs. wake-up penalty tradeoff. State-retentive power-gating of register files featuring multiple sleep modes is presented in (ROY, RANGANATHAN e KATKOORI, 2011). However, to control these memories an efficient power management is required. In (MONDAL e MEMIK, 2005) the hardware power-gating is controlled by monitoring the underlying hardware. These observation-based techniques may lead to miss-predictions, especially in case of sudden variations. The techniques in (LIU, SHENOY e CORNER, 2008) and (RAJAMANI, HANSON, *et al.*, 2006) consider application-knowledge for a video decoder case study, but they only exploit the knowledge at frame level. These techniques consider longer periods and may not cope with severe variations at the MB-level. Authors in (JAVED, SHAFIQUE, *et al.*, 2011) presented an adaptive pipelined MPSoC for H.264/AVC with a run-time system that exploits the knowledge of macroblock characterization based on their spatial and temporal properties (SHAFIQUE, L. BAUER e HENKEL, 2010)(SHAFIQUE, MOLKENTHIN e HENKEL, 2010) to predict the workload. Based on this knowledge, unused processors are clock-gated or power-gated. These techniques provide limited energy-efficiency in MVC as they cannot exploit the MVC-specific knowledge such as (a) distribution of memory usage at frame and MB levels, (b) memory usage correlation in the 3D-Neighborhood, (c) memories with multiple sleep modes.

2.9.4 Energy-Efficient Video Architectures

The work of (SHAFIQUE, L. BAUER e HENKEL, 2010) presents an energy budgeting scheme for the H.264 ME. This solution considers the total energy available along with the video properties to dynamically define a search pattern able to deal with the energy versus quality tradeoff. Each frame is classified into one of six energy classes and further classification refinement is performed at MB level. The highest complexity class performs a search composed of three search patterns (Octagon Star, Polygon and Diamond) without samples subsampling. The lowest complexity class employs a Diamond-shaped search using 4:1 subsampling. The highest complexity class requires 17x more energy when compared to the lowest complexity class.

The authors in (CHEN, HUANG, *et al.*, 2006) evaluated different state-of-the-art data reuse schemes (Level-A, Level-B, Level-C and Level-D) and proposed a new search window-level data reuse for H.264 ME (Level-C+) in order to reduce the energy consumption related to external memory access and on-chip memory storage. Level-A and Level-B solutions are based on candidate blocks. While Level-A fetches and stores on-chip a single candidate blocks, Level-B fetches a whole candidates stripe (inside the search window). They require frequent external memory access and only fit with regular search patterns which is not the case for state-of-the-art ME/DE algorithms. Level-C and Level-D follow the same logic but at search window level. Level-C stores one search window (avoiding the retransmission of overlapping search window regions accessed by neighboring MBs in the same line) and Level-D a search window stripe for the whole frame. Observe that Level-D requires a extremely large on chip memory for large search window or frame size. As Level-C presents a reasonable tradeoff between external memory access and on-chip memory size it was extended in Level-C+. Level-C only exploit the data reuse between horizontal neighboring MBs. Levels-C+ proposes to increase the vertical on-chip storage to include the search window of the MB line bellow. This allow exploiting the vertical data reuse at the cost of increased on-chip memory and out-of-order processing (two MB lines are processed using double-Z order).

In (WANG, TAI e CHIANG, 2009) a bandwidth-efficient H.264 ME architecture using binary search is proposed. This solution employs a frame-level preprocessing that downsamples the image twice in a factor 2. It results in three images (or three layers), the original image, the downsampled image, and the twice downsampled image. After that, a search is performed in the three layers. This technique is also modified to allow parallel processing and easy hardware implementation. A hardware architecture is presented targeting low power through low memory access, efficient hardware utilization, and low operation frequency.

An complete MVC encoder targeting low power operation is presented in (DING, CHEN, *et al.*, 2010) employing eight pipeline stages, dual CABAC and parallel MB interleaving. A cache-based solution is used for the search window reading along with a specific prefetching technique. The cache tags are formed by the frame index and x and y block position. Also, each cache entry stores an image block (instead of words like in generic caches) following the same concept proposed in (ZATT, AZEVEDO, *et al.*, 2007). The search is constrained to a [±16,±16] search window with a predicted center point. The ME/DE architecture is described in more details in the previous work from the same group (TSUNG, CHEN, *et al.*, 2009). This approach might lead to quality loss when the center point prediction is not accurate. Also, the authors ignored the fact that fast ME/DE schemes already consider this information to start the search. The MVC encoder is able to real-time encode 4 views HD720p at 317mW.

Generally, the search window-based data reuse approaches suffer from excessive leakage resulting from big on-chip SRAM memories required to store the complete rectangular search windows. This point becomes crucial for MVC as the DE requires relatively large search windows (mainly for high resolutions) such as [±96,±96] to accurately predict high disparity regions (XU e HE, 2008). In this case, even considering asymmetric search windows incurs in large on-chip storage overhead, thus suffering from significant leakage.

The authors in (SHIM e KYUNG, 2009) use multiple on-chip memories to realize a big logical memory or multiple memories (one for each reference frame) according to the frame motion. A search window centered prediction is employed for data prefetching while the size of search window is dynamically adjusted at frame level using the size of motion vectors found in previous frames. The data reuse scheme Level-C is employed.

A data-adaptive structured search window scheme is presented in (SAPONARA e FANUCCI, 2004). A adaptive window size approach is proposed considering the spatial/temporal correlation of the motion field. If the vectors of the current and past frames do not exceed a given value, there is no need to search in a region larger than this vector size and the fetching of a reduced window is necessary. In case the window is too small and the error starts do increase, a test detects it and the search window is increased regardless of the neighborhood. This solution leads to reduced external memory access but its potential for on-chip memory reduction is not discussed.

The work in (CHEN, CHEN, *et al.*, 2007) proposed a candidate-level data reuse scheme and a Four Stage Search algorithm for ME. Firstly, multiple search start points are predicted from the neighboring MBs motion activity. The predicted points are evaluated and the best one is selected for a Full Search around its position. A ladder-shaped data arrangement is also proposed in order to support random access for the proposed algorithm. The candidates parallel processing is performed using a systolic array.

In (TSAI, CHUNG, *et al.*, 2007) a caching algorithm is proposed for fast ME. Additionally, a prefetching algorithm based on search path prediction is proposed in order to reduce the number of cache misses. The work (TSAI, CHUNG, *et al.*, 2007), however, is limited to a fixed Four Step Search pattern and it does not consider disparity estimation and power-gating.

2.10Summary of Background and Related Works

The MVC is the most efficient video coding standard focusing on 3D-video coding. It is able to provide 20-50% of coding efficiency increase, if compared to H.264 simulcast, by employing inter-view prediction, the disparity estimation. Mode decision and motion and disparity estimation represent the most complex modules in the MVC encoder and bring big challenges for their real-world implementation.

The implementation of MVC encoders may exploit different multimedia processing architectural solutions. Currently, the most preeminent alternatives are multimedia processors/DSPs, reconfigurable processors, ASICs, and heterogeneous multicore SoCs. Each solutions presents positive and negative points. At the one hand, ASICs provide the highest performance and energy efficiency at the cost of no flexibility. At the other hand, multimedia processors/DSPs are totally flexible but deliver low performance and reduced energy efficiency. Heterogeneous multicore and reconfigurable processors provide tradeoff points between ASICs and processors. By employing units specialized in each kind of task, the heterogeneous multicore SoCs improve the performance in relation to multimedia processors but typically present issues related to programming and portability. Reconfigurable processors can cover this gap by employing extensible instruction set and defining, at run time, if regular or custom instructions should be used in that specific time instant. Still, these solutions is unable to met the performance and energy efficiency required for MVC encoding without application specific ASIC acceleration. Therefore, considering the current technology, a complete ASIC encoder or an heterogeneous SoCs with hardware specific accelerators seen to be the most feasible solutions for embedded mobile devices.

Multiple proposals targeting on complexity and energy reduction for the MVC are available in the current literature. These contributions are centered in two abstraction levels, the algorithmic and architectural levels. At the coding algorithms perspective, complexity reduction is most frequently addressed at the mode decision and motion and disparity estimation because they represent the most complex MVC blocks. The mode decision solutions used distinct side information in order to reduce the number of coding modes tested during the coding process. Video properties such as texture, edges, luminance, and motion/disparity activity are used to predict the most probable coding modes in each image regions. Additionally, extensive analysis has been done to learn how neighboring views and frames are correlated. This correlation also useful to predict the coding modes. As the ME/DE spends about 90% of the total encoding time, the same kind of information is used to predict the most probable motion and disparity vectors and reduce the ME/DE complexity. However, the related works do not fully exploit the correlation available within the 3D-neighborhood and perform badly under content changing scenarios. Moreover, these solutions are not developed considering the energy perspective and cannot react to battery level changing situations by dynamically adapting the complexity to the available energy.

Generally, the complexity reduction techniques lead to uncontrolled quality degradation and coding efficiency losses. The rate control becomes a key tasks in order to minimize this complexity reduction drawback. The majority of rate control solution currently available target the H.264 or are simple extensions from H.264 solutions. The few rate control algorithms designed for MVC focus only on frame-level or basic unit-level actuation levels. Additionally, theses algorithms do not use the intra- and inter-GOP bitrate correlation in the 3D-neighborhood.

At the hardware architectural perspective ME/DE is the most studied MVC coding block. The ME/DE is a processing and memory intensive task requiring massively parallel processing and efficient memory access and management. The resulting high energy consumption is mainly related to external memory access and on-chip video memory size. Diverse related works propose ME/DE processing hardware architectures, memory hierarchies and data reuse techniques. However, they share limitations related to the complexity reduction algorithms implemented (leading to quality losses), excessive external memory accesses, or large on chip-memory resulting in high. Moreover, most of the available architectures lack the ability to dynamically adapt its operation according to changing coding parameters or video content characteristics.

Therefore, there is a demand for novel and energy-efficient MVC encoding solutions able to significantly reduce energy consumption under changing video and system scenarios. For this reason, this thesis targets on jointly addressing the energy issues at algorithmic and architecture levels while sustaining the video quality.

3 MULTIVIEW VIDEO CODING ANALYSIS FOR ENERGY AND QUALITY

The Multiview Video Coding (MVC) standard brings high coding efficiency gains reducing the bitrate in 20-50% for similar video quality if compared to the H.264/AVC simulcast. The coding efficiency gains are driven by novel high-complexity coding tools that drastically increase the overall encoding processing effort and, consequently, the energy consumption. In this section an extensive analysis of the energy requirements for real-time MVC encoding and the energy consumption breakdown are presented. The goal is to provide a better comprehension on the MVC performance and energy requirements. Additionally, the requirements in terms of objective video quality are discussed in the following.

3.1 Energy Requirements for Multi View Video Coding

Encoding MVC at high definitions has shown to be an unfeasible task for mobile devices when all coding tools are implemented without energy-oriented optimizations. State-of-the-art embedded devices are unable to provide the processing performance or to supply the energy required by the MVC encoder. To demonstrate the energy-related challenges to MVC encoding a case study is presented in the following.

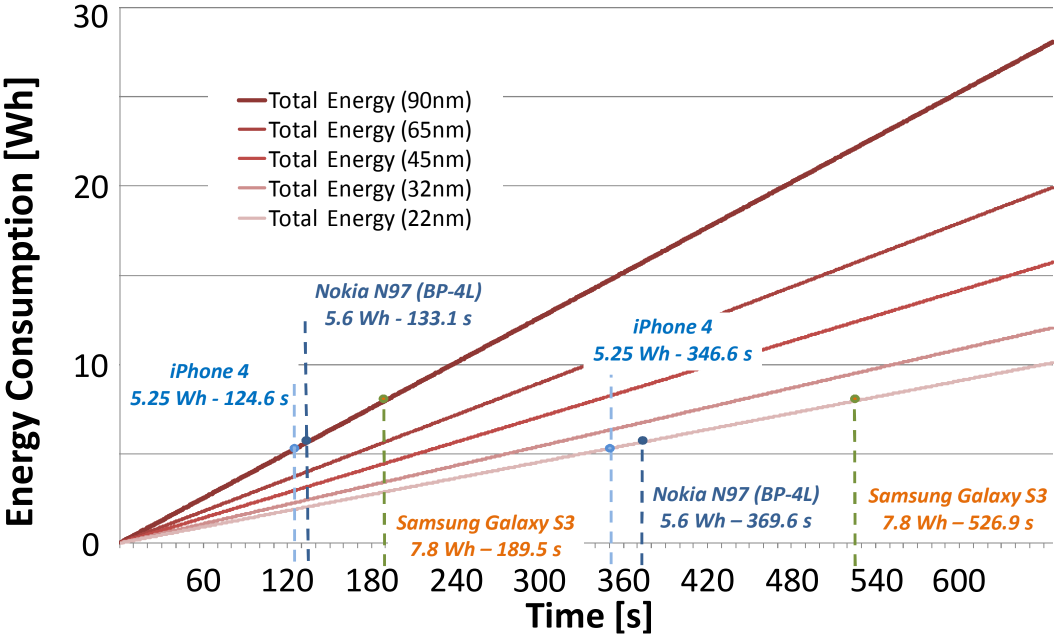


Figure 3.1: MVC energy consumption and battery life

Figure 3.1 presents the energy consumption to encode a 4-view HD1080p video sequence using the MVC encoder while considering four fabrication technologies. Also, the battery draining time for state-of-the-art smartphone batteries are presented. Note, these smartphones are unable to attend the MVC constraint. Despite the technological scaling that provides meaningful energy reduction for deep-submicron technologies, the energy consumption remains high considering embedded devices constraints. For instance, let us analyze the best-case scenario where a device is fabricated with a state-of-the-art 22nm fabrication node and features a 7.8Wh battery as available in the latest Samsung Galaxy S3 (SAMSUNG, 2012) released in Q3 2012. For a scenario where MVC encoder is the only task draining the battery, only 526.9s (8 min, 46.9s) of recording would be possible before the battery was completely drained. The presented battery life is not acceptable and does not attend market and user requirements. Meaningful energy reduction techniques are required to bring the MVC consumption to a feasible energy envelope. For this, a better understanding on the energy consumption sources is required.

Figure 3.2 demonstrates the motion and disparity estimation task (ME/DE) is responsible for about 90% of the total energy. These numbers were measured using the Orinoco(KROLIKOSKI, 2004) simulation environment and might present errors in the actual numbers. This simulation, however, is worth for a relative comparison of the energy consumed by MVC encoding tools. This numbers consider the fast search algorithm TZ Search (TANG, DAI e CAI, 2010) as search pattern. Motion compensation (2.5%), deblocking filter (2.5%) and intra-frame prediction encoder (2%) are the following in terms of energy consumption while representing less than 2.5% each. Thus, reducing ME/DE consumption is of key importance to reach energy efficiency. ME/DE consumption is directly related to the size of search window (SW), that is, the size of the region to perform the search. The increase in ME/DE search window leads to energy increase due to increased number of matching candidates and larger amount of data required (memory accesses) to perform the task. Figure 3.3 quantifies the energy consumption for five distinct sized SWs. Comparing the corner cases, a small search window [±16, ±16] to a big [±128, ±128] SW, the energy increases in a factor of 6.5x. From single-view knowledge it is possible to affirm that there is no need for using SWs larger than [±64, ±64]. However, disparity vectors tend to have larger magnitude and the ME/DE task requires increased SW to find these matching candidates. According to (XU e HE, 2008), for a good disparity estimation performance in HD1080p video sequences, the search window should be at least [±96, ±96].

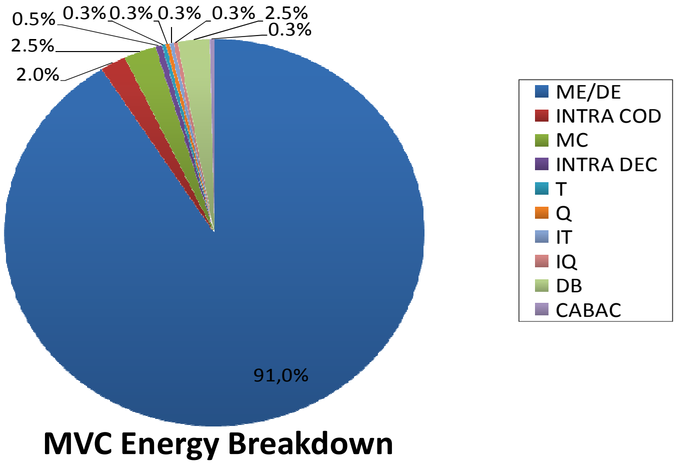


Figure 3.2: MVC component blocks energy breakdown

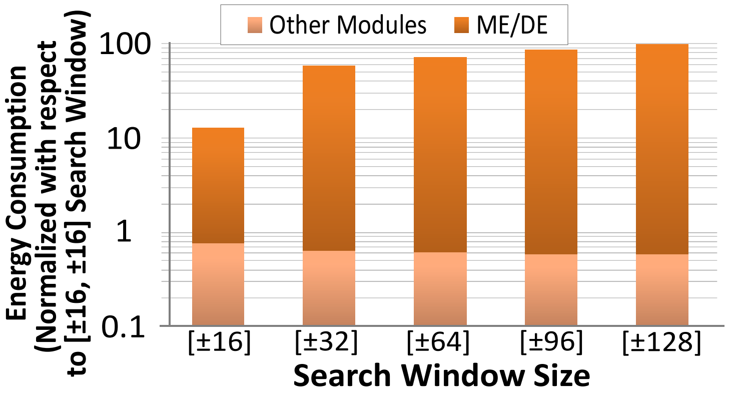


Figure 3.3: MVC energy breakdown for multiple search window sizes

Although the analysis correctly depicts some sources of energy consumption, a deeper knowledge of the application behavior is mandatory. The MVC encoder hides in the encoding process a control function that controls the complexity of each and single module discriminated in Figure 3.2. The mode decision (MD) defines how many modes are tested and how many times the ME/DE search is performed, how frequently the intra-frame encoder is used, etc. The relation between the exhaustive mode decision, the Rate-Distortion Optimized MD (RDO-MD), to the simplest possible MD that tests a single coding mode is in the order of 100x energy consumption, as shown in Figure 3.4. Obviously, the single mode MD is not used in practice under penalty of poor quality and coding efficiency results. Nevertheless, this example highlights the optimization space for energy-efficient solutions in the MD control.

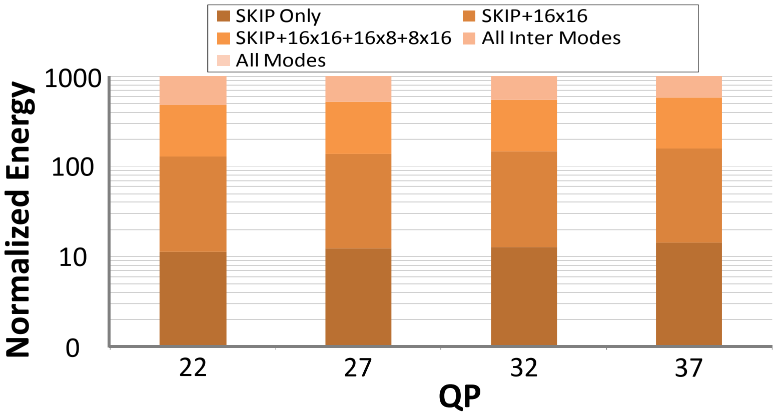


Figure 3.4. MVC energy for distinct mode decision schemes

At the architectural perspective, computation and memory (external memory access and on-chip memory storage) are the two energy consumption sources. Here, dynamic and static energies are jointly considered. As shown in Figure 3.5, the energy breakdown is composed of 90% memory-related energy consumption while 10% are represented by the computation itself. Typically, for a rectangular search window on-chip memory using Level-C data reuse (CHEN, HUANG, *et al.*, 2006), the on-chip memory energy and external memory access are evenly distributed but may vary according to design options (on-chip memory size, data reuse scheme, etc). The presented energy breakdown highlights the importance of reducing memory-related energy. Even so, the reduction of computational complexity stands as key challenge for low-energy MVC. Observe that multimedia processing applications are typically data-oriented applications and require intense memory communication. However, the complexity reduction leads to a win-win situation where both less data is processed and less memory accesses are required. Thus, complexity reduction positively impacts computation and memory energy consumptions.

The following sections discuss on how computational complexity and memory access influence the overall energy consumption in MVC and how these components are distributed among the MVC modules.

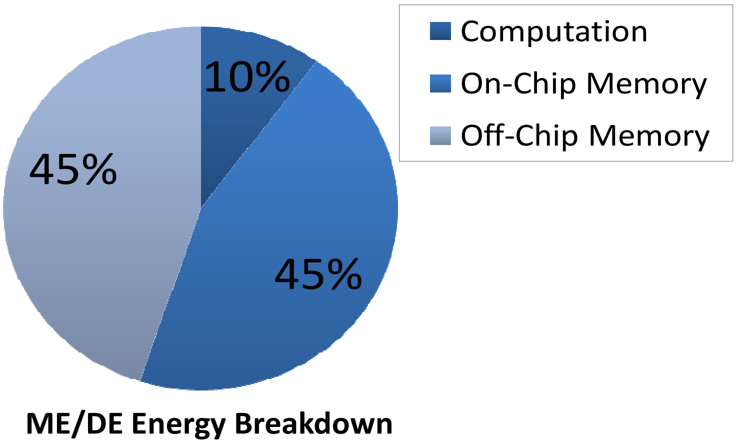


Figure 3.5: MVC energy breakdown

3.1.1 MVC Computational Complexity

The MVC high energy consumption is driven by the complexity associated to the MVC video encoder. In this section we compare the complexity in relation to previous standards and quantify the main sources of complexity within the video encoder. The experiments here presented consider the fast search algorithm TZ search for motion and disparity search.

Figure 3.6 compares the MVC encoder in three distinct scenarios compared to the H.264-based simulcast encoding. The 8-view video sequences were encoded independently (simulcast) and using inter-view prediction with one, two or four reference views (when available). Custom extensions to the JMVC (JVT, 2009) reference software were done to support more than two reference views. Although using more than two reference views is not a common practice in current encoding systems, the increase in reference views is expected for many-view systems, especially for 2D-array camera arrangements where the four surrounding neighbor views are closely correlated to the current view. The measured complexity to encode eight views using four reference views is 19x more complex than encoding a single H.264 view. Even using two reference views, as current multiview systems, the complexity exceeds in 14x the H.264 single-view complexity. To understand what this complexity represents it is important to consider that real-time H.264 encoding for HD1080p still pose interesting challenges in the embedded devices development and require application specific hardware acceleration (see discussion in Section 2.5.4). Moreover, according to (OSTERMANN, BORMANS, *et al.*, 2004) the H.264 encoder is about 10x more complex then the MPEG-4 Part 2 encoder. If compared to the simulcast encoding of eight views (8x compared to H.264 single-view), the MVC is 1.75x and 2.37x more complex for two and four reference views, respectively.

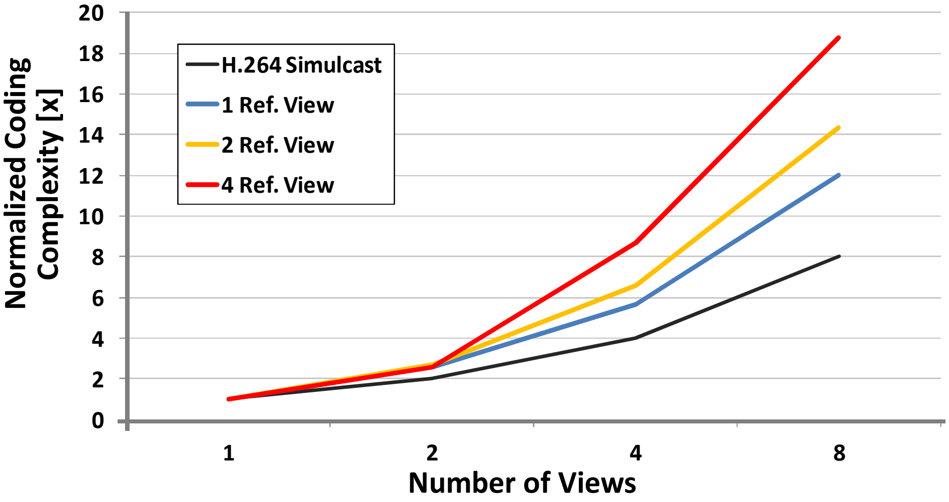


Figure 3.6: MVC vs. Simulcast computational complexity

The total encoder complexity is mainly concentrated in the motion and disparity estimation (ME/DE) unit responsible for about 90% of the total processing, as depicted in Figure 3.7. The deblocking filter (DF) and motion compensation (MC) blocks are the more complex blocks after ME/DE. The MVC encoder complexity measured from the JMVC (JVT, 2009) reference software without optimizations leads to 2 GIPS (Giga Instructions per Second) for only 4-view real-time MVC encoding at HD1080p resolution. This throughput is unfeasible even for high-end desktop computers. For instance, the latest Intel Core i7 3960X (BENNETT, 2011) processor with six physical cores running at 3.3GHz is able to provide about 180MIPS. Thus, the state-of-the-art high-end processors are orders of magnitude bellow the performance requirements for real time MVC encoding if no application/architectural optimizations are performed. The task is even more challenging for embedded processors.

For energy-efficient MVC there is a need to drastically reduce the computational complexity. Based on the presented observations, ME/DE and MD modules have the highest potential for complexity reduction and, for this reason, are explored in this work. Therefore, deep application knowledge is required to design efficient complexity reduction algorithms able to avoid objective and subjective video quality losses.

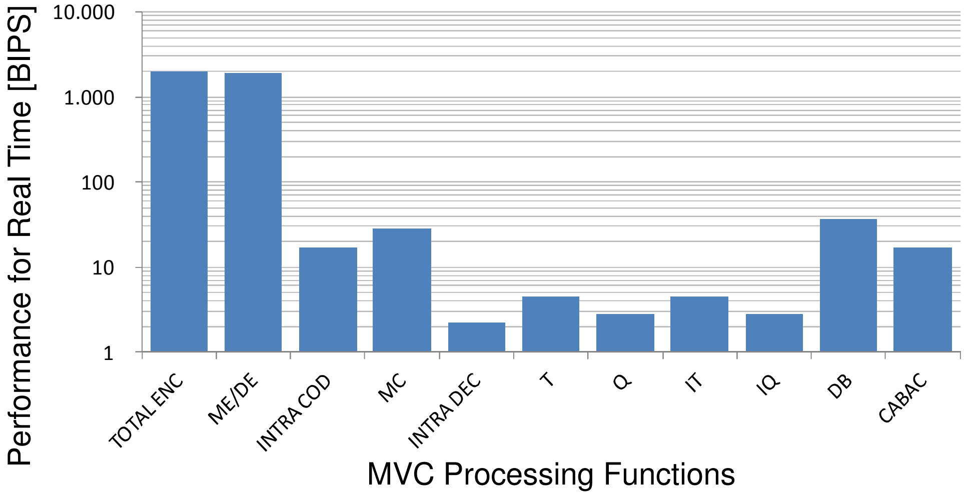


Figure 3.7: MVC computational complexity breakdown

3.1.2 MVC Memory Access

The other major component of energy consumption is related to data access. Multimedia applications are known for their data-oriented nature and consequent intense memory communication. The MVC encoder for 4-views HD1080p requires a memory bandwidth of 41 GB/s, as pointed in Figure 3.8. It can be met using GDDR5 memory interfaces available in high-end GPUs such as Nvidia GeForce GTX 690 (384 GB/s @ 300w) (NVIDIA, 2012) at the cost of high energy consumption. For embedded systems, however, the memories interfaces are limited by power constraints and deliver a reduced bandwidth. Thus, this bandwidth is not feasible for embedded devices. For instance, the Nvidia ULP GeForce embedded in Tegra 3 SoC (NVIDIA, 2012) provides a theoretical limit of 4.26 GB/s employing a LPDDR2-1066 memory interface. In this scenario, MVC encoding in embedded devices pose the need for drastically reducing the memory bandwidth through algorithmic and architectural optimizations.

In video encoding systems, mainly the MVC video encoder, the access to the DPB (Decoded Picture Buffer) is the memory bottleneck. The DPB stores all reference frames used for inter-frame and inter-view (ME/DE) prediction. The frames are written in the DPB after the DF processing and the ME/DE block reads the stored data to perform motion/disparity search. ME/DE unit is responsible for about 68% of the encoder total memory access requiring a 28GB/s memory bandwidth for 4-view encoding, as shown in Figure 3.8. The measured memory bandwidth is far higher compared to the raw video data input (355 MB/s) because the reference frame data may be requested multiple times in order to perform the motion/disparity search for distinct MBs. Aware of this behavior, multiple techniques try to reduce external memory accesses through employing on-chip video memories and data-reuse techniques. Even though effective at the external memory perspective, these solutions significantly increase the on-chip energy consumption. Therefore, energy-efficient external and on-chip memory reduction must be jointly considered at design time and at ruin time. Moreover, the complexity reduction design, discussed in Section 3.1.1, must consider the memory access behavior to optimize the overall energy consumption.

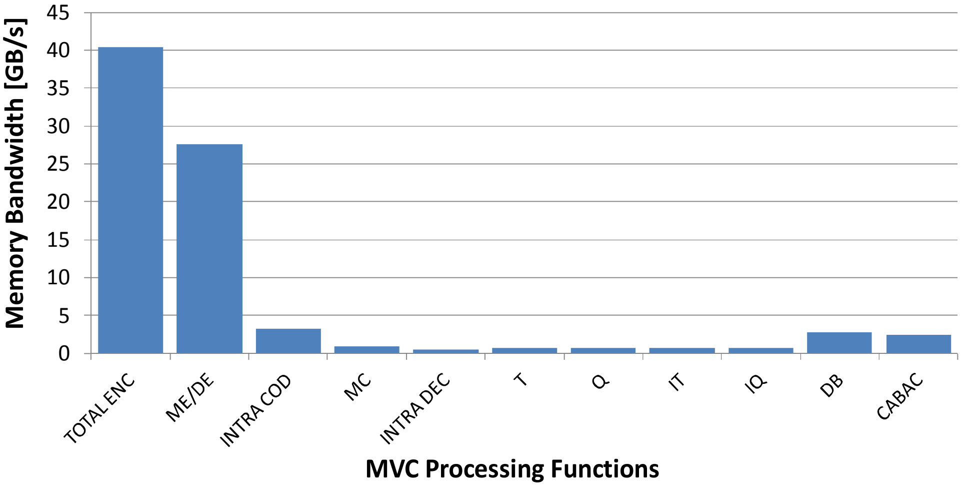


Figure 3.8: Memory bandwidth for 4-views MVC encoding

3.1.3 Adaptivity in MVC Video Encoder

The high complexity and memory requirements posed by the MVC encoder are not the only challenges related to its realization. MVC energy consumption is unevenly distributed along the time. Processing and memory energy components vary depending upon coding parameters, user's definitions, system state and, video content. These run-time variations make the MVC encoder design even more challenging. If on the one hand, an under-dimensioned encoder leads to performance issues and does not guarantee reduced energy consumption due the need of additional buffering. On the other hand, over-dimensioned encoders face under-utilization and unnecessary energy consumption.

The MVC prediction structure is a dominant factor in terms of energy variation once distinct frame types (I, P or B) present distinct processing and memory access behaviors. I frames are the lightest frames once the ME/DE (that represents 90% of the encoder complexity) is completely skipped. P frames employ ME/DE to a single direction. In this scenario the P frames search in a single reference frame. The B frames require heavy processing, in comparison to I and P frames, and intense memory access while executing ME/DE search in multiple reference frames/views. In Figure 3.9 the frame-level energy consumption for seven GGOPs is presented. Each bar represents the sum of energies spent to encode the frames from all four views that belong to the same time instant (i.e., S0Tx+S1Tx+S2Tx+S3Tx). GGOP borders (anchors), for the experimented prediction structure, have one I frame, two P frames and, one B frame (that performs only DE once there are no temporal references available), for more details see prediction structure in Figure 2.7. Consequently, GGOP borders drain reduced energy amount (1.5 Ws/frame), as shown in Figure 3.9. All other relative positions within the GGOP are composed only by B frames and the energy consumption drastically increases in comparison to GGOP borders. Typically, the center of GGOP is the energy hungriest time instant once temporal references are far and more extensive motion search is required to find a good matching. According to the experiment presented in Figure 3.9, the energy consumption may exceed 7 Ws/time instant in this case. It represents a 4.7x instantaneous energy variation within the same GGOP.

Although prediction structure-related energy variations may be easily inferred from the coding parameters, there is another important variation source that may not be easily obtained, the video content-related variations. The video content variations occur at multiple levels: (i) view level: distinct views may present distinct video content such as textures, motion and disparity behavior; (ii) frame level: video properties vary along the time; (iii) MB level: within the frame distinct regions or object may present distinct image properties. Figure 3.9 depicts the energy variations along the time. For instance, GGOP #6 (frames 41-49) drains reduced energy amount if compared to previous GGOPs due to reduced coding effort resulting from easier-to-encode video content. The MB-level variations are shown in Figure 3.10 in terms of ME memory requirements. The memory usage changes along the time depending on the video content motion intensity. High motion regions present increased memory usage in relation to low motion regions within the same frame.

Therefore, energy-efficiency in MVC encoding requires the understanding of the energy sources variations and the design of adaptive architectures able to manage, at run time, the energy consumption while considering dynamically varying parameters (such as video content) and system state.

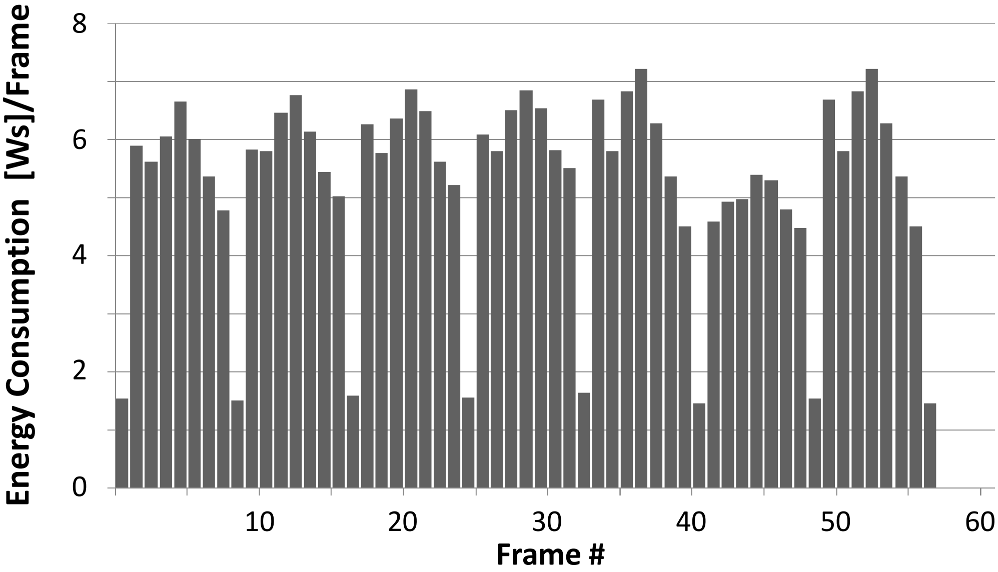


Figure 3.9: Frame-level energy consumption for MVC

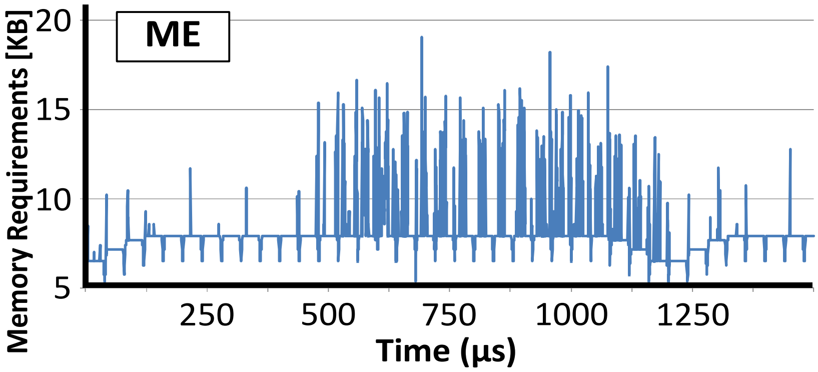


Figure 3.10: Memory requirements for motion estimation at MB-level

3.2 Energy-Related Challenges in Multiview Video Coding

The large computational complexity and intense memory communication related to MVC poses a series of challenges related to real-time encoding for high-definitions mainly at the embedded systems domain. Energy consumption represents the most challenging issue related to embedded MVC encoding. Thus, there is a dire need for energy reduction of the MVC video encoder through computational complexity and memory access reduction. Energy-efficient solutions must jointly consider optimizations at algorithmic and architectural levels. Coupling deep application knowledge to intelligent employment of low-power design techniques is a key enabler for energy-efficient embedded MVC encoder realization.

Based on the discussion presented along Section 3.1 the energy-efficient MVC requires the following optimizations at algorithmic level:

- *Energy-efficient mode decision scheme*: The MVC defines an increased optimization space for the optimal prediction mode selection leading to high complexity and energy requirements, as demonstrated in Section 3.1. An efficient fast mode decision scheme is needed to reduce the optimization space through heuristics able to accurately anticipate the coding mode selection. The neighborhood information and image/video properties may provide hints to completely avoid the evaluation of unlikely prediction modes

- *Energy-efficient motion and disparity estimation*: ME/DE is the most complex and energy hungry module in the entire MVC encoder. Intelligent optimizations in ME/DE lead to meaningful overall energy reduction. Energy-efficient ME/DE may be reached by applying ME or DE elimination, search direction elimination, motion/disparity vector anticipation, object motion/disparity field analysis, etc.
- *Dynamic complexity adaptation*: The energy-efficient MD and ME/DE can be designed considering distinct strengths in order to handle the energy versus quality tradeoff. Additionally, the MVC presents a dynamically varying behavior along the time depending on coding parameters, user's constraints and video content. An energy-aware complexity adaptation scheme must be able to predict these variations and to react at run-time through reduction/increase of complexity budget by setting MD and ME/DE parameters. The dynamic complexity adaptation scheme may also exploit asymmetric coding properties such as the binocular suppression theory (STELMACH e TAM, 1999).

The energy-efficient algorithms described above must be designed considering their impact in the architectural implementation. At architectural level, the energy-efficient solution must employ:

- *Low-energy motion and disparity estimation architecture*: The ME/DE task requires high throughput but typically allows a high level of parallelism. To attend the throughput requirements at a reasonable frequency of operation while reducing energy multiple levels of parallelism must be exploited including (i) pixel-level, (ii) MB-level, (iii) reference frame-level, (iv) frame-level and (v) view-level parallelisms. It allows operating in a reasonable range of operation frequency and voltage. The processing units should be designed to enable power-gating and/or DVS to adapt to the performance variations.
- *Energy-efficient on-chip video memory hierarchy*: Simply feeding the highly parallel ME/DE processing units while avoiding performance losses is typically a very challenging task. The on-chip video memory, however, has to deal with the high memory-related energy consumption and memory requirements variations. For that, an accurate memory sizing strategy is required. Also, the on-chip video memory must support partial power gating and/or DVS to adapt to memory requirement variations while minimizing static energy consumption.
- *Data reuse and prefetching technique*: Neighboring MBs tend to access repeated times the same data from reference frames during the ME/DE process. To avoid additional external memory access the reference data must be stored in the local memory. However, increased local memory leads to increased static energy. Hence, only the actually required data must be read from external memory and stored locally. The energy-efficient MVC solution requires a memory-friendly data reuse technique able to reduce external memory access without employing increased local memory. To avoid performance losses due to local memory misses the required data must be prefetched accordingly. Thus, it demands an accurate memory behavior predictor that understands the ME/DE search pattern. Accurate prefetching becomes even more challenging for state-of-the-art adaptive and customizable search algorithms.
- *Dynamic Power Management*: Supporting power gating and/or DVS in memory and processing units does not directly lead to energy savings. To reach energy-efficiency an intelligent dynamic power management scheme is required to define the proper power states at each given time instant. The DPM must apply deep

application knowledge including offline statistical analysis, neighborhood history and, image/video characteristics in order to accurately predict performance and memory requirements and take proper action.

Addressing each challenge related to energy-efficient MVC brings a contribution to the overall energy reduction. A balanced combination of energy-efficient techniques may lead to drastic MVC energy reduction. The energy reduction, however, shall not be built upon meaningful coding efficiency/video quality losses. Otherwise, the se of JMVC over simulcast is no more justified. Video quality issues are discussed in details in the following section.

3.3 Objective Quality Analysis for Multiview Video Coding

In the previous subsections the need for energy-efficient MVC encoding was motivated and justified. To reach such efficiency, complexity reduction, efficient architecture, and efficient memory management techniques including run-time adaptations are required. These techniques, however, may lead to undesirable rate-distortion (RD) performance losses. In other words, the optimizations techniques may lead to reduced video quality for the same output bitrate. For simplicity, in this section we discuss the impact of optimizations algorithms in terms of video quality variation. However, it is necessary to keep in mind that, for a more general analysis, the rate-distortion performance must be evaluated by jointly considering the objective video quality and the generated bitrate. The RD tradeoff can be managed through Quatization Parameter (QP) adaptation by employing an efficient rate control (RC) scheme (see 2.3.4).

To enable the use of MVC in real-world solutions, its implementation must be energetically feasible and the resulting video quality (for similar bitrate) must be significantly improved in relation to previous coding standards applying simulcast-based coding. According to this assumption, the energy reduction techniques must aggressively reduce the total energy consumption at the cost of none or reduced quality loss. Figure 3.11 (extended from Figure 3.4) depicts the impact of some simplified mode decisions in terms of video quality versus bitrate. Take the example of the "SKIP only" MD, which represents 1% of the total coding energy compared to the exhaustive RDO-MD at the cost of nearly 3dB quality loss. Remarkably, this is not a reasonable solution due high quality loss. According to the experiments presented in (MERKLE, BRUST, *et al.*, 2009) and (OH, LEE e PARK, 2011), the MVC provide about 1dB quality increase in relation to H.264 simulcast. In case the energy-efficient optimizations lead to a quality drop at the order of 1dB there is no reason for using the MVC. In this scenario multiple state-of-the-art H.264 encoders should be employed avoiding the 1.75x-2.37x complexity increase (Section 3.1.1) driven by MVC in relation to simulcast. Intermediary solutions are also presented in Figure 3.11 dealing with the relation between energy and video quality. The same kind of energy vs. quality observations are noticed in the ME/DE optimizations.

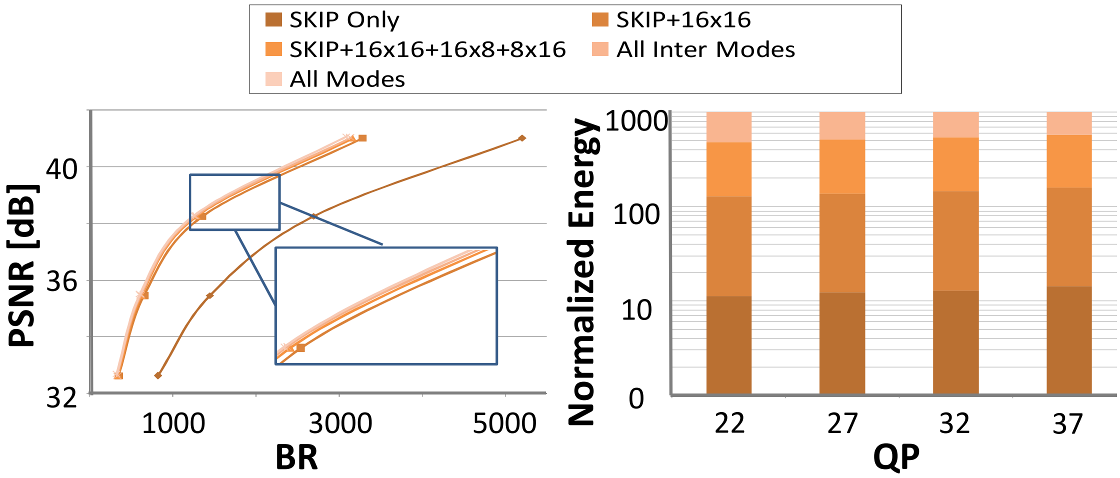


Figure 3.11: Objective video quality in relation to coding modes

Additionally, the 3D video quality includes additional properties in relation to the regular 2D videos. Blocking artifacts are severely undesirable in 3D videos and must be avoided during the encoding process. Such artifacts may lead to problems for intermediate viewpoints generation and/or to the stereo pair mismatch problem, as described in (STELMACH e TAM, 1998). Quality drop due to blurring effect in certain views, however, is tolerable and is attenuated according to the binocular suppression theory which is based on the psycho-visual studies of stereoscopic vision (STELMACH e TAM, 1998). According to it, if the video qualities of left and right eye views differ, the overall perceived quality is close to the high quality of the sharper view. In other words, there is space for controlled quality losses in odd or even views while sustaining the perceived quality and reducing overall energy consumption.

3.4 Quality-Related Challenges in Multiview Video Coding

To reduce the possible quality losses inserted by the energy-efficient optimizations related to the challenges pointed in Section 3.2, there is a need to define quality protection mechanisms able to manage the energy versus quality tradeoff. Such mechanisms must consider the application dynamic behavior in order to optimize the video quality for a given energy constraint. To sustain the overall video quality the energy-efficient MVC must employ:

- *QP-based thresholding*: Most of the energy reduction schemes proposed in the current literature are unable to react to changing QP scenarios due to fixed thresholding. This limitation leads, for corner case scenarios (low or high QPs), to very high quality losses or to limited energy reduction. To deliver high video quality while providing meaningful energy reduction an energy-efficient MVC must control the energy reduction schemes through QP-based threshold equations. Moreover, the thresholds must be defined based on extensive statistical analysis to avoid biasing.

- *Frame-level rate control*: Some energy optimizations may prioritize key frames by providing higher energy/processing budgets for such frames. The drawback of these approaches is the uneven quality distribution, at frame level, inside the same view or between neighboring views. If these quality variations are not properly controlled they may lead the observer to experience some discomfort (STELMACH e TAM, 1998). In order to avoid such quality variations, a frame-level rate control unit must be implemented. The RC task is to predict and control the bitrate versus quality tradeoff and to distribute the amount of bits available (according to a given bandwidth limitation) in such a way to reduce the video quality oscillation and maximize the overall perceived quality.

- *Basic unit-level rate control*: A rate control is also required at basic unit level once energy-efficient optimizations are also defined at MB level. In this scenario, the basic unit RC must be designed to optimize the overall video quality within each frame while considering image/video properties of the image regions.

The challenges described above are critical to deliver high video quality even under a series of energy restrictive constrains and simplifications along the video coding process. In the following section is presented an overview on this thesis contribution. It describes, at high level, the main energy-efficient algorithms and architectures proposed in this volume along with the video quality control strategies.

3.5 Overview of Proposed Energy-Efficient Algorithms and Architectures for Multiview Video Coding

Figure 3.12 presents the overview of this thesis contribution related to the energy-efficient realization of Multiview Video Coding. The high level diagram presents the algorithmic and the architectural contributions along with the conceptual contribution related to the 3D-Neighborhood correlation. Each contribution is detailed in the Chapters 4 and 5, as pointed in Figure 3.12.

The energy reduction and management algorithms, hardware architecture design, memory designs and data-reuse schemes are based on the application knowledge to deliver more efficient results. In this thesis we define the 3D-Neighborhood concept that is widely used to guide the algorithmic and architectural contributions of this thesis. The 3D-Neighborhood is defined as the MBs belonging to neighboring regions at spatial, temporal and view/disparity domains. The analysis of the 3D-Neighborhood space is powerful information to better understand the MBs correlation and to accurately predict the future MBs behavior, as detailed in Chapter 4.

The algorithm-level contribution is centered in energy reduction through complexity reduction and management. The complexity reduction is reached through a multi-level fast mode decision (see Section 4.1) and fast motion and disparity algorithms (for details refer to Section 4.3) both based on the 3D-Neighborhood exploitation and image/video properties analysis. The fast MD and ME/DE algorithms are controlled by an energy-aware complexity adaptation scheme (detailed in Section 4.2) able to handle the energy versus video quality tradeoff while considering battery level and encoder state along with external constraints and user's preferences. To avoid the possible quality losses inserted by complexity reduction techniques a hierarchical rate control (detailed in Section 4.4) featuring both frame- and basic unit-level rate control is proposed in order to guarantee a smooth video quality and output bitrate through QP adaptation. Additionally, to provide efficient energy reduction under varying QP scenarios our proposals employ QP-based thresholding according to the methodology presented in Section 4.1.2.2.

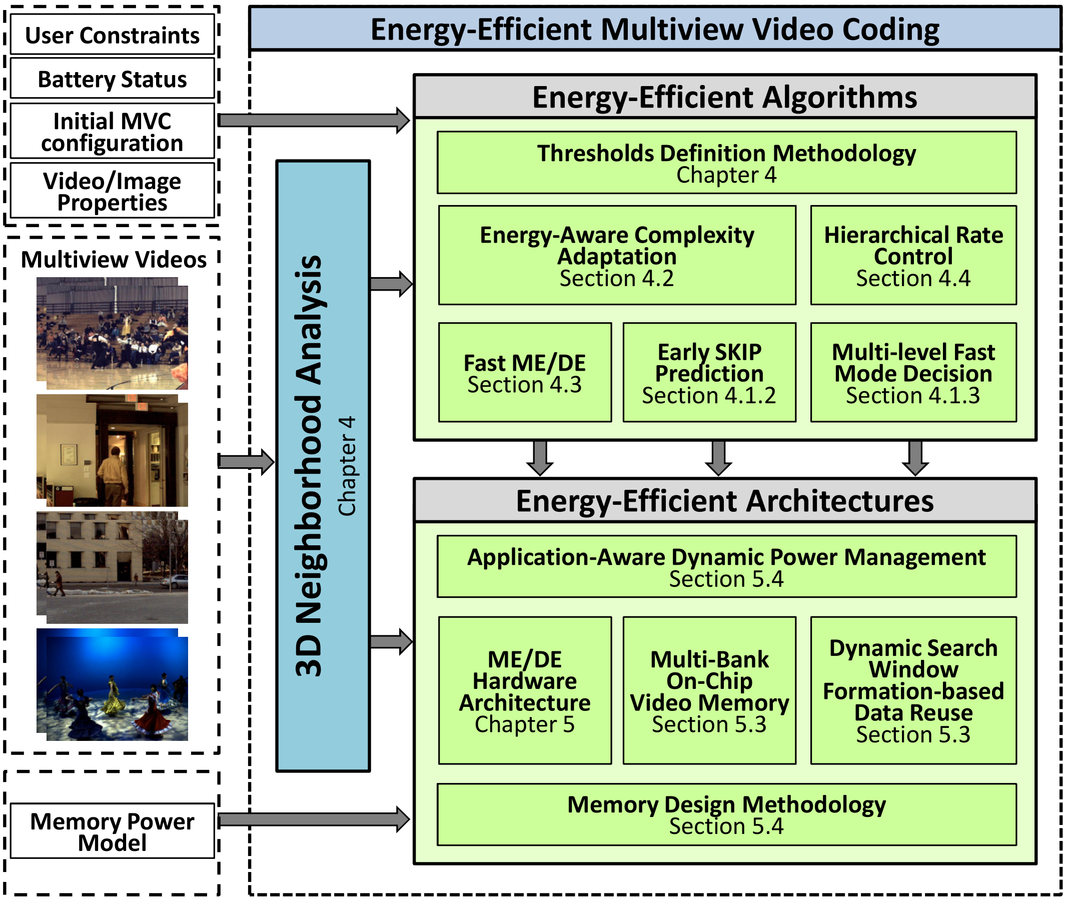

User Constraints
Battery Status
Initial MVC configuration
Video/Image Properties
Multiview Videos
Memory Power Model
Energy-Efficient Multiview Video Coding
3D Neighborhood Analysis
Chapter 4
Energy-Efficient Algorithms
Thresholds Definition Methodology
Chapter 4
Energy-Aware Complexity Adaptation
Section 4.2
Hierarchical Rate Control
Section 4.4
Fast ME/DE
Section 4.3
Early SKIP Prediction
Section 4.1.2
Multi-level Fast Mode Decision
Section 4.1.3
Energy-Efficient Architectures
Application-Aware Dynamic Power Management
Section 5.4
ME/DE Hardware Architecture
Chapter 5
Multi-Bank On-Chip Video Memory
Section 5.3
Dynamic Search Window Formation-based Data Reuse
Section 5.3
Memory Design Methodology
Section 5.4

Figure 3.12: Energy-efficient Multiview Video Coding overview

The architectural contribution is focused on the motion and disparity estimation unit and is composed of the ME/DE hardware architecture itself, the dynamic power and data-reuse management techniques and, the memory design methodology. A multi-level pipelined ME/DE architectural template is proposed (see details in Section 5.1) featuring parallel processing elements, search control and parallel memory interface initially designed to fit to the fast ME/DE algorithm. The on-chip video memory (see Section 5.1.3) sizing and organization was designed considering extensive offline analysis with real video content following our memory design methodology. The on-chip video memory allows sector-level power gating to optimize the energy consumption through implementing a dynamic power management scheme based on the 3D-Neighborhood knowledge (presented in Section 5.4.4). The external memory communication is optimized while employing reduced on-chip memory through a data-reuse technique based on dynamic search window formation that also exploits the 3D-Neighborhood concept.

3.5.1 3D-Neighborhood

The 3D-Neighborhood is defined as the set of MBs belonging to the neighborhood of the current MB in relation to spatial, temporal and view/disparity domains. The high coding properties correlation available in the 3D space is discussed and quantified through the statistical analysis presented along Chapter 4. For this reason the 3D-Neighborhood is used to design and control multiple algorithms and architectural decisions proposed in this thesis.

At design time, the offline statistical analysis is used to understand which coding modes are more frequent, the range of motion and disparity vectors, performance and memory requirement variations, bitrate distribution, which neighboring regions are more correlated for distinct cases and video inputs. Such information is used to guide the complete design of the algorithms. Additionally, the offline analysis is used to define the threshold equations according to our thresholding methodology.

The 3D-neighborhood data are also analyzed at run-time to perform the actual predictions related to the fast mode decision, fast ME/DE algorithms and rate control. Also, the data reuse and the memory requirements prediction used to control the power states of the on-chip video memory employ the neighborhood knowledge.

3.5.2 Energy-Efficient Algorithms

In this section are presented the main energy-efficient algorithms proposed in this thesis and detailed along the Chapter 4.

Early SKIP Prediction: The Early SKIP prediction algorithms exploits the high occurrence of SKIP MBs in order to reduce the MVC encoder complexity. It also considers the 3D-Neighborhood correlation and image properties to take the early SKIP decision and avoid the evaluation of all other encoding modes. This techniques is later incorporated in the Multi-Level Fast Mode Decision algorithm.

Multi-Level Fast Mode Decision: We propose a novel dynamic complexity reduction scheme for non-anchor frames in Multiview Video Coding (MVC). Our scheme exploits different video statistics and the coding mode correlation in the 3D-Neighborhood to anticipate the more-probable prediction modes. Our scheme employs a candidate mode-ranking mechanism reinforced with an RDCost-based neighbor confidence level to determine the more-probable and less-probable prediction modes. Two complexity reduction levels named *Relax* and *Aggressive* with different threshold equations are employed. These levels provide a tradeoff between energy/complexity reduction and video quality. To limit the propagation of prediction error, the anchor frames are encoded using exhaustive RDO-MD. In this case the prediction error is propagated less due to the availability of a better prediction from the anchor frames of the neighboring GOPs.

Fast Motion and Disparity Estimation: Our fast ME/DE algorithm computes the confidence of predictors (motion/disparity vectors of the neighboring MBs) in the 3D-Neighborhood to completely skip the search step. The predictors are classified according to a confidence level and the search pattern is replaced by a reduced number of candidate vectors (up to 13). To exploit this knowledge, accurate motion and disparity fields must be available. Therefore, at least one frame using DE and one using ME must be en-coded with a near-optimal searching algorithm. In our scheme, to avoid a significant quality loss, all anchor frames and the frames situated in the middle of the GOP are encoded using the TZ Search algorithm (TANG, DAI e CAI, 2010). Once the motion and disparity fields are established, all remaining frames are encoded based on predictors available in these fields.

Energy-Aware Complexity Adaptation: The energy-aware complexity adaptation scheme for MVC targeting mobile devices employs several Quality-Complexity Classes (QCCs), such that each class evaluates a certain set of coding modes (thus a certain complexity and energy requirement) and provides a certain video quality. It thereby enables a run-time tradeoff between complexity and video quality. To support asymmetric view quality and exploit the binocular suppression properties, views for one eye are encoded with high quality class and views for the other eye are encoded using a low-quality class. Our scheme adapts the QCCs for different views at run time depending upon the current battery level.

Hierarchical Rate Control: The Hierarchical Rate Control (HRC) for Multiview Video Coding employs a joint solution for the multiple actuation levels of rate control. The proposed HRC employs a Model Predictive Control-based rate control that jointly considers GOP-phase and frame-level stimuli to accurately predict the bit allocation and define an optimal control action at coarse-grain. This guarantees smooth bitrate and video quality variations along time and view domains while supporting any MVC hierarchical prediction structure. To further optimize the bit allocation within the frames, the HRC implements a Markov Decision Process to refine the control action at BU-level taking into consideration image properties to define and prioritize Regions of Interest (RoI). The fine-grained adaptation promotes an increase in objective and subjective video qualities inside the frame. The target bitrate at each time instant is predicted based on the bitrate distribution within the 3D-Neighborhood.

Thresholds Definition Methodology: The energy-efficient algorithms, mainly those based on statistic-based heuristics, are very sensible to the thresholds. For this reason we consider the threshold definition methodology as part of this work. Our schemes employ QP-based threshold equations in order to guarantee proper reaction to changing QP values and keep the energy-efficiency. The thresholds for a subset of QPs are derived from extensive correlation statistical analysis of the 3D-Neighborhood. Probability Density Functions (PDF) considering a Gaussian distribution are typically used to model the coding properties distribution. The QP-based threshold equations are then modeled and formulated using polynomial curve fitting from the set of thresholds statically defined.

3.5.3 Energy-Efficient Architectures

The overview of our architectural contribution to the energy-efficient MVC realization is presented in the following. The implementation details are given in Section 5.

Motion and Disparity Estimation Architectural Template: In this thesis we define a hardware architectural template for the ME/DE unit. This template was designed to facilitate the development and validation of our novel energy-efficient techniques. The hardware is composed of four main modules: (a) programmable search control unit, (b) shared SAD calculator, (c) on-chip video memory, and (d) address generation unit. Additional control and management hardware units may be added to this architectural template to implement novel techniques.

Motion and Disparity Estimation Hardware Architectures: A pipelined hardware architecture was designed to fit the fast ME/DE algorithm introduced in 3.5.2. A multi-level-pipelined parallel hardware architecture for ME/DE exploits four levels of parallelism inherent to the MVC prediction structure which are view, frame, reference frame and, MB levels. To improve reduce the energy consumption related to the memory leakage, another architecture is presented. The second solution features a multi-bank on-chip memory and the dynamic window formation-based power gating control. Finally, a application-aware dynamic power management is proposed and integrated to the third architectural proposal. The goal of the ME/DE architectures is to deliver the performance for real-time ME/DE for up to 4 views HD1080p while reducing the overall energy consumption.

Multi-Bank On-Chip Video Memory: Our multi-bank on-chip memory is designed to feed the SAD calculation by employing 16 parallel banks and provide high throughput in order to meet high definitions requirements. Each bank is partitioned into multiple sectors, such that each sector can be individually power-gated to reduced energy through leakage saving. The on-chip video memory behaves as a cache. Thus, it does not require complete reading of the entire search window. Only the required data is prefetched according to an application-aware prefetching technique such as dynamic window formation. The control of the power-gating is obtained from the application-aware dynamic power management. The size and the organization of the memory are obtained by an offline analysis of the ME/DE memory and energy requirements within the 3D-Neighborhood.

Memory Design Methodology: Based on the offline memory usage analysis, an algorithm is proposed to determine the size of the on-chip memory by evaluating the tradeoff of leakage reduction (on-chip energy) and cache misses (off-chip access energy; result of reduced-sized memory). Afterwards, the organization (banks, sectors) is obtained by considering the throughput constraint. Each bank is partitioned into multiple sectors to enable a fine-grained power management control. The data for each prediction direction is stored in distinct memory sections.

Dynamic Search Window Formation-Based Date Reuse: Instead of prefetching the complete rectangular search window, a selected partial window is dynamically formed and prefetched for each search stage of a given fast ME/DE search pattern depending upon the search trajectory inferred within the 3D-Neighborhood. In other words, the search window is dynamically expanded depending upon the search history of neighboring MBs and the outcome of previous search stages. The search trajectories of the neighboring MBs and their spatial and temporal properties (variance, SAD, motion and disparity vectors) are considered to predict at run time the shape of the search window for the current MB. The goals are significantly reducing energy for off-chip memory accesses and reducing the total amount of on-chip memory bits.

Application-Aware Dynamic Power Management: One key source of leakage is the big on-chip SRAM memory required to store a big rectangular search window, which is inevitable in case of DE. The unused regions of the rectangular search window indicate a waste of on-chip memory hardware. Therefore, significant leakage reduction may be obtained by reducing the size of the on-chip memory, while considering an analysis of the memory requirements of fast ME/DE schemes. Thus, an application-aware power-management scheme is employed. Depending upon the fast ME/DE search pattern, search direction, MB properties, and 3D-Neighborhood memory usage, the amount of required data is predicted. Only the sectors to store the required data are kept powered-on and the remaining sectors are voltage scaled to sleep power states.

Each energy-efficient algorithm and architectural contribution introduced in this section is detailed in Chapters 4 and 5. They were designed and evaluated through simulations considering benchmark video sequences and recommended test conditions (SU, VETRO e SMOLIC, 2006)(ISO/IEC, 2011). The simulation setup and the energy reduction gains are presented, discussed and compared to the state-of-the-art in Chapter 6.

3.6 Summary of Application Analysis for Energy and Quality

The computational and energy requirements demanded for optimal MVC encoding are orders of magnitude beyond the reality of current embedded systems. As demonstrated along this section, MVC optimal encoding requires up to 1000BIPS while current processors delivers about 180MIPS. In this scenario, state-of-the-art batteries would be able to power the MVC encoder for just a few minutes. Thus, there is a need to reduce the MVC complexity and attack the main sources of energy consumption.

As quantified along this section, mode decision and ME/DE represent more than 90% of MVC encoder consumption. Moreover, in the ME/DE block, the memory-related energy is dominant in relation to the computation-related energy. Aware of this behavior, a series of energy-oriented contributions are presented.

Along this thesis are presented energy-efficient algorithms and hardware architectures to enable the real-world implementation of the MVC video encoder. Among the algorithms are a Multi-Level Fast Mode Decision and a Fast ME/DE algorithms. These solutions employ the 3D-neighborhood correlation to predict the full RDO-MD or to avoid unnecessary ME/DE searches. Additionally, an Energy-Aware Complexity adaptation algorithm is proposed to enable run-time adaptation in face of varying coding parameters and video inputs. To avoid eventual quality losses posed by these heuristic-based algorithms, a Hierarchical Rate Control is presented.

Motion and disparity estimation architectures are proposed in order to provide real-time performance and increased energy-efficiency to the most complex MVC encoding block. The Fast ME/DE algorithm is considered along with on-chip memory design techniques to reduce energy consumption. Moreover, the on-chip memory employs multiple power states controlled by our Application-Aware Dynamic Power Management. The external memory accesses are reduced by the Dynamic Search Window Formation algorithm.

4 ENERGY-EFFICIENT ALGORITHMS FOR MULTIVIEW VIDEO CODING

The energy consumption in MVC encoding is directly related to the high computational effort and the intense memory access driven by the data processing. Therefore, the energy-efficient algorithms for the Multiview Video Coding proposed in this thesis are based on complexity reduction and complexity control techniques. Moreover, in addition to the energy consumption perspective, meaningful complexity reduction is also required at the performance perspective in order to make MVC real-time encoding feasible for real-world embedded devices.

Therefore, this chapter presents the proposed energy-efficient algorithms targeting complexity reduction for the Multiview Video Coding through fast mode decision and fast motion and disparity estimation techniques. An energy-aware complexity adaptation algorithm designed to offer run-time adaptivity to changing scenarios (battery level, user constrains, video content) of battery-powered embedded devices is further presented. Aware of the rate-distortion losses posed by such complexity reduction techniques we also present a video-quality management technique to avoid visual degradation. The quality management employs a rate control unit able to maximize the video quality for a given target bitrate while providing smooth quality and bitrate variations at spatial, temporal and disparity domains.

The studies of correlation within the 3D-Neighborhood build the foundation for all algorithms proposed in this chapter. These studies are detailed along the chapter and contemplate the coding mode, motion and disparity fields and bitrate allocation. Additionally, the profiling of the mode distribution and motion/disparity vectors are key enablers for energy-efficient solutions able to provide high complexity reduction at a negligible cost in terms of coding efficiency.

4.1 Coding Mode Decision

Two coding mode decision algorithms are presented in the current section, the Early SKIP and the Multi-level Fast Mode Decision that also encapsulates the concepts exploited in the Early SKIP algorithm. Before moving to the algorithms description we present the analyses that build the ground foundation behind these algorithms.

4.1.1 Coding Mode Correlation Analysis

In this section is discussed the 3D-Neighborhood correlation considering the coding mode used for the different macroblocks in a video sequence. It discusses the mode distribution profiling and presents a statistical analysis considering coding modes, RDCost correlation and video properties for multiple multiview video sequences.

4.1.1.1 Coding Mode Distribution Analysis

The graph presented in Figure 4.1 quantifies the mode distribution in anchor and non-anchor frames of the *Ballroom* and *Vassar* sequences for various QP values (22-37). In anchor frames the mode distribution follows the typical distribution trend of H.264/AVC-based encoding at lower QPs (HUANG, HSIEH, *et al.*, 2006), i.e. more Intra-coded MBs at lower QP values and more SKIP and large block partitions of Inter-coded MBs at higher QP values. On the contrary, for non-anchor frames, a major portion of the total MBs (50-70%) is encoded as SKIP for QP>22. The percentage of the SKIP-coded MBs goes up to 93% (average 63%) in *Vassar*, a well-known test video which has slow-motion sequences. The second dominant mode is Inter-16x16. Notice that, for QP>27, the percentage distribution of the Intra-coded MBs in non-anchor frames diminishes to less than 1%.

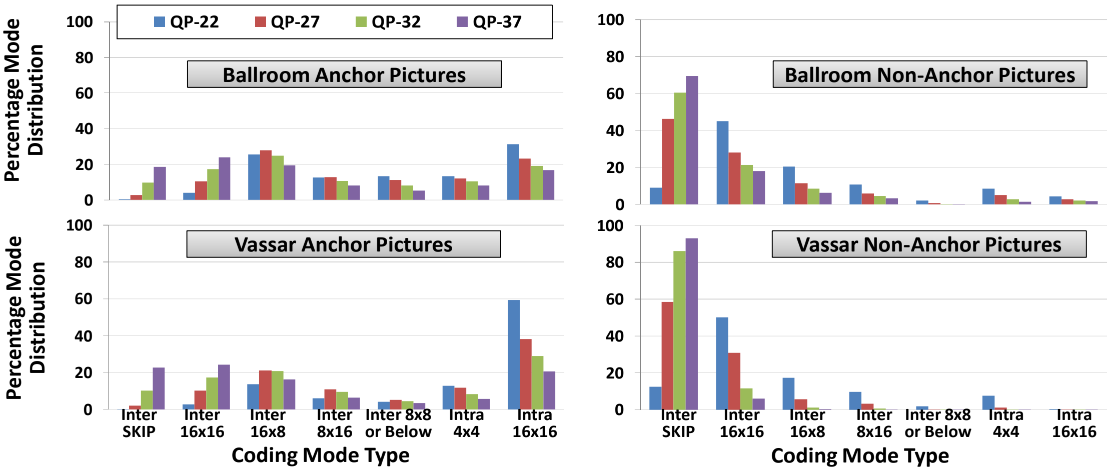


Figure 4.1: Coding mode distribution in test video sequences

The uneven mode distribution for non-anchor shows there is a great potential of complexity reduction in the non-anchor frames if the SKIP or Inter-16x16 coding mode are correctly predicted for a MB. In the following analysis, we show that variance/gradient information in conjunction with the coding mode and RDCost correlation in the 3D-Neighborhood provides a good prediction of the SKIP and/or Inter-16x16 coding modes.

The mode distribution analysis provides high-level information about the features of a video sequence. This analysis is required for relating the distribution of predictions modes to the video features for a given QP. An in-deep analysis is provided in Section 4.1.1.2 where Figure 4.2 provides a subjective analysis of the optimal mode distribution in the *Ballroom* sequence encoded using the exhaustive RDO-MD.

4.1.1.2 Analyzing the Coding Mode Correlation

The first observation provided by Figure 4.2 is the distinct mode distribution in the anchor and non-anchor frames. It is noteworthy that the number of SKIP coded MBs is much higher in the non-anchor frames. This is due to the fact that a higher correlation space is available for non-anchor frames compared to the anchor ones and, consequently, there is higher likelihood to provide a better prediction employing the SKIP mode.

The upscaled frame (S0T1) of *Ballroom* sequence in Figure 4.2 demonstrates that most of the MBs in the background of the scene (spectators and wall) and partially foreground objects (suits of the dancers and floor) of a non-anchor frame are encoded using the SKIP mode. The MBs at the object borders (dancers) are encoded using temporal-/view-prediction modes (i.e. Inter-coded MBs) or spatial-prediction modes (i.e. Intra-coded MBs). Only a few high-textured MBs containing moving spectators in the background are encoded using spatial-/temporal-/view-prediction modes.

Note in Figure 4.2 that the MBs belonging to the same region tend to use the same coding mode when considering spatial, temporal or disparity collocated MBs. For instance, consider frame S0T1, the dancer borders share the same coding mode used by the spatial neighboring MBs that belong to this border. Also, the same coding mode tends to be shared with temporal and disparity collocated MBs in frames S0T2 and S1T1, respectively.

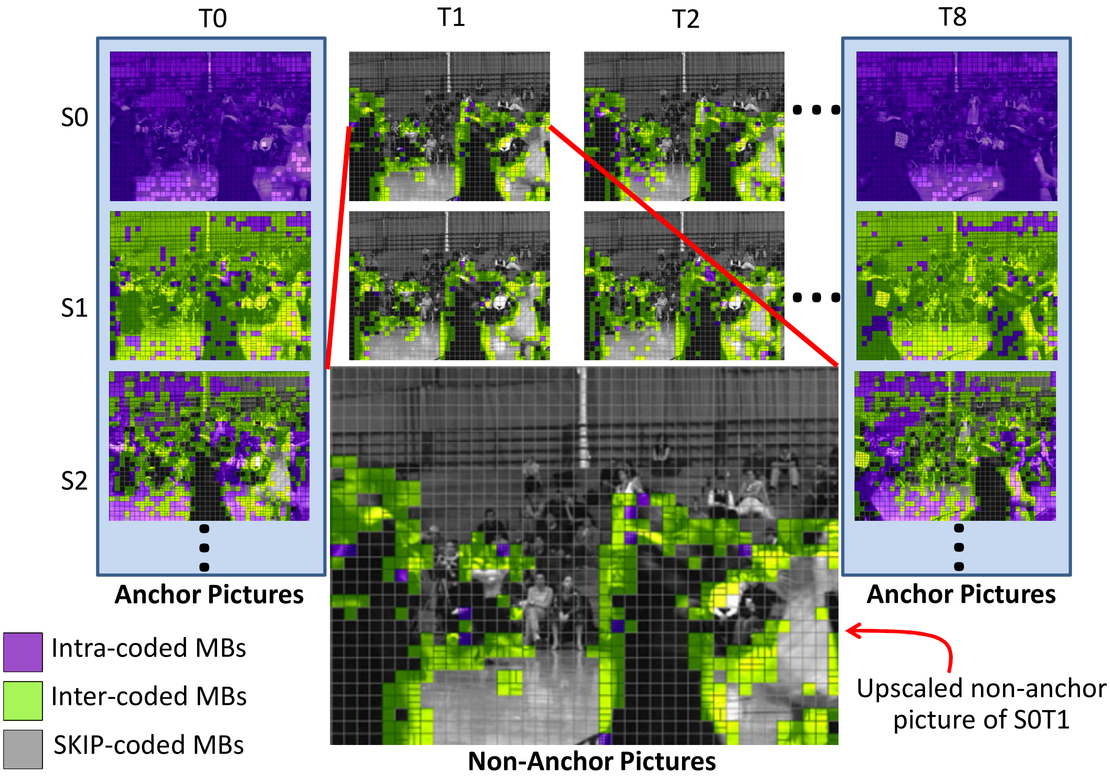
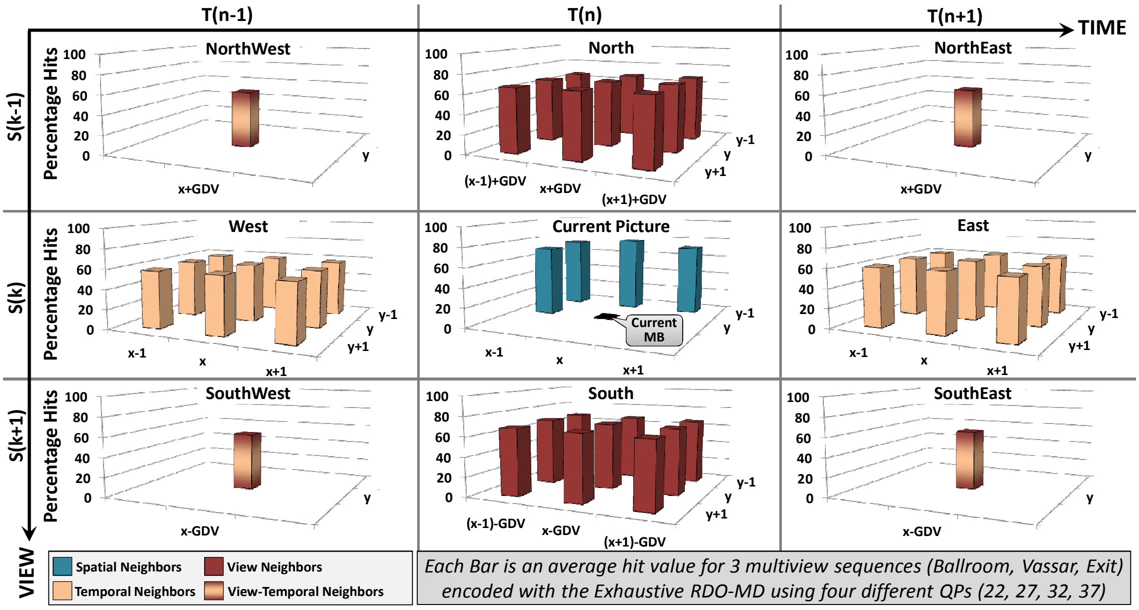


Figure 4.2: Visual coding mode correlation

However, different neighboring MBs in the 3D-Neighborhood exhibit different amount of correlation to the current MB. Figure 4.3 shows the coding mode *hits* (averaged over various QPs and video sequences) using the exhaustive RDO-MD. A coding mode *hit* corresponds to the case when the optimal coding mode of a neighbor is exactly the same as that of the current MB. Otherwise it is given as a coding mode *miss*. The coordinates on the x- and y-axis correspond to the MB number in the corresponding column and row of a frame, e.g., (2,4) means the 2nd MB of the 4th row. The eight neighbor frames in the 3D domain are evaluated and named according to the cardinal points presented in Figure 7. There are total 44 neighbors: 4 spatial, 18 temporal, 18 disparity, and 4 disparity-temporal. Note, the disparity and disparity-temporal neighbors consider the GDV rounded to an integer number of MBs.

Figure 4.3 illustrates that the spatial neighbors in the current frame exhibit the highest coding mode correlation to the current MB (i.e. *hits*>70%) followed by the disparity neighbors in the *North* and the *South* view frames (i.e. *hits*>66%). The coding mode *hits* of the disparity neighbors is less than that of the spatial neighbors due to the variations near the object borders and an inaccuracy in the GDV. The lower number of *hits* for the temporal and disparity-temporal neighbors is basically due to the motion properties. On overall, for non-anchor frames, in more than 98% of the cases the optimal coding mode of an MB is present in the 3D-Neighborhood. It means that by testing the coding modes of all 44 neighbors it is highly probable to find the optimal coding mode for the current MB (more than 98% of the MBs in the current frame). Moreover, due to the availability of a limited set of optimal coding modes in the non-anchor frames (typically much less than the number of modes tested in an exhaustive RDO-MD), a significant complexity reduction may be achieved.



Each Bar is an average hit value for 3 multiview sequences (Ballroom, Vassar, Exit) encoded with the Exhaustive RDO-MD using four different QPs (22, 27, 32, 37)

Figure 4.3: Coding mode hits in the 3D-Neighborhood

As discussed above, there is a big potential of finding the optimal encoding mode in the 3D-Neighbohood. However, a big number of different coding modes may exist in this neighborhood. Thus, in order to reduce the number of probable modes, additional information is needed. In this thesis we consider video and image properties and the RDCost as additional information. The study related to these properties is presented in the following sections.

4.1.1.3 Analyzing the Video Properties

Along our studies multiple video and image properties - including variance, brightness, edges and gradient - were evaluated in order to provide useful information to build fast mode decision algorithms. Among these properties, variance and gradient information showed to be the most helpful to identify highly correlated neighboring MBs and their possible coding modes. The complete evaluation based on statistical analysis for the variance is presented below. For that, multiple video sequences were considered while assuming a Gaussian distribution for the video properties. The variance is defined in Eq. (4.1) while horizontal (Δx) and vertical (Δy) gradients are determined by Eq. (4.2), where ρ_i represents the pixels of a MB.

$$Var_{MB} = \sum_{i=1}^{256} (\, \rho_i - \rho_{AVG}\,)^2 ; \quad \rho_{AVG} = (\sum_{i=1}^{256} \rho_i + 128\,) >> 8 \tag{4.1}$$

$$\begin{gathered} \Delta x = \left(\sum_{i=0}^{15} \sum_{j=0}^{15} \left| \frac{\partial f}{\partial x} \right| + 128 \right) \Big/ 256, \quad \Delta y = \left(\sum_{i=0}^{15} \sum_{j=0}^{15} \left| \frac{\partial f}{\partial y} \right| + 128 \right) \Big/ 256; \\ \frac{\partial f}{\partial x} = \rho(i,j) - \rho(i-1,j), \quad \frac{\partial f}{\partial y} = \rho(i,j) - \rho(i,j-1) \end{gathered} \tag{4.2}$$

Figure 4.4 shows different PDF (Probability Density Function) plots for the variance related to various coding modes. It is noticeable that the peaks for the SKIP and Inter-/Intra-16x16 modes are at 400 and 700, respectively. Therefore, MBs with low variance are more likely to be encoded as SKIP than Inter-/Intra-16x16. On the contrary, MBs with high variance (1500-2500) are more likely to be encoded using smaller block partitions. The PDFs for gradient are omitted, however, they have a similar distribution to that of the variance.

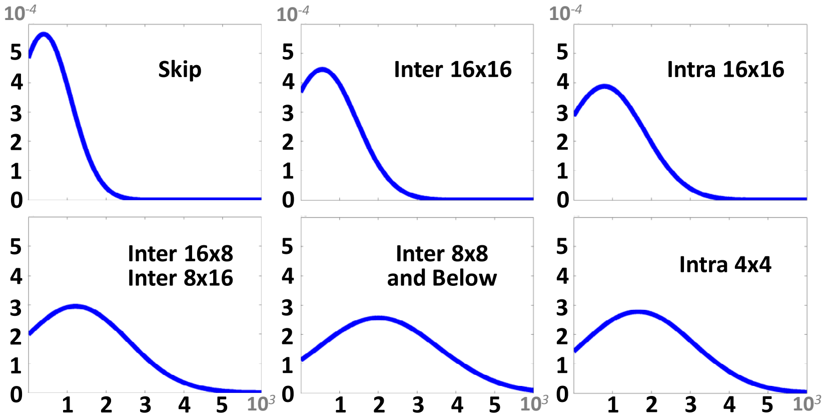


Figure 4.4: Variance PDF for different coding modes

Since there is a considerable overlap between the PDFs of 16x16 and smaller block partitions, in order to obtain a more robust/accurate prediction about the coding modes, RDCost (see Section 4.1.1.4) and coding mode correlation in the 3D-Neighborhood are considered along with the variance and gradient information.

4.1.1.4 Analyzing the RDCost

In order to determine which neighbor has a probable coding mode *hit* or *miss*, we compute the difference between RDCost of a neighbor and the predicted RDCost of the current MB (as the actual RDCost is not available before the RDO-MD process). In the following we analyze the relationship between the RDCost difference and the above discussed coding mode *hit*/*miss*. Figure 4.5a presents the PDF for the RDCost difference for coding mode *hits* and *misses* in case of the SKIP mode. The PDF shows that a SKIP coding mode can be predicted with a high probability of a *hit* when the variance of an MB is low. Figure 4.5b and Figure 4.5c show MB-wise surface plot of the RDCost difference (averaged over all frames of the *Ballroom* sequence) for *hits* and *misses*, respectively. These plots demonstrate that most of the *hits* occur when the RDCost difference is below 10K, while the number of *miss* increases when the value of RDCost difference goes above 70K. This behavior also conforms to the PDFs in Figure 4.5a. This analysis shows that the value of RDCost difference provides a good hint for a *hit* in case of a SKIP coding mode. Similar behavior was observed in the *hit* and *miss* PDFs for other coding modes. Here, we discuss the PDFs for the SKIP mode as an example since it is the dominant coding mode in non-anchor frames, especially for higher QP values.

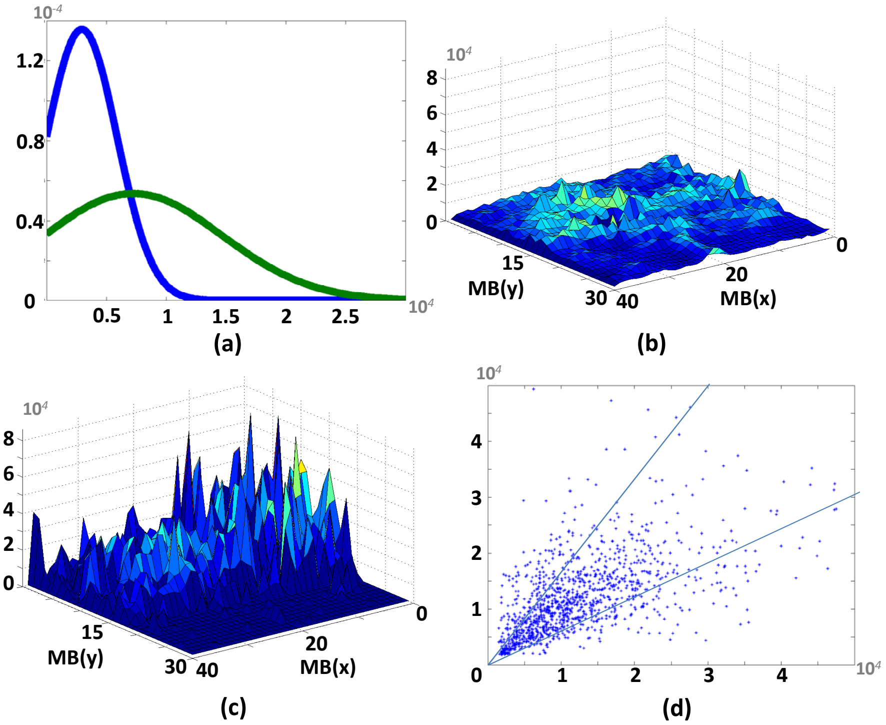


Figure 4.5: (a) PDF for RDCost difference (between the current and the neighboring MBs) for SKIP *hit* and *miss*; (b, c) Surface plots of RDCost difference for the SKIP coding mode hit and miss; (d) RDCost predition error for spatial neighbors

Figure 4.6 presents the PDFs of predicted RDCost for different coding modes. Variable shapes of the PDFs already hint towards the exclusion of improbable mode for a given value of the predicted RDCost. Since a good prediction is important to determine a near-optimal coding mode, we have evaluated the accuracy of the predicted RDCost and optimal RDCost to analyze the risk of misprediction.

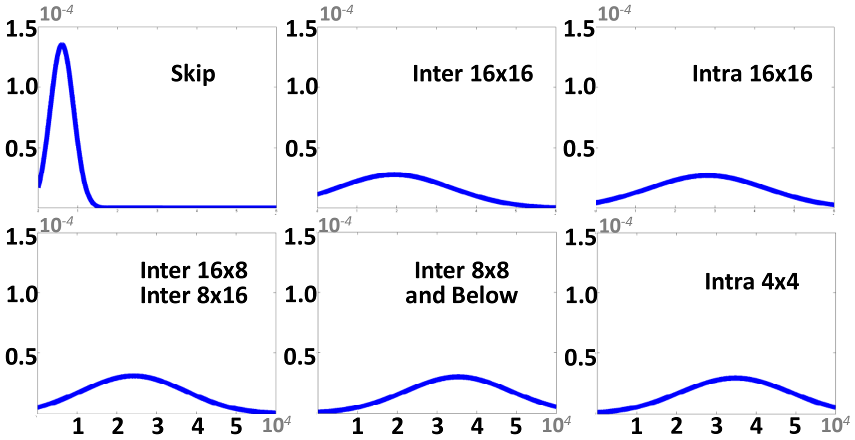


Figure 4.6: PDF of RDCost for different prediction modes of the of Ballroom sequence

Once the RDCost is not available without the exhaustive RDO-MD we tested different predictors for the current MB RDCost in the 3D-Neighborhood. After analyzing the mean and median RDCosts predictors, we have determined that the median RDCost of the spatial neighbors (see Eq. (4.3)) provides the closest match to the optimal RDCost. In Eq. (4.3) S_L, S_T, and S_{TL} represent left, top and top/left spatial neighbors, respectively.

$$RDCost_{PredCurr} = Median(S_L, S_T, S_{TL}) \tag{4.3}$$

Figure 4.5d shows the optimal RDCost vs. predicted RDCost for each MB in 36 frames of the *Vassar* 3rd view. It illustrates a high correlation between the two values (approximately 0.88). Figure 4.7 shows the error surface for the predicted RDCost compared to the optimal RDCost highlighting the regions of misprediction (i.e. borders of the moving objects).

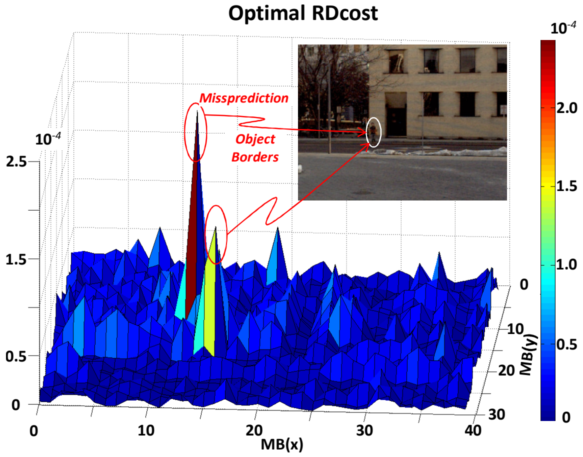


Figure 4.7: Average RDCost prediction error for spatial neighbors in Vassar Sequence

4.1.1.5 Coding Mode Analysis Summary

Our detailed analysis illustrates that it is possible to accurately predict the optimal coding modes, mainly for non-anchor frames, if the coding mode distribution, video statistics, and RDCost correlation in the 3D-Neighborhood are considered. It leads to a high potential of complexity and energy consumption reduction during the MVC encoding process. The main conclusions that enable our fast algorithms are summarized below.

- SKIP MB is the dominant prediction mode (47-97%) in non-anchor frames for QP>27.
- The inter-coded MBs with big partitions are dominant over the smaller partitions and intra-coded MBs in non-anchor frames.
- Different prediction modes exhibit different variance, gradient, and RDCost properties which may be used to identify more- and less-probable coding modes for fast mode decision.
- The spatial, temporal, and view neighborhood exhibit up to 77%, 62%, and 69% coding mode hits, thus there is a high-probability to find a correct prediciton of the coding mode in the 3D-Neighborhood.
- RDCost provide means to identify neighbors with relatively high *hit* probability at run-time.
- The median RDCost of the spatial neighbors provide an accurate RDcost prediction for the current MB.
- Mispredictions may occur at the object borders, objects with high motion, and in the foreground objects where the displacement is different from GDV.

4.1.2 Early SKIP Prediction

In this section is presented the adaptive early SKIP mode decision scheme for Multiview Video Coding based on 3D-Neighborhood correlation (ZATT, SHAFIQUE, *et al.*, 2010), variance and RDCost properties, as discussed in previous section. The algorithm handles an adaptive QP-based thresholding in order to react to the changing QP sustaining the complexity reduction for all QP range.

4.1.2.1 Algorithm for Adaptive Early SKIP Mode Decision

Figure 4.8 presents the flow of our proposed adaptive early SKIP mode decision algorithm. It evaluates the RDCost, variance, and mode *hits* in the 3D-Neighborhood using QP-based adaptive thresholds. To ensure a high probability of correct SKIP mode decision at least two out of three criteria must be satisfied, as shown in Step 2-3 below. The steps are:

Step 1) The RDCost of SKIP mode ($RDCost_{SKIP}$) is computed for the current MB.

Step 2) If $RDCost_{SKIP}$ is less than TH_{RD} (Eq. (4.5)), to select the SKIP mode, the MB is additionally checked for variance (using TH_{Var}, Eq. (4.6)) or percentage neighbors with mode *hits* (using TH_{NbHits}, Eq. (4.7)).

Step 3) Otherwise, conditions for both variance and mode *hits* in the 3D-Neighborhood need to be satisfied to select the SKIP mode.

Step 4) If the early SKIP mode is selected, the current MB is encoded using the SKIP mode and all other modes are not evaluated, i.e. DE and ME are completely bypassed.

Step 5) Otherwise, the exhaustive RDO-MD is used to select the optimal coding mode.

Note, in the step 5, any fast RDO-MD may also be employed. However, we use exhaustive RDO-MD to demonstrate the benefit of our early SKIP mode decision scheme. The thresholds are defined using the offline statistical analysis detailed in Section 4.1.2.2 and Section 4.1.2.3. The goal is to define QP-based thresholds that lead to decisions with high probability hit considering a Gaussian distribution model.

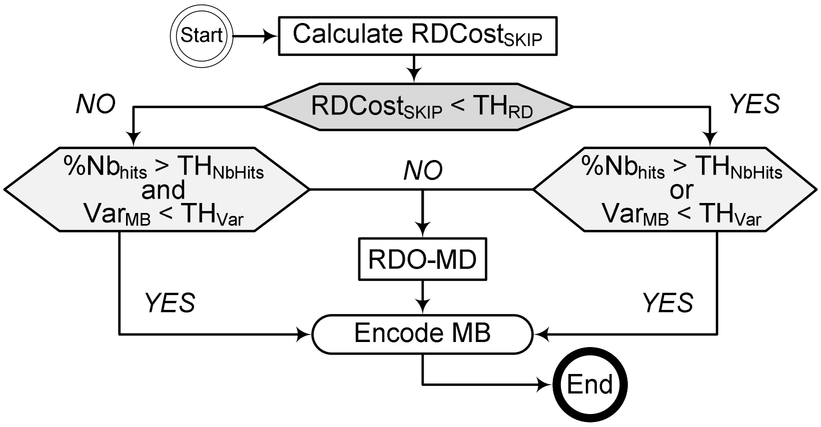

Start
Calculate RDCost$_{SKIP}$
RDCost$_{SKIP}$ < TH$_{RD}$
NO
YES
%Nb$_{hits}$ > TH$_{NbHits}$ and Var$_{MB}$ < TH$_{Var}$
NO
%Nb$_{hits}$ > TH$_{NbHits}$ or Var$_{MB}$ < TH$_{Var}$
RDO-MD
YES
YES
Encode MB
End

Figure 4.8: Early SKIP prediction algorithm

4.1.2.2 RDCost and Variance Thresholding for Early SKIP

To determine the thresholds for early SKIP mode decision, we analyze the Probability Density Function of the RDCost of MBs that are encoded as SKIP MBs when using the exhaustive RDO-MD. Figure 4.9 shows the PDFs for the *Vassar* sequence encoded using various QP values. Notice that the PDF for QP 27 shows a concentrated distribution centered in a relatively low RDCost range, i.e. a small average (μ) and standard deviation (σ). Contrarily, the PDFs for relatively high QPs (32-42) exhibit a low peak centered in a relatively high RDCost range.

For an accurate early SKIP mode decision, the SKIP mode RDCost of the current MB needs to lie in the zone of high probability in the PDF, i.e. $RDCost_{SKIP} < \mu+\sigma$. Assuming a Gaussian distribution, we can compute the area of the high probability zone as follows in Eq. (4.4).

$$F(\mu+\sigma;\, \mu,\, \sigma^2) - F(0;\, \mu,\, \sigma^2) \approx 0.84 \qquad (4.4)$$

Eq. (4.4) shows that up to 84% MBs have the high probability to be coded as SKIP. We define these points of high probability as the RDCost thresholds (TH_{RD}) for predicting a SKIP coding mode (diamond points in the Figure 4.9). Different points for different QPs are used to derive the QP-based threshold Eq. (4.5) using polynomial curve fitting.

$$TH_{RD} = 4.06QP^3 - 279.89QP^2 + 6755.90QP - 53541 \qquad (4.5)$$

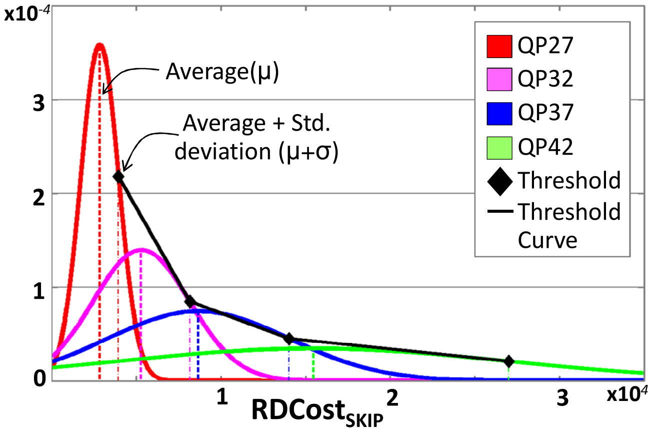


Figure 4.9: PDF of RDCost for SKIP MBs

Figure 4.10 demonstrates a similar statistical analysis using the variance property of a SKIP MB. It shows that the SKIP MBs have a PDF peak in the low variance range compared to other inter-/intra-coded MBs. The variance thresholds (TH_{Var}) for predicting a SKIP coding mode are computed in the same way as of TH_{RD}, i.e. $Var_{MB} < \mu+\sigma$. The QP-based threshold is given by Eq. (4.6).

$$TH_{Var} = 0.02QP^3 - 2.02QP^2 + 72.30QP + 196.04 \qquad (4.6)$$

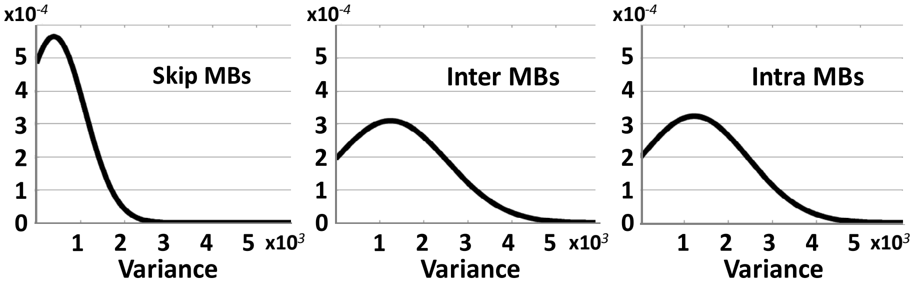


Figure 4.10: PDF of Variance for different prediction modes

From Figure 4.10 it is possible to notice the overlap between PDFs of SKIP MBs and other coding modes. Therefore, to obtain an accurate SKIP mode decision, in addition to the RDCost and variance properties, the SKIP mode correlation in the 3D-Neighborhood is also considered.

4.1.2.3 SKIP Mode Correlation Thresholding

To quantify the SKIP mode correlation in the 3D-Neighborhood we have considered a total of 44 neighbors: (i) 4 spatial, (ii) 18 temporal (previous and next frames), (iii) 18 disparity (top and bottom views), and (iv) 4 disparity-temporal (top and bottom views of the temporal frames). The disparity and disparity-temporal neighbors consider the Global Disparity Vector (GDV) displacement. If the optimal coding mode of a neighbor (out of many in a given set of neighbors) is the same as that of the current MB, it is called a mode *hit*. Figure 4.11 shows that the number of mode *hits* in the 3D-Neighborhood is greater than 88% for the *Vassar* sequence (QP 27). For a higher QP, the number of mode *hits* is even higher. It is noteworthy that when considering all of the 44 neighbors, the number of mode *hits* is almost 100%, i.e. at least one neighbor accurately predicts the SKIP mode.

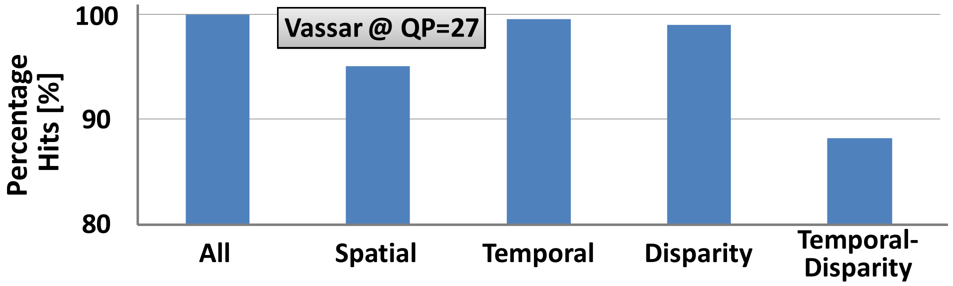


Figure 4.11: Analyzing the mode hits in the 3D-Neighborhood

If many neighbors exhibit a mode *hit*, there is a high probability that the current MB is also a SKIP MB. This assumption is more likely to be true for non-anchor frames. Figure 4.12 presents the PDFs of percentage neighbors encoded as SKIP MBs (i.e. mode *hit*s) for the *Vassar* sequence encoded using various QPs. Since in several cases some neighbors are not available (for instance, MBs at boundaries or MBs in the first encoded view), we consider percentage of SKIP-coded neighbors instead of an absolute number. A high probability zone can be defined as: $\%Nb_{hits} > \mu - \sigma$. The QP-based threshold equation is given in Eq. (4.7).

$$TH_{NbHit} = 0.006QP^3 + 0.64QP^2 - 19.31QP + 206.12 \quad (4.7)$$

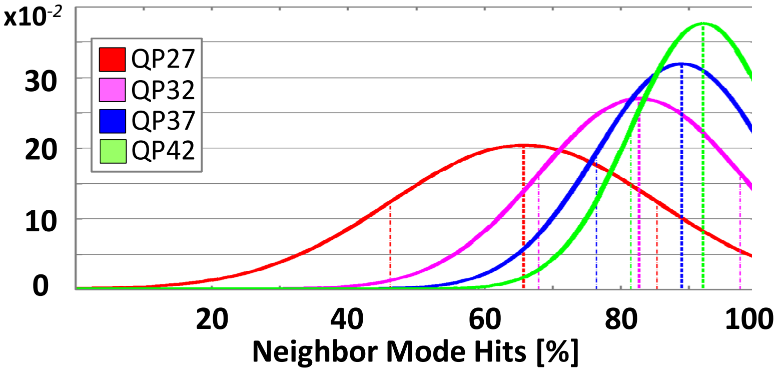


Figure 4.12: PDF of percentage neighbors with SKIP mode hits

4.1.2.4 Early SKIP Complexity Reduction Evaluation

While this thesis overall results are presented in Chapter 6, here are present some detailed results for the Early SKIP algorithm. Table 4.1 provides the detailed results for

ΔPSNR, ΔBitrate (BR), and time saving (TS) compared to the exhaustive RDO-MD. Our scheme achieves up to 77% time saving (avg. TS=56%). The early SKIP algorithm maintains the time savings for the complete QP range due to our adaptive QP-based thresholding.

Table 4.1: Detailed Results for ΔPSNR, ΔBitrate, and Time Savings (TS) compared to the exhaustive RDO-MD

Video	QP	Our Scheme		
		TS (%)	ΔPSNR (dB)	ΔBR (%)
Ballroom	27	39.72	0.095	-0.28
	32	37.66	0.091	-0.57
	37	40.54	0.085	-0.12
	42	43.82	0.108	0.20
Exit	27	57.41	0.128	-0.47
	32	62.14	0.157	1.07
	37	65.85	0.227	1.06
	42	70.90	0.268	-0.66
Vassar	27	59.25	0.076	-1.500
	32	70.68	0.075	-3.438
	37	75.40	0.073	-2.803
	42	77.56	0.063	-4.868
Rena	27	46.61	0.285	-1.242
	32	50.34	0.299	-2.277
	37	54.72	0.324	-2.489
	42	58.54	0.395	-2.706
Average	27	50.75	0.146	-0.87
	32	55.20	0.155	-1.30
	37	59.13	0.177	-1.09
	42	62.70	0.209	-2.01
	Avg.	56.95	0.172	-1.32

Figure 4.13 presents a deeper analysis chart of the time savings of our scheme for different views of two sequences. The first view presents less time saving due to the fewer number of available neighbors with mode *hits*. In *Exit* video, the odd views (i.e., 1, 3, 5) – with top and bottom views available – present higher time saving due to a relatively high number of mode *hits* in the neighboring views. The large number of SKIP MBs in *Vassar* results in high time saving for all views.

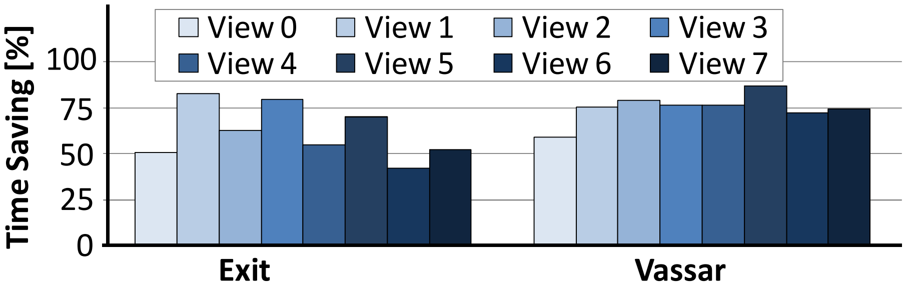


Figure 4.13: View-level time saving of our Scheme

In Figure 4.14 is presented the frame-level results of our early SKIP mode decision scheme for the first 50 non-anchor frames in the first 3 views of the *Exit* sequence. Anchor frames were omitted from these graphs due clarity reasons. The PSNR curves show that View 0 has relatively less PSNR loss compared to other views. However, it also provides relatively less time savings. Due to high SKIP mode *hits* in the 3D-Neighborhood, View 1 provides a relatively higher time saving and lower PSNR loss compared to View 2. Note that the sudden variations in the curves (i.e., valleys in Figure 4.14) occur at the middle of GOP, where the temporal neighbors are from the anchor frames (thus a low number of mode *hits*). The variations are higher in View 0 (TS ≈ 20%) due to the unavailability of the disparity neighbors.

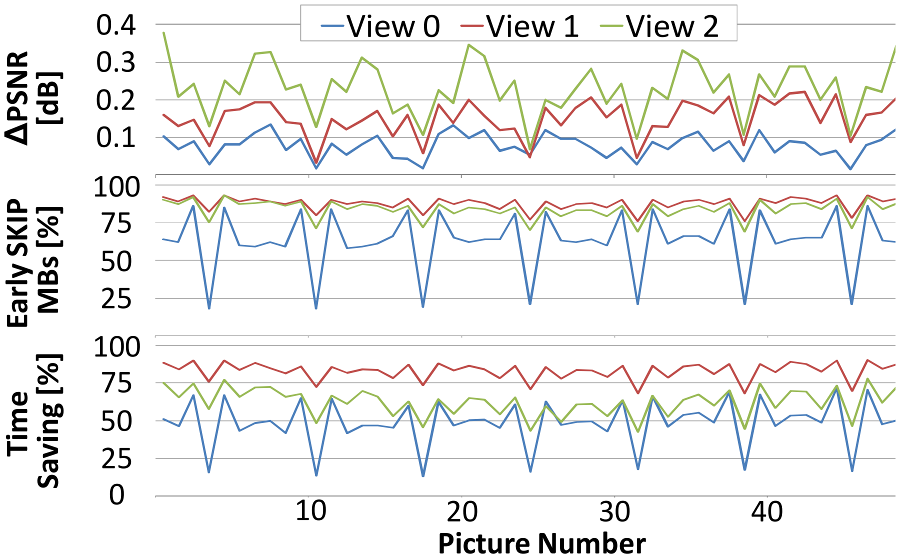


Figure 4.14: PSNR, percentage of MBs selected as early SKIP and time saving for non-anchor frames (Exit, QP=32).

This section presented the early SKIP mode decision algorithm that detects the high probability of SKIP occurrence and avoids the processing of all other coding modes. The 3D-Neighborhood correlation and video properties are used to guarantee algorithm accuracy while reducing RD losses.

4.1.3 Multi-Level Fast Mode Decision

In Section 4.1.2 an Early SKIP prediction algorithm was proposed. However, other optimizations may be proposed to further reduce the MVC computational complexity and energy consumption within the mode decision scope. Therefore, in this section we propose a complete multi-level mode decision scheme (ZATT, SHAFIQUE, *et al.*, 2010) based on the 3D-Neighborhood correlation and exploitation of additional statistical and video information.

The detailed flowchart of our multi-level fast mode decision for non-anchor in MVC is presented in Figure 4.15. The scheme operates in 6 phases: (i) RDCost-based confidence-level ranking, (ii) early SKIP prediction, (iii) evaluating high-confidence modes, (iv) evaluating low-confidence modes, (v) video properties based mode decision, and (vi) size/direction-based mode decision. At the end of each phase (except for phase i), a condition is evaluated for early termination of the scheme. We explain these phases in the subsequent sections.

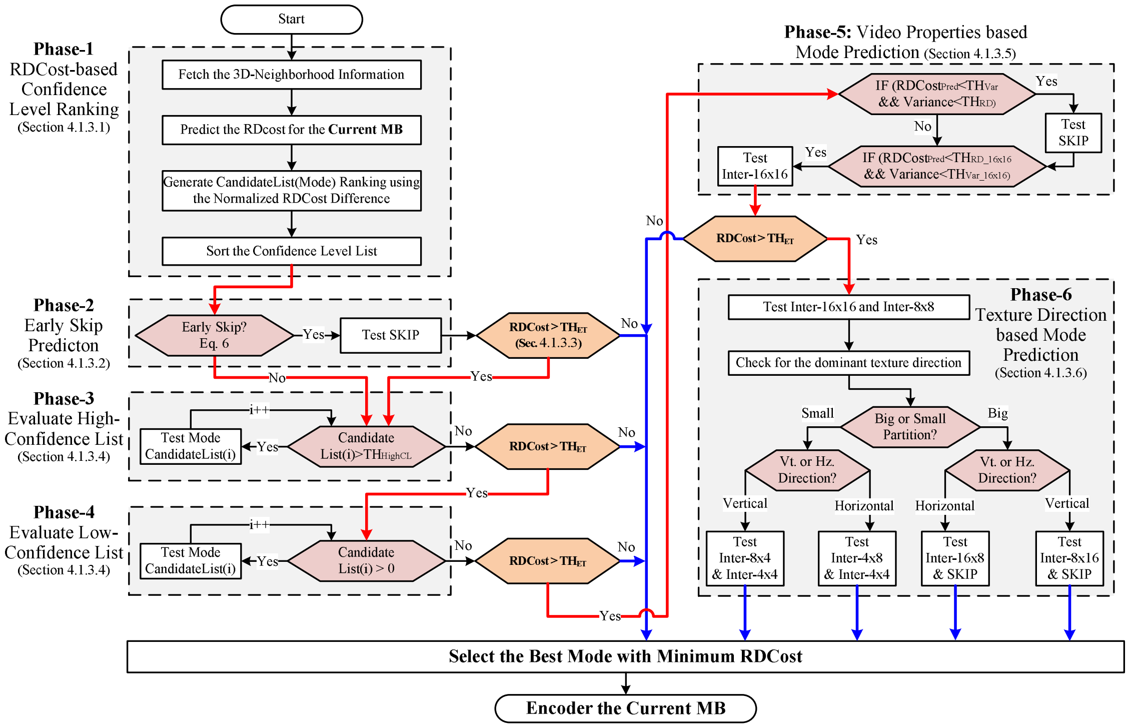

Start
Phase-1
RDCost-based Confidence Level Ranking
(Section 4.1.3.1)
Fetch the 3D-Neighborhood Information
Predict the RDcost for the Current MB
Generate CandidateList(Mode) Ranking using the Normalized RDCost Difference
Sort the Confidence Level List
Phase-2
Early Skip Predicton
(Section 4.1.3.2)
Early Skip? Eq. 6
Yes
Test SKIP
RDCost > TH_ET (Sec. 4.1.3.3)
No
Yes
Phase-3
Evaluate High-Confidence List
(Section 4.1.3.4)
i++
Test Mode CandidateList(i)
Yes
Candidate List(i)>TH_HighCL
No
RDCost > TH_ET
No
Yes
Phase-4
Evaluate Low-Confidence List
(Section 4.1.3.4)
i++
Test Mode CandidateList(i)
Yes
Candidate List(i) > 0
No
RDCost > TH_ET
No
Yes
Phase-5: Video Properties based Mode Prediction (Section 4.1.3.5)
IF (RDCost_Pred<TH_Var && Variance<TH_RD)
Yes
Test SKIP
No
IF (RDCost_Pred<TH_RD_16x16 && Variance<TH_Var_16x16)
Yes
Test Inter-16x16
No
RDCost > TH_ET
Yes
Phase-6
Texture Direction based Mode Prediction
(Section 4.1.3.6)
Test Inter-16x16 and Inter-8x8
Check for the dominant texture direction
Small
Big or Small Partition?
Big
Vt. or Hz. Direction?
Vt. or Hz. Direction?
Vertical
Horizontal
Horizontal
Vertical
Test Inter-8x4 & Inter-4x4
Test Inter-4x8 & Inter-4x4
Test Inter-16x8 & SKIP
Test Inter-8x16 & SKIP
Select the Best Mode with Minimum RDCost
Encoder the Current MB

Figure 4.15: Overview of the multi-level fast mode decision

4.1.3.1 RD-Cost Confidence Level Ranking

Firstly, the 3D-Neighborhood information is fetched and the RDCost for the current MB is predicted using the spatial neighbors considering their high ratio of coding mode *hits* (Eq. (4.8)). A list of candidate prediction modes (*CandidateList)* is formed from the 3D-Neighborhood. Each candidate mode is associated with a rank value (R_{MODE}). This value is calculated as the accumulated confidence level of the neighbors with the similar coding mode ($CL_{NBi}(Mode)$, Eq. (4.9) and (4.10)). This confidence level of a neighbor is computed by evaluating the normalized difference (*NDiff*) between its actual RDCost and the predictive RDCost for the current MB (Eq. (4.11)). Note that the confidence level calculation depends upon the quality of RDCost prediction (Section 4.1.1.4). The candidate list is then sorted according to the rank value (Eq. (4.12)).

$$RDCost_{PredCurr} = Median(\, S_L\,, S_T\,, S_{TL}\,) \tag{4.8}$$

$$R_{MODE} = \sum_{i=1}^{44} CL_{NBi}(\, Mode\,) \tag{4.9}$$

$$CL_{NBi}(\, Mode\,) = (\, Clip(\, NDiff(\, NB_i\,), 0\,, 1\,)) \tag{4.10}$$

$$NDiff(\, NB_i\,) = 1 - Abs(\, RDCost_{PredCurr} - RDCost_N\,)\,/\,Diff_{MAX} \tag{4.11}$$

$$CandidateList = Sort(\, R_{SKIP}\,, R_{INTER16x16}\,, \ldots, R_{INTRA4x4}\,) \tag{4.12}$$

where NB_i is the i^{th} neighbor, $Diff_{MAX}$ is the maximum RDCost difference. The Sort function in Eq. (4.12) sorts the values in a descending order.

4.1.3.2 Early SKIP Prediction

Based on the analysis of high SKIP MBs distribution in non-anchor frames (Section 4.1.1.1), our scheme employs an early SKIP prediction based on the algorithm presented in Section 4.1.2. In case a SKIP mode is correctly predicted, significant complexity reduction is obtained as the ME and DE are entirely skipped. This early

SKIP mode prediction is only performed if sufficient correlation is available in the 3D-Neighborhood. To avoid a misprediction (that may result in significant PSNR loss) the early SKIP prediction depends upon three conditions considering the mode rank, variance, and RDCost, as presented in Eq. (4.13).

$$EarlySKIP = \begin{array}{l} ((R_{SKIP} > TH_{Rank})\&\& \\ (Variance < TH_{Var})\&\& \\ (RDCost_{PredCurr} < TH_{RD})) \end{array} \tag{4.13}$$

The QP-based thresholds for RDCost (TH_{RDCost_ES}) and variance (TH_{Var_ES}) were obtained using the corresponding PDF analysis. The area of high probability (i.e. the grey-filled area in Figure 4.16) is considered as the average plus one standard deviation. A threshold is thereby given as $TH = \mu + \sigma$. The PDFs for four different QP values are used to determine four thresholds at different QPs. A QP-based threshold formulation is obtained using the polynomial curve fitting. Figure 4.17 presents thresholds (TH_{RDCost_ES} and TH_{Var_ES}) for four QPs and the corresponding curve fitting. The threshold for ranks (TH_{R_ES}) was obtained (using an exhaustive analysis) as 15% of the total confidence level accumulated on the entire *CandidateList* (i.e. sum the ranks of all modes).

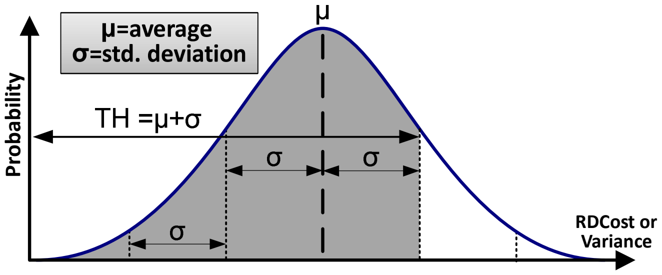


Figure 4.16: PDF showing the area of high probability as the shaded region

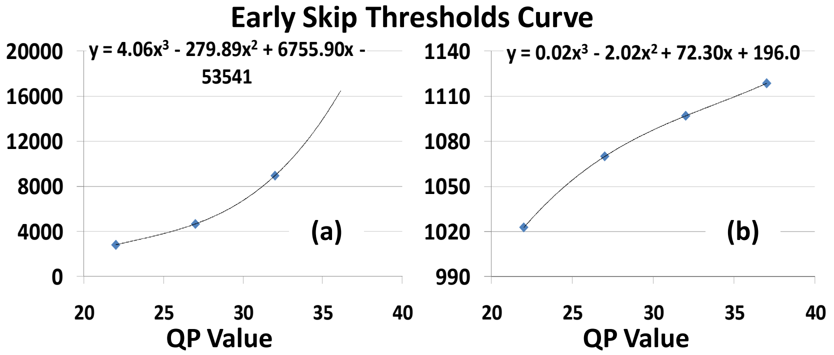


Figure 4.17: Early SKIP threshold curves for (a) RDCost and (b) Variance

4.1.3.3 Early Mode Decision Terminator

After the early SKIP mode prediction, the tested mode is evaluated for the early mode decision termination. If the tested RDCost is bigger than the threshold TH_{ET}, the mode decision proceeds to the next phase. Otherwise, the mode decision is terminated and the best tested mode is used for encoding the current MB.

The threshold for early mode decision termination controls the achieved complexity reduction and the resulting PSNR loss. An excessively high value threshold provides high complexity reduction at the cost of severe PSNR loss. We have performed an exhaustive analysis to determine these thresholds. Figure 4.18 shows the RD-curve for 5 different test threshold values and their corresponding complexity reduction (bars) for

QP=32. It is noted that TH_{ET} = *5000* provides minimal PSNR loss and low complexity reduction, while $TH_{ET} \geq$ *10000* provides a high complexity reduction at the cost of considerable PSNR loss (i.e. > 0.15 dB). In order to provide a tradeoff between achieved complexity and the resulting quality loss, we propose two complexity reduction levels or complexity reduction strengths:

- *Relax* complexity reduction: it provides a reasonable complexity reduction while considering a low PSNR loss.
- *Aggressive* complexity reduction: it provides a high complexity reduction at the cost of a slightly higher PSNR loss (but still visually un-noticeable in many cases, as we will show in results section).

From an exhaustive analysis of various multiview sequences (encoded using exhaustive RDO-MD), we obtained the plots and QP-based equations for *Relax* (blue) and *Aggressive* (red) complexity reduction (see Figure 4.19).

This early termination is employed after each phase of our dynamic complexity reduction scheme as explained in the subsequent sections.

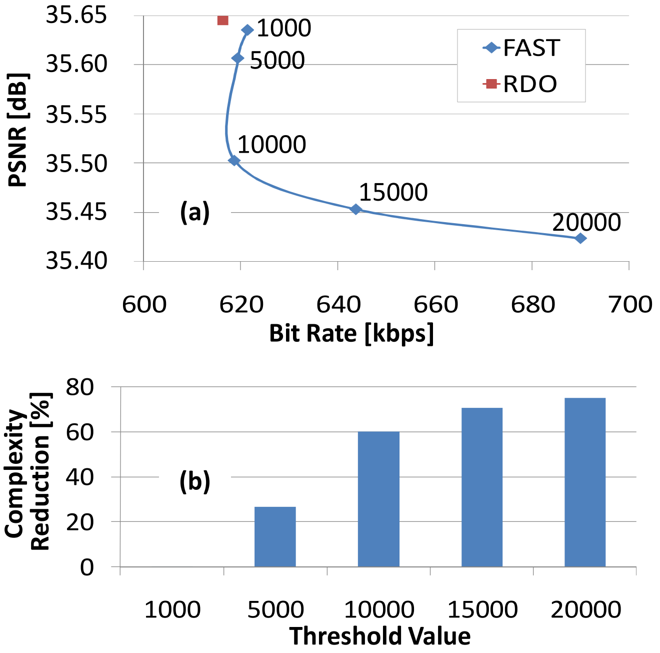


Figure 4.18: Evaluation of thresholds for Early Termination for (Ballroom, QP=32)

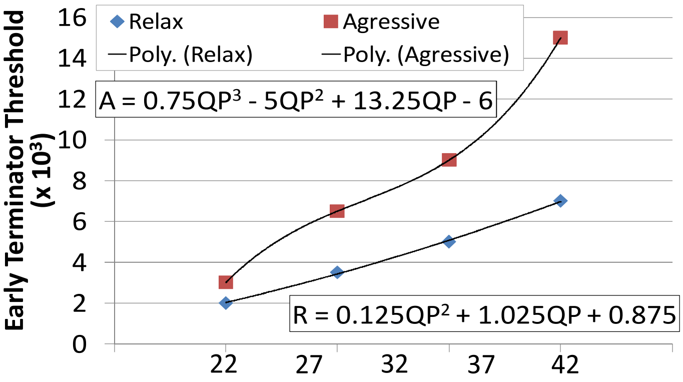


Figure 4.19: Early Termination threshold plots for Relax (blue) and Aggressive (red) complexity reduction

4.1.3.4 High Confidence Modes and Low Confidence Modes

The modes in the sorted *CandidateList* are partitioned into *high-confidence* and *low-confidence* modes using TH_{HighCL}. The threshold TH_{HighCL} is determined (using an exhaustive analysis) as 25% of the total confidence level accumulated on the entire *CandidateList*. First, all of the *high-confidence* modes (i.e. $R_{MODE} \geq TH_{HighCL}$) are evaluated. Afterwards, the condition for early termination is evaluated. If the condition is not satisfied, all of the *low-confidence* modes (i.e. $R_{MODE} < TH_{HighCL}$) are evaluated. If the termination condition is not satisfied after evaluating the *low-confidence* modes, the mode decision proceeds to the next phase.

4.1.3.5 Video Properties-based Mode Prediction

As discussed in Figure 4.1, SKIP and Inter-16x16 are the two most occurring modes in the non-anchor frames. In case sufficient correlation is not available in the 3D-Neighborhood, the variance property of a frame is considered to evaluate SKIP and Inter-16x16 coding modes (in case these were not evaluated in the previous phases). The thresholds used in the conditions of this phase are derived using the PDFs presented in section 4.1.1.3 considering the region of high-probability as discussed in Figure 4.16.

4.1.3.6 Texture Direction-based Mode Prediction

In the last phase a texture direction based prediction is employed to evaluate modes other than SKIP and Inter-16x16 (if they were not tested in the previous phases). The direction of the gradient is considered to exclude improbable modes. The RDCosts of Inter-16x16 and Inter-8x8 modes are compared to determine whether to evaluate bigger or smaller partitions for the current MB. If the RDCost of Inter-16x16 is less than that of Inter-8x8, larger partitions (Inter-16x8 or Inter-8x16) are evaluated depending upon the dominant direction of the gradient. Otherwise, smaller partitions (i.e. Inter-8x4 and Inter-4x8) are evaluated accordingly Figure 4.15.

In case of larger partition, SKIP mode is always tested (if not tested in the earlier phases). Similarly, Inter-4x4 is always tested in case of smaller partitions (if not tested in the earlier phases). Finally, after all the processed phases, the best mode (i.e. the mode with the minimum RDCost) is used for coding the current MB.

4.1.3.7 Multi-Level Fast Mode Decision Evaluation

The detailed results of our multi-level fast mode decision algorithm compared to the RDO-MD solution implemented in the JMVC are presented along this section. Table 4.2 presents the results for ΔPSNR, ΔBitrate, and complexity reduction (i.e. time saving, TS). For JMVC using the exhaustive RDO-MD the results are presented in coding time (column T, in seconds), PSNR (dB) and Bitrate (column BR, in kbps). The values for a certain QP value are obtained by averaging over all eight views. The last row named *Average* presents the average results over all sequence. The experiments were performed for 8 views considering IPB view coding order. For more details on the experimental setup refer to Section 6.1.

Figure 4.20 illustrates the PSNR (lines) and time savings (bars) comparison of *Relax* and *Aggressive* levels averaged over all views and QPs for *Ballroom* and *Exit* sequences. It is noted that the difference between the RD-curves of *Relax* and *Aggressive* is more significant at low-bitrates and this difference diminishes at higher bitrates. The time savings of the *Aggressive* level are significantly higher compared to

Relax level at higher bitrates while providing slight RD difference. Relax scheme was developed to keep video quality in all QP ranges and for this reason is forced to reduce TS for big and small QP ranges presenting higher TS for intermediate QPs. In *Aggressive* scheme the higher TS is prioritized for the whole QP range.

Table 4.2: Detailed Results for ΔPSNR, ΔBitrate, and Time Savings Compared to the Exhaustive RDO-MD

Video Sequence	**QP**	**JMVC**			**Proposed Relax**			**Proposed Aggressive**		
		T	PSNR	BR	TS	ΔPSNR	ΔBR	TS	ΔPSNR	ΔBR
		[sec]	[dB]	[kbps]	[%]	[dB]	[%]	[%]	[dB]	[%]
Ballroom	22	2682.53	41.111	3176.849	54.77	0.005	2.923	59.03	0.002	5.820
	27	2490.47	38.415	1319.736	61.23	0.039	3.015	70.12	0.038	10.150
	32	2315.22	35.667	654.338	57.04	0.039	0.934	65.71	0.084	3.060
	37	2121.62	32.884	360.162	52.67	0.025	0.453	63.07	0.065	1.500
Exit	22	2671.07	41.601	2114.453	60.29	0.006	3.937	67.68	0.045	7.510
	27	2268.79	39.456	652.491	71.36	0.016	2.402	80.08	0.099	12.090
	32	2065.18	37.508	292.673	70.19	0.030	1.101	78.10	0.109	5.910
	37	1900.21	35.293	163.436	67.92	0.043	0.357	78.37	0.123	2.060
Vassar	22	2963.66	40.743	3007.434	55.26	0.001	2.837	69.35	0.001	5.470
	27	2519.33	37.828	850.324	77.18	0.021	1.198	81.24	0.001	11.250
	32	2114.88	35.490	259.826	76.59	0.020	-0.028	82.44	0.055	3.450
	37	1827.57	33.294	117.428	74.69	0.008	-0.196	82.23	0.034	3.660
Race1	22	4908.26	42.340	2549.767	64.55	0.006	8.349	78.10	0.005	11.550
	27	4631.98	39.422	1182.855	71.70	0.036	6.514	80.09	0.028	17.800
	32	4269.55	36.501	552.949	70.49	0.036	3.443	78.13	0.064	9.800
	37	3806.22	33.795	294.763	69.05	0.028	1.543	74.92	0.074	6.650
Rena	22	2238.63	46.555	1347.801	68.54	-0.205	12.283	67.09	-0.212	22.960
	27	1960.61	43.846	587.514	70.55	-0.215	14.289	70.44	-0.310	36.550
	32	1685.64	40.535	293.333	71.79	0.028	6.971	70.87	-0.220	33.740
	37	1452.22	37.396	163.581	66.58	0.043	3.038	73.03	-0.154	26.880
Akko&Kayo	22	2644.20	43.53	1743.05	65.67	-0.056	10.152	66.81	-0.050	14.770
	27	2560.85	40.79	808.48	69.66	-0.015	9.395	74.27	-0.020	21.920
	32	2466.77	37.59	433.65	65.66	0.036	3.130	71.05	0.000	15.280
	37	2320.73	34.45	254.24	59.72	0.023	1.597	70.02	0.020	8.970
Breakdancers	22	5893.46	41.449	4899.089	53.81	0.002	5.393	62.39	0.004	7.150
	27	4817.77	39.841	1454.553	63.14	0.038	7.492	76.10	0.061	12.910
	32	4116.45	38.432	667.955	62.98	0.054	7.861	74.95	0.111	11.700
	37	3487.59	36.629	378.932	59.53	0.094	3.615	76.11	0.237	7.150
Uli	22	4826.40	40.476	8152.865	44.35	-0.001	2.024	47.53	0.007	3.545
	27	4638.51	38.591	3801.339	61.34	0.013	3.245	66.62	0.048	5.488
	32	4326.23	36.238	2056.013	57.22	0.002	2.115	61.41	0.037	4.280
	37	3945.21	33.554	1162.239	61.54	0.078	3.676	68.11	0.137	7.150
Poznan_ Hall2	22	10737.88	42.9111	5149.584	63.48	-0.003	5.649	67.68	0.045	7.510
	27	8911.15	41.693	1417.539	69.02	0.016	5.124	80.08	0.099	12.090
	32	7778.938	40.303	693.789	65.46	0.020	3.913	78.10	0.109	5.910
	37	6702.69	38.606	420.110	61.56	0.047	1.269	78.37	0.123	2.060
GT_Fly	22	6334.83	41.247	6437.935	55.16	0.017	9.547	64.68	0.028	12.975
	27	5168.38	39.705	2029.539	59.98	0.042	8.832	72.68	0.052	10.685
	32	4511.77	38.280	946.270	62.40	0.044	6.562	67.60	0.079	8.655
	37	2185.58	36.819	610.563	59.74	0.051	3.217	67.00	0.116	9.984
Average	22	3603.53	42.226	3373.914	58.59	-0.023	6.310	65.03	-0.013	9.926
	27	3236.04	39.774	1332.162	67.52	-0.001	6.151	75.17	0.010	15.093
	32	2919.99	37.245	651.342	65.98	0.031	3.600	72.84	0.043	10.178
	37	2607.67	34.661	361.847	63.30	0.044	1.857	73.12	0.077	7.606
	AVG	3091.81	37.183	1172.794	63.85	0.013	4.479	71.54	0.029	10.701

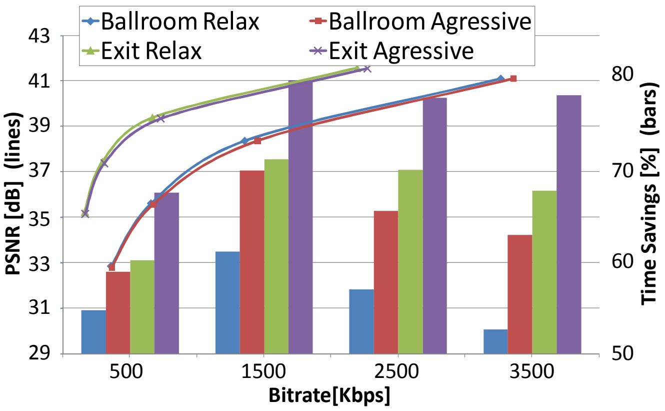


Figure 4.20: Average tested modes (QP={22,27,32,37,42}, GOP=8, Views=8)

View-Level Time Saving Evaluation

A view-wise ΔPSNR and time savings comparison of *Relax* and *Aggressive* levels is provided in Figure 4.21 for the *Exit* sequence encoded using QP=32. Odd views – with north and south views (i.e. Views 1, 3, 5) available in the neighborhood – present higher time savings compared to the views with just one (i.e. Views 2/4/6/7) or none available neighboring views (i.e. View 0). Additionally, Views 1, 3 and 5 also present a smaller PSNR loss. This higher complexity reduction and reduced PSNR loss is due to the larger correlation space in the 3D-Neighborhood. It implies that more neighboring MBs are available for the prediction. Consequently, a more accurate *CandidateList* is generated.

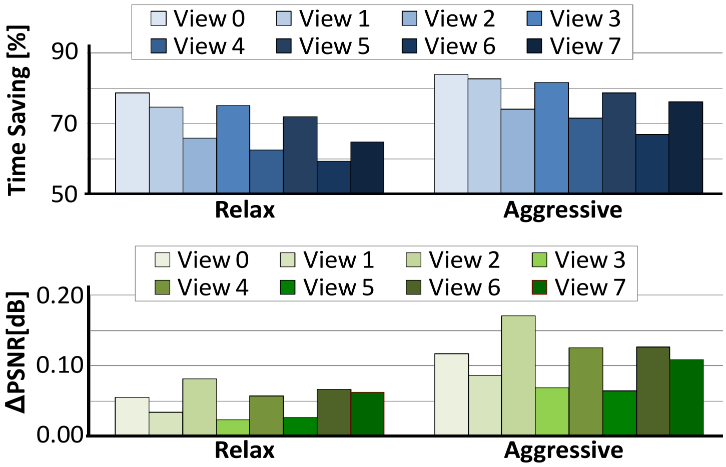


Figure 4.21: View-level time savings and ΔPSNR comparison of Relax and Aggressive levels (Exit sequence, QP=32)

Tested Modes Evaluation

The high time savings provided by the multi-level mode decision comes from the reduced number of coding modes tested. Figure 4.22 provides the result of the average number of modes evaluated per MB considering the different operation modes. The *Relax* and *Aggressive* levels of our complexity reduction scheme process only 3.7 and 2.3 modes per MB, respectively.

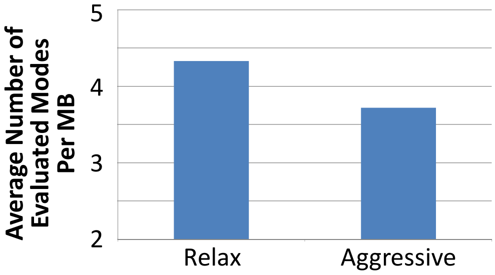


Figure 4.22: Average tested modes for all sequences

The distribution of the evaluated modes for *Relax* and *Aggressive* complexity reduction levels is presented in Figure 4.23. It is noted that the number of SKIP mode increases for higher QPs while the number of other modes decreases accordingly. This behavior confirms the analysis of optimal mode distribution discussed in Section 4.1.1.1. For QP 32 and above, the number of evaluated modes increases to maintain a high video quality.

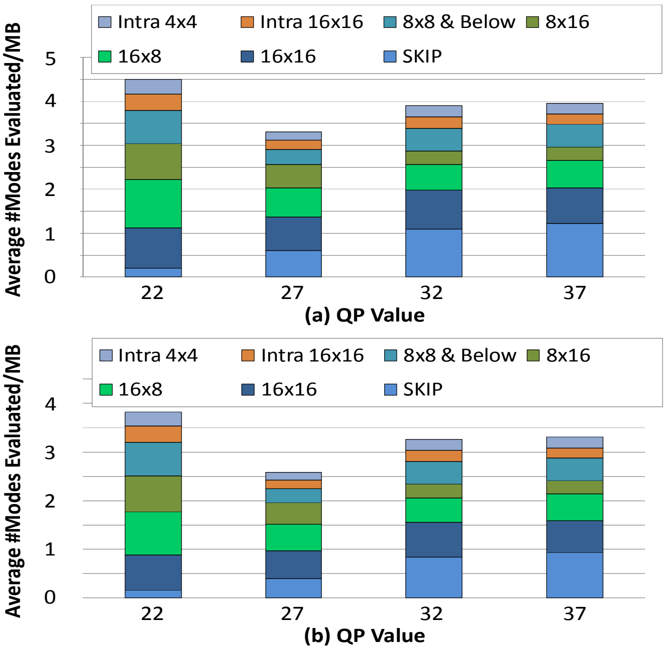


Figure 4.23: Detailed number of evaluated modes for (a) *Relax* and (b) *Aggressive* (Exit Sequence)

Frame-Level Time Saving Evaluation

To analyze the frame-wise comparison of *Relax* and *Aggressive* levels, we have plotted the PSNR and time savings for View 0, 1, and 2 of *Exit* test sequence encoded using QP=32, as shown in Figure 4.24 and Figure 4.25. Please note that the plots only contain results for non-anchor frames, which are the primary focus for complexity reduction in this algorithm.

There are 0, 1, and 2 available neighboring views available for the View 0, View 1 and View 2, respectively (representing all possible cases). View 2 exhibits a higher ΔPSNR and lower time savings while View 1 exhibits higher time savings and a lower ΔPSNR for most of the frames when compared to the other plotted views. This ratifies the view-level results from Figure 4.21.

The sudden variations (i.e. valleys) in Figure 4.25 correspond to the frames in the middle of the GOPs, i.e. frames that have temporal-neighbors from the anchor frames. In this case, more intra modes are evaluated (in phase 3 and 4) in addition to the inter modes leading to a lower complexity reduction. View 1 due to the availability of all view-neighbors suffers less with such variations.

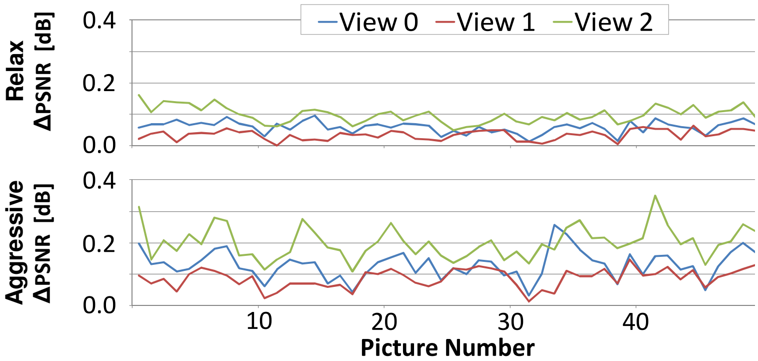


Figure 4.24: Frame-wise PSNR loss comparison of Relax and Aggressive levels (Exit, QP=32)

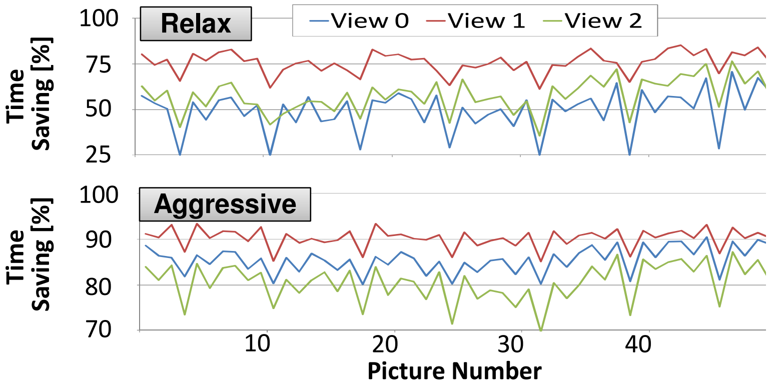


Figure 4.25: Frame-wise time saving comparison of Relax and Aggressive levels (Exit, QP=32)

Multi-Level Mode Decision Algorithm Overhead

The overhead of our complexity reduction scheme is already computed in the total processing time and time savings. Figure 4.26 compares the average overhead of the computational logic of our scheme with the average processing time of one MB encoded using different schemes. It is noted that the overhead is 0.15% of the average MB encoding time using the exhaustive RDO-MD. Figure 4.26 shows that the overhead of our scheme is insignificant compared to its time savings.

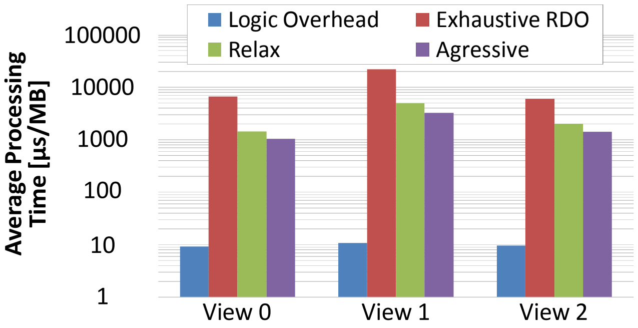


Figure 4.26: Overhead of our scheme

In this section was presented the multi-level fast mode decision algorithm focusing on complexity reduction MVC that exploits the image properties, RDCost and the correlation in the 3D-Neighborhood to provide complexity reduction with insignificant PSNR loss. Our detailed analysis provides the foundation for the proposed scheme. In order to react to the changing QP values, QP-based threshold equations are deployed.

For a tradeoff between the desired complexity reduction and the resulting quality loss, two different operational levels are proposed for our scheme, the *Relax* and *Aggressive* modes. However, to better exploit the complexity reduction vs. RD performance, a control algorithm able to select at run time the most appropriate complexity reduction level is desirable. In the following section an energy-aware complexity adaptation based on fast mode decision is proposed.

4.2 Energy-Aware Complexity Adaptation

Besides the algorithms able to perform the fast mode decision, a complexity adaptation algorithm (SHAFIQUE, ZATT, *et al.*, 2010) is required to adapt the mode decision at run time according to the changing application scenarios. Targeting MVC encoding systems where battery level, user constrains and video content may vary widely along the time, we propose in this section an energy-aware complexity adaptation for MVC targeting mobile devices. Our algorithm employs several *Quality-Complexity Classes* (QCCs), such that each class evaluates a certain set of coding modes (thus a certain complexity requirement) and provides a certain video quality. To support asymmetric view quality, views for one eye are encoded with high quality class and views for the other eye are encoded using a low-quality class. Our algorithm adapts the QCCs for different views at run time depending upon the current battery level.

4.2.1 Employing Asymmetric View Quality

A reduction in the complexity and energy consumption can be obtained by exploiting the binocular suppression theory which is based on the psycho-visual studies of stereoscopic vision (STELMACH e TAM, 1999). According to this study, if the video quality of left and right eye views differ, the overall perceived quality is close to the high-quality sharper view (STELMACH e TAM, 1999). However, for a blocky image, the perceived quality is the average of left and right eye views. Based on the binocular suppression theory (STELMACH e TAM, 1999) and considering the in-loop deblocking filter of MVC (JVT, 2008) (that reduces the blocking artifacts), views for two eyes can be encoded at different qualities (i.e., exploiting asymmetric view quality), thus requiring different computational complexity.

Figure 4.27 shows the MVC prediction structure for a four-view scenario employing asymmetric view quality. Assuming that the viewer is always exposed to adjacent views (S_n and S_{n+1}), the even views (S0 and S2) are encoded in higher quality while odd views (S1 and S3) are encoded in lower quality. In such way, the viewer sees one high quality and one low quality view resulting in a perception near to the high quality view. The use of high quality in even views is explained by the fact they are used as reference to odd views.

Although in (STELMACH e TAM, 1999) the low quality frames were synthetically blurred for analysis, this knowledge can be extended to a real scenario and applied in techniques to reduce the MVC coding complexity. In our scheme, the odd views will be submitted to more aggressive mode decision resulting in a lower quality in relation to their neighboring views.

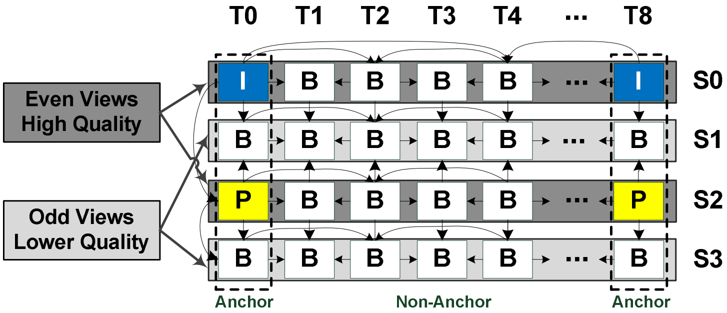


Figure 4.27: MVC coding structure for asymmetric coding

In the following section is presented the energy-aware complexity adaptation algorithm besides of the *Quality Complexity Classes* (QCCs) and *Quality States* (QS) description.

4.2.2 QCCs: Quality Complexity Classes

In order to employ the asymmetric view quality and the battery level sensitivity to our scheme we define three *Quality-Complexity Classes* (QCCs).

QCC 1: MBs of *QCC 1* are exposed to the more aggressive mode decision of our scheme including SKIP and Inter 16x16 modes. Therefore, they have the lowest video quality and the higher complexity reduction.

QCC 2: This class presents the intermediate video quality and complexity reduction. Modes of *QCC 1* plus Intra 16x16, Inter 16x8, 8x16 and 8x8 are evaluated.

QCC 3: More computational complex class and, consequently, the one that provides better video quality. It includes the coding modes available in *QCC 1* and *QCC 2* plus small blocks such as Intra 4x4, Inter 8x4, 4x8 and 4x4.

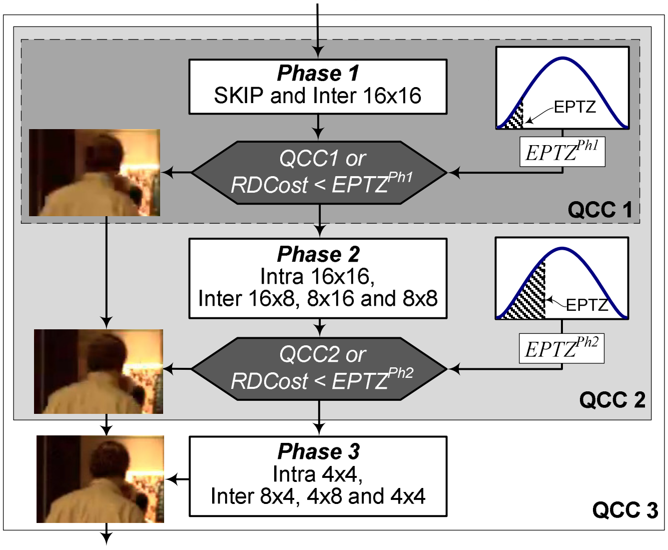

Phase 1
SKIP and Inter 16x16
EPTZ
EPTZ^Ph1
QCC1 or RDCost < EPTZ^Ph1
QCC 1
Phase 2
Intra 16x16, Inter 16x8, 8x16 and 8x8
EPTZ
EPTZ^Ph2
QCC2 or RDCost < EPTZ^Ph2
QCC 2
Phase 3
Intra 4x4, Inter 8x4, 4x8 and 4x4
QCC 3

Figure 4.28: Energy-Aware Complexity Adaptation MVC Scheme

Figure 4.28 presents the high level diagram of our scheme shown the mode decision flow. The QCCs are related to three different prediction phases according to the dashed blocks in Figure 4.28. *QCC 1* is subject to Phase 1; *QCC 2* to Phase 1 and Phase 2; and *QCC 3* is subject to Phase 1, Phase 2 and Phase 3. However, even for *QCC 2* and *QCC 3* big part of MBs are SKIP or Inter 16x16 and there is no need to test small block sizes. For this reason, the early prediction terminator zone (EPTZ) was defined.

4.2.3 Mode Decision Algorithm for Different QCCs

The proposed algorithm is presented in Figure 4.29. It is composed of three phases. In Phase 1 the RDCost of SKIP and Inter 16x16 are calculated. If the current MB is *QCC 1* or the RDCost of one of the predicted modes is under the $EPTZ^{Ph1}$ limit the MD is terminated. Phase 2 calculates the RDCost for Intra 16x16 and one out of three inter modes, 16x8, 8x16 and 8x8, depending upon the gradient direction. The MD is terminated if the MB is *QCC 2* or the best RDCost is lower than $EPTZ^{Ph2}$ limit. For MBs *QCC 3* with RDCost higher than $EPTZ^{Ph2}$ one of four small block modes (Intra 4x4, Inter 8x4, 4x8 and 4x4) is tested. Finally, the best prediction mode, i.e. mode of lowest RDCost, is used to encode the current MB.

```
MB Mode Decision (Current MB)
  //Phase 1
Calculate RDCost (SKIP, Inter16x16);
If (Class 1 or $RDCost<EPTZ^{Ph1}$)
  Exit;
End If
  //Phase 2
Calculate RDCost (SKIP, Intra16x16);
If (Gradient_Horiz > 1.25* Gradient_Vert)
  Calculate RDCost (Inter8x16);
Else If (Gradient_Horiz < 0.8* Gradient_Vert)
  Calculate RDCost (Inter16x8);
Else
  Calculate RDCost (Inter8x8);
End If
If (Class 2 or $RDCost<EPTZ^{Ph2}$)
  Exit;
End If
  //Phase 3
For (all 8x8 partitions)
  If (Gradient_Horiz > 1.25* Gradient_Vert)
    Calculate RDCost (Inter4x8);
  Else If (Gradient_Horiz < 0.8* Gradient_Vert)
    Calculate RCost (Inter8x4);
  Else
    If (Inter16x16< Intra16x16)
      Calculate RDCost (Inter4x4);
    Else
      Calculate RDCost (Intra4x4);
    End If
  End If
End For
Encode MB (Mode with lowest RDCost);
```

Figure 4.29: Pseudo-code of mode decision for different QCCs

4.2.4 RDCost-based Thresholding

Based on experiments considering the optimal RDO-MD we observed that the RDCost distribution varies according to QP values. To better evaluate and quantify this observation we analyzed the RDCost PDF. As presented in Figure 4.30, the RDCost for low QPs is concentrated in low range values (resulting from a small average μ and standard deviation σ) while for high QPs, the PDF is more disperse and centered around a larger value (larger μ and σ).

With the RDcost characterization we defined the EPTZ and being $RDCost_{SKIP} < \mu_{RD}-\sigma_{RD}$ for *QCC 2*. In other words, if the best RDCost for previously tested modes is within EPTZ (see PDFs in Figure 4.28 and Figure 4.30) the mode decision is terminated. For *QCC 3* there are two early termination points, one at Phase 1 defined as $RDCost_{SKIP} < \mu_{RD}-2*\sigma_{RD}$ and other in Phase 2 $RDCost_{SKIP} < \mu_{RD}-\sigma_{RD}$. Once the distribution is different for each QP, the EPTZ limit approximated by polynomial curve fitting is given by the following QP-based Eq. (4.14), (4.15) and (4.16).

$$(EMT^{Ph1})_{QCC2}=(EMT^{Ph2})_{QCC3}=RDCost_{AVG} - 1.5\ RDCost_{SD} \quad (4.14)$$

$$(EMT^{Ph1})_{QCC3}=RDCost_{AVG} - 0.5\ RDCost_{SD} \quad (4.15)$$

$$(EMT^{Ph1})_{QCC2}=(EMT^{Ph2})_{QCC3}=29.663QP^{2}-1409.1QP+18766 \quad (4.16)$$

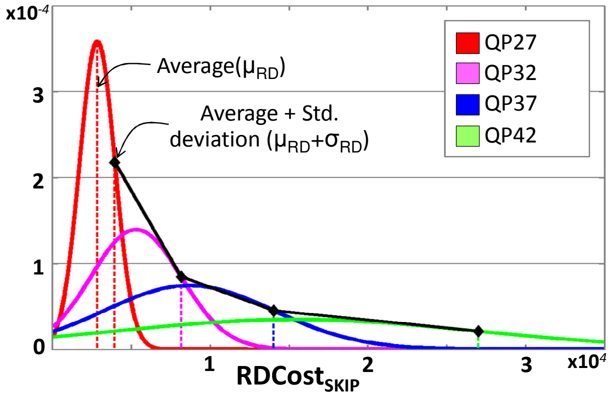


Figure 4.30: Probability Density Function for RDCost

4.2.5 Energy-Aware Complexity Adaptation

Associated to the QCCs, our scheme employs four different *Quality States* (QS). The QSs consider the binocular suppression theory (STELMACH e TAM, 1999) using asymmetric view quality and react, at run-time, to the changing battery level. As summarized in Table 4.3, *QS1* presents the highest quality and encode all views as *QCC3. In turn, QS2* and *QS3* use the view quality asymmetry encoding odd view in lower quality than even views. *QS4* provides the lowest quality and highest complexity reduction coding all views as *QCC1*.

Table 4.3: Quality States

Quality State	Video Quality	Even Views	Odd Views
Quality State 1(QS1)	High Quality	*QCC3*	*QCC3*
Quality State 2(QS2)	Medium Quality	*QCC3*	*QCC2*
Quality State 3(QS3)	Low Quality	*QCC2*	*QCC1*
Quality State 4(QS4)	Lowest quality for battery saving	*QCC1*	*QCC1*

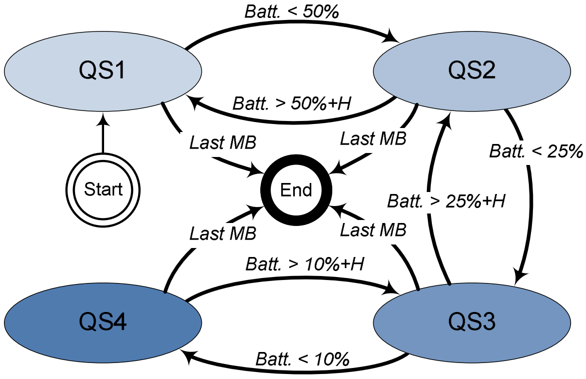


Figure 4.31: Run-time complexity adaptation state machine

The QS control is performed by one state machine that receives an indication of the battery level as input. Figure 4.31 presents the transitions between the four possible states. The quality states just change to the immediately superior or inferior quality in order to have smooth video quality variation. The hysteresis (*H*) is fixed as 5% in order to avoid quick oscillations between different states and, consequently, video quality fluctuations. This state machine can be easily adapted to consider other external parameters such as user presets and time constrains.

4.2.6 Energy-Aware Complexity Adaptation Evaluation

This section presents the detailed experimental results for each *Quality State* of the proposed energy-aware complexity adaptation algorithm compared to the RDO-MD. The, overall results for the complexity adaptation algorithm are presented in Section 6.2.1.2. The experiments used the experimental setup described in Section 6.1.

Table 4.4 presents the detailed PSNR, bitrate (BR) and time saving (TS) results of the 4 *Quality States* of our scheme compared to the exhaustive mode decision (RDO-MD). For the *QS1* state our scheme provides a TS of up to 77% with negligible PSNR loss (avg. 0.089 dB). The TS goes up to 87% for *QS4* with an average PSNR loss of 0.195 dB.

To calculate the objective quality of a sequence we consider the average PSNR between all possible stereo view points (VP) of a sequence. For example, a sequence with four views has three stereo VPs (View 0 and View 1, View 1 and View 2, View 2 and View 3). To calculate the PSNR of a given VP considering the binocular suppression we use the Eq. (4.17), as proposed in (OZBEK, TEKALP e TUNALI, 2007).

$$PSNR^{VP}=(1\text{-}\alpha).PSNR^{HighQuality}+\alpha.PSNR^{LowQuality};\ \alpha=1/3 \tag{4.17}$$

Table 4.4: Comparison between the Quality States (QS)

	QP	Ballroom			Exit			Vassar			Rena			QP Average			Total Average		
		TS	ΔPSNR	ΔBR	TS	ΔPSNR	ΔBR	TS	ΔPSNR	ΔBR	TS	ΔPSNR	ΔBR	TS	ΔPSNR	ΔBR	TS	ΔPSNR	ΔBR
QS1	22	75.71	0.095	-3.45	77.28	0.093	-4.13	75.21	0.084	-7.143	82.26	0.292	-3.221	77.62	0.14	-4.49	75.29	0.089	-5.48
	27	75.13	0.074	-5.33	77.28	0.072	-4.75	73.84	0.031	-8.527	79.64	0.207	-2.186	76.47	0.10	-5.20			
	32	73.15	0.075	-2.36	76.45	0.070	-5.70	71.97	0.015	-12.22	76.18	0.094	-1.493	74.44	0.06	-5.44			
	37	71.62	0.031	-1.24	75.95	0.003	-6.05	70.80	0.115	-15.49	72.24	0.073	-4.429	72.65	0.06	-6.80			
QS2	22	77.96	0.096	0.70	79.29	0.094	0.07	77.68	0.086	-2.344	84.18	0.291	0.305	79.78	0.14	-0.32	76.96	0.093	-0.53
	27	76.97	0.079	1.84	78.60	0.072	0.58	75.18	0.036	-2.381	81.21	0.210	1.193	77.99	0.10	0.31			
	32	75.22	0.089	2.01	78.08	0.077	0.69	73.49	0.023	-2.294	77.26	0.099	0.707	76.01	0.07	0.28			
	37	73.27	0.053	1.54	77.22	0.027	0.07	72.51	0.091	-8.604	73.30	0.063	-2.573	74.07	0.06	-2.39			
QS3	22	84.86	0.116	4.38	84.52	0.112	4.32	85.81	0.097	1.676	85.67	0.280	4.260	85.22	0.15	3.66	82.64	0.123	4.76
	27	83.98	0.088	6.42	83.43	0.084	7.45	83.61	0.050	2.660	83.31	0.228	3.919	83.58	0.11	5.11			
	32	82.86	0.124	6.14	82.73	0.112	8.20	81.57	0.056	3.974	80.23	0.186	3.536	81.85	0.12	5.46			
	37	81.43	0.131	5.48	81.53	0.113	7.83	80.03	0.059	3.790	76.66	0.127	2.116	79.91	0.11	4.80			
QS4	22	87.96	0.148	6.42	86.93	0.140	6.75	87.86	0.123	2.456	87.54	0.318	6.106	87.57	0.18	5.43	85.26	0.195	7.40
	27	87.29	0.112	8.95	85.40	0.117	11.39	85.59	0.065	3.690	85.83	0.320	6.175	86.03	0.15	7.55			
	32	86.42	0.173	9.24	84.57	0.205	13.08	83.58	0.086	5.536	83.53	0.364	6.530	84.53	0.21	8.60			
	37	85.44	0.239	8.96	83.32	0.264	13.02	82.20	0.090	5.277	80.74	0.356	4.782	82.93	0.24	8.01			

4.3 Fast Motion and Disparity Estimation

According to the motivational analysis presented in Section 3.1.1 and challenges discussed on Section 3.2, the two main sources of complexity and energy consumption in the MVC encoder are the mode decision and the motion and disparity estimation units. Along Section 4.1 distinct solutions for reducing the complexity and energy for the MD were proposed. Moreover, an energy-aware complexity adaptation based on mode decision was presented in order to enable run-time adaptivity to changing system and content scenarios. In this section the target is to present solutions to reduce the complexity and energy consumption associated to the second main complexity source, the ME/DE unit.

In this section is presented a correlation analysis related to motion and disparity vectors (MV, DV) followed by a Fast ME/DE algorithm (ZATT, SHAFIQUE, *et al.*, 2011). Our Fast ME/DE algorithm was designed taking into account a future hardware implementation.

4.3.1 Motion Vector Correlation Analysis

Before proceeding to the novel fast ME/DE algorithm proposed and to the motion/disparity vectors correlation analysis, we briefly recall the basic prediction structure of MVC to a level of detail necessary to understand the novel contribution. MVC uses the motion and disparity estimation tools to eliminate the temporal and view redundancies between frames, respectively. The prediction structure used in this work is presented in Figure 4.32. *I* squares represent intra-predicted frames (i.e. no ME/DE is used), *P* are frames using unidirectional prediction or estimation (in this example the *P* frames use only DE in one direction), and *B* frames use bidirectional prediction having reference frames in at least two directions. The arrows represent the prediction directions: frames at the tail side act as reference frames to the frames pointed by the arrowheads. Note that some frames have up to four prediction directions. In order to provide random access points, the video sequence is segmented in Groups of Pictures (GOPs) where the frames located at the GOP borders are known as anchor frames and are encoded with no reference to the previous GOP. All other frames are called non-anchor frames.

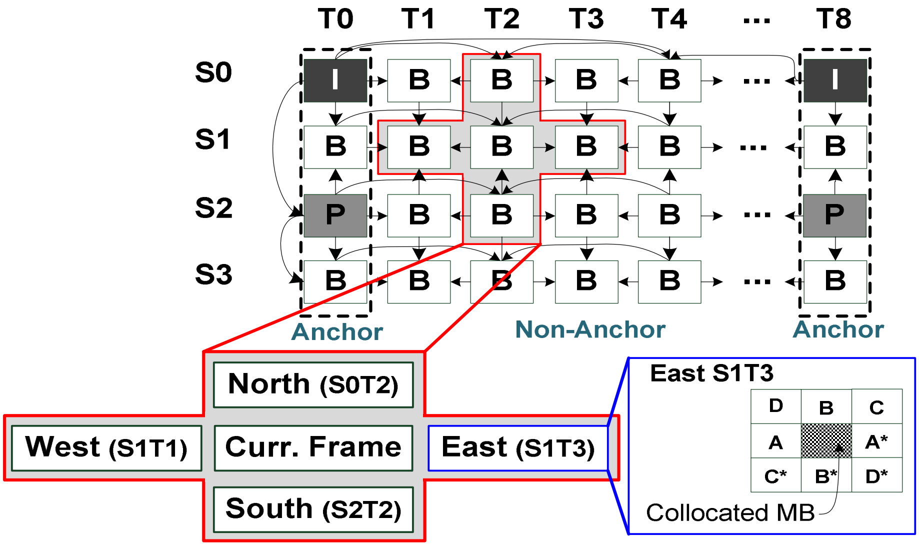


Figure 4.32: MVC prediction structure and 3D-Neighborhood details

In our observations we noticed that the same objects in a 3D scene are typically present in different views (except for occlusions). Consequently, the motion perceived in one view is directly related to the motion perceived in the neighboring views (DENG, JIA, *et al.*, 2009). Moreover, considering parallel cameras, the motion field is similar in these views (KIM, KIM e SOHN, 2007). Analogously, the disparity of one given object perceived in two cameras remains the same for different time instances when just translational motion occurs. Even for other kinds of motion the disparity is highly correlated.

Based on these observations an analysis of the motion and disparity vectors is performed to quantify the MV/DV correlation (here the term correlation is subjectively used, it is defined as the difference between the predictors and the optimal vector, i.e. MV/DV error) in the 3D-Neighborhood (i.e. spatial, temporal and view domains). A set composed of 1 spatial median predictor, 6 temporal predictors and 6 disparity predictors is analyzed. The temporal predictors are selected from the previous and next frames (in the displaying order) called West and East neighbor frames, respectively. For each neighboring frame, three predictors are calculated. They are (a) the collocated MB (MB in the reference frame with the same relative position of the current MB), (b) median up (using the median formula specified by the MVC standard (JVT, 2009) to calculate the spatial predictor), and median down (median of A*, B*, C* and D* as shown in Figure 4.32). The disparity predictors from the North and South neighboring view frames are obtained by considering the GDV.

Figure 4.33 illustrates the MV/DV error distribution for *Vassar* (low motion) and *Ballroom* (high motion) test video sequences in the 3D-Neighborhood. Each plot represents the difference between a given predictor (in this case for the spatial predictor and 3 collocated predictors in different neighboring frames) and the optimal vector of the current MB. It shows that for the majority of the cases, the predictor vectors have similar values in comparison to the optimal vector. Even, most of the predictors have exactly the same value of the optimal vector. Our analysis shows that this observation is valid for the other nine predictors in all direction of the 3D-Neighborhood as depicted in Figure 4.33 (only few error plots are shown here).

To quantify the MV/DV error distribution in the 3D-Neighborhood, several experiments were carried out to measure the frequency in which a given predictor is equal to the optimal vector (i.e. $MV_{Pred} = MV_{Curr}$). When this condition is satisfied, it is denoted as a so-called *hit*. A set of different conditions was defined including, for example, the case when all predictors of a given neighbor frame (collocated, median up and median down) are *hits*. Table 4.5 presents the detailed information on the vector *hits* where the *Availability* is the percentage of cases when that predictor is available. The disparity predictors present the higher number of *hits* followed by the spatial and temporal predictors. Considering the quality of predictors in the same neighboring frame, the collocated predictors present better results in relation to median up and median down. The latter two present similar number of *hits*. In the case where all predictors of a given neighboring frame are available, the predictor is highly accurate providing up to 98% *hits*.

In conclusion, there is a high vector correlation available in the 3D-Neighborhood that can be exploited during the ME/DE processing. Once the predictors point to the same region as the optimal vector, there is no need for search patterns exploiting a large search range. Moreover, for most of the cases the predictors' accuracy is enough to completely avoid the ME/DE search and refinement stages.

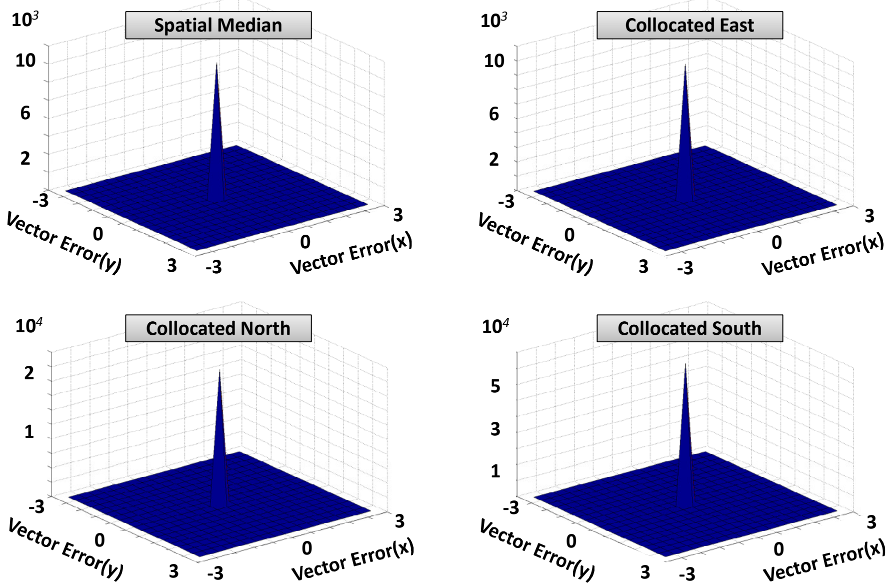


Figure 4.33: MV/DV error distribution between predictors and optimal vector (Ballroom, Vassar)

Table 4.5: Predictors Hit Rate and Availability

Predictor	Neighbor Frame	Hit [%]	Available [%]		Neighbor Frame	Hit [%]	Available [%]
Spatial	**n.a.**	94.12	99.90	**-**	**-**	-	-
All	***West***	96.94	51.52	**Median Up**	***West***	54.74	99.90
	East	97.93	60.30		***East***	63.78	99.90
	North	97.94	65.40		***North***	93.17	73.99
	South	98.67	21.29		***South***	94.61	23.98
Collocated	***West***	58.43	99.90	**Median Down**	***West***	54.99	99.89
	East	66.79	99.90		***East***	63.92	99.89
	North	95.39	72.39		***North***	93.21	74.13
	South	96.75	23.48		***South***	94.70	23.93

4.3.2 Fast Motion and Disparity Estimation Algorithm

Our Fast ME/DE scheme is based on the above-presented 3D-Neighborhood analysis. However, to exploit this correlation the motion and disparity fields must be available. In order to establish these fields at least one frame using DE and one using ME must be encoded with the optimal or a near-optimal searching algorithm. In our scheme, to avoid major quality loss, all anchor frames and the frames situated in the middle of the GOP are encoded using the TZ search algorithm (the fast ME/DE algorithm used in JMVC (JVT, 2009)). The anchor frames are encoded using DE, while the frames in the middle of a GOP use ME or ME and DE according to the view they belong. These frames encoded with high effort are herein referred as Key Frames (KF), while the others are the Non-Key Frames (NKF). Once the motion and disparity fields are available all NKF can be encoded based on these fields. The complete ME/DE search pattern is skipped for all NKF. It only uses the predictors inferred from the 3D-Neighborhood.

Figure 4.34 presents the flow diagram of our proposed fast ME/DE scheme based on the 3D-Neighborhood vectors correlation. It employs two different prediction classes: Ultra Fast Prediction and Fast Prediction. The scheme is composed of three main phases: (i) Frame Level MV/DVs evaluation; (ii) MB Level MV/DVs Evaluation and Prediction, and (iii) MV/DV Storage. Figure 4.34 considers only the NKF coding. For KF the TZ search algorithm is used (as discussed earlier) to provide a good motion/disparity field.

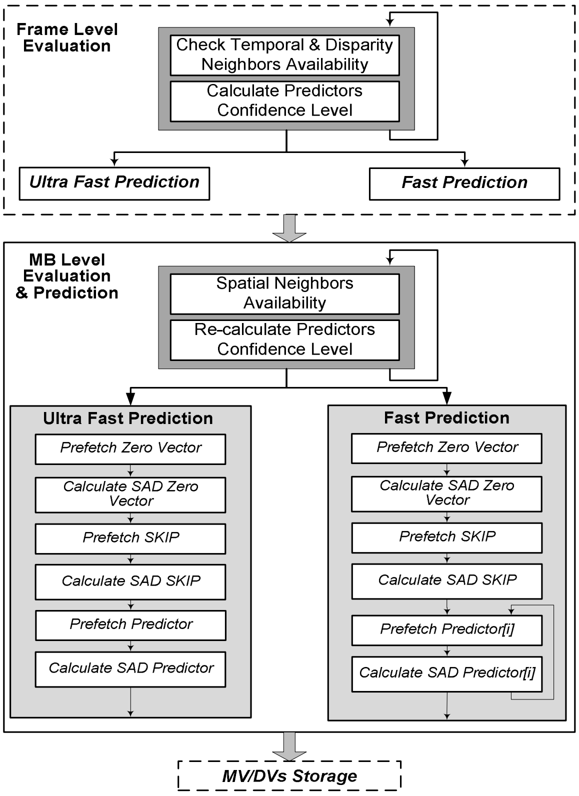


Figure 4.34: Flow diagram of the adaptive fast ME/DE

In Frame Level MV/DVs evaluation, the presence of all temporal-disparity predictors is checked. If available, they are read from a MV/DV memory. The spatial predictor is not loaded in this phase since it is not available due to the spatial dependencies, if a hardware architecture scenario is considered . With the available data, the current MB is pre-classified in one out of the two prediction classes according to the predictors' Confidence Level. Note, the spatial predictor is required for the SKIP vector calculation. So, this predictor is also considered in our algorithm to classify the current MB in the MB Level evaluation and prediction phases.

The predictors Confidence Level is calculated based on the offline *hit* value, as presented in Table 4.5. Each predictor is associated to a Confidence Level (*hit* value). If one predictor has a Confidence level higher than a threshold ($CL_{Pred} > CL_{TH}$), the current MB is classified to be encoded as Ultra Fast Prediction. Otherwise, the MB is classified to be encoded with the Fast Prediction. In case of Ultra Fast Prediction MBs, only three vectors are tested: the predictor with highest Confidence Level (also referred as Common Vector), the Zero vector and the SKIP vector. The Zero and SKIP vectors are tested because of their high occurrence. Fast Prediction MBs test all available predictors in addition to the Zero and SKIP vectors. It is important to mention that even if all predictors are available and different (this worst case rarely occurs), only 13 predictors are tested.

4.3.3 Fast ME/DE Algorithm Evaluation

In Table 4.6 the fast ME/DE results are detailed for the four evaluated sequences considering three different QPs (22,32,42). The TZ Search with a search range of [±64, ±64] is used for comparison as it is used for the Key frames and performs 23x faster compared to the Full Search (not used for performance comparison), while providing the similar rate-distortion results (YANG, 2009). Compared to the TZ Search, our fast ME/DE provides 83% execution time saving at the cost of 11% increase in bitrate and 0.114dB of PSNR loss. In the best case, the execution time savings go up to 86%, which represents a significant complexity reduction. Moreover, the reduced number of candidate blocks leads to a lower number of external memory accesses.

Table 4.6: Comparison of Our Fast ME/DE Algorithm to TZ Search

Video	QP	*TZ Search*			Fast ME/DE		
		Time [sec]	BR [kbps]	PSNR [dB]	TS [%]	ΔBR [%]	ΔPSNR [dB]
Ballroom	22	215.1	3298.026	40.709	85.9	8.4	0.011
	32	175.7	651.640	35.119	86.2	11.7	0.060
	42	127.1	188.178	29.318	84.9	19.8	0.190
Vassar	22	171.1	3415.744	40.456	83.0	1.1	0.010
	32	110.0	315.142	35.226	82.4	6.7	0.013
	42	72.1	64.888	30.563	79.7	6.7	0.043
Breakdancers	22	583.5	5680.204	41.172	86.0	15.7	0.087
	32	384.6	788.800	38.010	85.1	14.6	0.275
	42	262.9	277.970	33.803	82.8	8.9	0.304
Uli	22	600.9	12245.676	39.819	82.9	9.3	0.053
	32	516.8	2949.870	34.960	83.0	13.7	0.134
	42	406.5	849.462	28.944	82.4	8.5	0.191
Poznan_Hall2	22	1206.4	6245.748	42.654	86.8	8.2	0.012
	32	811.2	1002.354	40.138	86.1	10.3	0.017
	42	726.8	521.816	35.465	84.5	7.7	0.027
GT_Fly	22	1321.8	7123.971	40.980	81.3	10.0	0.103
	32	954.8	1113.591	38.057	81.9	17.3	0.265
	42	801.0	613.548	33.849	79.8	12.3	0.298
Average	22	683.1	6334.895	40.965	84.3	8.8	0.046
	32	492.2	1136.900	36.918	84.1	12.4	0.127
	42	399.4	419.310	31.990	82.4	10.7	0.176
	Avg.	524.9	2630.368	36.625	83.6	10.6	0.116

Figure 4.35 presents the RD-curves for two XGA (1024x768) video sequences *Uli* and *Breakdancers*. It is noted that the RD curves of our fast ME/DE algorithm are close to that of the Full Search (used for quality comparison only). Compared to the Full Search, our fast ME/DE algorithm suffers from an insignificant loss of 0.116dB (on average).

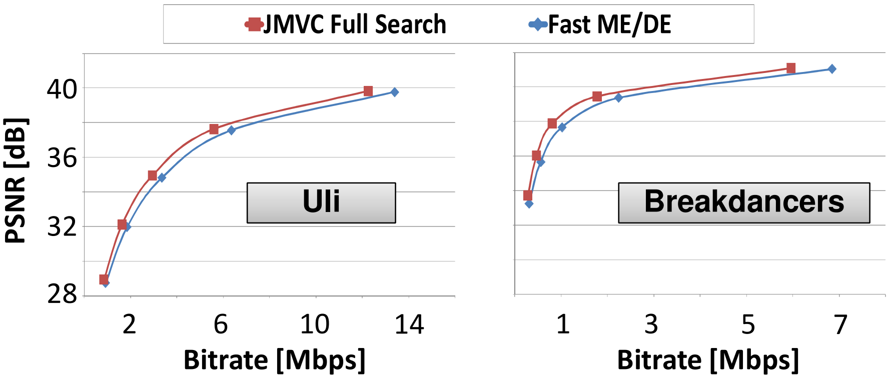


Figure 4.35: Rate-Distortion Comparison with Full Search

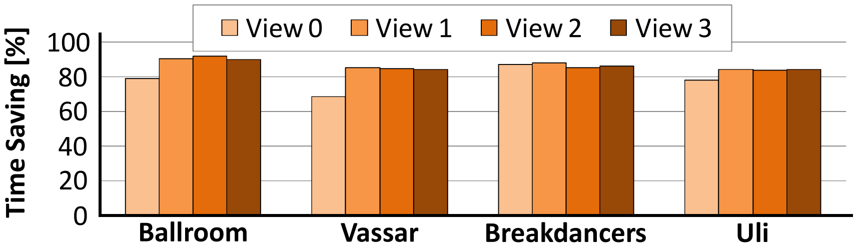


Figure 4.36: View-level execution time savings compared to TZ Search

The view-level execution time savings are presented in Figure 4.36. Note that for all views (except for View 0 of *Vassar* sequence) the time saving is ≥80%. The execution time savings for the high-motion sequences are slightly more than that in the low-motion sequences. Figure 4.37 presents the average number of SAD operations for ME/DE of one MB using Full-Search, TZ Search and, our fast ME/DE algorithm. Averagely, the proposed scheme reduces more than 99.9% in comparison to Full Search and 86% to TZ Search. Note, the detailed results in Figure 4.36 and Figure 4.37 are for QP32.

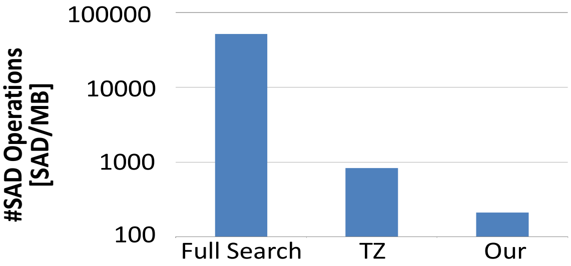


Figure 4.37: Comparison of the number of SAD operation

4.4 Video Quality Management for Energy-Efficient Algorithms

Although the energy and complexity reduction algorithms presented along this chapter were carefully designed to reduce undesirable effects to the coding efficiency, they lead to some level of quality drop due the heuristics and simplifications inserted in the encoding process. For this reason, an algorithm to manage the coding process and compensate eventual video quality losses is required. This management algorithm, however, must also consider and optimize the video quality and bitrate tradeoff in order to increase the coding efficiency while respecting to bandwidth constrains as discussed in the following.

Despite of the high coding efficiency provided by MVC, the transmission and storage of 3D videos remains a big challenge, especially for services operating over bandwidth/buffer-constrained infrastructures. It becomes even more challenging due to changing input video properties, run-time variations on video encoder state, battery level and user preferences. Thus, to provide high video quality while meeting channel bandwidth/buffering constraints it is necessary to further optimize the bandwidth usage by intelligently regulating the bits allocation. Therefore, a rate control algorithm is implemented to dynamically find a good compromise between the coding efficiency and video quality by adapting the QP.

In this section is presented the Hierarchical Rate Control (HRC) (VIZZOTTO, ZATT, *et al.*, 2012) for MVC that employs coupled Model Predictive Control-based frame-level RC and Markov Decision Process-based BU-level RC. Before presenting the HRC, however, a bitrate allocation study within the 3D-Neighborhood is detailed.

4.4.1 Bitrate Correlation Analysis

In this section we present a detailed bitrate distribution analysis to provide a better understanding towards the bitrate distribution during the MVC encoding process and its correlation with spatial, temporal and disparity neighborhood. The analysis is presented in a top-down approach starting with the view-level related discussion, following to frame-level and concluding with BU-level considerations. For that, we used eight views of the *Flamenco2* VGA video sequence encoded at a fixed QP, that is, without rate control, for an IBP view coding (0-2-1-4-3-6-5-7) order and Hierarchical Bi-prediction at temporal domain. One basic unit is defined as one macroblock.

Figure 4.38 shows the uneven bitrate distribution along different views. This distribution is highly related to the prediction hierarchy inside a GGOP. The View 0 or Base View is encoded independently with no inter-view prediction. It leads to reduced possibilities of prediction and, consequently, worse prediction, more residues and higher bitrate. B-Views (View 1, 3 and 5) fully exploit the inter-view correlation by performing disparity estimation (in addition to spatial and temporal predictions) to upper and bottom neighboring views. This increased prediction decision space results in improved prediction quality and tends to lead to reduced bitrates. P-Views (View 1, 3, 5, and 7) represent the intermediate case performing disparity estimation in relation to a single neighboring view. P-Views typically present bitrate in the range between Base View and B-Views bitrates. Note, in Figure 4.38 the View 7 is a P-View but its reference view is closer if compared to other P-Views. While View 2 is two views distant to its reference view (View 0), View 7 is just one view distant to View 6. It usually results in a reduced bitrate for View 7 due better disparity estimation prediction.

The bitrate relations associated to prediction hierarchy, however, are not always true and vary with the video/image properties of each view. For instance, in the example provided in Figure 4.38, View 6 (P-View) present reduced bitrate in relation to View 1 and View 3 (both B-Views). Thus, we may conclude that even employing Bi-prediction at disparity domain the View 1 and 3 are harder to predict in relation to View 6 and produce higher bitrate. Similar observation is the increased bitrate generated by View 7 if compared to other P-Views. Reduced bitrate is expected but for View 7 an increased bitrate is measured. These observations show that besides of the relation to the prediction structure (as discussed above), the bitrate distribution has a high dependence on the video content of each view. Hard-to-predict views typically present high texture and/or high motion/disparity objects and require more bits to reach a given video quality.

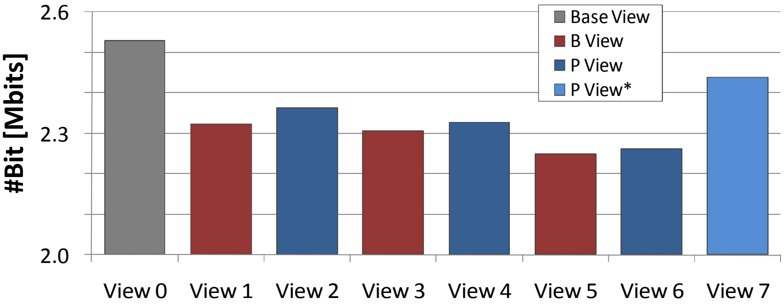


Figure 4.38: View-level bitrate distribution (Flamenco2, QP=32)

The bitrate distribution at frame level presented in Figure 4.39 shows that inside each GOP the frames that present higher bitrate are located at lower hierarchical prediction levels. This is related to the distance of temporal references, the farther the reference the more difficult is to find a good prediction. Therefore, more error is inserted resulting in higher bitrates. In B-Views this effect is attenuated once this view is less dependent to temporal references due to the higher availability of disparity references. Figure 4.39 illustrates that for neighboring GGOPs the frames at same relative position exhibit similar and periodic rate distribution pattern, the GOP-Phase.

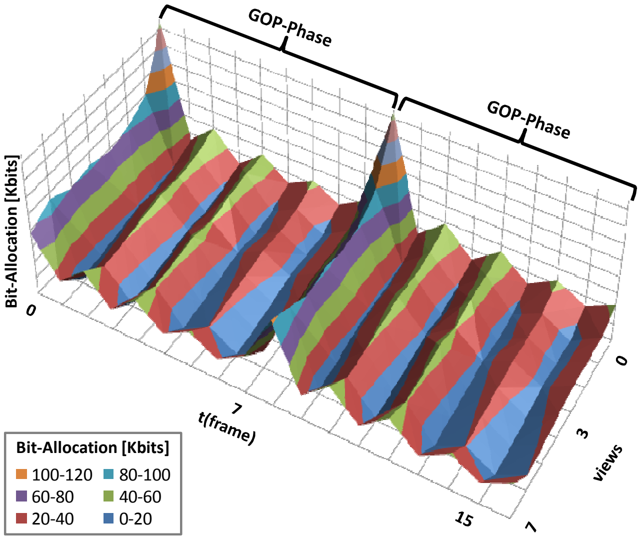


Figure 4.39: Frame-level bitrate distribution for two GGOPs (Flamenco2, QP=32)

Inside each frame the number of bits generated for each BU is also related to the video content. Figure 4.40 shows that the homogeneous and low motion/disparity background require lower bitrate if compared to the dancers region and to the textured floor for a similar quality. However, the Human Visual System (HVS) requires a higher level of details for textured and border regions to perceive good quality and, consequently, these regions deserve higher objective quality. Therefore, textured regions must be detected and receive further increased number of bits during the encoding process through QP reduction.

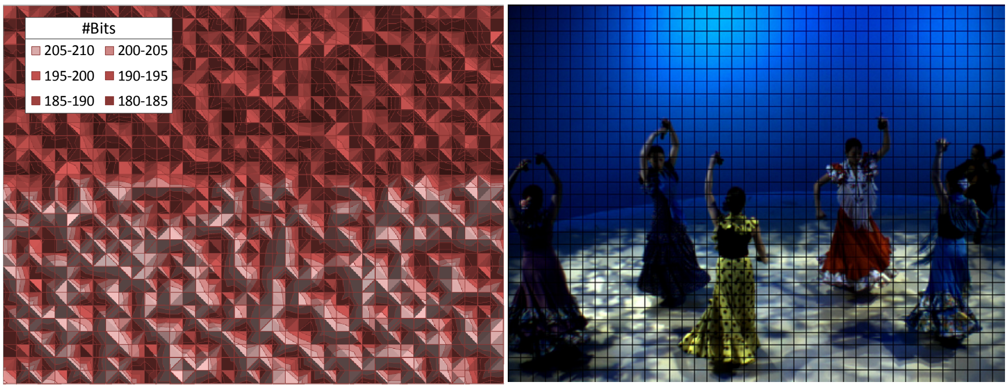


Figure 4.40: Basic Unit-Level bitrate distribution (Flamenco2, QP=32)

Summary: The frame-level bitrate distribution depends on the prediction hierarchy and the video content of each frame. Due correlation of video content, an effective rate control must consider the neighboring frames at temporal, disparity and GOP-phase domains. The video properties have to be considered at BU-level in order to locate and prioritize regions that require higher quality.

4.4.2 Hierarchical Rate Control for MVC

In this section is presented the proposed Hierarchical Rate Control (HRC) for MVC, depicted in Figure 4.41. The HRC is responsible for controlling the encoder output bitrate, in accordance to the user preferences and/or channel limitations, by monitoring the MVC encoder and actuating through QP adaptation. It can be conceptually divided in two actuation levels: (i) frame-level (that encapsulates GOP and frame levels) at coarse grain and; the (ii) basic unit-level at fine grain. The MVC encoder receives the video sequences as input along with all user preferences and configurations to start the encoding process. The Model Predictive Control-based frame-level RC models the system behavior considering the encoding hierarchy and predicts the bitrate allocation at frame-level considering temporal, view and GOP-phase (inter-GOP) correlation. It defines the optimal QP for the predicted frames, the base QP, and forward it to the Markov Decision Process-based basic unit-level RC. At BU-level, a fine grained-decision is taken to define the QP variation considering the image properties in terms of regions of interest. The fine-grained adaptation promotes an increase in objective and subjective video qualities inside the frame by allocating more bits to the RoI (in our case the hard-to-predict regions, see Section 2.8.1.4 and 4.4.1). The decision maker considers the previous knowledge, by implementing the Reinforcement Learning (RL) method, to increase or decrease the QP in relation to the base QP. To couple the frame- and BU-level in HRC, the RL unit feedbacks both the MPC and the MDP to keep system consistency and avoid mismatches. The HRC employs an observer unit able to read, store and manage the MVC encoder feedback (generated bitrate) and variables that define the encoder system state (target bitrate, QP, input constraints, etc) in order to support the bitrate prediction and actions/decision taking. Also, an image properties extractor is employed to build the RoI map used for BU-level RC. This integration allows HRC to properly exploit the influence of spatial, temporal, view and GOP-phase inputs to define a global optimal control action.

MPC-based frame-level Rate Control: It is responsible for predicting the bitrate allocation and defining an optimal QP value for the current frame while minimizing a performance cost function. Our MPC-based RC deals with multiple stimuli superposition building the input horizon using previously encoded frames from temporal and view neighborhood. The proposed scheme also incorporates the GOP-phase for accurate bitrate prediction.

MDP-based Basic Unit-level Rate Control: The BU-level RC receives the QP defined at frame level and adjusts the QP for each BU. The proposed Markov Decision Process-based RC takes the decisions over a map of states based on a set of possible actions (QP adaptations) and the associated rewards. The texture-based map of states is linked to the map of RoI and provides the structure to make decision.

Coupled Reinforcement Learning: It is responsible for adapting MPC and MDP models to the dynamic system behavior. After an action is taken at BU-level, the RL reads the system response and, updates the transition probabilities and the associated rewards in the MDP model. Once the frame is totally encoded, the resulting map of states is used to update the fame-level MPC. This strategy integrates frame-level and BU-level guaranteeing consistency and avoiding modeling mismatches.

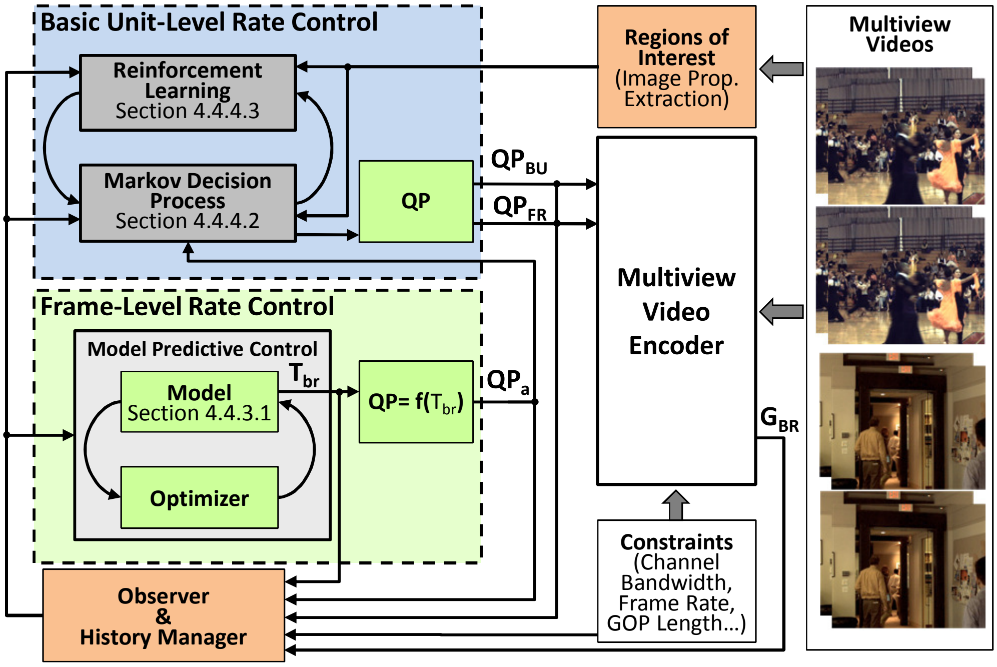

Basic Unit-Level Rate Control
Reinforcement Learning Section 4.4.4.3
Markov Decision Process Section 4.4.4.2
QP
QP$_{BU}$
QP$_{FR}$
Regions of Interest (Image Prop. Extraction)
Multiview Videos
Multiview Video Encoder
Frame-Level Rate Control
Model Predictive Control
T$_{br}$
Model Section 4.4.3.1
Optimizer
QP= f(T$_{br}$)
QP$_{a}$
G$_{BR}$
Constraints (Channel Bandwidth, Frame Rate, GOP Length...)
Observer & History Manager

Figure 4.41: Hierarchical Rate Control system diagram

On the following sub-sections the Hierarchical Rate Control will be presented in details along with the equations that describe the whole controller behavior. For simplicity we provide, in Table 4.7, the definitions of the main variables used in the HRC description.

Table 4.7: Variables Definitions

Variable	Description
	Frame-Level Rate Cotnrol
T_{BR}	Target bitrate for one frame (bits per frame)
BW	Channel bandwidth (bits per second)
FR	Frame rate (frames per second)
BA	Bit allocation (absolute)
w_I, w_P, w_B	I, P and B weight respectively (absolute)
$\overline{w}_{GOP}$	Average w for the current GOP (absolute)
L_{GOP}	GOP Length (# of frames)

ω	Frame weight (absolute)
N_A	Number of anchor frames (# of frames)
BR	Bitrate (#bits)
H_{QP}	QP History (absolute)
QP_{FL}	Quantization Parameter at Frame-level RC (discrete)
QP_{CLP}	Quantization Parameter in last process (discrete)
QP_{st}	Initial Quantization Parameter (discrete)
Q	Quantization Parameter in the optimization loop (discrete)
N_{FR}	Number of frames
Basic Unit-Level Rate Cotnrol	
M_S	RoI- Normalized Variance Matrix (absolute 0 – 1)
$M(\delta)$	MDP Reward Matrix (matrix of absolute RD)
BU	BU variance
μ	Avarage of $\mathrm{BU_i}$
N_{BU}	Number of BUs
QP_{BU}	Quantization Parameter at Frame-level RC (discrete)
T_{BR}	Target bitrate for one frame (bits per frame)
R_S	BU Reward “Shared” (absolute)
R_L	Reinforcement learning Value (vector of H_R)
$f(s,\delta)$	Probability of state transition
P_R	Probability results from $\mathrm{R_L}$ vector of “phase” actions. Actions of $\mathrm{R_L}$ in a range of at least 2 horizons.
$\Delta\ \delta$	Variation between actual BU δ and the δ of anchor frame
M_f	Variation of variance matrix values
H_R	History of R_L
G_{BR}	Generated bitrate (bits per frame)
$U(s,s')$	Function to update the matrix from *s* to *s'*

4.4.3 Frame-Level Rate Control

The frame-level MVC Rate Control problem matches the control-theory superposition principle (TATJEWSKI, 2010) defined as the response at a given place and time of the linear system caused by multiple stimuli. Model Predictive Control (MPC) techniques (GARCÍA, PRETT e MORARI, 1989)(MORARI e LEE, 1999) have demonstrated to accurately predict the response of multiple stimuli dynamic systems such as MVC encoder while incorporating the phase concept (periodic behavior) present in GGOP-level RC (see Section 4.4.1). MPC outperforms traditional feedback controllers by efficiently integrating input stimuli to state space constrains while providing flexibility by employing rolling input and output horizons (see Section 2.8.1.1).

As discussed in Section 2.8.1.1, the main goal of a Model Predictive Controller is to predict the future behavior of a system state and/or output over a finite time horizon as well as compute the future input signals at each step. These actions occur by minimizing a cost function under inequality constraints on the manipulated control or the controlled variables. In this work the MPC operates at frame-level predicting the bitrate and providing the QP for each frame to be encoded. The rate controller tries to define a sequence of actions and then induce the system to a desired state while the negative effects of this action are reduced respecting restrictions and taking constraints into account. In other words, the RC defines a QP that optimizes the bandwidth or bit allocation while maximizing the visual quality and reducing bitrate/quality sudden variations.

The bitrate prediction is performed considering the neighborhood correlation at temporal, view and inter-GOP domains. As discussed in Section 4.4.1, there is a high correlation in the temporal and view neighboring frames inside the same GOP. Moreover, there is also a periodic pattern that repeats at GOP level, the GOP-Phase. Our MPC-based RC is able to exploit this correlation in order to accurately predict the future bitrate. Figure 4.42 represents the previously encoded frames used for prediction (control horizon) and the current frame to be predicted (prediction horizon) for a given MVC prediction structure. Our method employs a variable weighting factor for frames considering their positions in relation to the current frame. The variable weighting factor is calculated considering the number of references and their distance to the current frame. With this extension our fame-level RC may be directly implemented in any hierarchical bi-prediction structure (HBP) while still catching the GOP-phase correlation.

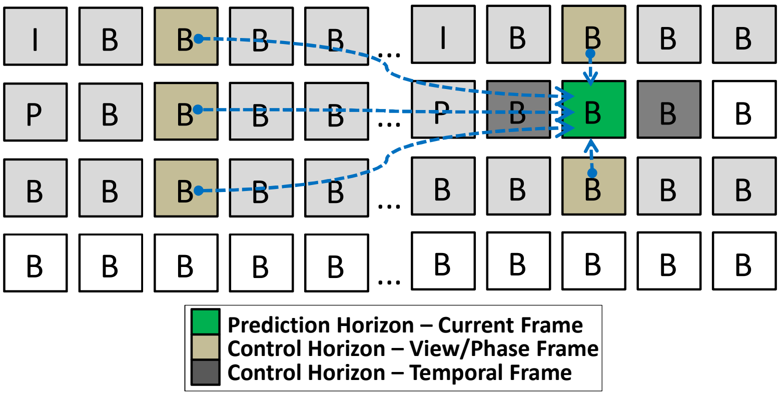

I
B
P
...
Prediction Horizon – Current Frame
Control Horizon – View/Phase Frame
Control Horizon – Temporal Frame

Figure 4.42: MPC-based RC horizons

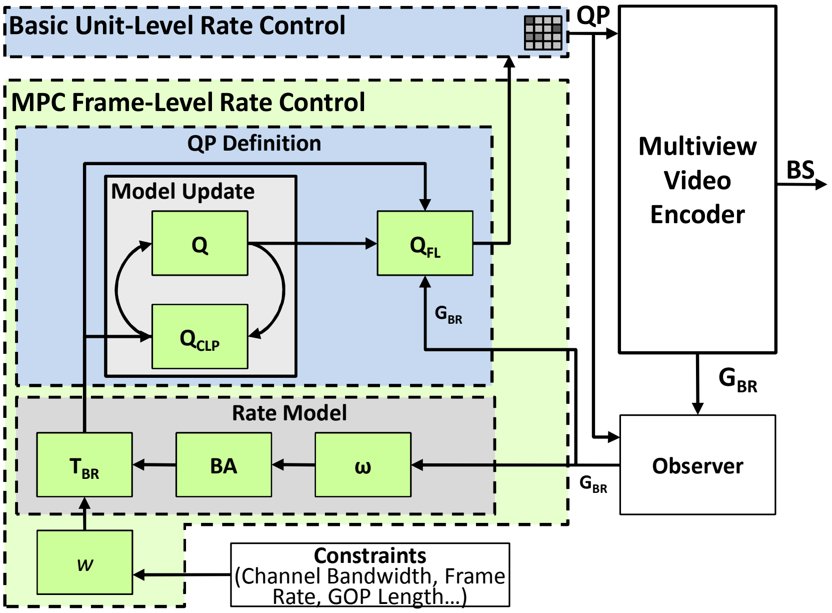


Figure 4.43: Frame-level rate control diagram

Figure 4.43 shows in details the MPC optimization process and how the component functions interact to each other. The Rate Model generates, based on the neighborhood correlation, a bitrate prediction for the current frame, the target bitrate. Based on the prediction an optimal QP is defined and the internal model is updated. The system feedback and the actually used QP defined in the BU-level RC are received through the observer.

4.4.3.1 Rate Model

The MPC-based Rate Control defines the target bitrate ($T_{BR(f)}$) considering the bandwidth (*BW*) and frame rate (*FR*) constrains along with the neighboring frames weights (*w*) and their frames bit-allocation (*BA*), as shown in Eq. (4.18).

$$\mathrm{T}_{\mathrm{BR}(f)} = \frac{BW}{FR} \pm w(\mathrm{BA}) \tag{4.18}$$

The feedback and the correlation between frames vary with the type of each frame. The bitrate range of distinct frame types (I, P and B) lie in different ranges, see Figure 4.39. Thus, the weighting factors for each frame type must be different. A static weight (w_{I}) is statically predefined for I frames (LI, PAN, *et al.*, 2003) while P and B-frame weights (w_{P} and w_{B}) are calculated dynamically considering the weights of temporal neighboring frames. Eq. (4.19) shows how the weights are calculated considering the HBP in order to respect the local linearity inside the current GGOP; where $\overline{w}_{\mathrm{GOP}}$ is the average of *w* in the current GOP, *f* represents the *f-th* frame of a given type (I, P or B) in the processing order, u=1/(L_{GOP-1}) and L_{GOP} denotes the GOP length. For a smooth weighting propagation, *w* is limited according to a statistically-defined range.

$$\begin{aligned} w_{\mathrm{I}} &= 0.75 \\ w_{\mathrm{P}} &= max\left\{w_{\mathrm{f}-1} - 2\mathrm{u}, min\left\{\overline{w}_{\mathrm{GOP}} - .25, w_{\mathrm{I}} - 2\mathrm{u}\right\}\right\} \\ w_{\mathrm{B}} &= max\left\{w_{\mathrm{f}-1} - 4\mathrm{u}, min\left\{\overline{w}_{\mathrm{GOP}} - .25, w_{\mathrm{P}} - 2\mathrm{u}\right\}\right\} \end{aligned} \tag{4.19}$$

The target bit-allocation (*BA*) is given by a history-based weighted model to optimize MPC for best target bit-allocation, as shown in Eq. (4.20). The proposed MPC-based RC was designed to differentiate the frames according to their number of references (0..2 temporal + 0..2 disparity reference frames) as it is an important data to understand how the bit allocation propagates within the 3D-Neighborhood. It allows HRC to respond to variations inside the GGOP and to become more flexible by adapting, without further extensions, to any HBP structure.

The weights $\omega_{i,j}^{m,n}$ (where *i* and *j* are the frame time instant and view; *m* and *n* denotes the number of references in the temporal and view domains, respectively) calculation is presented in Eq. (4.21).

$$\mathrm{BA}_{(\mathrm{f})} = \left(\mathrm{BA}_{(\mathrm{f}-1)} - \frac{\mathrm{BA}_{(\mathrm{f}-1)}}{\mathrm{N}_{\mathrm{A}} - 1} + \frac{\omega_{i,j}^{m,n}}{\sum_{0}^{m}\sum_{0}^{n} \omega^{m,n}} - 1 \right) \times \frac{\mathrm{BW}}{FR} \times L_{\mathrm{GOP}} \tag{4.20}$$

$$\omega_{i,j}^{m,n} = \frac{(BR_{i,j}^{m,n} \times \mathrm{QP}_{i,j}^{m,n}{}_{(\mathrm{f}-1)}) + (L_{\mathrm{GOP}} - 2)\omega_{i,j}^{m,n}{}_{(\mathrm{f}-1)}}{L_{\mathrm{GOP}} - 1} \tag{4.21}$$

4.4.3.2 Quantization Parameter Definition

Once the prediction is performed, the RC must define a proper action in terms of QP. The QP is determined by summation of all target bitrate ($T_{BR(f)}$) in the prediction horizon, the summation of all generated bitstream in the control horizon (*BR*) and, the history of QPs (H_{QP}), as shown in Eq. (4.22). Note, the QP defined in the frame-level (QP_{FL}) RC is not directly used by the MVC encoder but forwarded to the BU-level RC to refine the QP selection.

$$QP_{FL} = H_{QP} \times \frac{\sum_{i=1}^{p} T_{BR}}{\sum_{i=1}^{m} BR} \tag{4.22}$$

To maintain the performance of our MPC-based controller there is a need to update the QP model. For that, the HRC implements an optimization loop with non-discrete steps (k) where Q_{CLP} denotes the quantization parameter for the frame coded in the last process. Eq. (4.23) and Eq. (4.24) describe the update process where the QP value is constrained to a variation range of ±2 QP points for smooth update. In Eq. (4.24) M is the transposed matrix of ω multiplied by target bitrate variation ($\Delta T_{BR(f)}$) for the frames belonging to the control horizon. Q_{st} is the initial QP defined by the user.

$$\begin{aligned} Q_k &= min\left\{Q_{(k-1)} + 2, max\{Q_{(k-1)} - 2, Q\}\right\} \\ Q_{(k-1)} &= min\left\{QP_{max}, max\{QP_{min}, Q_{CLP}\}\right\} \end{aligned} \tag{4.23}$$

$$Q_{CLP} = \sum_{i=L} Q_k \times \det(M(\omega \times \Delta T_{br})^T) \times Q_{st} \times \frac{\sum \Delta \tilde{Q}_k^j}{N_{Fr}} \tag{4.24}$$

4.4.3.3 Frame-Level Rate Control Evaluation

In the following are presented the detailed results of the frame-level only HRC. Table 4.8 presents the bitrate results generated using SMRC (Single-View Mode Rate Control) extrapolated from the H.264 reference software (JM) using the quadratic MAD prediction (LI, PAN, *et al.*, 2003). To measure the target bitrate accuracy, we use the Mean Bit Estimation Error (MBEE) metric presented in Eq. (4.25). On average, the proposed frame-level RC provides 1.13% (up to 1.58%) of bitrate error while the SMRC provides 2.46% (up to 2.91%). The results show that the frame-level HRC predicts more accurately the bitrate behavior and is able to adapt the QP in order to reduce the output error.

$$MBEE = \left\{ \sum_{i=0}^{GOP_{size} \times N_v} \frac{|R_t - R_a|}{R_t} \times 100 \right\} \Big/ N_{Fr} \tag{4.25}$$

Table 4.8: Comparison of Frame-Level HRC Bitrate Accuracy

Video	Target [kbps]	Bitrate [kbps]		Error (MBEE) [%]	
		SMRC	Frame-Level HRC	SMRC	Frame-Level HRC
Ballroom	256	263	259	2.63	1.17
	392	402	396	2.61	1.07
	512	523	518	2.16	1.13
	1024	1048	1032	2.35	0.81
Exit	256	261	258	2.10	0.88
	392	402	397	2.55	1.29
	512	523	519	2.25	1.36
	1024	1048	1038	2.34	1.38
Flamenco2	256	262	258	2.30	0.81

	392	402	396	2.50	1.00
	512	525	517	2.54	1.07
	1024	1049	1035	2.46	1.10
Vassar	256	263	258	2.91	0.84
	392	402	397	2.56	1.25
	512	526	519	2.68	1.36
	1024	1049	1040	2.44	1.58
Average	256	262	258	2.49	0.93
	392	402	397	2.55	1.15
	512	524	518	2.41	1.23
	1024	1049	1036	2.40	1.22
Total Average				2.46	1.13

The proposed frame-level RC also provides rate-distortion (RD) results that outperform SMRC and the fixed-QP solution (non-RC). Table 4.9 summarizes the quality and bitrate outputs in terms of BD-PSNR (Bjøntegaard Delta PSNR) and BD-BR (Bjøntegaard Delta BR) (TAN, SULLIVAN e WEDI, 2005) in relation to the non-RC solution. Compared to SMRC the proposal provides 0.6dB BD-PSNR increase. The BD-BR reduction is 19.28% in relation to SMRC.

Table 4.9: Comparison of BD-PSNR

Video	**SMRC**		**MPRC**	
	BD-BR [%]	**BD-PSNR [dB]**	**BD-BR [%]**	**BD-PSNR [dB]**
Ballroom	10.902	-0.328	28.603	-0.939
Exit	11.542	-0.368	36.920	-1.089
Flamenco	9.630	-0.217	29.852	-0.880
Vassar	6.514	-0.183	20.333	-0.596
Average	9.647	-0.274	28.927	-0.876

4.4.4 Basic Unit-Level Rate Control

Markov Decision Process (MDP) is a mathematically-based optimization model of discrete state, sequential decision making in a stochastic environment that depends only on the current state and not in previous states. However, if a controlled MDP is considered, the transition probabilities are affected by previous actions. According to this definition, the controlled MDP perfectly fits to the BU-level rate control where a decision among a set o discrete QP values has to be made considering the neighborhood history. However, in MVC, the transition probabilities between the possible states are not known *a priori* and vary for distinct time instants and video content. Reinforcement learning can solve MDP with no explicit probabilities definition. It calculates the probabilities of transition based on the Law of Effect theory that states: in case an action is followed by satisfactory state, the probability taking the same action again is increased. It is also possible to incorporate additional variables such as image properties into the reinforcement learning definition.

As part of the HRC we propose a BU-level Rate Control employing Markov Decision Process along with RL able to consider the image properties through a texture-based RoI map, as detailed along this section.

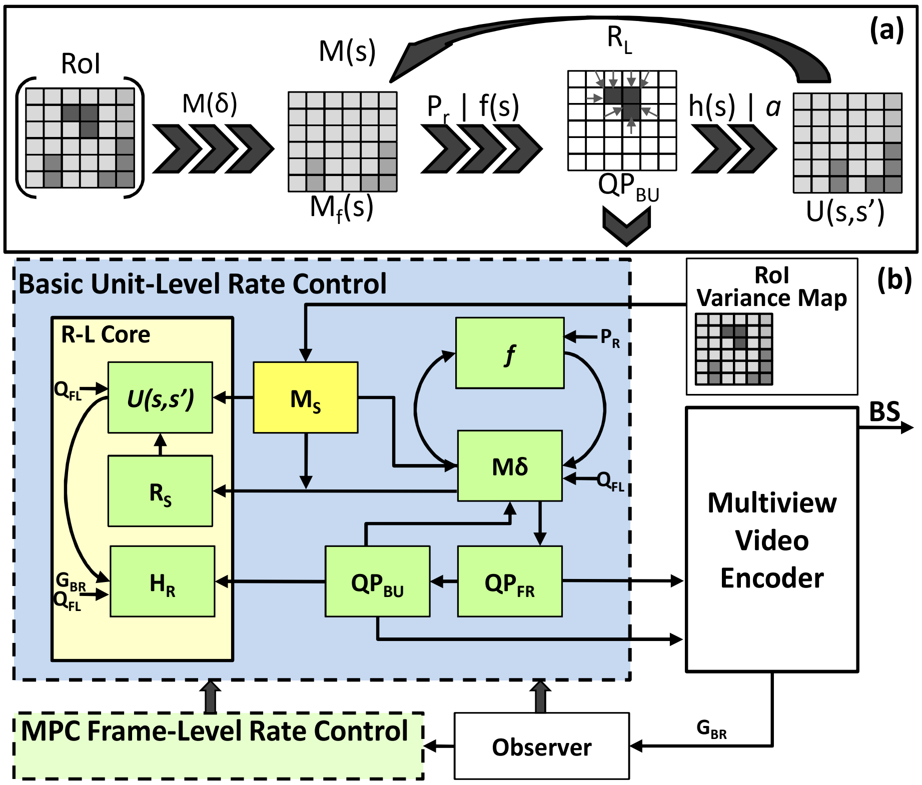


Figure 4.44: Basic unit-level rate control diagram

Figure 4.44 depicts the diagram of the proposed BU-level RC that works as a refinement of the frame-level RC. In order to refine the accuracy of bit allocation and provide smooth visual quality, our BU-level RC includes the concept of region of interest (RoI) into a Markov Decision Process that employs reinforcement learning for adapt to dynamic encoder and input variations. At each decision step, the RC monitors the state of the system and determines the next action to take based on constraints observations and the control policy. Firstly, the HRC defines the RoIs for anchor frames generating a map of weights M_S that will determine the importance of each BU inside the frame. Secondly, the weights map is linked to a map of states *M(δ)* in the MDP that corresponds to the QP for each BU. The MDP fits to the MVC encoder behavior by providing the structure to make decisions partly random and partly under a control. Finally, to dynamically adjust the matrix of states for next decision, the RL is responsible to feedback the system response to the current state for both BU-level and frame-level control.

4.4.4.1 *Regions of Interest*

As discussed in Section 2.8.1.4, frames are composed of regions with distinct image properties requiring a variable number of bits to be encoded. Regular video encoders use the same QP to encode all basic units within a frame leading to inefficient bitrate distribution and undesirable quality variations inside the frame. However, it is possible to define regions to receive special treatment, the regions of interest. The BUs belonging to RoIs may be prioritized by the rate control unit in order to protect the quality of those regions. In this work, the whole frame is considered to have the same semantic relevance (this leave space for further application specific extensions) but regions that present a hard-to-predict content must be allowed to use more bits through QP reduction. According to our analysis (see Section 4.4.1), textured regions tend to generate more residue and, consequently, require higher bitrate.

In our solution, the RoI is determined by a normalized variance map – given by M_S in Eq. (4.26) – for all anchor frames. Additionally, HRC also keeps a second matrix of states where each value represents a bitrate of a frame inside a GGOP encoding history to incorporate temporal and view neighborhood information to the MDP process. The matrixes data are used by the MDP and RL to define the rewards associated to each state and the actions taken by the control. For non-anchor frames are used the statistics given by anchor frames considering the reinforcement learning R_L.

$$M_{S(i,j)} = \frac{(BU_i - \mu)^2}{N - 1} \qquad (4.26)$$

4.4.4.2 Markov Decision Process

The HRC implements the BU-level RC by employing the Markov Decision Process. The MDP works over a matrix of independent states $M_f(s)$ representing the QPs of each BU within a frame. Each BU has a set of possible actions A with associated rewards R_S and transition probabilities $f(s,\delta)$. In our model the possible actions are increase, decrease or maintain the QP value defined at frame-level, as shown in Eq. (4.30) and Eq. (4.31). A matrix of coefficients $M(\delta)$ is used to define the reward for each action according to Eq. (4.27). The rewards R_S are calculated based on the RoI map M_S, matrix of coefficients $M(\delta)$ and the reinforcement learning R_L (see Section 2.8.1.3), as shown in Eq. (4.28). For each action there is a probability of transition $f(s,\delta)$ defined by Eq. (4.29).

$$M(\delta) = \sum \frac{QP_{BU} \times BS}{Max_{QP} \times (T_{BR} \,/\, N_{BU})} \qquad (4.27)$$

$$R_S = R_L \times | M(\delta) - M_S | \qquad (4.28)$$

$$f(s,\delta) = P_R \mp \Delta\delta \qquad (4.29)$$

$$QP_{BU} = \begin{cases} QP_{FR} + 1 \,\forall\, f(s,\delta) > +1 \\ QP_{FR} - 1 \,\forall\, f(s,\delta) < -1 \\ QP_{FR} \,\forall\, -1 < f(s,\delta) < +1 \end{cases} \qquad (4.30)$$

$$QP_{FR} = \mathrm{trunc}\left(\sum M_f(s) \middle/ N_{BU} \right) \qquad (4.31)$$

4.4.4.3 Coupled Reinforcement Learning

The RL agent incorporates the knowledge of previous events in the decision making process through monitoring the MVC system response and updating state transitions probabilities and rewards at both frame- and BU-level. The BU-level feedback happens by updating the history of reinforcement learning h_R, see Eq. (4.32). Eq. (4.33) gives the final MDP state matrix that is used as obtained knowledge for the upcoming frames. The QP of the frame updated using Eq. (4.33) and calculated according to Eq. (4.29) QP_{FR} provides feedback to the MPC at frame-level.

$$H_R = \frac{\Delta T_{BR} \times \sum QP_{BU\,L}^{\;k}}{\sum G_{BR\,L}^{\;k} \times \Delta QP_{FL}} \qquad (4.32)$$

$$U(s,s') = QP_{FL}\begin{cases} M_f\left(s,s'\right)\forall \; -1 > f\left(s,\delta\right) > +1 \\ M_f\left(s,s\right)\forall \; -1 < f\left(s,\delta\right) < +1 \end{cases} \quad (4.33)$$

4.4.5 Hierarchical Rate Control Evaluation

In this section are presented the detailed results of the proposed HRC compared to baseline solution, the JMVC without RC and the SMRC (Single-View Mode Rate Control). The comparison with the state-of-the-art is presented in Chapter 6. Table 4.10 presents the accuracy in terms of MBEE (less is better) for our HRC compared to baseline RC solutions. On average, our Hierarchical Rate Control provides 1.6% MBEE decrease while raging from 0.7%-1.37%. The superior accuracy is a result of the ability to adapt the QP jointly at frame and BU-levels considering the neighborhood correlation and the video content properties.

Table 4.10: Comparison of Proposed HRC Bitrate Accuracy

Sequence		Bit-Rate [kbps]				MBEE [%]		
		Target	JMVC	SMRC	HRC	JMVC	SMRC	HRC
VGA	*Ballroom*	256	268	263	258	4.64	2.63	0.75
		392	408	402	395	4.06	2.61	0.78
		512	529	523	516	3.33	2.16	0.78
		1024	1058	1048	1032	3.30	2.35	0.78
	Exit	256	267	261	258	4.29	2.10	0.94
		392	408	402	396	3.99	2.55	0.92
		512	528	523	516	3.21	2.25	0.83
		1024	1056	1048	1031	3.14	2.34	0.72
	Flamenco2	256	268	263	258	4.79	2.91	0.71
		392	409	402	395	4.34	2.56	0.71
		512	530	526	516	3.56	2.68	0.84
		1024	1059	1049	1031	3.41	2.44	0.70
	Vassar	256	267	262	258	4.27	2.30	0.75
		392	407	402	395	3.73	2.50	0.72
		512	528	525	516	3.13	2.54	0.86
		1024	1056	1049	1033	3.15	2.46	0.86
Average		256	268	262	258	4.50	2.49	0.79
		392	408	402	395	4.03	2.55	0.78
		512	529	524	516	3.31	2.41	0.83
		1024	1057	1049	1032	3.25	2.40	0.76
XGA	*Break dancers*	512	525	524	518	2.47	2.41	1.23
		768	801	788	776	4.33	2.54	1.08
		1024	1052	1050	1034	2.72	2.56	1.00
		2048	2101	2109	2070	2.58	2.99	1.06
	Uli	512	525	525	519	2.46	2.54	1.37
		768	801	789	776	4.28	2.72	1.08
		1024	1052	1052	1034	2.74	2.72	0.95
		2048	2101	2101	2069	2.59	2.60	1.05
Average		512	525	525	519	2.46	2.48	1.30
		768	801	788	776	4.30	2.63	1.08
		1024	1052	1051	1034	2.73	2.64	0.97
		2048	2101	2105	2070	2.58	2.80	1.05
HD	*GT Fly*	1024	1050	1049	1037	2.54	2.44	1.27
		1536	1581	1575	1553	2.93	2.54	1.11
		2048	2104	2101	2069	2.73	2.59	1.03
		4096	4202	4219	4140	2.59	3.00	1.07
	Poznan Hall2	1024	1049	1050	1038	2.44	2.54	1.37
		1536	1582	1578	1553	2.99	2.73	1.11
		2048	2104	2104	2068	2.73	2.73	0.98
		4096	4202	4203	4139	2.59	2.61	1.05
Total Average						3.40	2.55	0.95

Table 4.11 presents the objective rate-distortion in BD-PSNR (Bjøntegaard Delta PSNR) and BD-BR (Bjøntegaard Delta Bitrate) (TAN, SULLIVAN e WEDI, 2005) in relation to JMVC without RC. The HRC provides 1.86dB BD-PSNR increase along with BD-BR reduction of 40.05%, on average. If compared to SMRC, the HRC delivers 1.6dB increased BD-PSNR and 31.08% reduced BD-BR. Remember, besides superior RD performance the HRC also outperforms SMRC in terms of accuracy.

Table 4.11: BD-PSNR and BD-BR Comparison

JMVC 8.5 vs.		VGA				XGA		HD1080p		AVG
		Ballroom	Exit	Flamenco2	Vassar	Bdancer	Uli	Poznan	GTGly	
SMRC	BD-PSNR	0.328	0.368	0.217	0.183	0.215	0.208	0.253	0.012	0.223
	BD-BR	-9.831	-10.348	-8.784	-6.116	-8.963	-9.805	-12.180	-6.711	-9.092
HRC	BD-PSNR	1.585	2.375	2.103	1.176	2.060	1.870	2.085	2.055	1.914
	BD-BR	-31.588	-47.458	-38.199	-27.335	-46.112	-49.660	-48.760	-47.250	-42.045

In the following we present HRC detailed results for *Flamenco2* sequence encoded at 1024kbps. For simplicity, we analyze only the first 4 views. Figure 4.45 shows the target bitrate, the total accumulated bitrate and the accumulated bitrate for each view. The presented bitrate distribution is smooth also at view level without abrupt oscillations. As expected from the discussion in Section 4.4.1, the base view (View 0, I-view) is more bitrate hungry followed by P-views (View 2) and B-views (View 1, 3).

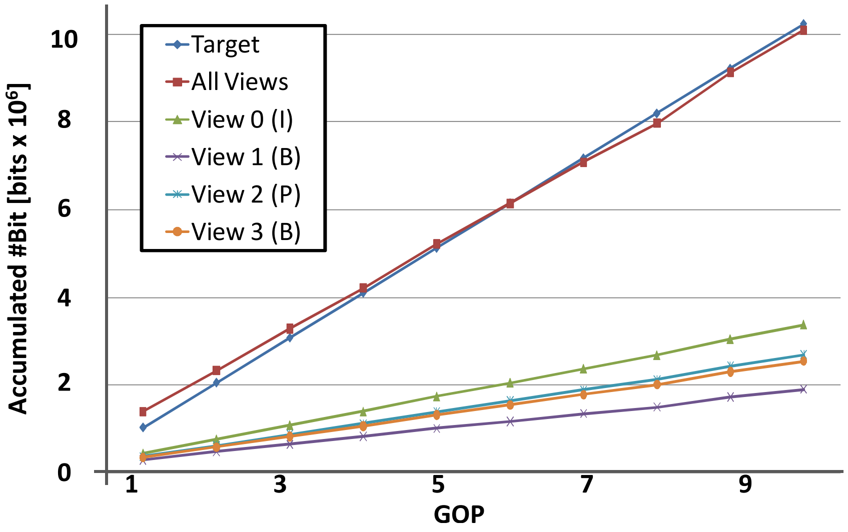


Figure 4.45: View-level bitrate distribution (Flamenco2)

The frame-level bitrate distribution is further detailed for the GOP #8 in Figure 4.46. It shows, graphically, the smooth bitrate and PSNR variations delivered by our solution considering frame-level. Note, the HRC surface presents no sudden variations for both bitrate and PSNR. Compared to the other solutions, it is clear that the bitrate and quality provided by our HRC are significantly smoother even when compared to SMRC.

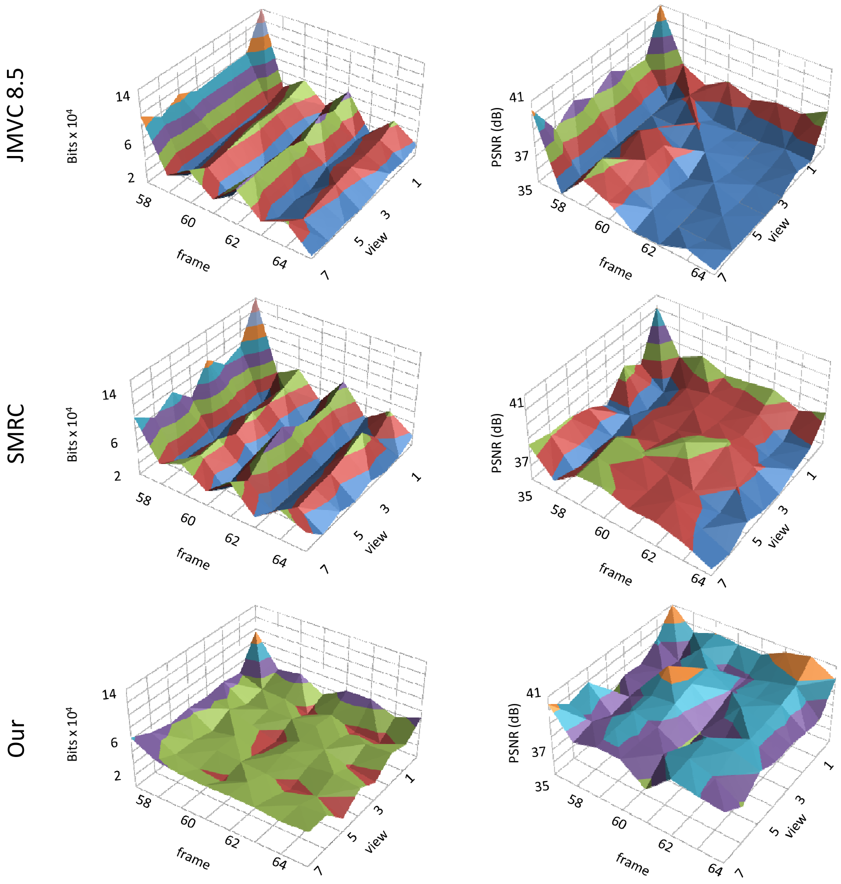


Figure 4.46: Bitrate and PSNR distribution at frame level (GOP #8)

Analogous analysis was performed to demonstrate the behavior of our RC at BU level. Figure 4.47 shows the bitrate distribution for a frame region (zoomed image) in sequence *Flamenco2*. Observe that for HRC the bitrate varies with the texture complexity due to our RoI-aware MDP implementation. For the homogeneous background, reduced number of bits is spent while for textured objects and borders (dancer) more bits are allocated. Note that, in Figure 4.47, the HRC bitrate distribution surface plot accurately fits the objet shapes. This behavior prioritizes the regions where the HVS requires a higher level of details tending to lead to a superior overall perceived quality. SMRC is unable to accurately react to the image content. In addition, the HRC also results in smoother variations within the same region (dancer's body or background), as shown in Figure 4.47. It avoids sudden quality variations and the resulting coding artifacts inside those regions.

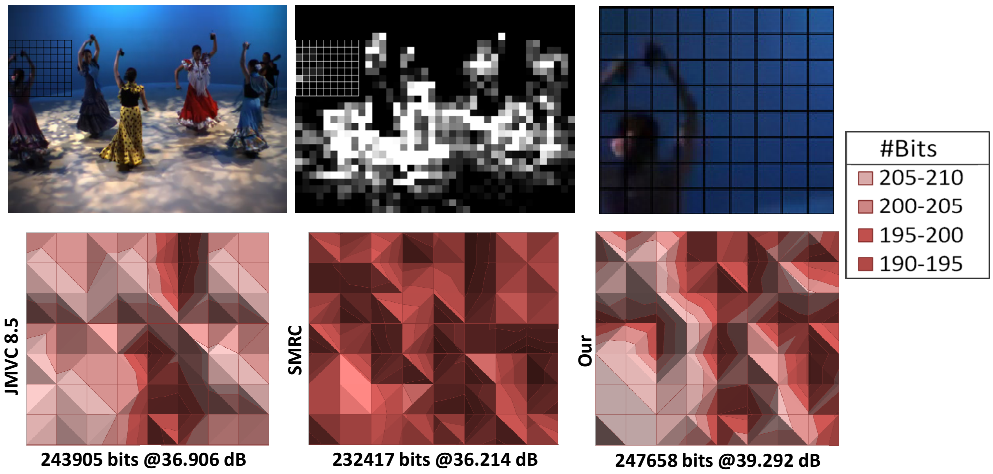


Figure 4.47: Bitrate distribution at BU level (GOP #8)

The evaluation presented demonstrates that is possible to maximize the video quality while obeying to bandwidth constrains by implementing an efficient RC algorithm. The proposed HRC is a powerful tool in order to protect the MVC encoder from quality losses typically posed by fast MD and fast ME/DE heuristics such those shown in Section 4.1, 4.2 and 4.3.

4.5 Summary of Energy-Efficient Algorithms for Multiview Video Coding

To provide basis to the mode decision algorithms presented in Section 4.1, a complete coding mode correlation analysis was presented. After that, the Early SKIP algorithms was presented in details along with the QP-based threshold derivation technique. The Multi-Level Fast Mode Decision that incorporates the Early SKIP concept describes a 6-step sophisticated algorithm for complexity reduction. It employs two complexity reduction operation modes while exploiting the 3D-neighborhood correlation along with video properties. To handle the energy versus quality tradeoff an Energy-Aware Complexity Adaptation algorithm is presented.

Targeting the ME/DE complexity reduction the Fast ME/DE is presented in Section 4.2. This algorithms defines two classes of frames, the key and non-key frames. Depending on the prediction mode inferred from 3D-Neighborhood information, the MBs belonging to non-key frames are submitted to the evaluation of 3 or 13 candidate blocks. It represents a meaningful overall complexity reduction.

Aware of the video quality drawback posed by the energy-efficient MD and ME/DE algorithms, a Hierarchical Rate Control was developed in order to manage and compensate eventual quality drawbacks. The goal is to improve the video quality by optimizing the bit allocation. For that, the HRC operates in frame-level and BU-level rate control actuation levels. At frame level a Model Predictive Controller is used while the BU-level RC exploits a Markov Decision Process along with Reinforcement Learning.

5 ENERGY-EFFICIENT ARCHITECTURES FOR MULTIVIEW VIDEO CODING

Although the fast ME/DE provides significant complexity reduction, a high throughput hardware architecture is required for real-time ME/DE in MVC. Without a dedicated hardware, ME/DE for MVC in real-time mobile application is unfeasible. Therefore, in addition to our fast ME/DE algorithm we propose novel motion and disparity estimation hardware architectures designed to provide real-time MVC encoding for up to four views HD1080p (1920x1080) based on the proposed fast ME/DE algorithm. As the architectural solutions share some similar architectural blocks we are going to start presenting a high-level architectural template description in order to avoid redundancies along the architectures description. The architectural template, presented in Section 5.1, will give the required basis for a better understanding of the three proposed architectures. Note that the main architectural contributions along this thesis do not lie in the design of the processing units but in the MVC parallelism exploitation, energy management, on-chip memory design, etc.

In Section 5.2 a ME/DE architecture implementing in hardware the fast ME/DE algorithm proposed in Section 4.3. This first solution exploits the multiple levels of parallelism available in the MVC encoding structure. To reduce the on-chip video memory size and reduce external memory communication is presented in Section 5.3 an ME/DE hardware architecture with dynamic search window formation. The dynamic search window formation accurately predicts the memory access pattern from the 3D-Neighborhood to manage the on-chip video memory. Finally, in Section 5.4 a complete application-aware dynamic power management algorithm and its architectural hardware implementation are presented. Also, the memory sizing, partitioning and, management techniques are detailed along this chapter.

5.1 Motion and Disparity Estimation Hardware Architectural Template

Our custom motion and disparity estimation hardware architectures typically employ a similar structure and processing units. For this reason, before moving to the each architecture and the energy-efficient techniques employed, we summarize in this section the architectural template and the design of the main ME/DE hardware building blocks.

The architectural template overview is presented in Figure 5.1. It is composed of 5 main blocks named: (i) Energy/Complexity-aware Control Units; (ii) Programmable Search Pattern Control Unit; (iii) Address Generation Unit (AGU); (iv) On-chip Video Memory and; SAD Calculator. Energy/Complexity-aware Control Units box is not detailed in this section since this block represents the implementation of all energy/complexity-aware control techniques presented in Sections 5.2, 5.3 and 5.4. Our

ME/DE architectures communicate with the remaining MVC encoder blocks by providing SAD values and motion/disparity vectors for the mode decision unit. The reference frames data is read from the external memory that stores the Decoded Picture Buffer (DPB). The MVC encoder writes the DPB after the encoded frames are reconstructed and filtered by the deblocking filter.

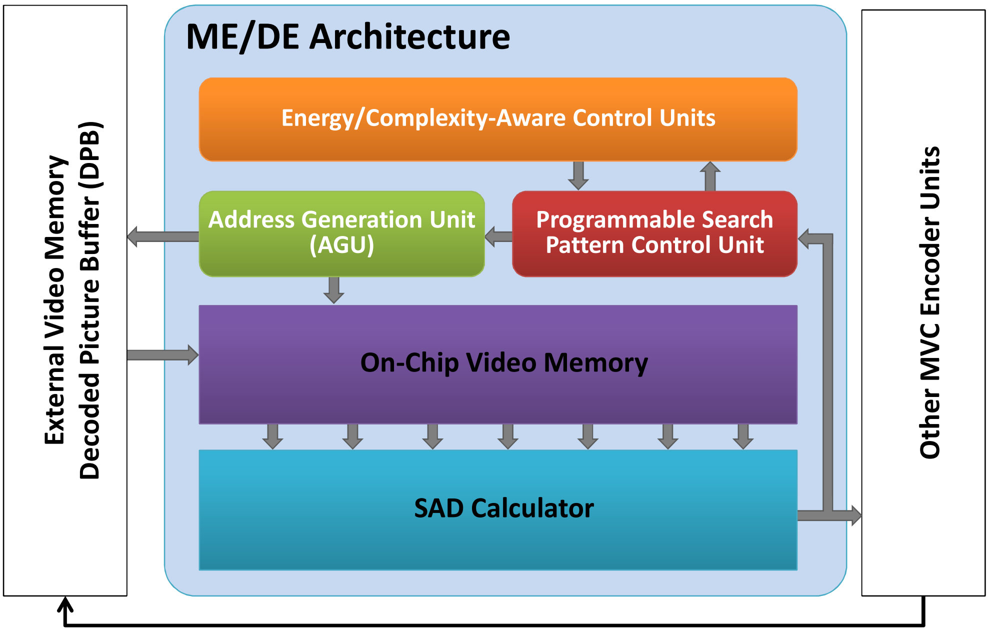

External Video Memory
Decoded Picture Buffer (DPB)
ME/DE Architecture
Energy/Complexity-Aware Control Units
Address Generation Unit (AGU)
Programmable Search Pattern Control Unit
On-Chip Video Memory
SAD Calculator
Other MVC Encoder Units

Figure 5.1: ME/DE hardware architecture template

The proposed Programmable Search Pattern Control Unit was designed employing a microprogrammed style in order to facilitate the implementation and experimentation of multiple search patterns. It communicates with the Energy/Complexity-aware Control Units in order to provide search pattern information such as search pattern used, memory regions accessed, number of candidate blocks tested. Energy/Complexity-aware Control Units feedbacks the Programmable search pattern control unit with energy/complexity budget, search pattern to be employed for future MBs, vector predictors, search directions to be exploited, active on-chip memory sectors, etc. This communication and the hardware actually implemented inside the Energy/Complexity-aware Control Units depend on which energy-efficient techniques are designed for the specific architectural solution.

Once the search pattern is defined, the candidate blocks are forwarded to the AGU as a set of points inside the search window. The AGU is responsible for translating these points into a sequence of actual memory addresses. As the On-chip Video Memory is implemented in a cache fashion, the cache tags are generated using the address provided by the AGU according to a predefined tag format defined in Section 5.1.3. The On-Chip Video Memory is implemented using SRAM memory to locally store samples belonging to the search window. The samples are brought from the external memory in block-based read operations. To check if the samples required by the search control are available on-chip, the above mentioned cache tags are tested employing a fully associative approach.

The SAD (Sum of Absolute Differences) Calculator is composed of an array of 4-sample SAD Processing Elements (PEs), an array of adder trees, and comparator trees.

The number of PEs depends on the throughput require. The PEs connectivity depends on the block sizes supported by the architecture and the number of candidate blocks processed in parallel. The SAD Calculator is fed in parallel by the On-chip video memory. The number of PEs and the memory width must be jointly defined in order to maximize the hardware usage and processing throughput.

In the following sections the ME/DE hardware modules are presented in details.

5.1.1 SAD Calculator

All the data processing itself is performed in the SAD Calculator unit. It receives the current MB samples, that are stored in a small local buffer (omitted in Figure 5.1), and the reference samples to determine the SAD between the original block and the reference block according to Eq. (5.1).

$$SAD=\sum_{i=1}^{n}\left|Orig(i)-Ref(i)\right| \tag{5.1}$$

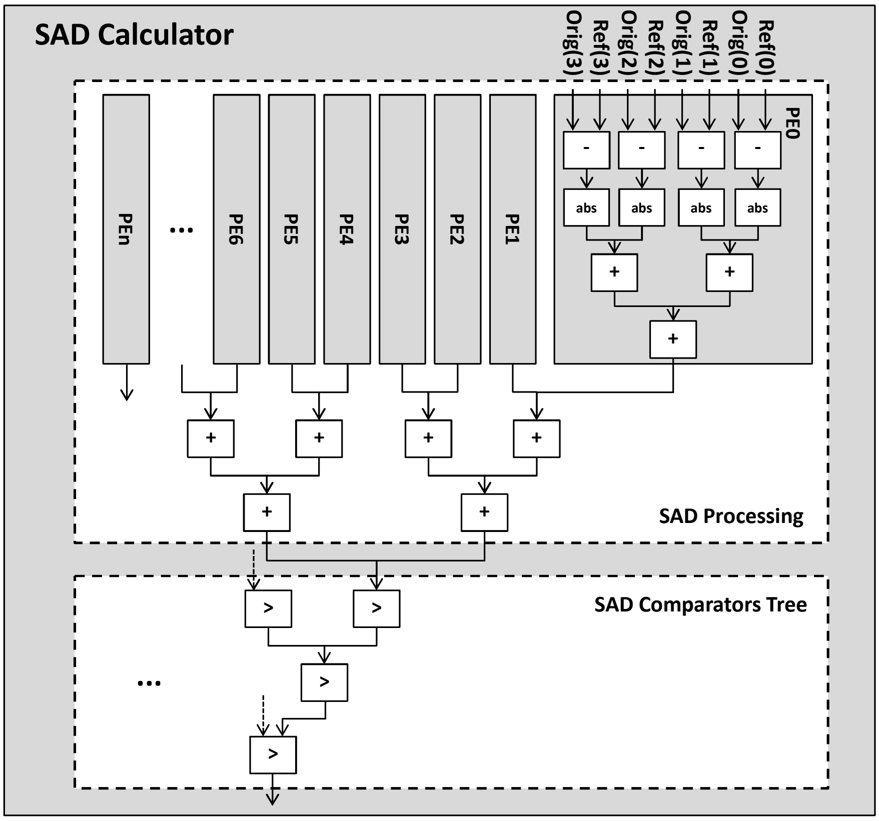


Figure 5.2: SAD Calculator architecture

Each Processing Element, as depicted in Figure 5.2, calculates the SAD for four samples in parallel. PEs are composed of four subtractors, one absolute operator and three adders. Although the hardware description supports multiple sample bitdepth the implementation was limited to 8-bit sample inputs. The PEs are associated using adder trees to generate the SAD for a whole block of NxN. In the example presented in Figure 5.2, the SAD Calculator is designed to process a 4x4 block in parallel by associating four PEs (PE0..PE3 process one 4x4 block). In this scenario, each adder three requires further three adders in two logic levels. The larger the block to be processed, the bigger the adder tree. For 16x16 blocks, 63 adders are required in six logic levels. Therefore, pipelining is required for bigger block sizes in order to avoid operation frequency reduction. For simplicity, Figure 5.2 omits pipeline barriers.

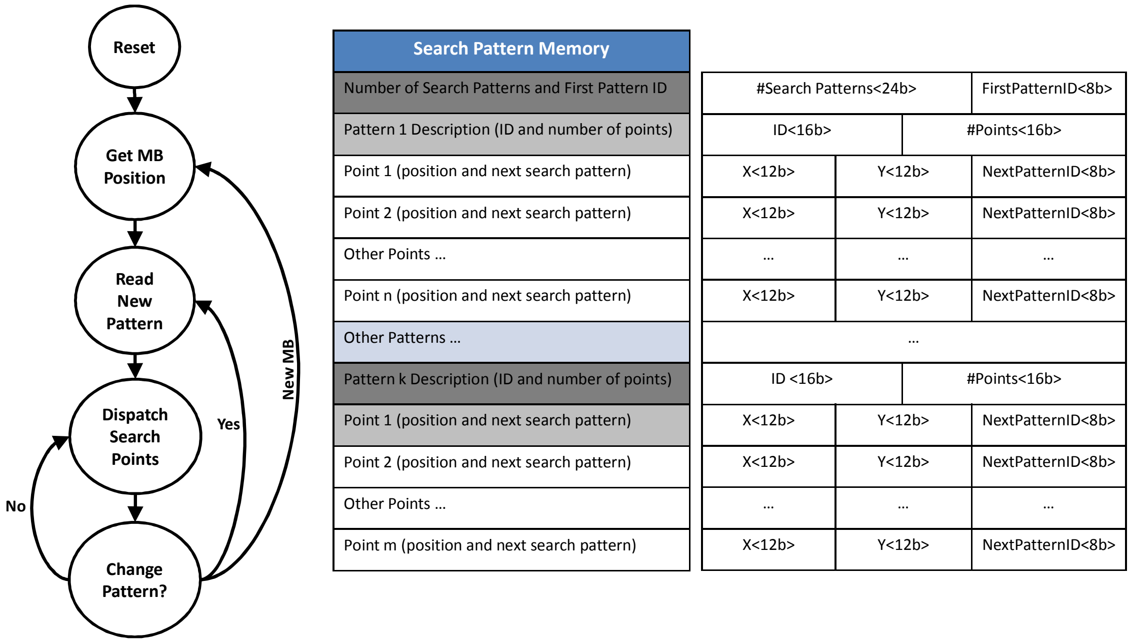
After the SADs are calculated for the multiple block processed in parallel, the SAD Comparators Tree is used to select the smallest SAD values. Along with the SAD value the SAD Calculator feedbacks the Programmable Search Pattern Control with the position where the smallest SAD was found. This information is used do decide the following steps of the search process. The SAD Comparators Tree size and logic depth depends on the number of blocks processed in parallel. Thus, pipelining might be necessary in case of a large number of blocks.

5.1.2 Programmable Search Control Unit

Hardware implementations for the search control unit are typically limited to one single search pattern. In the architectures proposed in this thesis we implement a Programmable Search Control Unit able to support multiple search patterns without hardware redesign by employing the microprogramming concept. By simply reprogramming the Search Pattern Memory it is possible to change the search pattern (or shape). It allows fast hardware ME/DE algorithms design and verification.

The Programmable Search Control Unit is composed of a Finite State Machine (FSM) and a Search Pattern Memory (SPM), both presented in Figure 5.3. Firstly, the FSM identifies the current MB position and reads, form the SPM, the first search pattern. By adding the current MB position and the coordinates of each search point defined in the SPM, the Programmable Search Control Unit determines and dispatch, in parallel, all the search points for that specific search pattern step. Depending on the feedback from the SAD Calculator, the next search step is selected among three options: repeat the same pattern, use another pattern described in the memory or process next MB.

The SPM program memory organization is presented in Figure 5.3. The left table shows the description of each line while the right table specifies the fields and bitdepth of each field (a number o *x*-bits is represented as <*x*b>). A 32-bits program memory is used. The first SPM line brings the total number of patterns programmed and the ID of the first pattern (*FirstPatternID*) to be used, where each search pattern has a unique ID sequentially defined. The search pattern is described starting by a line containing a 16-bit ID (actually only the 8 LSB are considered) and the number of points belonging to that specific search pattern. In the following, each point is described using a (*X*, *Y*) coordinates pair and the *NextPatternID*. *X* and *Y* are 12-bits integer numbers representing the displacement between the search point and the search pattern central reference point. The search pattern central reference is initially defined as the MB position and evolves according to the algorithm interaction assuming the best SAD point as center. The 12-bit coordinates enables a search range of up to [±2048, ±2048] in relation to the search pattern central reference. The *NextPatternID* specifies the next pattern to be used in case this point presents the lowest SAD among all points of the current pattern. In case the search point represents a terminating point (in case it is the lowest SAD the search ends) the *NextPatternID* is defined as the reserved value 0xFF.



Search Pattern Memory			
Number of Search Patterns and First Pattern ID	#Search Patterns<24b>		FirstPatternID<8b>
Pattern 1 Description (ID and number of points)	ID<16b>	#Points<16b>	
Point 1 (position and next search pattern)	X<12b>	Y<12b>	NextPatternID<8b>
Point 2 (position and next search pattern)	X<12b>	Y<12b>	NextPatternID<8b>
Other Points …	…	…	…
Point n (position and next search pattern)	X<12b>	Y<12b>	NextPatternID<8b>
Other Patterns …	…		
Pattern k Description (ID and number of points)	ID <16b>	#Points<16b>	
Point 1 (position and next search pattern)	X<12b>	Y<12b>	NextPatternID<8b>
Point 2 (position and next search pattern)	X<12b>	Y<12b>	NextPatternID<8b>
Other Points …	…	…	…
Point m (position and next search pattern)	X<12b>	Y<12b>	NextPatternID<8b>

Figure 5.3: Programmable Search Control Unit (a) FSM and (b) program memory

Table 5.1 provides a simple example using a two-step Log Search (MARLOW, NG e MCARDLE, 1997) with window size W=16 (search range [±8, ±8]) and finishes with a local Cross-Search (GHANBARI, 1990) refinement. The first Log Search step (ID 0x0000) leads to the second Log Search step (ID 0x0001) except for the central position (line 2) that leads to the Cross-Search refinement (ID 0x0002). After the second Log Search step all points lead to the Cross-Search refinement (ID 0x0002). The terminating step Cross-Search points to the reserved terminating pattern ID 0xFF.

Although it is possible to extend the Programmable Search Control Unit in order to program new test conditions or thresholds, the current implementation requires additional modifications in the FSM to support features such as early termination and thresholds adaptation.

Table 5.1: Search Pattern Memory example

Addr	Instruction		
0	0x000003	0x000001	
1	0x0000	0x0009	
2	0	0	0x02
3	-8	0	0x01
4	-8	-8	0x01
5	0	-8	0x01
6	8	-8	0x01
7	8	0	0x01
8	8	8	0x01
9	0	8	0x01
10	-8	8	0x01
11	0x0001	0x0009	
12	0	0	0x02
13	-4	0	0x02
14	-4	-4	0x02
15	0	-4	0x02
16	4	-4	0x02
17	4	0	0x02
19	4	4	0x02
20	0	4	0x02
21	-4	4	0x02
22	0x0003	0x0004	
23	-1	1	0xFF
24	1	1	0xFF
25	1	-1	0xFF
26	-1	1	0xFF

5.1.3 On-Chip Video Memory

The On-Chip Video Memory used in this thesis works as a cache memory composed of an Address Comparator block and the On-Chip SRAM memory itself, as represented in Figure 5.4. The address requested by the Search Control and forwarded by the AGU (still using video representation) is compared to the Tags of each memory entry. Each entry represents a full MB of the reference frame and the Tags comparison is performed in parallel. Case the reference is available on-chip the requested data is transmitted to the SAD Calculator unit. Otherwise, the read request is sent to external memory. The addresses are provided by the AGU that translates the address from video representation to a burst of addresses mapped to the actual memory address space (see Section 5.1.4). After updating the data, the samples are sent to the SAD Calculator.

The Tag in the On-Chip Video Memory is defined in Figure 5.5 where the <*n*b> represents a value with *n* bits wide. The Tag is composed by a unique view identifier, six LSBs of the frame Picture Order Counter (POC) within that specific view and the X and Y coordinates of the reference MB. By using this Tag is possible to support up to 16 views, access reference frames within a 64-frames temporal window and support up to 2kx4k (QDH) video resolutions. This definition, however, can be easily extended by increasing the bitdepth of each field in order to handle increased demands.

Remember this is just a template description, in the following sections we are going to present, in details, the SRAM organization, sizing and energy management of the On-Chip Video Memory for different scenarios.

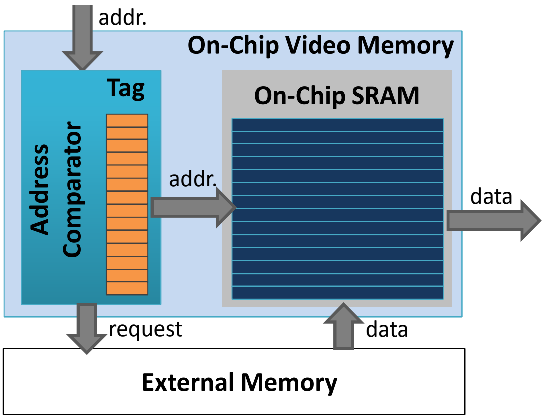


Figure 5.4: On-Chip Video Memory organization

Cache Tag:

ViewID<4b>	FramePOC<6b>	MBPosX<8b>	MBPosY<8b>

Figure 5.5: On-Chip Video Memory cache tag

5.1.4 Address Generation Unit (AGU)

The AGU is used to convert the addresses defined in video representation provided by the Search Control to a linear memory representation. Video content is represented using 2D arrays, however, when mapped to the external memory these 2D arrays must be translated to a sequence of 1D addresses. This process is depicted in Figure 5.6, where the circles represent video samples. The math for generating the memory mapped addresses is detailed in the following paragraph. Note, this AGU is used for reading-only purpose since other blocks of the MVC encoder are responsible for writing the external video memory.

The view (v) and frame (f) identifiers associated to the frame resolution (*FrameWidth* and *FrameHeight*) are used to define the frame base address (*FrameBaseAdd*), as presented in Eq. (5.2). Also, the line Line*Stride* is defined by the frame width. To read a block pointed by positions (*PosX*, *PosY*) and size (*SizeX*, *SizeY*) a sequence of linear memory addresses (Add_0, Add_1, etc) are generated by the AGU. If the block requires more than one access per sample line (as the example Figure 5.6) two sequential addresses are generates by the AGU. This process is repeated *SizeY* times, always considering the *LineStride* displacement, to complete the block reading. The general linearization definition is provided in Eq. (5.2). Note that the video representation addresses refer to sample positions, to map it to memory positions the memory word size (*MemWordSize*) in number of samples is taken into consideration.

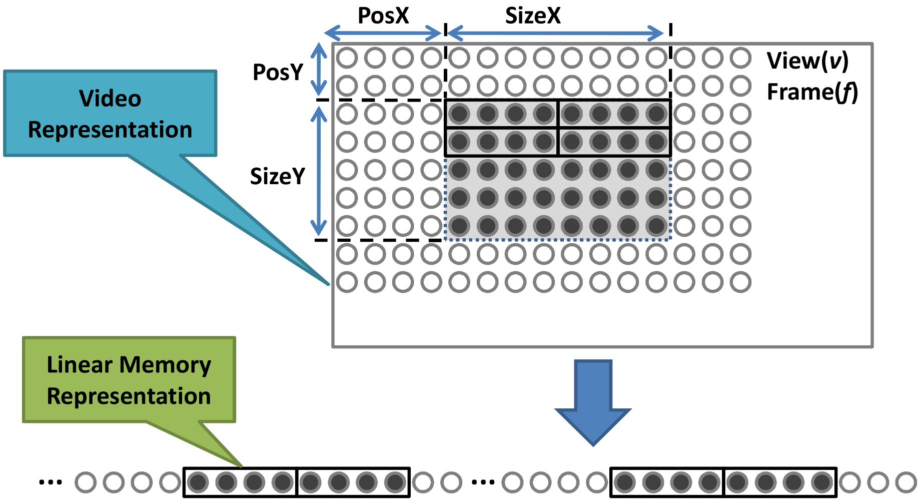


Figure 5.6: Address Generation Unit (AGU) for dedicated video memory

$$\begin{aligned}
&LineStride = \lceil FrameWidth / MemWordSize \rceil\\
&FrameStride = LineStride * FrameHeight\\
&FrameBaseAdd = (v * FramesperView + f) * FrameStride\\
&Add_0 = FrameBaseAdd + (PosY * LineStride) + \lfloor PosX / MemWordSize \rfloor\\
&Add_1 = Add_0 + 1\\
&Add_2 = Add_0 + LineStride\\
&Add_3 = Add_2 + 1\\
&\ldots
\end{aligned} \qquad (5.2)$$

5.2 Multi-Level Pipelined HW Architecture with Fast Motion and Disparity Estimation

Although significant complexity reduction was achieved through the Fast ME/DE algorithm presented in Section 4.3, real-time motion and disparity estimation feasibility for mobile devices depends on energy-efficient dedicated hardware architectures.

In this section is presented a hardware architecture (ZATT, SHAFIQUE, *et al.*, 2011) that integrates our Fast ME/DE algorithm with a multi-level pipelined parallel hardware to expedite the ME/DE process jointly at the algorithm and hardware levels. This solution is able to exploit the four levels of parallelism available in the MVC encoder and discussed in Section 5.2.1. The architecture scheduling and design are presented in Section 5.2.2 and Section 5.2.3, respectively. Preliminary results are discussed in Section 5.2.4.

5.2.1 Parallelism in the MVC Encoder

Due to the prediction structure used in the MVC as depicted by the arrows in Figure 5.7, four levels of parallelism can be exploited to achieve high throughput. For an easy understanding, the frames in Figure 5.7 are ordered according to the coding sequence using numbers for the KF and the alphabet order for NKF. The I frames are not processed by ME/DE and are considered available. Frames 2', 4' and 6' belong to the previous GOP.

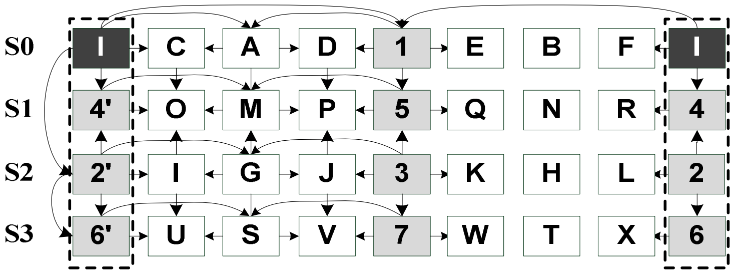


Figure 5.7: MVC prediction structure in our Fast ME/DE algorithm

View-Level Parallelism: Although MVC defines the *Time First* decoding order (i.e. all frames inside a GOP of a given view are processed and then the next view is processed), this order is not mandatory (i.e. not forced by the standard) during the encoding process, as far as the bitstream respects it. For instance, views S1 and S3 can be encoded completely in parallel after S0 and S2 reference views are available.

Frame-Level Parallelism: Within a view there are frames with no dependencies between them. For example, using one reference frame per prediction direction (1 west, 1 east, 1 north and 1 south) frames A and B can be processed in parallel. Analogously, it is possible to process the frames C, D, E, and F in parallel.

Reference Frame-Level Parallelism: Every single frame can be predicted using up to four reference frames to be encoded. The search in different reference frames has no data dependencies allowing the parallel processing. For instance, while encoding Frame P the search in reference frames 5, D, M, and J may be performed in parallel.

MB-Level Parallelism: Each MB has data dependencies to the previous encoded MB due to motion vector prediction process and SKIP vector prediction. However, using the Fast ME/DE scheme (see section 4.3.2) it is possible to start the predictors evaluation before the spatial neighboring MB. Additionally, the prefetching and SAD calculation may be started before obtaining the previous MB results (also possible for Zero Vector).

5.2.2 Multi-Level Pipelined Parallel Scheduling

A novel scheduling scheme is proposed for the MVC ME/DE as shown in Figure 5.8. The numbers and letters are consistent to Figure 5.7. The letters between parentheses represent the prediction direction E (East), W (West), N (North) and S (South). The dotted boxes represent a frame that belongs to the next GOP. This scheduling assumes the existence of two hardware modules executing/operating in parallel: one processing the TZ search for KF and the other processing the fast ME/DE for NKF. However, it can be easily be mapped to a serial architecture or extended to a more parallel hardware that exploits the four level parallelism knowledge.

Each time slot of the TZ Module is dedicated to perform the search for a complete KF in one reference frame. It is noticeable that the coding time for a given GOP is the time to perform 16 TZ searches. This number represents a reduction of 81% in the number of complete TZ searches if compared to a system without our fast ME/DE search (that performs 88 complete TZ searches).

For NKF encoding there is a Fast ME/DE module. After the required reference frames are processed by the TZ module (solving the data dependencies) all NKF in the same view are processed following the pre-defined coding order (as shown by the alphabetic order). The data dependencies between the KF and NKF are represented in Figure 5.8 by dashed arrows. To avoid pipeline stalls, the GOP-level pipeline starts the TZ search in KF of next GOP while the fast ME/DE Module concludes the current GOP processing. Since the Fast ME/DE represents less than 1% of the processing effort of TZ Search the fast ME/DE module process the NKF in a burst and, in the following, it is clock-gated (CG) until the next usage. For simplification, in Figure 5.8, the encoding of NKF does not present the details showing which prediction direction is tested. However, in the slot of given frame A, all required prediction direction are serially tested (in the specific case of frame A, West and East directions).

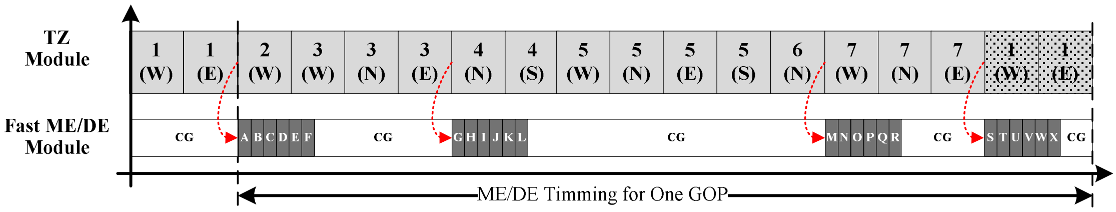


Figure 5.8: GOP-level pipeline schedule

The internal scheduling of the TZ Module operates at MB-level, as presented in Figure 5.9a. The three main tasks are: the algorithm control which is always active; the TZ search window prefetch logic which can be clock-gated after bringing the search windows from the external reference memory (DPB); and the TZ SAD computation that starts processing as soon the first useful reference block is available.

As the Fast ME/DE scheme has two prediction modes (the Fast and Ultra Fast prediction modes) two distinct pipeline scheduling are required for the Fast ME/DE Module. The Ultra Fast scheduling is presented in Figure 5.9b and the Fast scheduling in Figure 5.9c. Three tasks are considered: Fast ME/DE Vector Calculation, Data Prefetching, and SAD calculation. First, the Zero Vector is tested because it has no data dependencies with the spatial neighbors. Afterwards, the predictors are evaluated and the Common Vector (if it exists, for algorithm details check Section 4.3) or Predictor Vector 1 are processed (represented by the gray blocks in Figure 5.9b and Figure 5.9c). This is the second vector evaluation step (MB-Level Evaluation in Figure 4.34) once the vectors can be calculated based on the Frame-Level Evaluation (Section 4.3.2). If the spatial vector points to a different position, additional data is then fetched and processed (Predictor Vector N). The last vector to be tested is the SKIP predictor that depend upon the previous MB. In this pipeline stage, the previous MB MV/DV information must already be available to avoid pipeline stalls. The MB time borders (indicated by the vertical dashed lines in Figure 5.9b and Figure 5.9c) interfaces are the same for both prediction modes allowing the mode exchange (Fast↔Ultra Fast) with no pipeline stalls.

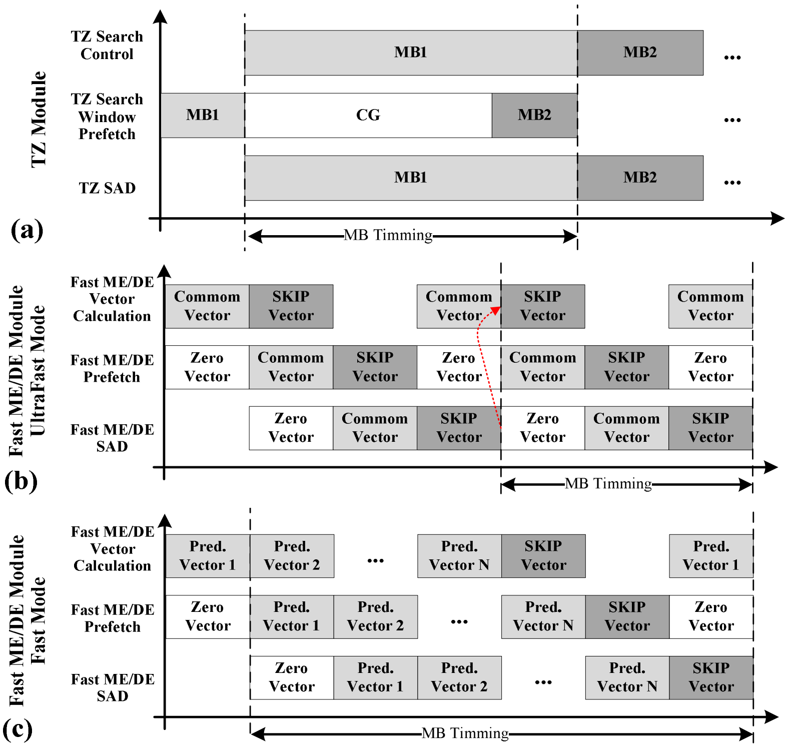


Figure 5.9: MB-level pipeline schedule for (a) TZ Module and Fast ME/DE Module in (b) Ultra Fast and (c) Fast operation modes.

5.2.3 Hardware Architecture

Based on the above-discussed pipeline schedule that exploits the MVC parallelism, a multi-level pipelined parallel hardware architecture was designed, described in RTL and synthesized down to logic synthesis level. Figure 5.10 shows the block diagram of our ME/DE hardware architecture. It is composed of three main modules: (a) TZ Search Module, (b) Fast ME/DE Module, and (c) Shared SAD calculator, which consists of SAD operator and adder trees. A SKIP Vectors Prediction module and MV/DVs storage memory shared by the TZ and fast ME/DE modules were also designed.

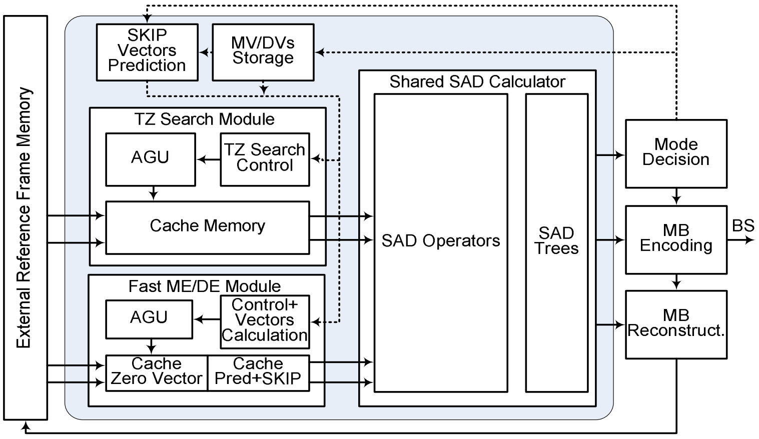


Figure 5.10: ME/DE hardware architecture block diagram

TZ Search Module: The TZ Module is composed of three internal modules: (i) TZ Search Control, (ii) Address Generating Unit (AGU), and (iii) Cache Memory. The TZ Search Control requires the search windows to be stored in the Cache Memory and the candidate blocks to be sent to SAD calculation. The implemented cache memory exploits the regularity of TZ algorithm by using the Level-C data reuse scheme as described in (CHEN, HUANG, *et al.*, 2006).

Fast ME/DE Module: The Fast ME/DE differs from TZ module in the control module and the cache memory organization. The control module is responsible for (i) reading the MV/DVs from a dedicated memory, (ii) calculating the Common Vector (in case Ultra Fast Prediction is used), and (iii) to fetch the reference frame data. However, since the Fast ME/DE module evaluates candidate blocks with very different regularity behavior, the cache is composed of two small memories. Since the Zero Vectors show regular distribution, the Level-A (CHEN, HUANG, *et al.*, 2006) caching technique is used for the Zero Vector Cache. For the predictors from the 3D-Neighborhood, the access to different reference frame areas may be required. The same behavior is noticed for SKIP predictor that cannot be anticipated. For this reason, our architecture employs the 3D-Cache (ZATT, AZEVEDO, *et al.*, 2007) originally developed for the H.264 decoder motion compensation. The 3D-Cache is used to store the data required for SKIP and 3D-Neighborhood predictors. In our ME/DE the 3D-cache is composed of 16 sets instead of the original 32 because our solution tests the predictors for a single reference frame at a time while the original memory was designed to support multiple reference frames.

Shared SAD Calculator: The relation between the processing time for one complete TZ search and one Fast Prediction lies in about 1:100. Therefore, to balance the parallel modules of TZ search and Fast ME/DE without extra hardware, the TZ SAD throughput should be 100x higher than the Fast ME/DE Module. Alternatively, a Shared SAD Calculator hardware was developed guaranteeing the fully usage of the SAD operators (and the adder trees) along the coding process. If the Fast ME/DE is performed, some operators are allocated for this task, otherwise all SAD operators are allocated to the TZ Search module. Moreover, this solution allows exploring the parallelism by simply varying the number of SAD operators (and Adder Trees) and the SAD operator allocation logic. In order to obtain 4-views HD1080p real-time MVC encoding, 64 4x4-SAD operators were instantiated, i.e., 256x the 4-sample SAD PEs described in Section 5.1.1.

5.2.4 Multi-Level Pipelined ME/DE Architecture Evaluation

The proposed architecture is designed, implemented, and verified in VHDL. It is synthesized and implemented (place and route) for a *Xilinx Virtex-6 xc6vlx240t* FPGA and an ASIC using *IBM 65nm LPe LowK* standard technology. For the *Xilinx Virtex-6 xc6vlx240t* FPGA, our architecture processes real-time ME/DE for HD1080p at a maximum frequency of 258MHz. It requires 4,308 Slices (9,876 LUTS) and 103 BRAM modules. The ASIC on-chip memory is based on the 1 Mb SRAM scheme for low power SoCs proposed in (FUKANO, KUSHIDA, *et al.*, 2008) that consumes 60mW at 300 MHz. The ASIC implementation details are presented in Table 5.2. Our design is able to provide real-time ME/DE for up to four-views HD1080p videos. Besides the faster hardware and the pipelined schedule considering four levels of parallelism, the throughput increase is due to the reduced number of candidate blocks per MB provided by our fast ME/DE algorithm.

Memory Overhead: While providing very high throughput and requiring a relatively low gate count compared to the current state-of-the-art solutions (as will be discussed in Chapter 7) the proposed architecture requires a high number of on-chip memory bits. The large memory is justified by the increased search range of [±64,±64] (required to encode increased resolutions) supported by our architecture, the capability to encode HD1080p (thus support for more MBs), and the support for three different caching schemes. Level-A requires 2 Kbits, Level-C 131 Kbits and 3D-cache 82 Kbits. Additionally, the MV/DV memory used by our fast ME/DE algorithm requires 522 Kbits (a limitation of our approach, but justified by the high throughput achieved). The prefetching schemes enable to reduce the external memory bandwidth by up to 65%.

Table 5.2: Comparison of Our Fast ME/DE Algorithm to TZ Search

	Multi-Level Pipelined Fast ME/DE Architecture
Technology	IBM 65nm LPe LowK
Gate Count	211k
SRAM	737 Kbits
Max. Frequency	300 MHz
Power	81mW, 0.8v
Proc. Capability	4-views HD1080p

Remember from our discussion in Section 3.2 that the energy drained by the on-chip video memory represents a meaningful share of total memory consumption. For this reason, Section 5.3 and Section 5.4 present alternative architectural solutions that emphasize the on-chip memory design and dynamic power management focusing on reducing the on-chip energy consumption. To provide overall energy reduction, the on-chip memory energy reduction is considered along with the external memory access energy consumption.

5.3 Motion and Disparity Estimation Hardware Architecture with Dynamic Search Window Formation

Aware of the dominant memory-related energy consumption we present in this section a hardware architecture featuring novel data prefetching scheme and on-chip memory management solution (ZATT, SHAFIQUE, *et al.*, 2011). Considering the previous MBs memory access our architecture is able to build a search map and predict the memory usage. It enables to read only the required data from external memory, avoiding performance drawback. Additionally, the mentioned memory access prediction allows the run-time management of the on-chip memory by adapting the power states of each memory sector, resulting in reduced energy consumption. The features of this architecture are summarized below.

Dynamically Expanding Search Window Formation Algorithm: Instead of prefetching the complete rectangular search window, a selected partial window is formed and prefetched for each search stage of a given fast ME/DE scheme depending upon the search trajectory, i.e., the search window is dynamically expanded depending upon the outcome of each search stage. An analysis is performed to highlight the issues related to the expansion of the partial window at each search stage. The search trajectories of the neighboring MBs and their spatial and temporal properties (variance, SAD, motion and disparity vectors) are considered to predict at run time the form of the search window for the current MB. This results in a significantly reduced energy for off-chip memory accesses.

Hardware architecture with Multi-Bank On-Chip Memory: A hardware architecture with parallel SAD modules is proposed. A pipelined schedule is proposed to enable high throughput. Moreover, the hardware is equipped with a multi-bank on-chip memory to provide high throughput in order to meet high definition requirements. The size and the organization of the memory is obtained by an analysis of the fast ME/DE scheme. Each bank is partitioned into multiple sectors, such that each sector can be individually power-gated to save leakage. The control of the power-gates is obtained from the application layer.

Application-Aware Power-Gating Scheme for the On-Chip Memory: Depending upon the fast ME/DE scheme and the macroblock properties, the amount of required data is predicted. Only the sectors to store the required data are kept powered-on and the remaining sectors are power-gated.

Before moving to the hardware description, the memory organization, and the power-gating algorithm, we present the access pattern analysis that provides the basis for our memory access pattern prediction. After that, the search maps used to predict the memory access pattern are introduced followed by the Dynamic Search Window formation algorithm. The memory hierarchy and its application-aware power gating are also presented in this section with detailed results.

5.3.1 ME/DE Memory Access Pattern Analysis

Real encoding systems do not implement the exhaustive search (Full Search, FS) but fast search algorithms. Fast ME/DE search algorithms usually are based on multiple search interactions following a given geometric shape and may employ start point selection and early stop criteria to reduce the computational complexity. These algorithms can provide expressive speedup and reduced number of memory accesses at the cost of negligible quality loss. However, real systems may suffer due to the irregular memory access pattern of external memory.

As a case study we present the memory access pattern for two fast ME/DE algorithms, TZ Search and Log Search, considering low and high motion MBs (see Figure 5.11). These search algorithms are implemented in the MVC reference software (JMVC) and their behavior represent a wide family of search algorithms. During ME the high motion areas perform higher number of memory accesses compared to low motion areas in which the search converges quickly. Analogous behavior happens for DE where objects with high disparity require more effort to find a good match. In Figure 5.12 the memory access profile for one frame is presented. The flat regions represent the low motion/disparity areas while the peaks are located at high motion/disparity ones. Other important observation is that for a same image region or object the number of memory access and the search pattern behavior is similar, i.e. neighbor MBs that belong to the same object tend to have similar memory and search pattern behavior.

Even considering high motion/disparity regions it is noticeable that big part of the search window is not accessed resulting in communication and storage energy wastage. Averagely, the ME/DE access 19.85% of the total search window using TZ search and 1.01% using Log Search. This represents that most of the search window is read and stored in vain, as detailed in Figure 5.13. The search pattern also is of key importance in the accuracy vs. memory access tradeoff. Compared to Log search, the TZ requires more memory accesses (see Figure 5.13), reaches extended search area (see Figure 5.11) and tends to provide more accurate search results.

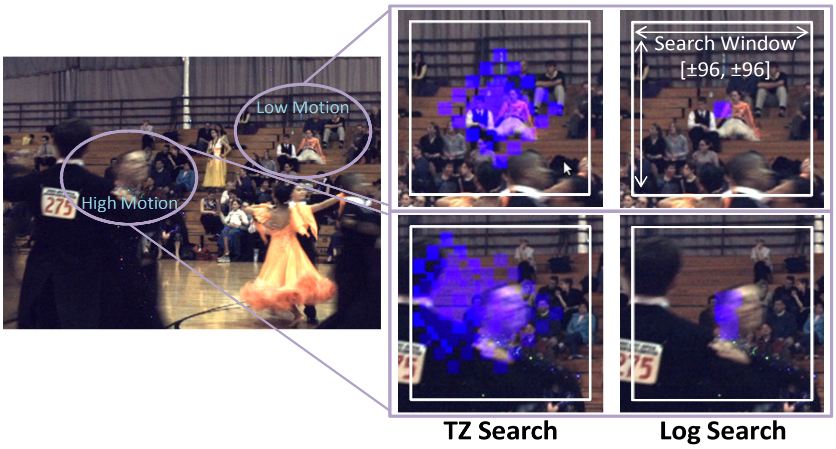


Figure 5.11: ME/DE search pattern for TZ Search and Log Search

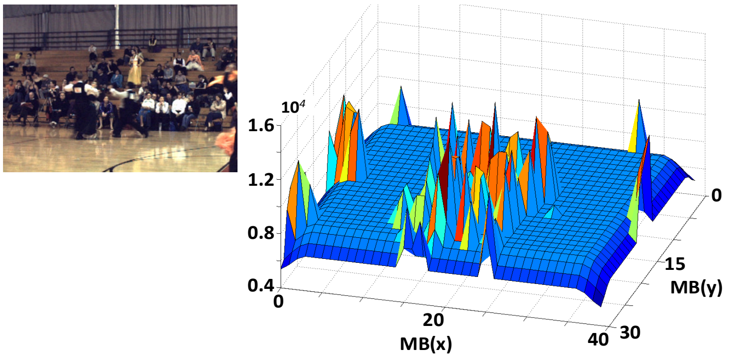


Figure 5.12: Number of pixels accessed in external memory

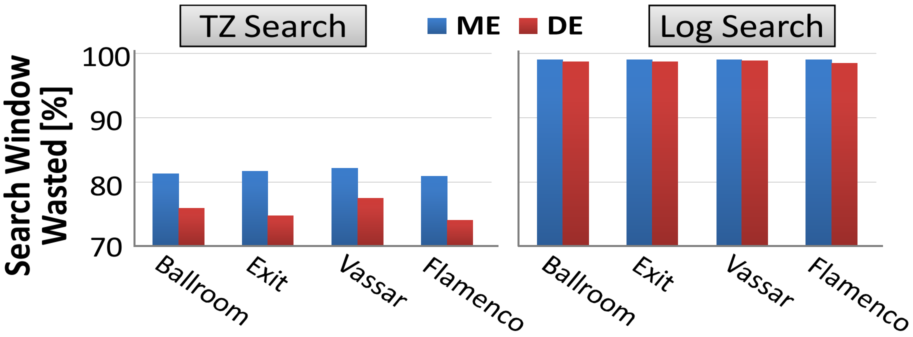


Figure 5.13: ME/DE search window wastage

5.3.2 Search Map Prediction

Figure 5.14 presents the *Search Map* for two neighboring MBs (denoted as MB_x and MB_{x+1}) using the Log Search algorithm. After the ME search is performed for the MB_x, a Search Map is built based on the search trajectory (i.e., the ID of the selected candidate search points at each search stage of the ME/DE scheme). As shown in Figure 5.14a, the first search stage selects the candidate with ID 6 as the best candidate. Similarly, candidates with ID 3 (at stage 2), ID 4 (at stage 3), and ID 4 (at stage 4) are selected as the best candidates at their respective search stages. This provides a Search Map of [6,3,4,4] (the trajectory is shown by the red arrows). Note, for each search stage there is an entry in the Search Map with the ID of the candidate with minimum SAD at that particular search stage.

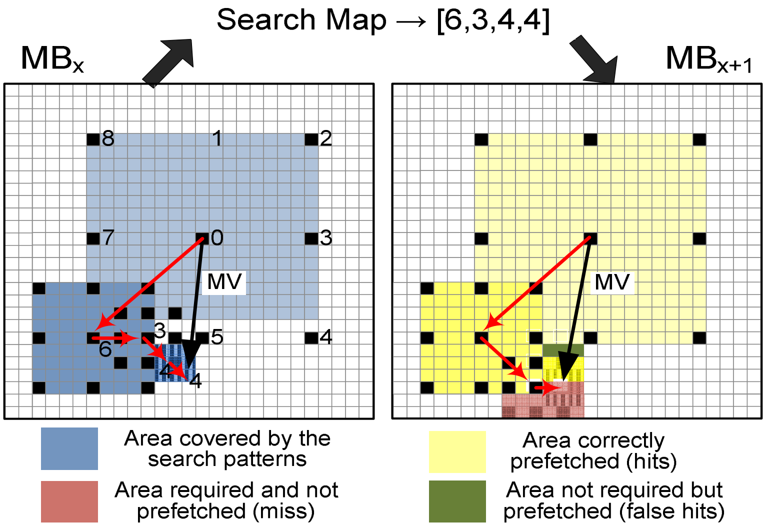

Search Map → [6,3,4,4]
MBx
MBx+1
8
1
2
7
0
3
MV
3
5
4
6
4
4
MV
Area covered by the search patterns
Area required and not prefetched (miss)
Area correctly prefetched (hits)
Area not required but prefetched (false hits)

Figure 5.14: Search Map Prediction for the Log Search

Considering the analysis of the MB neighborhood, a Search Map for the MB_{x+1} can be predicted from the Search Map of MB_x. In case there is a deviation in the search trajectory of these two MBs, there will be a miss in the on-chip memory due to the prefetch of the false region (see the green box in Figure 5.14b). Typically, these misses are at the boundaries of the moving objects and occur in relatively few MBs along the whole video frame. In case of a miss, there will be a stall only for the prefetching of the first candidate data on the new trajectory (i.e., 16x16 pixel data). All other candidates on the new trajectory will be then prefetched correctly (before their respective SAD computations, thus not causing any stall) as the search pattern design of the fast ME/DE schemes is fixed at design time (see the brown box for the new prefetched data). Typically a miss in the trajectory depends upon the motion/disparity difference of two MBs, which is significantly smaller in most of the neighboring MBs due to high correlation between them.

5.3.3 Dynamic Search Window Formation

Figure 5.15 depicts the pseudo-code for the algorithm of the dynamic search window formation and expansion. The algorithm works in two major steps. First it predicts the Search Map from the spatial predictors (lines 3-21). Afterwards, it checks if the search pattern matches the Search Map, prefetches the appropriate partial search window, and updates the *Search Map* (lines 23-33).

Four spatial predictor with presenting high correlation with the current MB are used to predict the Search Map (line 4). Afterwards, variance of these predictors is computed and motion and disparity information is obtained (lines 5-6). Based on the spatial, temporal, and view information, a matching function is computed that provides a hint that predictors may belong to the same object or may exhibit similar motion/disparity properties (lines 7, 9-12). Afterwards, the predictors are sorted with regard to their similarity to the current MB (line 14). The closest predictor is determined by computing the SAD with the current MB (line 15). In case the closest predictor also belongs to the same object or exhibit similar motion/disparity, its Search Map is considered as the predicted Search Map (line 17). Otherwise, the closest map is found in the remaining predictor set (line 19). If none of the predictors exhibit similarity to the current MB, then the predicted Search Map is empty.

```
1.  // Predict the Search Map from the Neighboring MBs
2.  PredictorSet ← Ø;
3.  If (PredictorsAvailable) Then
4.     PredictorSet = {$MV_{Left}$, $MV_{Top}$, $MV_{TopRight}$, $MV_{SpatialMedian}$};
5.     computeVariance (PredictorSet); //Compute Variance of all predictors
6.     getTemporalInfo (PredictorSet, currMB); //Get MV, DV, SADs
7.     $Motion_{currMB}$ = ($SAD_{currMB}$ > $TH_{SAD}$)? 1: 0;
8.     For i = 0 to all Predictors //Compute the Similarity of predictors,i.e., check if predictors
    belong to the same object as of the current MB
9.        $diffVar_{predi}$ = $Var_{currMB}$ - $Var_{predi}$;
10.       $Motion_{predi}$ = ($SAD_{predi}$ > $TH_{SAD}$)? 1: 0;
11.       $diffMotion_{predi}$ = $Motion_{currMB}$ - $Motion_{predi}$;
12.       $predDiff_{predi}$ = $\alpha$*$diffVar_{predi}$ + $\beta$*$diffMotion_{predi}$;
13. End For
14.    PredictorSet = sortPredictors (predDiff, PredictorSet);
15.    bestPred = determineBestPredictor (PredictorSet, currMB);
16.    If ($predDiff_{bestPred}$ < $TH_{diff}$) Then
17.       predSearchMap = $SearchMap_{bestPred}$;
18.    Else
19.       predSearchMap = findClosestSM (PredictorSet, $TH_{diff}$);
20. End If
21. End If
22. // Perform Dynamic Formation and Expansion of the Search Window
23. For all SearchStages        // Depending upon the fast ME/DE scheme
24. SM_Miss = checkSearchMap (searchStageID, predSearchMap);
25.    If ((PredictorSet == Ø) or SM_Miss) Then
26.       SWBuffer = prefetchPartialWindow (refFrame, searchStageID,
             searchStagePattern);
27.    Else
28.       SWBuffer = prefetchPartialWindow (refFrame, searchStageID,
             predSearchMap);
29. End If
30.    bestCand = performMEDE (currMB, SWBuffer, SearchAlgorithm);
31.    Build_CurrMB_SearchMap (bestCand, searchStageID);
32.    If (earlyTermination)     return;     End If
33. End For
34. return;
```

Figure 5.15: Algorithm for Search Map Prediction and the Dynamic Formation and Expansion of the Search window

After the Search Map is predicted, it is used to form the search window. For each search stage, the partial search window is determined according the Search Map and prefetched. In case the search candidates of the search pattern are present in the Search Map (i.e., the search trajectory falls in the predicted region), the partial search window is simply constructed according to the predicted Search Map and the prefetched data is used (i.e., a case of *hit*) (see line 28). Otherwise, if the Search Map is empty or does not contain the search candidate, the Search Map is ignored for this stage onwards (see line 26). In this case the prefetched data is wasted and it is considered as a *miss*. The partial search window is then constructed according to the search pattern for the miss parts (see line 31, it can also be seen in the example of Figure 5.14b).

In the following section we discuss the architecture of our joint ME/DE scheme along with the design of the multi-bank on-chip memory and application-aware power-gating.

5.3.4 Memory Hierarchy

On-Chip storage of rectangular search windows incurs in increased area and leakage of on-chip memory, like those presented in (CHEN, HUANG, *et al.*, 2006)(DING, CHEN, *et al.*, 2010)(SAPONARA e FANUCCI, 2004)(CHEN, CHEN, *et al.*, 2007)(TSUNG, CHEN, *et al.*, 2009)(TSAI, CHUNG, *et al.*, 2007). The size of the dynamically formed search window is significantly lower compared to the rectangular search windows. This scenario becomes even more critical in MVC where ME and DE are performed for multiple views using larger search windows (for instance [±96, ±96] to capture high disparity regions in DE). Depending upon the MB properties, the sizes of dynamically expanding search windows may vary significantly. However, the size of on-chip memory that stores this window must be fixed at design time. Therefore, we firstly perform a design space exploration to obtain a reasonable size of the on-chip memory (that provides leakage power reduction and area savings). In case the MB exhibit low motion and the size of the prefetched window is still less than the on-chip memory, the remaining parts of the memory are power-gated to further reduce leakage.

Figure 5.16 demonstrates the design space exploration for the memory access distribution using *Ballroom* video sequence (a fast motion sequence). Figure 5.16a shows the number of MBs for which ME and DE require less than 96 MBs. Here, a MB-based memory fetching is considered. Please note that the reduced number is mainly due to the adaptive nature of fast ME/DE schemes and it does not mean that this is within a smaller search range. A rectangular search window of [±96, ±96] size requires 37KB of on-chip memory. Figure 5.16b shows that even for such a large search range, at most 96 candidates are evaluated per MB. This corresponds to an on-chip memory of 24KB, i.e., a reduction of 35% area. When scrutinizing the Figure 5.16b, it is noticed that in more than 95% cases a storage of only 64MBs is required (i.e., 16 KB → 57% savings). We have performed such an analysis for various video sequences with diverse motion (not shown here due to space reasons). Similar observations were made in all of the cases. Therefore, we have selected an on-chip memory of 16KB, which provides significant leakage reduction in the on-chip memory. In rare cases, where the ME and DE may require more MBs, misses may happen (as we will show in section 5.3.7.1).The on-chip memory is organized in 16 banks, where one 16 pixel row of an MB is stored in each of the banks, in order to guarantee high parallel throughput.

As discussed above, even 16KB memory may not be completely used to store the dynamically expanding search window as the size of prefetched search window highly depends upon the MB properties and the fast ME/DE scheme (it can be seen in Figure 5.16b that in more than 20% of the cases storage for 32MBs is used, i.e., only half of the memory). Therefore, each bank is partitioned into multiple sectors (8 sectors in this case) where each sector can be individually power-gated to further reduce the leakage (see Figure 5.17). The main challenge here is to incorporate the application and video knowledge to determine the power-gate control, such that the power-gating signals may be determined depending upon the predicted memory requirements of the current MB.

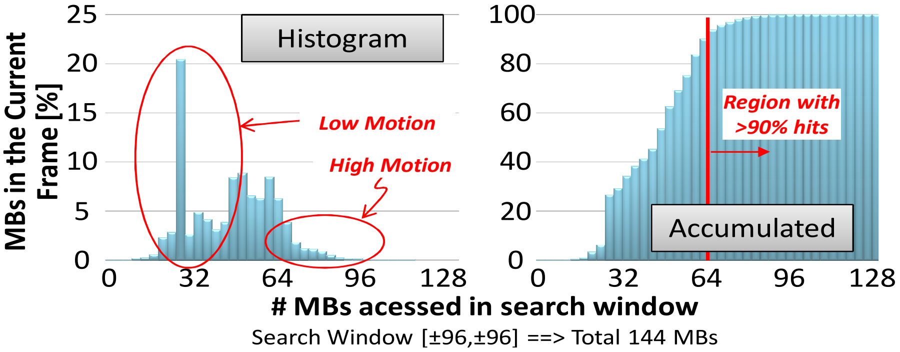


Figure 5.16: Analyzing the memory requirements for ME/DE of different MBs in Ballroom sequence

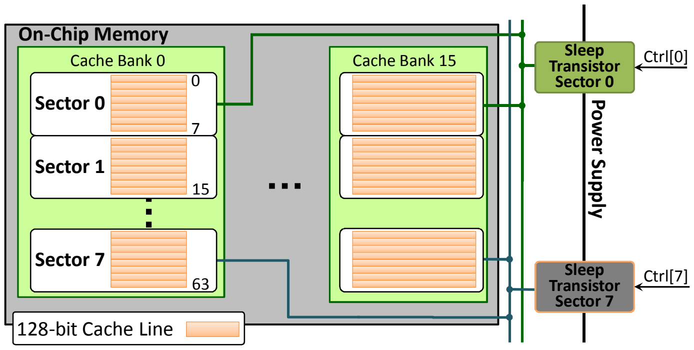


Figure 5.17: Search Window Memory organization with power-gating

5.3.5 Application-Aware Power Gating

A previously discussed, MB properties provides a relatively high potential for leakage reduction by accurately predicting the memory requirements of MBs before their actual ME/DE. However, frequent on-off switching needs to be avoided to reduce power-gating wakeup energy overhead. Therefore, our scheme predicts the sleep time as function of *n* consecutive MBs whose sectors can be jointly power-gated. Considering the worst case of stationary MBs, to overcome the wakeup energy overhead, the condition defined in Eq. (5.3) must hold.

$$P_{leak\_onChipMem} * T_{minMEDE} * n > E_{wakeup} \qquad (5.3)$$

However, the minimum ME/DE time depends upon the deployed fast ME/DE scheme. For instance, in case of a stationary MB there will be a minimum of 9 SAD computations for the Log Search and for the TZ Search it is 46 SADs. Therefore, considering the minimum number of SADs for a stationary MB, Eq. (5.3) can be re-written as Eq. (5.4) where *minNumberSADs* is 9 and 46 for Log and TZ Searches, respectively. For a given sleep transistor design and a given SRAM memory, the *n* can be determined. In reality, MBs exhibit diverse motion and spatial properties. Therefore, the number of consecutive MBs that require a certain amount of on-chip memory may be even less than *n*.

$$n > (E_{wakeup} / P_{leak\_onChipMem} * minNumberSADs * T_{SAD}) \qquad (5.4)$$

Let's assume *n* consecutive MBs require at most *R* KB of on-chip memory for their search window prefetching. For a given on-chip memory of size S_{memory} KB with N_{Sec} number of S_{Sec} KB sectors, the amount of power-gateable memory sectors is computed by Eq. (5.5).

$$N_{gateableSectors} = (S_{memory} - R) / S_{Sec} \qquad (5.5)$$

The control signal is generated by the Power Gating Control unit by simply reading the motion and disparity vectors from 3D-Neighborhood and counting the number of consecutive low motion/disparity MBs.

5.3.6 Hardware Architecture

Figure 5.18 shows the hardware architecture with our proposed dynamic search window formation scheme. It employs the above-discussed dynamically expanding search window prefetching and a multi-banked on-chip memory with application-aware power gating control. In order to obtain high throughput, a set of 64 (4x4-pixel) SAD operators and SAD trees is provided as the main computation block. An ME/DE search control unit is integrated which can be programmed to realize various fast ME/DE schemes. This unit controls the search stages and patterns, and it provides the required algorithmic information to various other modules. The search window formation unit predicts the Search Map and dynamically constructs the search window structure. This data corresponding to the window is prefetched in the multi-bank search window memory which consists of various sectors that can be individually power-gated (ZHANG, BHATTACHARYA, *et al.*, 2005) depending upon the ME/DE requirements of the current MB.

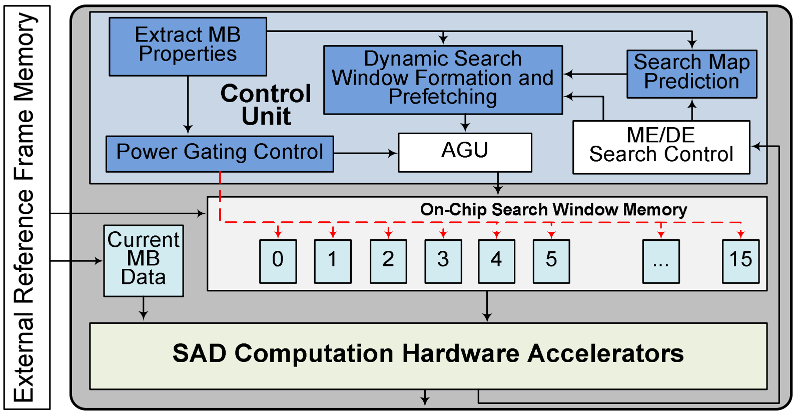

External Reference Frame Memory
Extract MB Properties
Control Unit
Dynamic Search Window Formation and Prefetching
Search Map Prediction
Power Gating Control
AGU
ME/DE Search Control
On-Chip Search Window Memory
Current MB Data
0
1
2
3
4
5
...
15
SAD Computation Hardware Accelerators

Figure 5.18: ME/DE hardware architecture block diagram

In Figure 5.19 is presented the MB-level ME/DE processing pipeline scheduling including the data prefetching and SAD computation for different search stages. During the SAD computations of the preceding search stage, the partial search window data is prefetched for the succeeding search stage. However, in case of a Search Map miss, stall for one candidate data prefetch happens (see *A* in Figure 5.19). In case the fast ME/DE scheme stops the search due to early termination criteria, the prefetch data in the search window is wasted (see *B* in Figure 5.19).

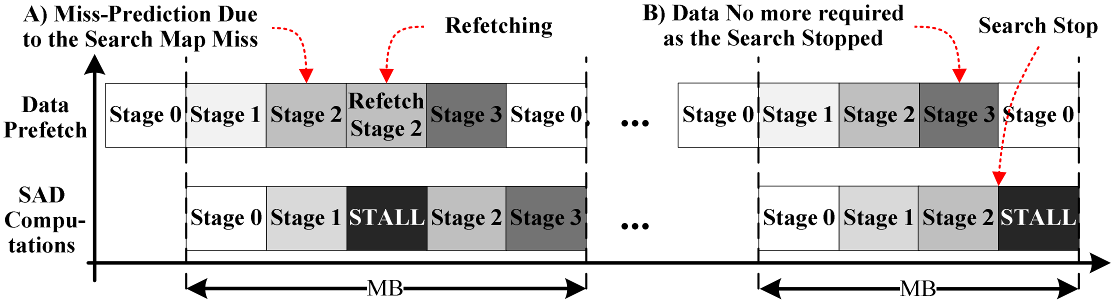


Figure 5.19: Pipeline Processing Schedule of our ME/DE Architecture

5.3.7 Hardware Architecture Evaluation

5.3.7.1 Dynamic Window Formation Accuracy

For the detailed experimental results presented in this section a set video sequences with 4 views each was used. The search algorithm used were *TZ Search* (YANG, 2009) and *Log Search* (JVT, 2009) considering three QP values (22,27,32) and search in the four possible directions with a search window of [±96, ±96]. The thresholds set used were: N=6, α=1, β=500 and TH_{SAD}=400.

Figure 5.20 presents details for the Search Map and on-chip memory evaluation. Figure 5.20 shows that the accuracy of Search Map prediction is higher for low motion sequences (e.g., *Vassar*) compared to high motion sequences (e.g., *Flamenco2*) as the search trajectory is shorter and easier to be predicted (due to a higher number of stationary/slow-moving MBs). However, even for the worst case the hits are around 80% (see Figure 5.20a). In case of off-chip memory accesses, the misses are higher for low motion sequences because the search trajectory tends to converge to the center (only the central region of search window is accessed) reducing the overlapping accessed area with the neighboring MBs. The higher number of memory misses for low motion sequences, however, does not limit off-chip energy savings achieved for the same sequences. The reason is that the percentage of misses is calculated over a much smaller number of total memory accesses for low motion sequences.

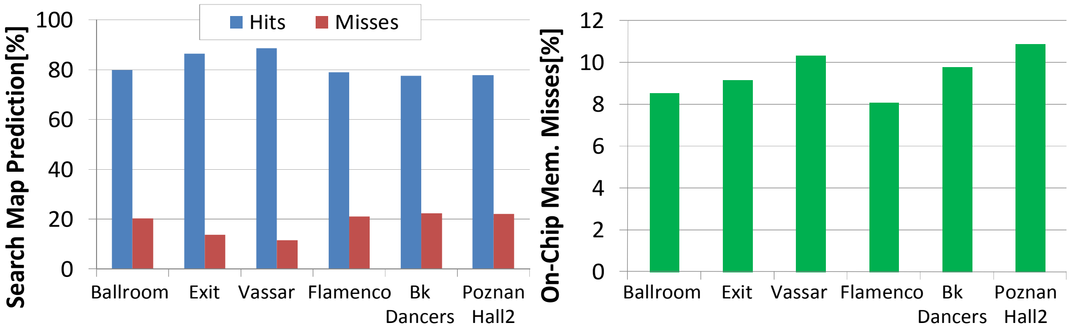


Figure 5.20: Search Map Prediction accuracy and on-chip memory misses

5.3.7.2 Hardware Architecture Evaluation

Table 5.3 presents the ASIC implementation results of our architecture. The hardware implementation executes at 300MHz and provides real-time ME/DE for up to 4-views HD1080p consuming 74mW. Reduced power is reached mainly due to the employment of dynamic search window formation, power-gating, smaller logic, and fast ME/DE scheme. Please, refer to Section 6.2 for comparison to state-of-the-art ME/DE architectures.

Table 5.3: Comparison of Our Fast ME/DE Algorithm

	Fast ME/DE Architecture w/ Dynamic Search Window Formation
Technology	ST 65nm Low-Power 7 metal layer
Gate Count	102k
SRAM	512 Kbits
Max. Frequency	300 MHz
Power	74mW, 1.0v
Proc. Capability	4-views HD1080p

5.4 Motion and Disparity Estimation Hardware Architecture with Adaptive Power Management

Based on in-depth memory access correlation analysis (see Section 5.4.1) it was possible to conclude that the memory access prediction can be further improved in relation to the solution presented in Section 5.3. As a result, novel memory power-gating control schemes may be proposed. In this section we present a ME/DE hardware architecture featuring an adaptive power management scheme (ZATT, SHAFIQUE, *et al.*, 2011) able to consider the 3D-Neighborhood correlation and reduce the energy consumption related to memory. The memory hierarchy and power-management are carefully explained along this section. Below the main features implemented in this architectural solution are summarized.

An On-Chip Multi-banked Video Memory: based on the offline memory usage analysis, an algorithm is proposed to determine the size of the on-chip memory by evaluating the tradeoff of leakage reduction and misses (as a result of reduced-sized memory). Afterwards, the organization (banks, sectors) is obtained by considering the throughput constraint. Each bank is partitioned into multiple sectors to enable a fine-grained power management control. The data for each prediction direction is stored in distinct sections.

An Application-Aware Power-Gating Scheme for the On-Chip Memory: A multi-level power-management scheme is employed. First, depending upon the current prediction direction (top, left, down, right, i.e., using the knowledge from the application that determines a prediction direction), different sectors can be completely power-gated. Afterwards, frame-level memory requirements are predicted by taking the weighted-average of the neighboring frames in the 3D-Neighborhood. Then, the consecutive MBs with similar spatial and temporal properties are grouped together and sleep modes for their idle sectors are determined by evaluating a cost function of leakage savings and wakeup overhead. In the last step, the power-gating control of different sectors is refined at the MB level.

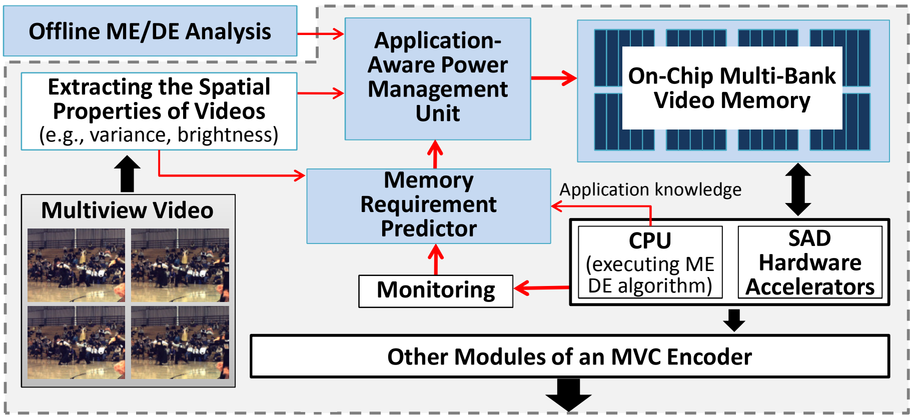

Offline ME/DE Analysis
Application-Aware Power Management Unit
On-Chip Multi-Bank Video Memory
Extracting the Spatial Properties of Videos (e.g., variance, brightness)
Multiview Video
Memory Requirement Predictor
Application knowledge
CPU (executing ME DE algorithm)
SAD Hardware Accelerators
Monitoring
Other Modules of an MVC Encoder

Figure 5.21: MVC with motion and disparity estimation hardware

To the best of our knowledge, this is the first multi-banked video memory architecture that employs a multi-level application-aware power management scheme to enable low-power motion and disparity estimation in MVC. The proposed architecture and power-management scheme require the knowledge of ME/DE algorithm and the search window perfecting technique in order to perform a memory-requirement analysis, though our concept is not limited to any fixed algorithm. Figure 5.21 presents an MVC encoder with joint ME/DE hardware architecture, showing our novel contribution in blue filled boxes.

5.4.1 Motion and Disparity Estimation Memory Access Analysis

Instead of a *Full Search* (impracticable due to very high computation and energy requirements), an adaptive fast ME/DE algorithm (*TZ Search*) is deployed for this analysis (to represent a real-world embedded system scenario). These adaptive algorithms are typically based on multiple search stages and patterns, and process different number of search candidates for different MBs, thus exhibit highly-varying memory usage profile, as shown in Figure 5.22 and Figure 5.24.

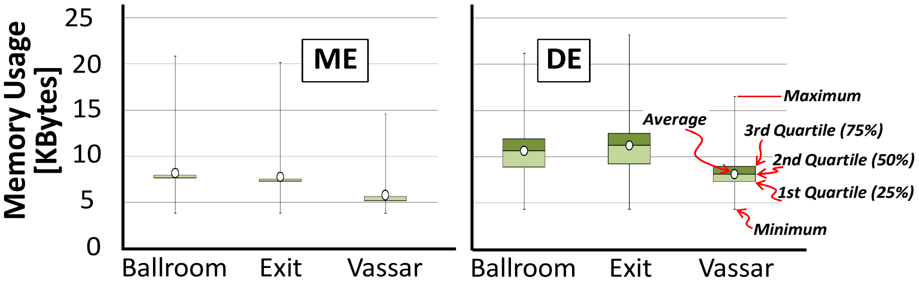


Figure 5.22: Summary of memory usage of various macroblocks for ME and DE

Figure 5.22 shows the box-plot summary of memory usage for different MBs in various video sequences for an on-chip memory of size 37.25 KB (storing a search window of [±96,±96], which is recommended for ME/DE (XU e HE, 2008)). However, the maximum measured memory usages are 20.9 KB (i.e., memory wastage of 44%) and 23.2 KB (i.e., memory wastage of 38%) for ME and DE, respectively. Still, most of the MBs require much less memory than the maximum requirements. The minimum and maximum memory requirements vary for different video sequences due to their spatial and temporal properties. In the worst case, more than 80% of the on-chip memory may be idle, thus leading to significant energy wastage due to leakage. Figure 5.22 shows that, in case of ME, the box plot is less scattered and close to the average. It demonstrates a high correlation in the memory usage profile for ME. However, the observation is different for DE, where the range between the 25% and 75% quartiles is relatively wider compared to that of ME. Still, the 75% quartile is much below than the maximum usage. However, care needs to be taken, as a misprediction may incur significant misses, thus a high penalty in terms of re-fetching from the external memory and wakeup of additional memory sectors. The less scattered distribution in the box-plot hints towards the fact that there is an extensive correlation in the 3D-Neighborhood, i.e., MBs in the neighboring frames and views exhibit similar memory requirements, as they belong to the same object. This fact become apparent in the 3D plot of Figure 5.23. Therefore, the memory requirements of a frame may be predicted (with a high accuracy) by exploiting the correlation in the 3D-Neighbohood (i.e., memory requirements of the neighboring frames). The frame-level prediction can be further refined considering the MB-level properties.

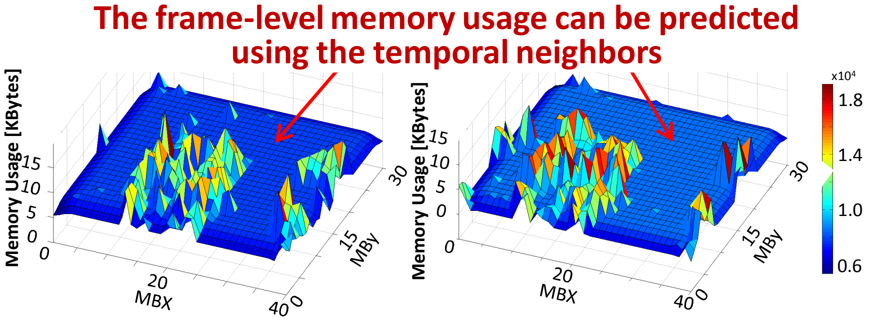


Figure 5.23: 3D- Plots showing the similarity in memory usage

When further analyzing the memory requirements within a frame (see Figure 5.24 for *Ballroom* sequence), two different variation zones are noticed in ME that correspond to two different groups of MB, where MBs in a group have similar spatial and temporal properties. MBs in the group-1 exhibit a low-variation in their memory usage, while

MBs, in the group-2 exhibit high-variation. The distinction between two groups can be made by evaluating the average spatial and temporal properties of MBs. Depending upon the group-level variations, low-leakage or high-leakage sleep mode may be selected. The large variations for DE are primarily due to the bigger search performed by the TZ algorithm for capturing longer disparity vectors.

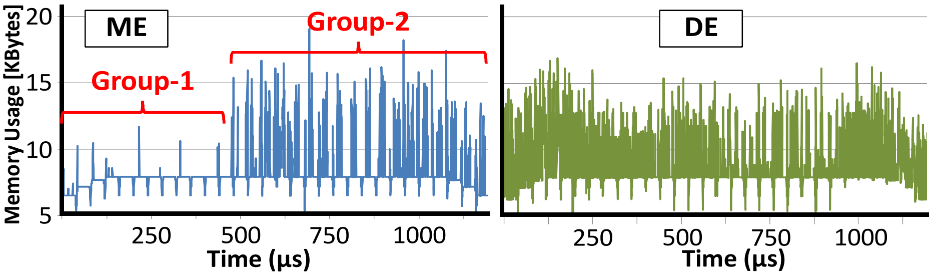


Figure 5.24: Memory usage variation within one video frame

Figure 5.25 shows four excerpts of the *Ballroom* sequence illustrating the memory access behavior (within the white lines) of MBs with low and high ME/DE. The memory access behavior of the MB with low motion/disparity is less spread and focused towards the centre (i.e., less memory is used in a smaller vicinity). In contrast, the memory access behavior of the MB with high motion/disparity shows that the memory from a wider region is accessed (see multiple displaced diamond patterns). Figure 5.24 and Figure 5.25 demonstrate that the memory requirements of an MB can be accurately prediction by considering its spatial and temporal properties and the memory requirements of MBs of the same group.

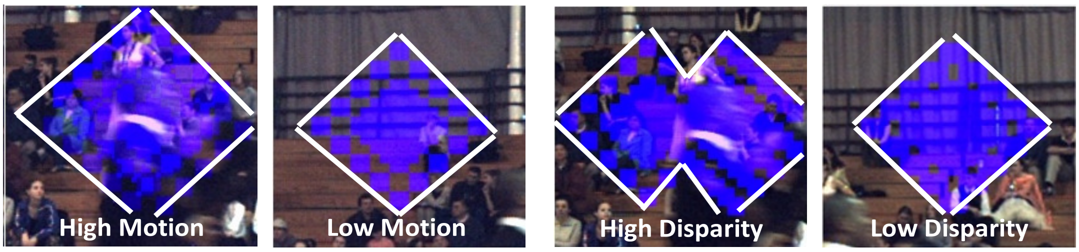


Figure 5.25: Comparing the memory access patterns of MBs with slow and fast motion/disparity

Summarizing, an application-aware power management scheme for an on-chip video memory needs to consider the knowledge of ME/DE algorithm, spatial and temporal video properties (at both frame and MB levels), and correlation in the 3D-Neighborhood to determine the number of idle sectors and an appropriate sleep mode for each sector.

5.4.2 Memory and Power Model

The on-chip video memory is partitioned into N_{Banks} banks, such that the rows of an MB are stored in different banks to provide parallel data access for the SAD accelerator hardware in order to support high-throughput constraints. Each bank $B_{i;\ i\in[1...NBanks]}$ is composed of N_{Sector} equally-sized sectors. Each sector consists of S_{Sector} number of bytes organized in memory lines, where the size of one memory line is given as NB_{Line}. This implies that the number of lines in a sector S_{ij} is $\mathrm{S}_{\mathrm{Sector}}/NB_{Line}$. Figure 5.26 shows an abstract diagram of our memory organization.

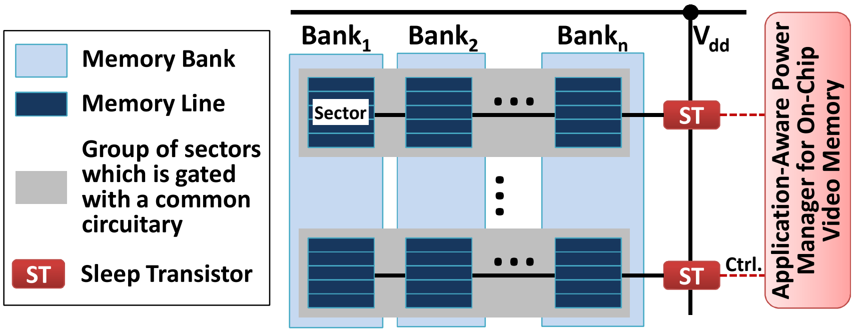

Memory Bank
Memory Line
Group of sectors which is gated with a common circuitary
ST
Sleep Transistor
Bank1
Bank2
Bankn
Vdd
Sector
ST
ST
Ctrl.
Application-Aware Power Manager for On-Chip Video Memory

Figure 5.26: Architectural model of the on-chip multi-banked memory with sleep transistors and application-aware power manager

All $S_{ij;\ i\in[1...NSector]}$ sectors are connected to a power gate circuitry ST_j in order to simultaneously power gate the S_{ij} sectors of all banks. In this thesis, we assume the power-gate model with multiple sleep modes (like in (SINGH, AGARWAL, *et al.*, 2007)(AGARWAL, NOWKA, *et al.*, 2006)(ROY, RANGANATHAN e KATKOORI, 2011)), where each sleep mode has a certain leakage savings at the cost of a wakeup energy and latency overhead. Therefore, using multiple sleep modes provide the foundation to exploit the wake-up overhead vs. leakage saving tradeoff. Different sleep modes are typically realized by controlling the virtual ground bias using footer transistors.

Figure 5.27 shows the power state machine (PSM), where each sector can be power gated in one of the three sleep modes, i.e., S_1, S_2, and S_3. The S_0 mode corresponds to the *powered-on* state. PSM is given as $P_{SleepMode} = \{S_0, S_1, S_2, S_3\}$. For the S_0 mode, the leakage energy is computed based on the drain current 'I' and V_{dd}, i.e, $E_{S0} = \Sigma V_{dd}.I_i.t_i$. The S_1 and S_2 modes are intermediate state-retentive sleep modes, i.e., data inside the memory cells is preserved and this mode does not require re-fetching of data from the off-chip memory. For these sleep modes, the total energy is computed as $ES_1=E_{S0}\,.\Phi_{S1}$ and $E_{S1}=E_{S0}\,.\Phi_{S1}$, where Φ_{S1} and Φ_{S2} are calculated using the design curves for footer gate bias vs. normalized leakage and footer gate bias vs. virtual ground voltage, as discussed in (SINGH, AGARWAL, *et al.*, 2007). The S_3 mode is a non-retentive state, i.e., data is lost, requires re-fetching from the off-chip memory. It is also termed as an *powered-off* state and the wakeup energy from S_3 to S_0 depends upon the capacitance ($C_{circuit}$) and V_{dd}, $E_{wake\_up}=½.C_{circuit}.V_{dd}^2$ (see Figure 5.27). The wake-up penalty for other transitions depends on the scale factor ξ_x for wake-up energy and ρ_x for wake-up latency, where x represent the transition $x \in \{T_1, T_2, T_3\}$. The scale factors are obtained from the design curves for normalized leakage vs. normalized wakeup-penalty, as discussed in (SINGH, AGARWAL, *et al.*, 2007). The wakeup latencies of S_1 and S_2 are quite short (for values, see Table 5.4), thus these modes are beneficial for short sleep durations, i.e., in the Group-2 with fast variations of memory usage by different MBs (as shown in Section 5.4.1). In contrast, S_3 is beneficial for longer sleep durations. Correspondingly, S_1 and S_2 modes also provide reduced leakage savings compared to S_3.

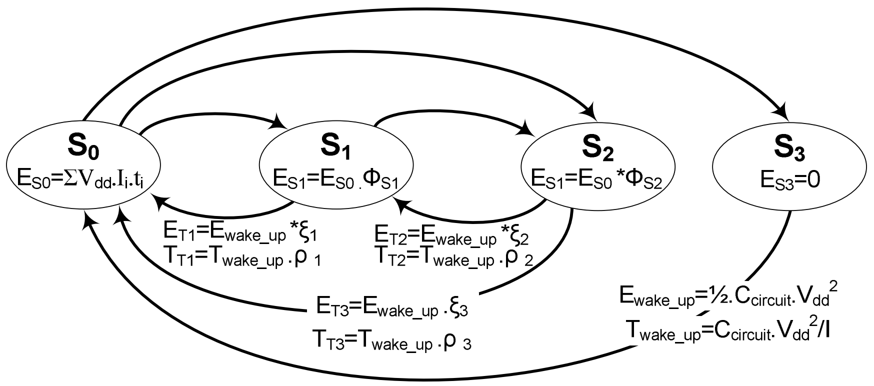


Figure 5.27: PSM of power-gate with multiple sleep modes

The leakage energy of the total on-chip video memory and the energy for memory misses are given by Eq. (5.6) and Eq. (5.7). P_{Leak} is the accumulated leakage power for the total on-chip video memory, T_{MEDE} is the time for performing Motion and Disparity Estimation (ME, DE), and E_{Miss} is the energy required for one memory miss that includes the energy to fetch data from the off-chip memory ($E_{offChipAccess}$), additional energy due to the stalling of the SAD hardware ($E_{HWstall}$), and energy to fill the memory line ($E_{lineFilling}$). Table 5.4 shows the set of thresholds and power mode parameters used in our experiments.

$$E_{Leak} = P_{Leak} \times T_{MEDE} \tag{5.6}$$

$$\begin{aligned} E_{MissTotal} &= E_{Miss} \times N_{Miss} \\ E_{Miss} &= E_{offChipAccess} + E_{HWstall} + E_{lineFilling} \end{aligned} \tag{5.7}$$

Table 5.4: Power Model Parameters and Thresholds

Φ_{S1}	0.5	α	0.65
Φ_{S2}	0.3	β	0.35
ξ_1	0.35	ρ_{S1}	0.1
ξ_2	0.35	ρ_{S2}	0.2
ξ_3	0.6	ρ_{S3}	0.3

5.4.3 Multi-Bank Video Memory Architecture

The goal is to determine an appropriate size of the on-chip video memory and its organization in terms of number of banks, sectors in a bank, etc. The parameters that can affect the size of the on-chip video memory are: (a) ME/DE search algorithm, number of search candidates in different search stages; (b) search range that also depends upon the video resolution; and (c) spatial and temporal video properties. Figure 5.28 shows the histogram of memory accesses during ME and DE for a search range of [±96, ±96]. The memory usage in ME/DE is much less than the size of a rectangular search window (37.25 KB). The maximum requirement is < 20 KB what represents 54% of the total size of a rectangular search window. This leads to an increased leakage. However, using a rectangular search window ensures no misses, as all the data is always available in the on-chip memory. In case a reduced-sized memory is used, the probability of misses increases. This fact is illustrated in Figure 5.29 with the help of two histograms from our memory miss analysis. The histograms show that the number of misses decreases exponentially with a linear increase in the memory size. Especially the reduction rate is significant for ME. The challenge is to obtain the size of on-chip video memory for ME/DE, such that the leakage savings (due to reduced size compared to the rectangular search window) are balanced by the energy overhead due to misses.

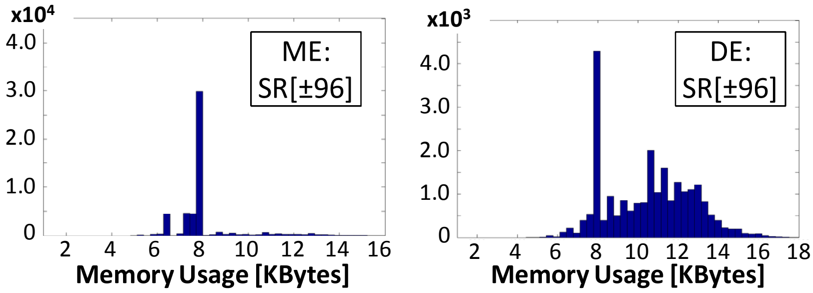


Figure 5.28: Histograms of memory usage during ME and DE

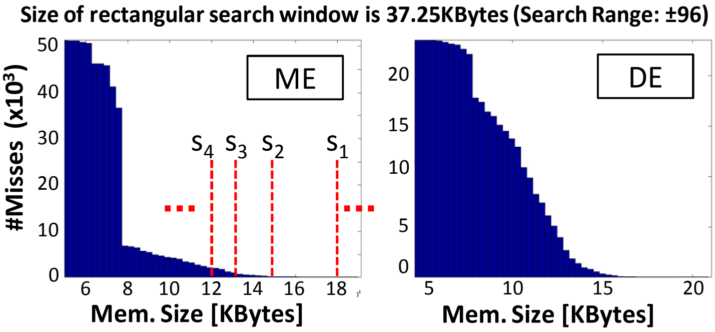


Figure 5.29: Histograms of memory misses (ME and DE) identifying potential size options for the on-chip memory

Figure 5.30 presents the pseudo-code for the proposed algorithm to perform memory size exploration for a given prediction direction. The input is a set of different size options $S=\{s_1, s_2, ..., s_n\}$, obtained from an extensive memory usage analysis for a set A of various video sequence (slow-fast motion), as exemplified in Figure 5.28 and Figure 5.29. Further inputs are (a) leakage energy of the rectangular search window ($E_{LeakRecSW}$), a set B of test video sequences which are different from the set A to avoid biasing towards the offline analysis), and the prediction direction (dir). Different on-chip memory sizes are evaluated in a loop in a decreasing order (lines 4-11). The candidate size $\boldsymbol{s}$ is evaluated for a miss-analysis by performing video encoding tests for set B of video sequences and the energy for misses ($E_{MissTotal}$) is estimated using Eq. (5.7) (line 5). Depending upon the duration of ME and DE, the leakage energy of the on-chip memory (E_{Leak}) for a given size is estimated using Eq. (5.6) (line 5). Afterwards, the energy profit (E_{Profit}) is computed in relation to the rectangular search window size considering the leakage-energy saving and miss-energy overhead (line 6). The memory size with the best energy profit ($\boldsymbol{s}_{\text{Best}}$) is selected and returned (lines 7-10, line 12).

After the size for a given prediction direction (S_{dir}) is obtained, for N_{dir} number of prediction directions, the total memory size is computed as $S_{Total} = \sum_{Ndir} S_{dir}$. As discussed in Section 5.4.3, the memory is partitioned into banks to provide MB parallel access for parallel SAD computation. The number of banks is computed (Eq. (5.8)) depending upon the given throughput constraints (as frame rate F_{Rate} in fps) and video resolution (WxH: width and height of the video in pixels).

$$N_{Banks} = \frac{1}{NB_{Line}} \times \left(\frac{f \times 10^6}{\left(W \times H/_{256}\right) \times F_{Rate} \times N^{SAD}_{Avg\_dir} \times N_{dir}} \right) \qquad (5.8)$$

$N^{SAD}_{Avg\_dir}$ is the average number of SAD per MB and it depends upon the search algorithm. The frequency of the ME/DE hardware is denoted as *f* in MHz. Afterwards, the size of a sector (S_{Sector} in bytes, Eq. (5.9)) is computed by considering the variations in the usage profile (see Figure 5.28) in order to increase the potential of power-gating for different MBs that exhibit diverse spatial and temporal properties. Total number of sectors for each direction (N^{dir}_{Sector}) is computed according to Eq. (5.10).

$$S_{Sector} = \lfloor (Usage_{Max} - Usage_{Min}) / Usage_{Std} \rfloor \tag{5.9}$$

$$N^{dir}_{Sector} = \lceil S_{dir} / (N_{Banks} \times S_{Sector}) \rceil \tag{5.10}$$

```
DetermineVideoMemorySize($E_{LeakRecSW}$, B, dir, $\mathcal{S}$)
BEGIN
$E_{BestProfit} \leftarrow 0$; $s_{Best} \leftarrow 0$;
For all $s \in \mathcal{S}$                    // evaluate sizes in a decreasing order
  ($E_{Leak}$, $E_{MissTotal}$) $\leftarrow$ performMEDE($s$, B, dir ); // see Eq.Error! Reference source not found. & Error! Reference source not found.
  $E_{Profit} = (E_{LeakRecSW} - E_{Leak}) - (E_{MissTotal})$;
  If ($E_{Profit} \geq E_{BestProfit}$) Then;
    $s_{Best} = s$;
    $E_{BestProfit} = E_{Profit}$;
  End If;
End For
return $s_{Best}$;
END
```

Figure 5.30: Pseudo-code of the algorithm for finding the memory size for a given prediction direction

As discussed in Section 5.4.1 and illustrated by Figure 5.28, not all MBs use the complete on-chip memory and despite of a reduced-size memory, major parts (in several cases more than 40%) of the memory may not be used (see usage variations in Figure 5.28). Furthermore, the memory usage in ME is much less than that in DE, see Figure 5.28. Therefore, our proposed power-management scheme performs power-gating to the unused sectors. The key challenge is to determine an appropriate sleep mode depending upon the predicted memory requirements considering the spatial and temporal properties of frames/MBs, thus raising the abstraction level of power-gating to the application-level.

5.4.4 Application-Aware Power Management

First, a prediction direction is obtained from the application level. Since the search window for each prediction direction is stored in distinct sectors, the sectors of the unused prediction directions are put in S_2 state-retentive mode (as the data will be required in the MB loop, different search predictions are processed sequentially for each MB). Afterwards, the application-aware power management is employed for each prediction direction independently.

The primary input for the application-aware power management scheme is an offline analysis of the memory requirements (like in Figure 5.35). From this analysis, three different memory requirement predictions are obtained by performing a PDF analysis over various test video sequences. First prediction is about the maximum memory requirement which is denoted as PM_3. Considering a Gaussian distribution, two further highly-probable memory requirement predictions (PM_1 and PM_2) are computed using Eq. (5.11) and Eq. (5.12), where the high-probability zones cover the area under the curve considering $PM_1=\mu+\sigma$ and $PM_2=\mu+2\sigma$. Here, μ denotes the average of the distribution and σ denotes the standard deviation. Figure 5.32 shows an abstract example for computing PM_1, PM_2, and PM_3. PM_1 covers 84% of the area under the curve, while PM_2 covers 97.5% of the area under the curve.

$$PM_1 \leftarrow F(\mu+\sigma;\, \mu,\, \sigma^2) - F(0;\, \mu,\, \sigma^2) \approx 0.84 \tag{5.11}$$

$$PM_2 \leftarrow F(\mu+2\sigma;\, \mu,\, \sigma^2) - F(0;\, \mu,\, \sigma^2) \approx 0.975 \tag{5.12}$$

These predicted memory requirements are then forwarded to the algorithm of the power management scheme (Figure 5.33) as a tuple: $\mathcal{MR}_{Offline} = \{PM_3, PM_2, PM_1\}$. Further inputs are: (a) prediction direction (dir), (b) camera view (v), (c) video frame (f), (d) total size of the on-chip memory (S_{Total}), and (e) size of a sector (S_{Sector}). The algorithm in Figure 5.33 performs the power management in five main phases, as explained in the following.

Phase 1 - Frame-Level Power-Management: the memory requirements for the current frame *f* in a view *v* ($MR_{Current}$) are predicted from the neighboring frames in temporal (left, right) and disparity (top, down) domains using a weighted prediction of their respective MR_n (as shown in Figure 5.33, line 5). Firstly, the neighboring frames are obtained (line 3). In case the information about the memory requirements of a certain neighboring frame is not available, its memory requirements are initialized with the offline memory requirements ($MR_{Offline}$), see line 4. Figure 5.34 presents the pseudo-code for frame-level memory requirement prediction. Each predicted memory requirement $\{PM_3, PM_2, PM_1\}$ is computed as the weighted average of the corresponding PM of the neighboring frames, using Eq. (5.13) (see line 5 in Figure 5.34). $d_{n,\forall n\in\{Left,Right,Top,down\}}$ denotes the temporal/disparity distance (in terms of number of frames between the current frame and the prediction frame), while α and β are given as the motion and disparity weighting factors, respectively.

Note, during the encoding of Intra-frames, the complete on-chip memory for ME/DE is kept is the S_3 mode, as no ME/DE is performed for Intra-frames.

$$\begin{aligned} MR_{Current} = [(&MR_{Left} * d_{Left} + MR_{Right} * d_{Right})* \alpha \\ &+ (MR_{top} * d_{Top} + MR_{Down} * d_{Down})* \beta] / 4 \end{aligned} \tag{5.13}$$

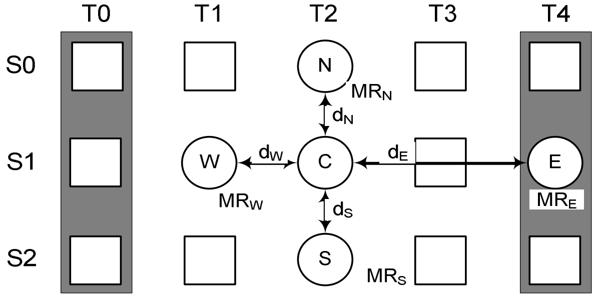


Figure 5.31: 2D-weighted prediction using the memory usage of the frames in the 3D-Neighborhood

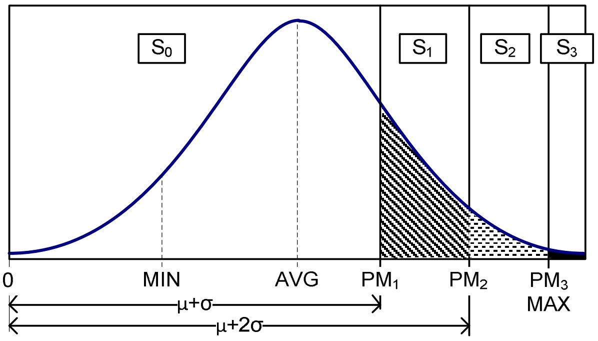


Figure 5.32: Statistical distribution of memory requirements (ME, DE)

```
1.  ApplicationAwarePowerManager(dir, v, f, $S_{Total}$, $S_{Sector}$, $\mathcal{MR}_{Offline}$)
2.  BEGIN
3.  list<NeighboringFrames> $\mathcal{N}$ ← getNeighboringFrames (dir, v, f );
4.  $\forall n \in \mathcal{N}$   $\mathcal{MR}_n \leftarrow$ ( n is Available )? getMemReq(n) : $MR_{Offline}$;
5.  $\mathcal{MR}_{Current}$ ← frameMemReq($\mathcal{MR}_{Left}$, $\mathcal{MR}_{Right}$, $\mathcal{MR}_{Top}$, $\mathcal{MR}_{Down}$); // Figure 5.34:
6.  list<MBGroups> $\mathcal{G}$ ← getMBGroups ($f$); // combine MBs in Groups
7.  For all $g \in \mathcal{G}$
8.    $\mathcal{MR}_{Group}$ ← reAdjustMemReq($g$, $\mathcal{MR}_{Current}$, $E_{MissGroup}$); // see Figure 5.35:
9.    list<Sectors> $\mathcal{PS}$ ← setSleepModes($S_{Total}$, $S_{Sector}$, $\mathcal{MR}_{Group}$);    // Figure 5.36:
10.   For all mb $\in g$
11.     ($E_{MissGroup}$, $E_{LeakGroup}$, $memUsed_{MB}$) ← performSearch( ); // perform ME/DE search and log memory requirements of the
          current MB
12.     $\mathcal{MR}_{Current}$ ← mbLevelPowerGating($\mathcal{PS}$, $memUsed_{MB}$); // Figure 5.37:
13.   End For
14. End For
15. $\mathcal{MR}$ ← computeMemStatistics($PM_3$, $PM_2$, $PM_1$);
16. return $\mathcal{MR}$;
17. END
```

Figure 5.33: Pseudo-code of the Application-Aware Power Manager

*Grouping of MBs***:** Since different MBs in a frame exhibit different spatial and temporal properties, not all MBs of a frame use same amount of memory for ME and DE. Therefore, the frame-level memory requirement prediction is adapted for different MBs in order to determine the sleep mode. Since state transitions (especially from S_3 to S_0) incur a wakeup overhead (in terms of energy and latency), consecutive MBs (sharing the same spatial and temporal video properties) are grouped together (using Eq. (5.14)) in order to increase the sleeping duration (see line 6 in Figure 5.33). Figure 5.35 shows the PDF of memory requirements for two different groups of MBs, where Group-I contains the homogeneous MBs with slow motion/disparity and the Group-II contains highly-textured MBs with medium-fast motion/disparity. It is noteworthy that the distribution of MBs in the Group-I is more centered compared to the Group-II in case of ME. Therefore, the frame-level prediction is readjusted considering the MB group using Eq. (5.16) (see line 8 in Figure 5.33), where ξi is given as the difference between the average textures of the complete video frame and the MB Group. Where, TH_{SAD} and TH_{Var} are computed using a statistical analysis over various test video sequences and are derived as Eq. (5.15) and Eq. (5.16).

$$\begin{cases} if\,(\,SAD_{MB} > TH_{SAD}\ \ \&\ \ Var_{MB} < TH_{Var}\ )\ \ Group = I \\ Else,\ \ \ Group = II \end{cases} \tag{5.14}$$

$$TH_{Var} = \mu_{Var} + 1.5 * \sigma_{Var}\,; TH_{SAD} = \mu_{SAD} + 1.5 * \sigma_{SAD} \tag{5.15}$$

$$\forall i \in [1...3] \quad PM_{i\text{-}Group} = \xi_i * PM_{i\text{-}Current} \tag{5.16}$$

```
frameMemReq($\mathcal{MR}_{Left}$, $\mathcal{MR}_{Right}$, $\mathcal{MR}_{Top}$, $\mathcal{MR}_{Down}$)
$\forall n \in$ {Left,Right,Top,down}
  {$PM_3$, $PM_2$, $PM_1$}$_n$ ← getMemReqSteps($\mathcal{MR}_n$); // see // see Figure 5.32:
$\forall i \in$ [1…3]          // compute the weighted average using Eq. 3
  $PM_{i\text{-Current}}$ ← weightedAvg($PM_{i\text{-Left}}$, $PM_{i\text{-Right}}$, $PM_{i\text{-Top}}$, $PM_{i\text{-Down}}$, $\alpha$, $\beta$);
$\mathcal{MR}_{Current}$ ← {$PM_3$, $PM_2$, $PM_1$}$_{Current}$;
return $\mathcal{MR}_{Current}$;
```

Figure 5.34: Pseudo-code of Frame-Level Memory Requirement Prediction

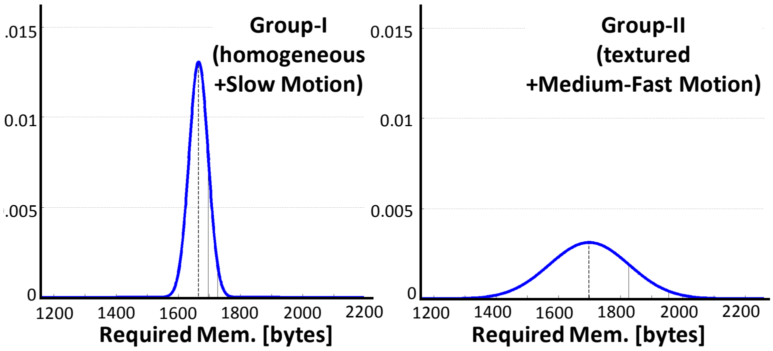


Figure 5.35: Statistical distribution of memory requirements for homogenous and textured MBs for ME and DE

Phase 3 - Group-Level Power-Management: Pseudo-code in Figure 5.36 provides the flow for making power-gating decisions. First, the group-level memory requirement prediction in terms of {PM_3, PM_2, PM_1} is obtained. Afterwards, different sets of sectors candidate for power gating are obtained, (as denoted by the 3 different types of filled area under the curve in Figure 5.32). The memory exciding the maximum requirements (M_3) is gated in S_3 mode, as it is highly improbably to be used by the MB-group. The other two sets of sectors – M_2 and M_1 – are candidates for being gated in S_2 and S_1 state-retentive modes, as they store the data which might be used later by other MBs of the group. This leads to a reduced wakeup overhead and reduced leakage savings compared to S_3. Since wakeup incurs an energy overhead, our scheme predicts the sleep duration which is required to amortize the wakeup overhead as a function of number of MBs in the group, see Eq. (5.17). Due to the non-retentive nature of S_3 mode, there is a probability of memory misses. Therefore, in addition to E_{wakeup}, $E_{MissGroup}$ is also added for evaluating the sleep decision of the S_3 mode. The set of sectors in different power modes is saved and returned.

$$N_{Group} > \begin{cases} E_{wakeup} \,/\, E_{LeakGroup} & If \quad S_1 \;\; or \;\; S_2 \\ (\,E_{wakeup} + E_{MissGroup}\,) \,/\, E_{LeakGroup} & Else \end{cases} \tag{5.17}$$

After an MB group is encoded, the energy of misses ($E_{MissGroup}$) along with the wakeup energy overhead ($E_{wakeupS3\text{->}S2}$) and leakage savings ($E_{LeakGroup}$) are used to predict the number of sectors that should be moved from the sleep mode S_3 (state non-retentive) to S_2 (state-retentive), as $N_{PM3} = E_{MissGroup} \,/\, (\,E_{LeakGroup} \times E_{wakeupS3 \to S2}\,)$.

Phase 4 - MB-Level Power Management: Afterwards, for all MBs in an MB-Group, the ME/DE search is performed and $E_{MissGroup}$, $E_{LeakGroup}$, $memUsed_{MB}$ are obtained (see line 11 in Figure 5.33). Then, the number of sectors in the state-retentive sleep modes (S_0, S_1, and S_2) for the upcoming MB is re-adjusted depending upon the actually used memory of the currently-encoded MB (see 12). Figure 5.37 illustrates the procedure for readjusting the sleep modes for the upcoming MB in an MB-Group. Firstly, the difference between the used memory and predicted memory is computed in terms of number of sectors (line 4). If the difference is zero, no update in the sleep modes is performed (line 5). If the difference is positive, the used memory is less than the predicted memory in mode S_0 (i.e., powered-on) and additional sectors are put into the state-retentive sleep mode S_1 (lines 7). Otherwise, more sectors are powered-on in other state-retentive modes (lines 8-15).

```
1.  setSleepModes(S, $S_{Sector}$, $\mathcal{MR}$)
2.  BEGIN
3.     {$PM_3$, $PM_2$, $PM_1$} ← getMemReqSteps($\mathcal{MR}$);// see Figure 5.32:
4.     $M_3=\left\lfloor \frac{S-PM_3}{S_{Sector}} \right\rfloor$;  $M_2=\left\lfloor \frac{PM_3-PM_2}{S_{Sector}} \right\rfloor$;  $M_1=\left\lfloor \frac{PM_2-PM_1}{S_{Sector}} \right\rfloor$;
5.     PowerGate($M_3$, $S_3$);   PowerGate($M_2$, $S_2$);      PowerGate($M_1$, $S_1$); // using the cost function Eq.
           Error! Reference source not found.
6.     $M_0$; = (S/$S_{Sector}$) – ($M_3$+$M_2$+$M_1$);
7.     SwitchOn($M_0$, $S_0$);
8.     list<Sectors> $\mathcal{PS}$ ← {$M_0$, $M_1$, $M_2$, $M_3$};
9.     return $\mathcal{PS}$;
10. END
```

Figure 5.36: Pseudo-code for determining the number of sectors and their corresponding sleep modes

```
1.  mbLevelPowerGating($\mathcal{PS}$, $memReq_{MB}$)
2.  {$M_0$, $M_1$, $M_2$, $M_3$} ←$\mathcal{PS}$;
3.  $M=\left\lfloor \frac{memReq_{MB}}{S_{Sector}} \right\rfloor$;
4.  Δmem = $M_0$ – M;
5.  If (Δmem == 0) Then          return $\mathcal{PS}$;
6.  If (Δmem > 0) Then           // Put more sectors in $S_1$ gating mode
7.     $M_0'$ = $M_0$ – Δmem;     $M_1'$ = $M_1$ + Δmem;      $M_2'$ = $M_2$;
8.  Else
9.     $M_0'$ = $M_0$ – Δmem;             // switch ON more sectors
10.    If (|Δmem| ≥ $M_1$) Then     // re-adjust $S_1$-gated & $S_2$-gated sectors
11.       $M_1'$ = $M_2$ + $M_1$ – |Δmem|;               $M_2'$ = 0;
12.    Else
13.       $M_1'$ = $M_1$ – Δmem;                     $M_2'$ = $M_2$;
14.    EndIf
15. EndIf
16. $\mathcal{PS}$ ← {$M_0'$, $M_1'$, $M_2'$, $M_3$};
17. return $\mathcal{PS}$;
```

Figure 5.37: Pseudo-code of MB-Level Power-Gating

Phase 5 - Re-compute Statistics: After the frame ME/DE is completed, the probabilistic analysis (like in Figure 5.32 and Figure 5.35) is performed to obtain the $\mathcal{MR}$, which is used by the subsequent frames.

5.4.5 Hardware Architecture

Our on-chip memory with application-aware power management is integrated in the ME/DE hardware architecture presented in Section 5.3 that features an array of 64 4x4-sample SAD operators and SAD trees to provide high throughput and a TZ Search controller. All components, including the memory size are designed to support real-time encoding for up to 4-views HD1080p performing ME/DE for search ranges up to [±96,±96] pixels. The detailed results are explained here onwards while the comparison to the state-of-the-art is discussed in Chapter 6.

5.4.6 ME/DE Architecture with Adaptive Power Management Evaluation

The leakage reduction provided by our application-aware power management is presented in Figure 5.38. It shows the leakage reduction normalized to our architecture with no power management (*NC*). When the application-aware power-gating featuring only frame-level (*FC*) power management is integrated with our memory, the leakage energy reduction reaches more than 50%. The fine-grained power-management at the MB level (FMBC) provides further 5% leakage reduction, altogether providing up to 55% leakage reduction.

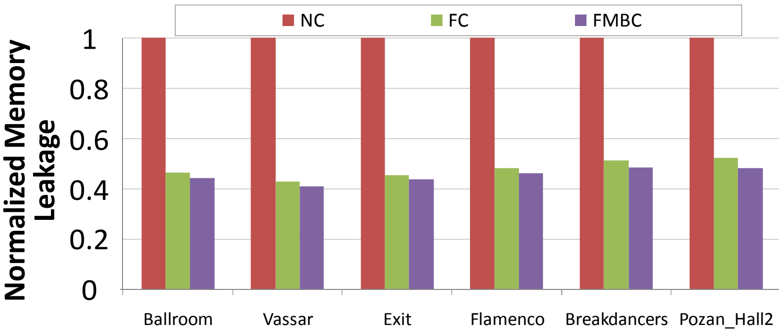


Figure 5.38: Comparing the leakage savings

Figure 5.39 presents the detailed analysis of the memory usage and the selected sleep modes and the corresponding energy savings for a series of MBs in *Flamenco2* sequence. Figure 5.39a shows different memory usages in ME/DE for different video sequence in terms of blocks (16x16 pixels). The power states of different blocks are shown by different colors. It is worthy to note that the sleep mode S1 is used too seldom. It is due to the fact that our scheme quickly transited between S_1 and S_2, as the difference in the wakeup overhead for S_1 and S_2 is insignificant. The decision of state S_3 is primarily at the frame-level as it is state non-retentive, which is visible by the transition in Figure 5.39b. At the MB and MB-Group levels, our scheme tends to choose S_2 mode due to its low wakeup overhead. The selection of S1 allows fine-grained power savings and accommodates sudden variations. Figure 5.39c shows the corresponding energy savings along the time. It is worthy to note that the variations in the energy savings are very frequent, it is due to the fact, that our scheme adapts very quickly to accommodate sudden variations in the memory requirements, thus frequently transiting between $S_0 \rightarrow S_1 \rightarrow S_2$. An interesting observation in Figure 5.39c is that the variations between S_1-S_2 do not touch the sectors gated in S_3 mode. This shows that the frame-level prediction of the maximum requirements is very accurate and the probability of powering-on the S_3 gated sectors is significantly low. As result, no data re-fetching is required.

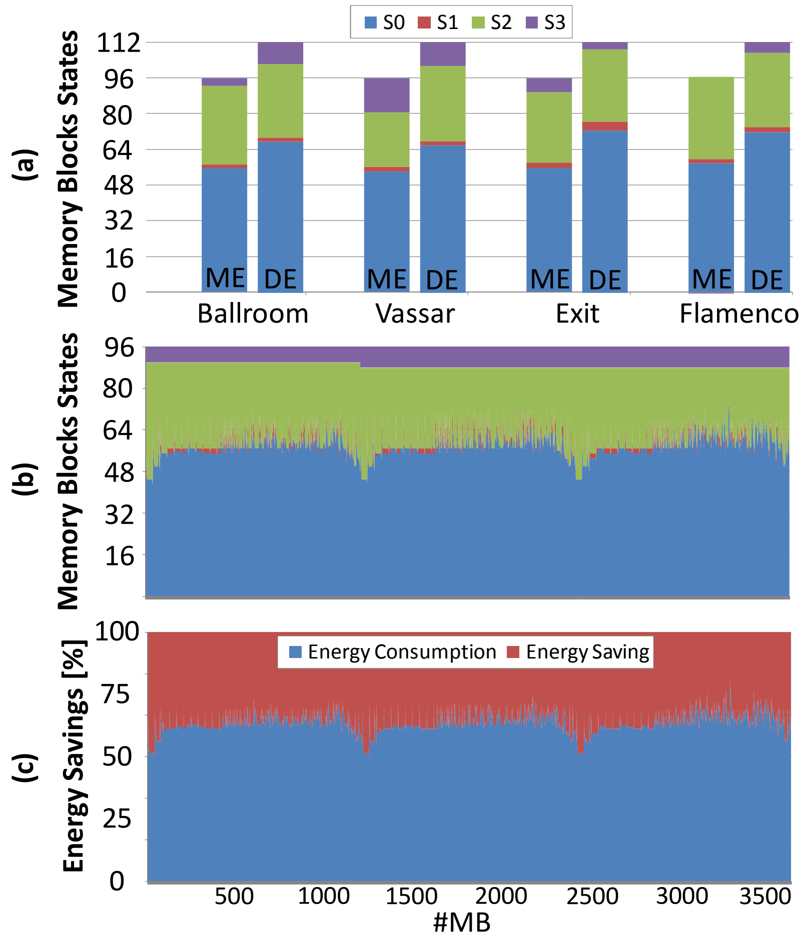


Figure 5.39: Detailed analysis of the memory usage and sleep modes

Figure 5.40 illustrates the comparison of our prediction accuracy with the actual memory usage and two history-based predictors. Observe that, in case of sudden variations, our application-based prediction follows the exact usage much accurately compared to the history-based predictors. This improved accuracy leads to the significant energy savings shown in Figure 5.38.

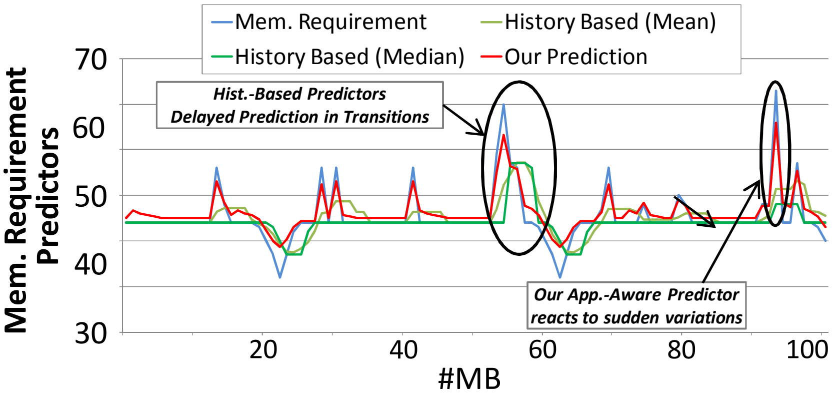


Figure 5.40: Comparing the accuracy of different predictors compared to the used memory (in case of a rectangular search window)

Table 5.5 presents the synthesis details for our ME/DE hardware architecture featuring application-aware dynamic power management using 65nm technology node. This implementation runs at 300MHz and provide the required throughput for real-time encoding 4-views HD1080p while consuming 57mW. This architecture requires 102k gates and 832Kbits of on-chip video memory. The comparison of these implementation results to the state-of-the-art is discussed in Section 6.2.

Table 5.5: Comparison the Hardware Results of the Fast ME/DE Architecture with Our On-Chip Memory

	ME/DE Hardware of with our On-Chip video Memory
Technology	ST 65nm LP 7 metal layer
Gate Count	102k
SRAM	832 Kbits
Max. Frequency	300 MHz
Power	57mW, 1.0v
Proc. Capability	4-views HD1080p

5.5 Summary of Energy-Efficient Algorithms for Multiview Video Coding

Three architectural solution to enable real-time ME/DE are presented along this chapter. Initially, the architectural template and the basic hardware building blocks are described in Section 5.1. Based on this structure a multi-level pipelined architecture implementing the Fast ME/DE algorithm is described in details along Section 5.2.

Targeting the reduction of the energy consumption related to the external memory accesses and on-chip video memory leakage, an architecture featuring the Dynamic Search Window Formation strategy is proposed in Section 5.3. This solution observes the search patterns of the neighboring MBs in order to anticipate the data required for the current MB. It allows accurate external memory data prefetching while reducing the on-chip memory size by avoiding the entire search window storage.

In Section 5.4 an Application-aware Dynamic Power Management algorithm is integrated to the ME/DE architecture. Assuming an on-chip memory with multiple power states and sector-level power gating, the DPM predicts the memory usage and power gate the memory sectors accordingly. By doing so, this architecture is able to significantly reduce the overall energy consumption through minimizing on-chip memory leakage.

6 RESULTS AND COMPARISON

In this chapter the overall results of this work and the comparison with the latest state-of-the-art approaches are presented. Before moving to the actual comparison, a description of the experimental setup is presented discussing the fairness of comparison in relation to the related works. The benchmark video properties, common test conditions, simulation environment, and synthesis tool chain are also introduced in this chapter. The results for energy-efficient algorithms are discussed in terms of complexity reduction while considering the coding efficiency and video quality in relation to state-of-the-art and optimal solutions. The video quality control algorithm based on rate control is compared to other rate control techniques described in current literature. Energy-efficient architectures are evaluated against the latest hardware solutions for ME/DE on MVC with emphasis on the overall energy consumption for both, memory access and processing datapath. Additionally, throughput and IC footprint area are discussed.

6.1 Experimental Setup

In this section are described the simulation, design and synthesis environment employed during the development of this work. Afterwards is presented a discussion on the test conditions and benchmark video sequences followed by the fairness of comparison with the state-of-the-art approaches. The hardware design method and synthesis tool chain is also presented in this section.

6.1.1 Software Simulation Environment

Each algorithm proposed along this thesis was implemented and evaluated using the reference software platform provided by the Joint Video Team (JVT, 2009), the Joint Model for MVC, also known as JMVC. The JMVC is provided in order to prove the concepts behind the MVC standard and facilitate the experimentation and integration of new tools to the MVC.

Initially, implementations were described on top of the JMVC 6.0, the latest version available by the time this work was started. In face of limitations related to the simulation of HD1080p sequences (note the use of these sequences were normalized in March 2011 (ISO/IEC, 2011) after this work was started), our algorithms were migrated to a more recent version, the JMVC 8.5, in order to extend our experimental results. Details on the JMVC software structure and the implemented modifications are detailed in Appendix A.

In Table 6.1 is presented a summary of the video encoder settings and parameters most commonly used for experimentation along this thesis. Note that some settings may vary depending on the experiments nature. These changes, however, are mentioned along the results discussion. Table 6.2 describes the computational processing resources used for simulation.

Table 6.1: Video Encoder Settings

Parameter	Setting
Entropy Encoder	CABAC
FRExt	Yes
QP (experiments w/o rate control)	22, 27, 32, 37, 42
Bitrate (experiments w/ rate control)	256, 392, 512, 768, 1024, 2048, 4096
GOP Size	8
Anchor Period	8
Temporal Coding Structure	IBP (Hierarchical B Prediction)
#Views	4/8
View Coding Structure	IBP (0-2-1-3 or 0-2-1-4-3-6-5-7)
Number of Reference Frames	Up to 4 (one per temporal/view direction)
Inter-frame/ Inter-view Prediction Pictures First	Inter-frame
B Pictures Reference	Yes
Search Mode	TZ Search
Search Reange	Up to [±96, ±96]
Distortion Metric	SAD
Weighted Prediction	No
Deblocking Filter	Yes

Table 6.2: Simulation infrastructure

Desktop for Simulation	
Processor	Intel Core 2 Duo-6600@2.4GHz
Main Memory	3.25GB DDR2
Operational System	Windows XP SP2
Mobile Device for Battery-Aware Experiments	
Device	HP Pavillion DV6000 Series
Processor	Intel Core-2 Duo T5500 @1.66GHz
Main Memory	2GB DDR2
Operational System	Windows XP SP2
Battery	6-cell lithium ion 4400mAh 10.8V

6.1.2 Benchmark Video Sequences

To allow other researchers to easily compare their results against ours and, consequently, make our results more meaningful to the current literature, the benchmark video sequences used in our experimental section were derived from the common test conditions recommendations provided by JVT (SU, VETRO e SMOLIC, 2006) and ISO/IEC (ISO/IEC, 2011). In Table 6.3 are presented the video sequence names along with the number of views, cameras organization and resolution. The considered video resolutions are VGA (640x480), XGA (1024x768) and HD1080p (1920x1088 – typically cropped to 1920x1080) featuring distinct number of cameras, camera spacing and organization. Although some sequences have up to 100 cameras, our experiments are constrained to four or eight depending on the algorithm under evaluation. Please consider that the main goal of this thesis is on MVC encoding for mobile devices that are not expected to feature more than 8 cameras. Nevertheless, the concepts behind the energy reduction algorithms proposed in this thesis are scalable to increased number of views for applications such 3DTV and FTV.

Table 6.3: Benchmark Video Sequences

Sequence	Resolution	# Views	Cameras Organization
Ballroom	640x480	8	20cm spacing; 1D/parallel
Exit	640x480	8	20cm spacing; 1D/parallel
Vassar	640x480	8	20cm spacing; 1D/parallel
Race1	640x480	8	20cm spacing; 1D/parallel
Rena	640x480	100	5cm spacing; 1D/parallel
Akko&Kayo	640x480	100	5cm horizontal and 20 cm vertical spacing; 2D array
Flamenco2	640x480	5	20cm spacing; 2D/parallel (Cross)
Ballet	1024x768	8	
Breakdancers	1024x768	8	20cm spacing; 1D/arc
Uli	1024x768	8	20cm spacing; 1D/parallel convergent
GT Fly	1920x1088	9	Computer generated
Poznan Hall2	1920x1088	9	13.75 cm spacing; 1D/parallel
Poznan Street	1920x1088	9	13.75 cm spacing; 1D/parallel
Undo Dancer	1920x1088	9	Computer generated

To support the reader that is not familiar with these video sequences, is provided, in Figure 6.1, the spatial, temporal and, disparity indexes (SI, TI and DI) for each video sequence referred in Table 6.3. The higher the index the more complex the sequence is in that specific dimension. The goal is to better understand why some sequences perform better than others under certain coding conditions and/or algorithms. The spatial and temporal indexes were proposed in (ITU-T, 1999) and have been used to classify the benchmark video sequences (NACCARI, BRITES, *et al.*, 2011) used to test the next video coding standard, the High Efficiency Video Coding (HEVC/H.265). Eq. (6.1) and Eq. (6.2) give the equations that define SI and TI extended for multiview videos where *ρ(i,j)* represents the pixel luminance value in coordinates *i* and *j*, *Sobel* denotes the Sobel filter operator and *n* is the frame temporal index. Additionally, in order to further adapt to multiview special needs we define the disparity index based on the same metric used for TI according Eq. (6.3) where *v* is the view index.

$$SI = max_{view}\{max_{time}\{\ std_{space}[\ Sobel(\ \rho(\ i,j\ ))]\}\} \qquad (6.1)$$

$$TI = max_{view}\{max_{time}\{\ std_{space}[\ \rho_n(\ i,j\ ) - \rho_{n-1}(\ i,j\ )]\}\} \qquad (6.2)$$

$$DI = max_{view}\{max_{time}\{\ std_{space}[\ \rho_v(\ i,j\ ) - \rho_{v-1}(\ i,j\ )]\}\} \qquad (6.3)$$

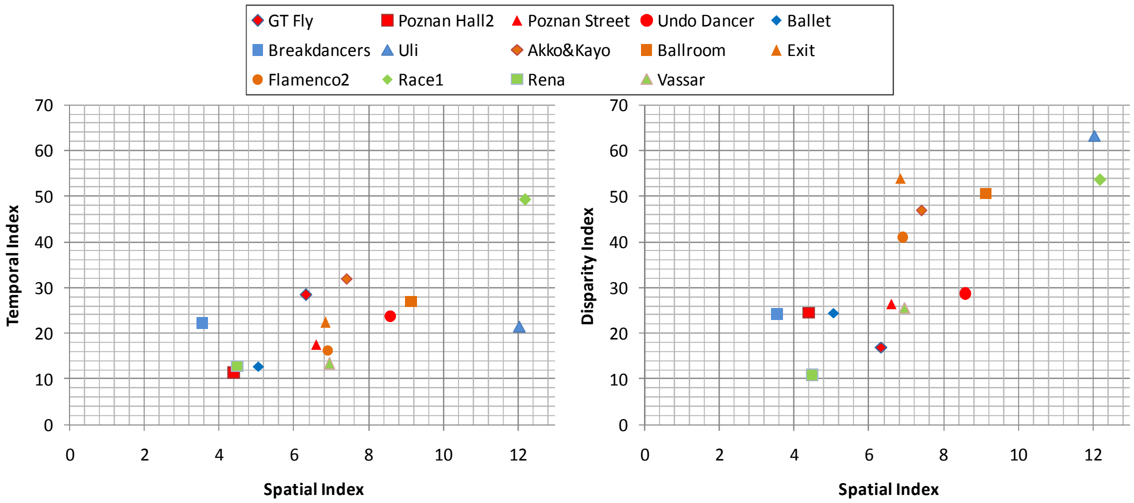


Figure 6.1: Spatial-Temporal-Disparity indexes for the benchmark multiview video sequences

6.1.3 Fairness of Comparison

Although the experimental results were generated using standard benchmark video sequences and standard coding settings, it is frequently not possible to directly compare our algorithms with the results provided by published the related works. For this reason, all state-of-the-art competitors were implemented using our infrastructure based on the information available in the referred literature. This approach requires significant implementation effort overhead, however, it ensures that all algorithms are tested under the same conditions and guarantees the fairness of comparison between all proposed solutions. The simulation infrastructure and modifications applied to the JMVC are presented in Appendix A.

6.1.4 Hardware Description and ASIC Synthesis

The architectural contribution proposed along this thesis includes complete RTL (Register Transfer Level) description, functional verification, and logical and physical synthesis. The hardware was described using VHDL hardware description language followed by functional verification with Mentor Graphics ModelSim (MENTOR GRAPHICS, 2012) using real video test vectors. The standard-cell ASIC synthesis for 65-nm technologies was performed using the Cadence ASIC Tool chain (CADENCE DESIGN SYSTEMS, INC., 2012). Two distinct processes and standard-cell libraries were considered in our hardware results, the *IBM 65nm LPe LowK* (SYNOPSYS, INC., 2012) and *ST 65nm Low-Power* (CIRCUITS MULTI-PROJECTS, 2012). For preliminary results, FPGA synthesis targeting Xilinx FPGAs was performed using the Xilinx ISE tool (XILINX, INC., 2012).

As mentioned above, the presented architectures were completely designed, integrated, and tested. The only exception is the on-chip SRAM memories featuring multiple power states. As far as our memory libraries and memory compiler were not able to generate such memories, regular SRAM memories were instantiated instead for connectivity and area approximation. The SRAM energy numbers were extracted from the related works that describe, implement and characterize the multiple power states SRAM memories for 65-nm (FUKANO, KUSHIDA, *et al.*, 2008) (ZHANG, BHATTACHARYA, *et al.*, 2005)(SINGH, AGARWAL, *et al.*, 2007). With the energy numbers and ME/DE memory traces, a memory simulator was designed to provide the energy saving results.

6.2 Comparison with the State-of-the-Art

6.2.1 Energy-Efficient Algorithms

In this section is presented the comparison between the energy-efficient mode decision algorithms proposed in this thesis and the state-of-the-art for fast mode decision. The efficiency of the algorithms is measured in terms of time savings compared to the JMVC using RDO-MD. Also, the video quality (PSNR in dB) and bitrate (BR in # of bits) variations are presented using RD curves and the Bjøntegaard rate-distortion metric (TAN, SULLIVAN e WEDI, 2005).

6.2.1.1 Comparing Our Mode Decision Algorithms to the State-of-the-Art

Figure 6.2 presents the percentage time savings compared to RDO-MD for the early SKIP mode decision (Section 4.1.2), the two strengths of our multi-level fast mode decision (*Relax* and *Aggressive*, Section 4.1.3) and, two related works (HAN e LEE, 2008) and (SHEN, YAN, *et al.*, 2009). Each bar represents the average for all QPs for that specific video sequence and mode decision algorithm. Even our simplest solution, the early SKIP algorithm, is able to outperform (SHEN, YAN, *et al.*, 2009) for most of the cases. The work proposed in (HAN e LEE, 2008) provides time savings superior to the early SKIP but pays a price in terms of video quality, as will be discussed soon. The multi-level fast mode decision shows a superior performance compared to all competitors and provides up to 79% time reduction. Additionally, it provides two complexity reduction strengths that allow handling the energy saving vs. quality tradeoff according to the system state and video content. The multi-level mode decision outperforms the state-of-the-art for all cases while keeping the video quality losses within an acceptable range, as discussed below.

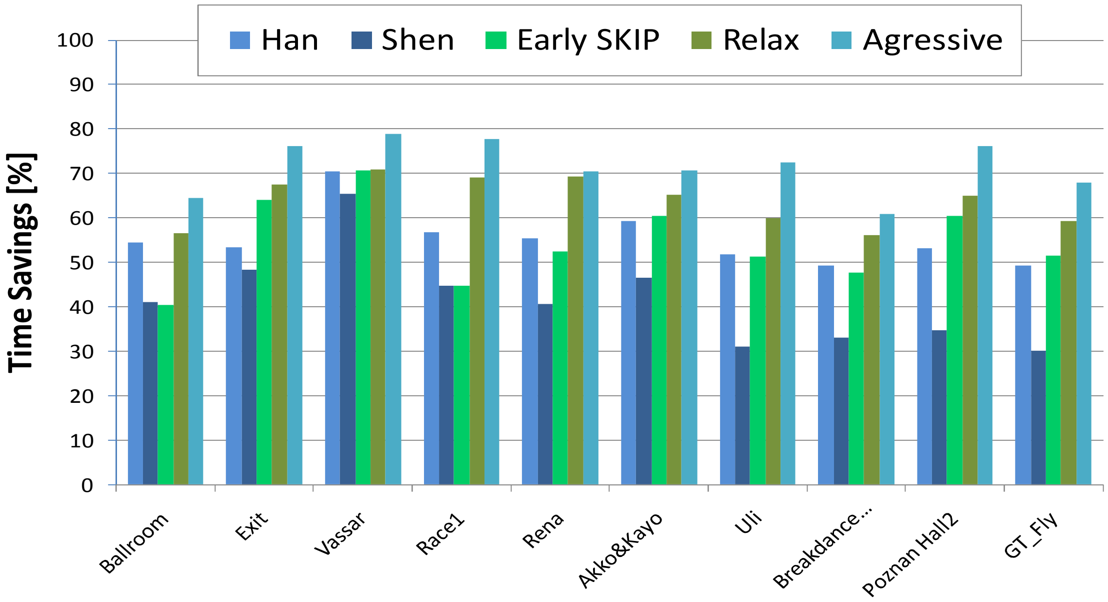


Figure 6.2: Time savings comparison with the state-of-the-art

The graph in Figure 6.3 brings the time savings information detailing its behavior for multiple QPs considering two video sequences, one VGA and one HD1080p. This plot shows that our fast MD algorithms are able to sustain the time savings for the whole QP range due to the QP-based thresholding employed. For instance, (SHEN, YAN, *et al.*, 2009) employs fixed threshold and suffers from reduced time savings specially for low QPs. At high QPs, the fixed thresholds tend to incur increased quality drop. To summarize the complexity reduction results, Figure 6.4 depicts the distribution of time savings provided by each competitor algorithm considering all video sequences and QPs tested. In summary, the algorithms proposed along this thesis provide averagely higher complexity reduction while sustaining significant complexity reduction for any encoding scenario. While (SHEN, YAN, *et al.*, 2009) shows scenarios with 10% reduction, the early SKIP prediction provides at least 38%. The multi-level fast mode decision ensures time savings between 55% and 90%.

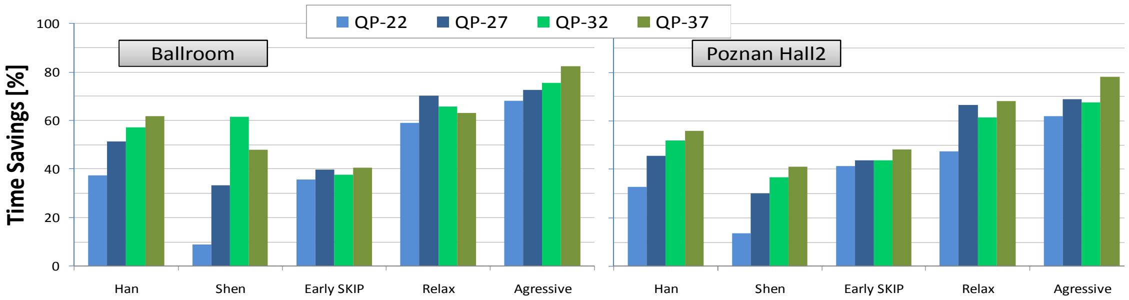


Figure 6.3: Time savings considering the multiple QPs

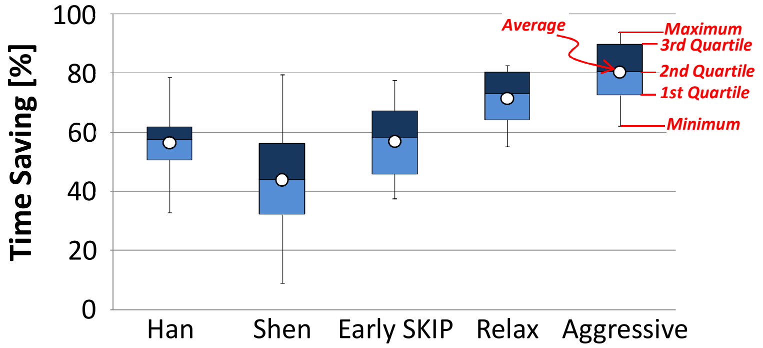


Figure 6.4: Time savings distribution summary

Beside of providing complexity reduction, fast mode decision algorithms must avoid significant video quality losses. In Figure 6.5 the rate-distortion curves show that for most of the tested video sequences there is a small displacement compared to the RDO-MD solution. The *Relax* level of our multi-level mode decision scheme provides RD results very close to the exhaustive RDO-MD for most of the cases. The *Aggressive* level incurs slightly worse RD results, especially for *Race1* and *Rena* sequences. Please note, our scheme with both *Relax* and *Aggressive* levels provides much higher complexity reduction compared to all schemes, as discussed earlier. The usage of the *Aggressive* level is recommended if high complexity reduction is desired (e.g. when the battery level of a mobile device is low). Under normal execution conditions, the *Relax* level is recommended as it provides superior complexity reduction compared to the state-of-the-art while keeping the RD performance close to the RDO-MD. In Table 6.4 is summarized the rate-distortion performance for the discussed mode decision algorithms. Averagely, the early SKIP and relax solutions present the best RD results. The *Aggressive* variant of the multi-level fast mode decision sacrifices RD performance, compared to other competitors, in order to provide the higher complexity reduction.

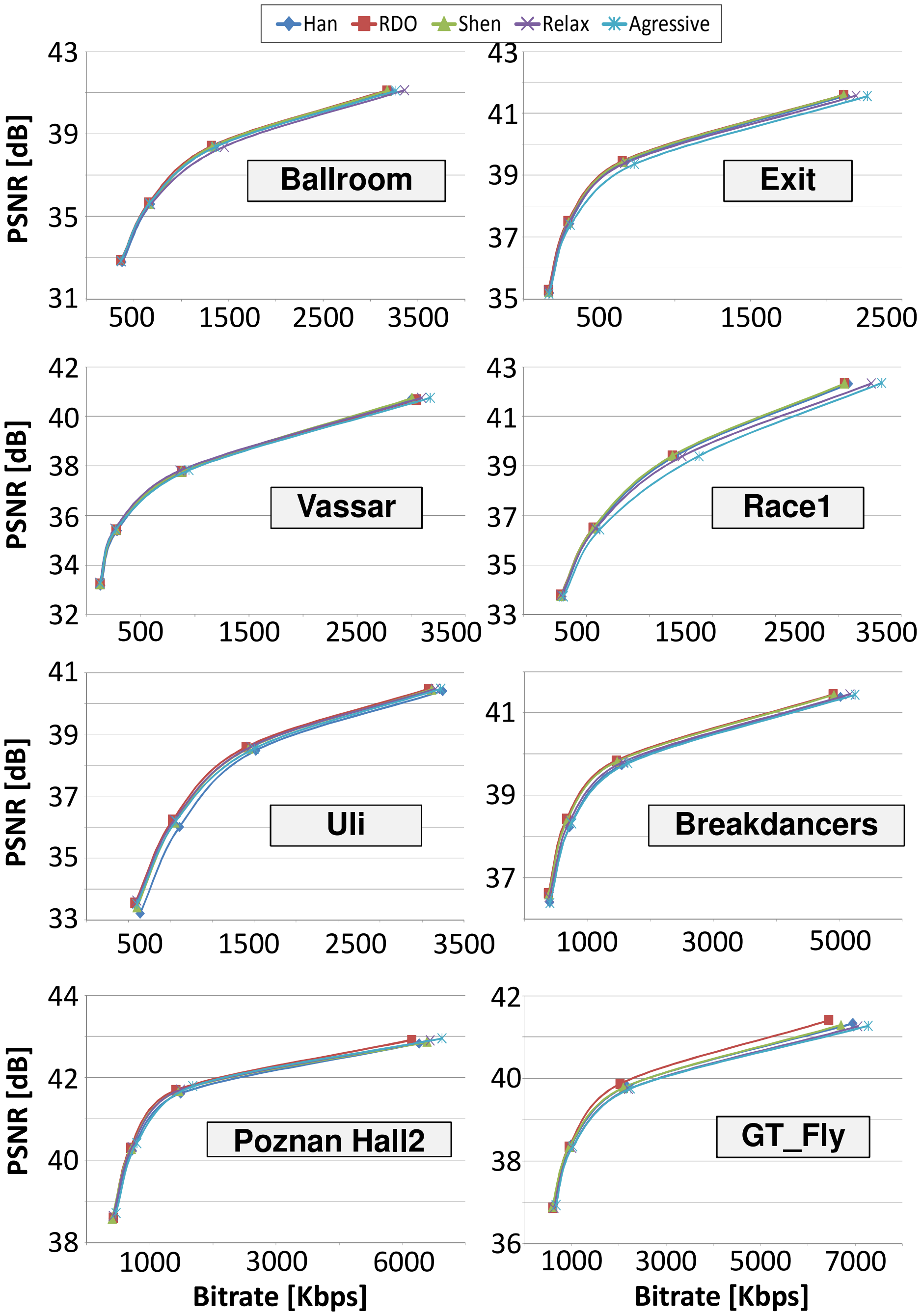


Figure 6.5: Rate-Distortion results for fast mode decision algorithms

Table 6.4: Bjøntegaard PSNR and BR for fast mode decision algorithm

Video Sequences	Han		Shen		Proposed Relax		Proposed Aggressive	
	BD-PSNR	BD-BR	BD-PSNR	BD-BR	BD-PSNR	BD-BR	BD-PSNR	BD-BR
Ballroom	-0.163	4.412	-0.054	1.458	-0.106	2.749	-0.272	7.221
Exit	-0.1234	5.278	-0.041	1.749	-0.097	3.960	-0.281	12.047
Vassar	-0.182	8.311	-0.122	5.582	-0.037	1.709	-0.172	8.189
Race1	-0.112	2.868	-0.024	0.600	-0.222	5.890	-0.514	14.019
Rena	-0.156	3.672	-0.022	0.514	-0.467	10.917	-1.031	25.585
Akko&Kayo	-0.298	6.444	-0.091	1.944	-0.278	5.852	-0.735	16.260

Breakdancers	-0.229	13.688	-0.039	2.301	-0.154	9.044	-0.268	15.314
Uli	-0.424	12.400	-0.149	4.234	-0.084	2.242	-0.202	5.521
Poznan_Hall2	-0.112	7.781	-0.027	3.137	-0.042	3.242	-0.140	8.780
GT_Fly	-0.134	7.273	-0.107	5.614	-0.232	12.886	-0.276	14.697
Average	-0.193	7.212	-0.067	2.713	-0.171	5.849	-0.389	12.763

6.2.1.2 Comparing the Energy-Aware Complexity Adaptation to the State-of-the-Art

The evaluation of the energy-aware complexity adaptation algorithm was done by experimentation on a battery-powered HP laptop (DV6000, Core-2 Duo). For accessing the battery level, we have used the *CallNtPowerInformation* windows API. In this experiment, the *Quality States* were forced to switch from *QS1* to *QS4* (simulating a battery discharge) and from *QS4* back to *QS1* (simulating battery charging). Figure 6.6 shows the frame-wise quality and time savings of our scheme encoding the *Ballroom* sequence. Two views are presented in Figure 6.6. Compared to the RDO-MD, the *Quality States QS1* and *QS2* incur a negligible quality loss while providing a TS of up to 75%. For *QS3* and *QS4* the TS go up to 79% and 88%, respectively. Due to the binocular suppression *QS3* maintains a negligible PSNR loss. The resulting quality for the resulting viewpoint (VP) is measured according to Eq. (6.4) (OZBEK, TEKALP e TUNALI, 2007).

$$PSNR^{VP}=(1-\alpha).PSNR^{HighQuality}+\alpha.PSNR^{LowQuality};\ \alpha=1/3 \qquad (6.4)$$

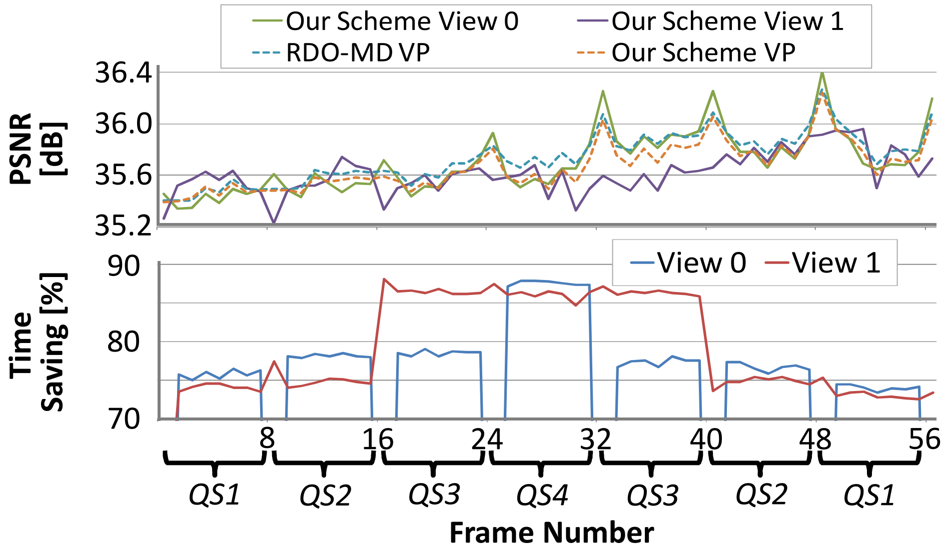


Figure 6.6: Complexity-adaptation for MVC for changing battery levels

The energy-aware complexity adaptation for MVC enables run-time tradeoff between complexity and video quality using different *Quality-Complexity Classes* (QCCs). Our scheme facilitates encoding of even and odd views using different QCCs (i.e., asymmetric view encoding) such that the overall perceived video quality is close to that of the high quality view. Our scheme is especially beneficial for next-generation battery-operated mobile devices with a support of 3D-multimedia.

6.2.1.3 Comparing the Fast Motion and Disparity Estimation to the State-of-the-Art

The comparison of our Fast ME/DE with the *TZ Search* algorithm and state-of-the-art complexity reduction schemes for ME/DE is presented in this section. Figure 6.7

shows the time savings of our Fast ME/DE algorithms for multiple video sequences and QPs for 4-view sequences. The *TZ Search* is used for comparison as it is 23x faster compared to the *Full Search*, while providing the similar rate-distortion (RD) results. Compared to the *TZ Search*, our fast ME/DE provides 83% execution time saving. In the best case, the execution time savings go up to 86%, which represents a significant computation reduction. These results are possible through drastic reduction in the number of SAD operations required, as shown in Figure 6.8. Compared to (LIN, LI, *et al.*, 2008) and (TSUNG, CHEN, *et al.*, 2009), the number of SAD operations is reduced in 99% and 94%, respectively. It also represents 86% complexity reduction compared to the original *TZ Search*.

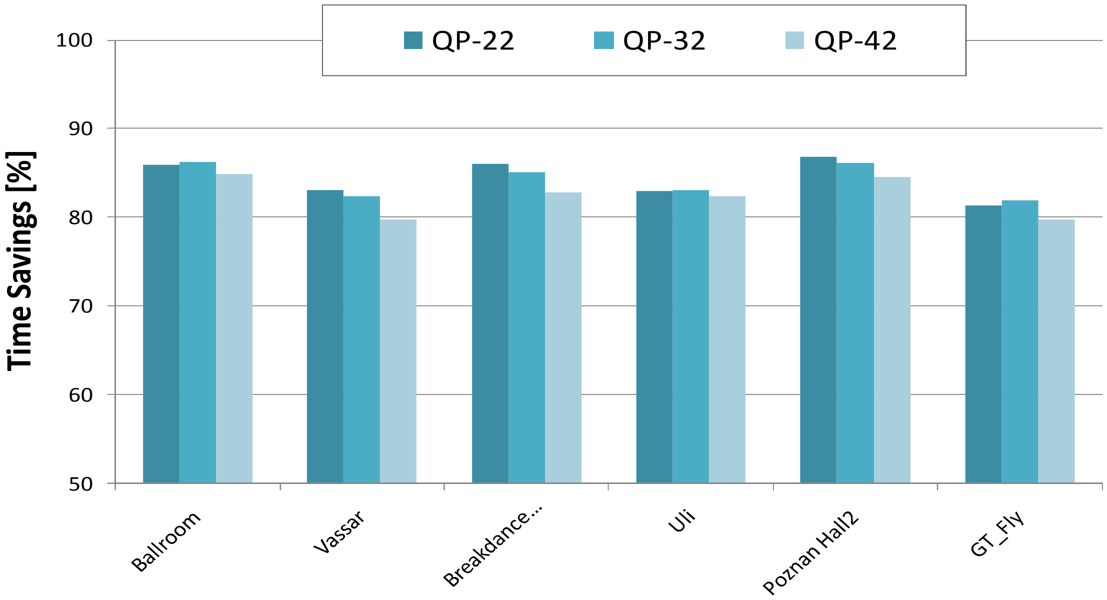


Figure 6.7: Complexity reduction for the Fast ME/DE

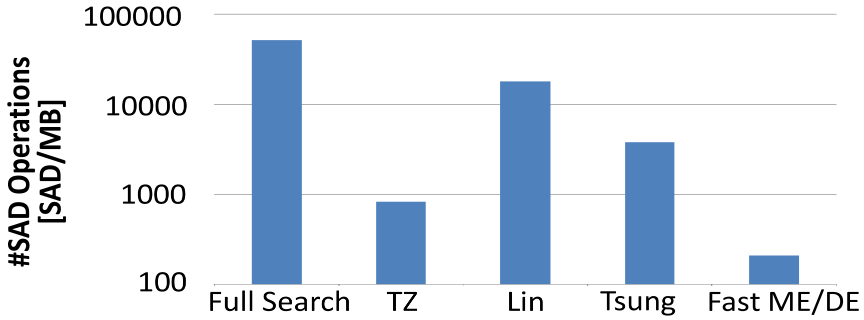


Figure 6.8: Average number of SAD operations

The Fast ME/DE algorithm was designed to avoid high quality drops and bitrate increases that surpass 10%, for this reason, it does not result in high rate-distortion losses. The RD curves presented in Figure 6.9 summarize the average 0.116dB quality reduction and 10.6% bitrate increase (see detailed table in Section 4.3.3) resulting from the aggressive complexity reduction provided by the proposed algorithm. Also, the Fast ME/DE demonstrate its robustness in terms of complexity reduction for all the tested video resolutions and QPs. This characteristic is desirable for real-time hardware architectures design.

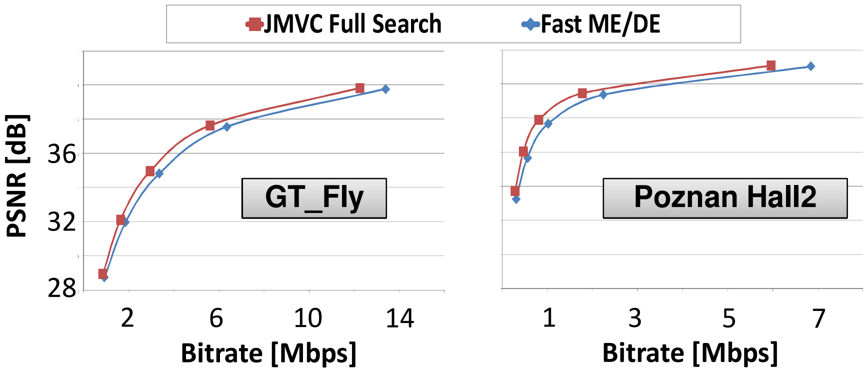


Figure 6.9: Fast ME/DE RD curves

6.2.2 Video Quality Control Algorithms

To deal with the quality losses posed by our fast algorithms, we propose, in Section 4.4, a complete rate control (RC) solution in order to efficiently manage the video quality vs. energy tradeoff. An efficient RC is supposed to sustain the bitrate as close as possible to the target bitrate (optimizing the bandwidth usage) while avoiding sudden bitrate variations. To measure the RC accuracy, that is, how close the actual generated bitrate (R_a) is in relation to the target bitrate (R_t), we use the Mean Bit Estimation Error (MBEE) (see Eq. (4.2)) metric. The averrage is calculated over all Basic Units (N_{BU}) along 8 views and 13 GOPs for each video sequence.

Figure 6.10 presents the accuracy in terms of MBEE (less is better) for our HRC compared to the state-of-the-art solutions (LI, PAN, *et al.*, 2003), (YAN, SHEN, *et al.*, 2009), (LEE e LAI, 2011), and our frame-level RC. On average, our Hierarchical Rate Control provides 0.95% MBEE, while raging from 0.7%-1.37%. The competitors (LI, PAN, *et al.*, 2003), (YAN, SHEN, *et al.*, 2009), (LEE e LAI, 2011), and the frame-level RC present, on average, 2.55%, 1.78%, 2.03% and 1.18%, respectively. The HRC reduces the state-of-the-art error on 0.83%, on average. The superior accuracy is a result of the ability to adapt the QP jointly at frame and BU levels while considering the 3D-Neighborhood correlation and the video content properties.

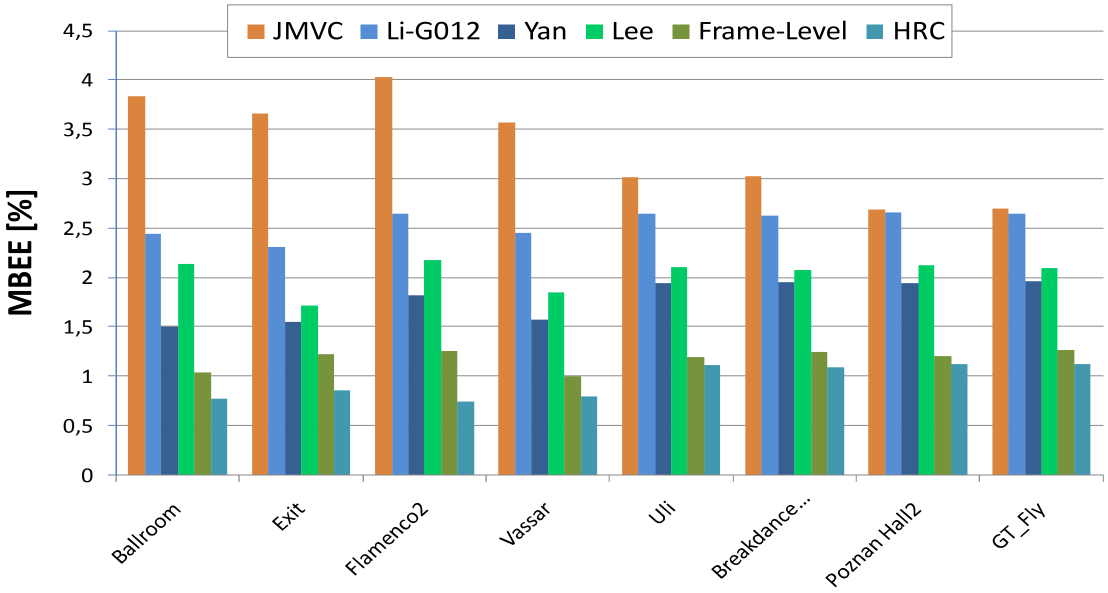


Figure 6.10: Bitrate prediction accuracy

In Figure 6.11 the long term behavior of distinct Rate Control schemes is presented in terms of accumulated bitrate. A more accurate RC maximizes the use of available bandwidth and, consequently, the accumulated bitrate tends to stay closer to the target bitrate line. After a few initial GOPs required for control stabilization, our HRC curve better fits to the target bitrate followed by our frame-level RC, as shown in Figure 6.11. JMVC without RC presents the worst bandwidth usage, as expected.

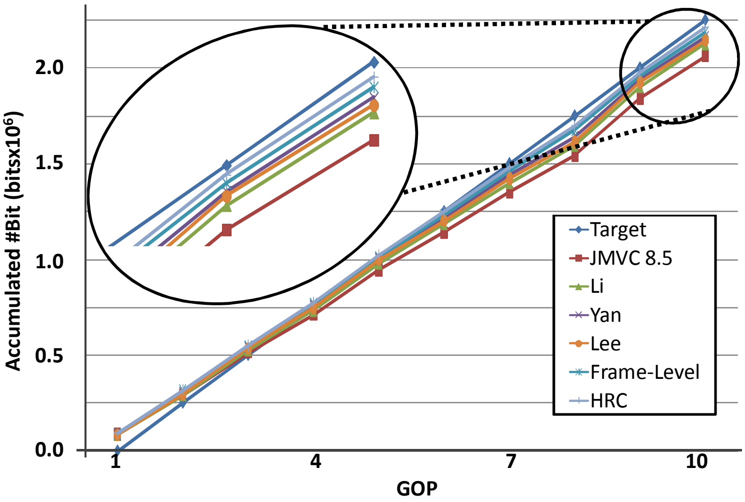


Figure 6.11: Accumulate bitrate along the time

Once the accuracy of our HRC is proven we present the rate-distortion (RD) results to show that overall video quality and quality smoothness are not compromised. Table 6.5 summarizes the objective rate-distortion in terms of BD-PSNR (Bjøntegaard Delta PSNR) and BD-BR (Bjøntegaard Delta Bitrate) in relation to JMVC without RC. The HRC provides 1.86dB BD-PSNR increase along with BD-BR reduction of 40.05%, on average. If compared to (LEE e LAI, 2011), that presents the best RD performance among the related works, the HRC delivers 0.06dB increased BD-PSNR and 3.18% reduced BD-BR. Remember, besides of superior RD performance, the HRC also outperforms (LEE e LAI, 2011) in terms of accuracy (1.08% MBEE).

Figure 6.12 shows the RD curves for different video sequences considering videos from distinct spatial, temporal and disparity indexes. The HRC shows its superiority in relation to the state-of-the-art for most of the RD curves. It is also important to highlight that HRC does not insert visual artifacts such as blurring and blocking noise. Moreover, our RC does not compromise the borders sharpness typically lost in case of bad QP selection.

Table 6.5: Bjøntegaard PSNR and BR for the HRC

JMVC vs.		VGA				XGA		HD1080p		AVG
		Ballroom	Exit	Flamenco2	Vassar	Bdancer	Uli	Poznan	GT_Fly	
Li	BD-PSNR	0.328	0.368	0.217	0.183	0.215	0.208	0.254	0.012	0.223
	BD-BR	-9.831	-10.348	-8.784	-6.116	-8.963	-9.805	-12.186	-6.711	-9.093
Yan	BD-PSNR	-0.090	0.073	0.114	0.051	-0.086	0.155	0.169	-0.118	0.034
	BD-BR	-4.156	-5.463	-3.346	-1.958	22.819	-5.293	-0.671	16.953	2.361
Lee	BD-PSNR	2.056	2.058	1.292	1.509	2.019	1.879	1.928	1.721	1.808
	BD-BR	-35.446	-43.167	-26.643	-33.474	-43.445	-39.110	-40.931	-38.134	-37.544
Frame-Level	BD-PSNR	0.939	1.089	0.880	0.596	0.881	0.670	0.750	0.614	0.802
	BD-BR	-22.241	-26.965	-22.989	-16.897	-22.818	-20.964	-17.184	-20.872	-21.366
HRC	BD-PSNR	1.585	2.375	2.103	1.176	2.060	1.870	2.086	2.056	1.914
	BD-BR	-31.588	-47.458	-38.199	-27.335	-46.112	-49.660	-48.766	-47.258	-42.047

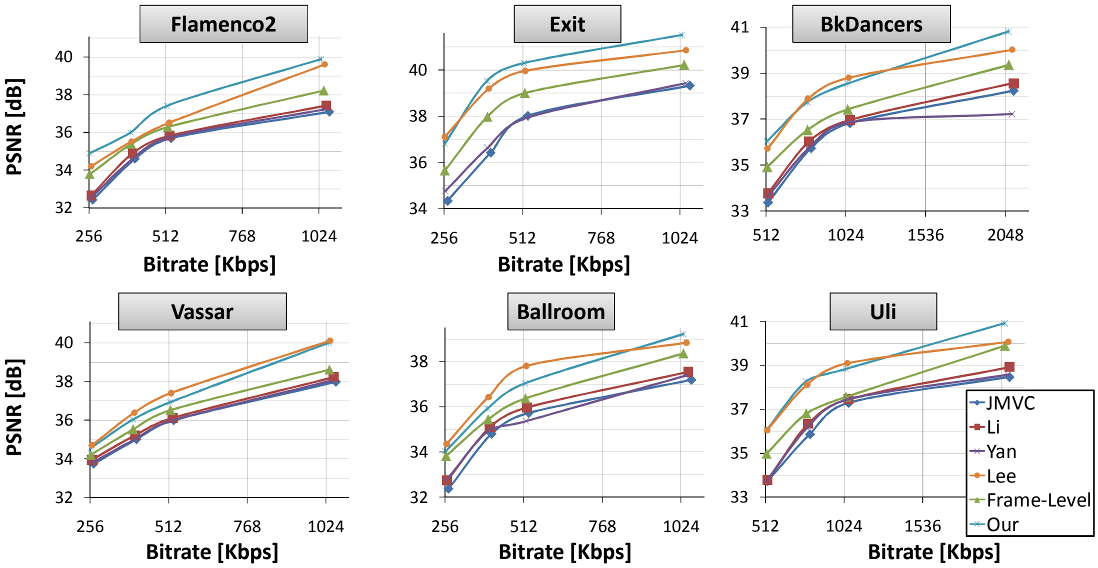


Figure 6.12: Rate-distortion results for the HRC

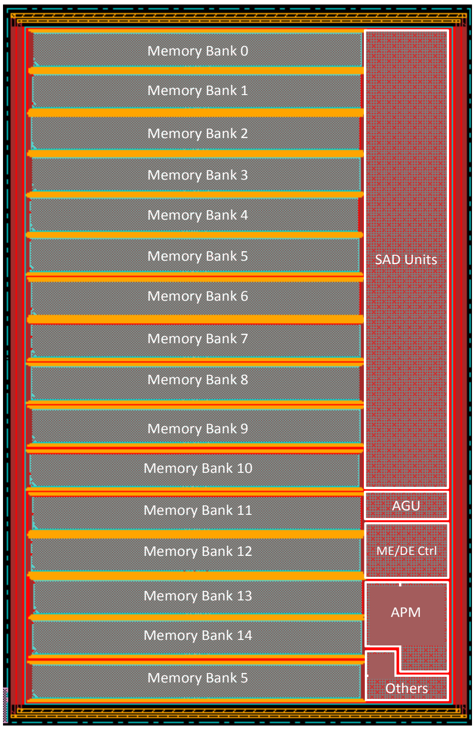
6.2.3 Energy-Efficient Hardware Architectures

Results and comparison to the state-of-the-art for the three proposed energy-efficient hardware architectures are presented in this section. Table 6.6 summarizes the hardware implementation results with details on gate count, size of on-chip memory, performance and power consumption (on-chip). For simplicity, along this section the proposed architectures are referred as: (A) Multi-Level Pipelined HW Architecture with Fast ME/DE (Section 5.2), (B) Motion and Disparity Estimation HW Architecture with Dynamic Search Window Formation (Section 5.3) and, (C) Motion and Disparity Estimation HW Architecture with Application-Aware Dynamic Power Management (Section 5.4). Figure 6.13 shows the physical layout of the ASIC implementing architecture (C). To the current stage, the IC was completely synthesized but not fabricated.

Compared to our architectures, the one presented in (CHANG, TSAI, *et al.*, 2010) requires more hardware resources while providing significantly low performance, attending only CIF resolution (352x258) requirements. Even assuming a frequency extrapolation, the performance provided by (CHANG, TSAI, *et al.*, 2010) is not comparable to the other discussed solutions. Comparing to (TSUNG, CHEN, *et al.*, 2009), our designs are able to provide real-time ME/DE for up to 4-views HD1080p videos compared to HD720p provided by (TSUNG, CHEN, *et al.*, 2009) at the same operation frequency. This represents a throughput increase (in terms of the processed MBs) of 2.26x obtained through Fast ME/DE and careful pipelining and scheduling in architecture. Additionally, architecture (A) reduces the gate count and power consumption compared to (TSUNG, CHEN, *et al.*, 2009). Architecture (A) requires 8% less gates compared to (TSUNG, CHEN, *et al.*, 2009). The power consumption (excluding the external memory accesses that are reduced by 65%) is also reduced by 69% (including the on-chip SRAM memory read/write power). The number of memory bits is increased once Level-A cache requires 2 Kbits, Level-C 131 Kbits and 3D-cache 82 Kbits. Additionally, the MV/DV memory used by our fast ME/DE algorithm requires 522 Kbits. In relation to (TSUNG, CHEN, *et al.*, 2009) the memory in (A) represents a 11.5x increase. It is also important to consider that our architecture is implemented in 65nm at 0.8v while (TSUNG, CHEN, *et al.*, 2009) uses a 90nm low power technology at 1.8v.

Further improvements were employed in architecture (B) in terms of control flow and memory design. A significant gate count reduction was possible by simplifying the two dataflow control units from architecture (A) – a single control unit remains in architecture (B) - and by merging the three cache memories in a single one. Architecture (B) reduces the gate count, number of memory bits and power consumption by 52%, 9% and 30%, respectively, compared to (A). It also increases the maximum search range from [±64,±64] to [±96,±96]. Compared to (TSUNG, CHEN, *et al.*, 2009) the area and power reductions are 66% and 72%, respectively, while providing higher throughput. This significant power reduction is mainly due to the employment of dynamic search window formation, on-chip memory power-gating, smaller logic, and fast ME/DE scheme. Note that the standard-cell library and fabrication technologies are different compared to (TSUNG, CHEN, *et al.*, 2009) and also to architecture (A).

Architecture (C) was designed based in the same architecture used in (B) except for the memory design and the more sophisticated application-aware power management. For this reason, (C) presents similar gate count compared to (B). The number of memory bits was increased in 62.5%, however, the improved power management led to a total power reduction of 27%, 30% and 79% if compared to (B), (A), and (TSUNG, CHEN, *et al.*, 2009), respectively. To the best of our knowledge, the Motion and Disparity Estimation HW Architecture with Application-Aware Dynamic Power Management (C) represents the most efficient architectural solution available in the current literature and guarantees the processing of 4-view HD1080p running at 300MHz and dissipation 57 mW.



SAD Units:	Sum of Absolute Differences Operators
ME/DE Ctrl:	Motion/Disparity Estimation Control
AGU:	Address Generation Unit
DPM:	Dynamic Power Management

Figure 6.13: ME/DE Architecture with Application-Aware Dynamic Power Management physical layout

Table 6.6: Motion and disparity estimation hardware architectures comparison

	(CHANG, TSAI, *et al.*, 2010)	(TSUNG, CHEN, *et al.*, 2009)	**Multi-Level Pipelined Fast ME/DE Architecture (A)**	**ME/DE Architecture w/ Dynamic Search Window (B)**	**ME/DE Architecture w/ app-aware DPM (C)**
Technology	UMC 90nm	TSMC 90nm Low Power LowK Cu	IBM 65nm LPe LowK	ST 65nm LP 7 metal layer	ST 65nm LP 7 metal layer
Gate Count	562k	230k	211k	102k	102k
SRAM	170 Kbits	64 Kbits	737 Kbits	512 Kbits	832 Kbits
Max. Frequency	95 MHz	300 MHz	300 MHz	300 MHz	300 MHz
Power	n/a	265mW, 1.2v	81mW, 0.8v	74mW, 1.0v	57mW, 1.0v
Search Range		[±16,±16]	[±64,±64]	[±96,±96]	[±96,±96]
Proc. Capability	CIF @ 42fps	4-views 720p	4-views HD1080p	4-views HD1080p	4-views HD1080p

At first analysis, the main drawback of the proposed ME/DE architectures lies in the increase on-chip video memory in comparison to the state-of-the-art. The on-chip memory in our hardware is relatively larger as it supports a much bigger search window of up to [±96,±96] compared to [±16,±16] in (TSUNG, CHEN, *et al.*, 2009) (which is insufficient to capture larger disparity vectors). However, the larger on-chip memory does not imply in increased power dissipation because of the dynamic power management and power-gating techniques employed in our solutions.

The authors of (TSUNG, CHEN, *et al.*, 2009) use a rectangular data reuse technique such as Level-C (CHEN, HUANG, *et al.*, 2006), which compared to our proposed solutions (search as dynamic window formation) perform inefficiently. Note, Level-C (CHEN, HUANG, *et al.*, 2006) with a search window of [±96,±96] would require four memories of 288Kb (i.e., a total of 1,115 Mb) to exploit the reusability in four possible prediction directions available in MVC. Our approaches implement it with 737 Kbits, 512Kbits and, 832Kbits, respectively. To perform a fair comparison, we have deployed the Level-C and Level-C+ (CHEN, HUANG, *et al.*, 2006) techniques in our hardware architecture.

Figure 6.14 shows the energy benefit of employing our dynamically expanding search window and multi-bank on-chip memory with power-gating (implemented in (B)). Compared to Level-C and Level-C+ (CHEN, HUANG, *et al.*, 2006) prefetching techniques (based on rectangular search windows), our approach presents energy reduction in on-chip and off-chip memories as shown in Figure 6.14. For a search window of [±96,±96], our approach provides an energy reduction of up to 82-96% and 57-75% for off-chip and on-chip memory access, respectively. These significant energy savings are due to the fact that Level-C and Level-C+ (CHEN, HUANG, *et al.*, 2006) suffer from a high data retransmission for every first MB in the row. Additionally, our approach provides higher data reuse and incurs reduced leakage due to a smaller on-chip memory and power-gating of the unused sectors.

Figure 6.15 shows the leakage reduction for the application-aware DPM (architecture (C)) normalized to Level-C+ (CHEN, HUANG, *et al.*, 2006). Due to its reduced size compared to Level-C+, our memory even without the power management (*NC*) is able to provide 50% leakage energy reduction compared to Level-C+. When the application-aware power-gating only at the frame level (*FC*) is integrated with our memory, the leakage energy reduction approaches 75%. The fine-grained power management at the MB level (FMBC) provides further 3%-5% leakage reduction, altogether providing up to 80% leakage reduction compared to Level-C+ (for more details kindly refer to memory miss results in Section 5.4.6). Altogether, our application-aware power management leads to reduced energy, which is the primary design concern in this work.

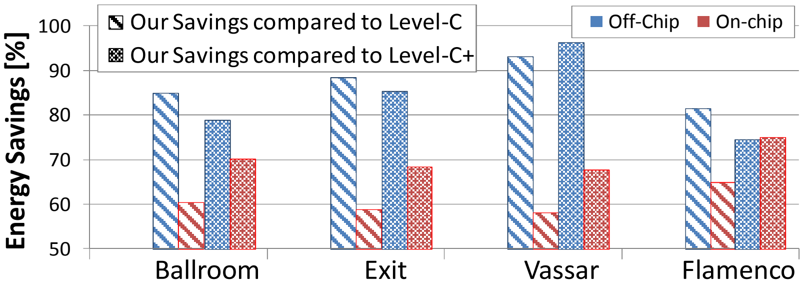


Figure 6.14: Memory-related energy savings employing Dynamic Search Window Technique

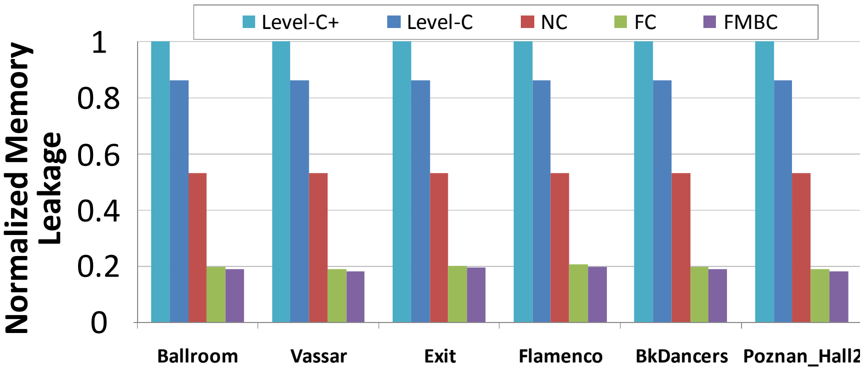


Figure 6.15: On-chip memory leakage reduction

6.3 Summary of Results and Comparison

To cope with comparison fairness issues, along this chapter were detailed all setting and videos used for comparison along this thesis. The video benchmark sequences were classified using the spatial, temporal and disparity indexes to quantify their complexity along theses axes. Additionally, the simulation infrastructure and tools employed along this work were presented in Section 6.1.

A complete comparison to the state-of-the-art was presented in Section 6.2 and showed the superiority of our solutions. The mode decision complexity reduction algorithms are able to provide average 71% complexity reduction with 0.17dB quality loss or 5.8% bitrate increase (BjØntegaard). In turn, the Fast ME/DE contributes with additional 83% complexity reduction with a drawback of 0.116dB quality loss or 10.6% bitrate increase. The energy/complexity versus quality tradeoff can be managed using the presented complexity adaptation algorithms whose stability and fast reacting to changing scenarios was demonstrated in Section 6.2.1.2. The quality drawback posed by our algorithms may be recovered by employing our HRC that provides an average video quality increase of 1.9dB (BjØntegaard).

Compared to the state-of-the-art architecture for ME/DE the solutions presented in this thesis are able to reduce the gate count in 56% while increasing the performance 2.26x. Most importantly, our latest architecture is able to provide real-time 4-views HD1080p at 300MHz with 57mW. It represents a 79% power reduction. This reduction is mainly due to intelligent on-chip video memory energy management. The chip physical layout is showed in Section 6.2.3.

The results presented in Chapter 6 for both, energy-efficient algorithms and energy-efficient architectures, demonstrate the superior performance of our solutions in face of the related works. Moreover, the results demonstrate that is possible to provide solutions able to encode MVC at real time while respecting to embedded devices energy constraints.

7 CONCLUSION AND FUTURE WORKS

The presented thesis focuses on the energy reduction of the Multiview Video Coding (MVC) encoder to enable the realization of real-time high-definition 3D-video encoding running on mobile embedded devices with battery-constrained energy. For that, novel energy-efficient techniques are proposed at both, algorithmic and architectural abstraction levels. The joint consideration of algorithms and underlying hardware architecture is the key enabler to provide improved energy-efficiency, as demonstrated along this thesis.

The strong correlation within the 3D-Neighborhood domain, concept defined in this work, has been the base for designing most of the algorithms and hardware architecture adaptation schemes proposed. An extensive study based on statistical analysis correlating MVC coding side information (such as coding modes, motion/disparity vectors and RDCost) to the video content properties is provided to justify the importance of the 3D-Neighborhood understanding and to demonstrate its potential to support energy reduction in the MVC encoder.

A set of *energy-efficient algorithms for MVC* compose one of the major contributions to the state-of-the-art proposed in this work. Two fast mode decision algorithms are described targeting energy-efficiency through complexity reduction. The Early SKIP prediction exploits the high occurrence of SKIP coded MBs to accelerate the encoding process by employing statistical methods that define if each MB is in the high SKIP probability region in order to avoid other coding modes evaluation. The early SKIP concept is integrated in the multi-level fast mode decision algorithm to further reduce the energy consumption. It eliminates coding modes evaluation even in the case where an early SKIP is not detected by analyzing the coding modes available within the 3D-Neighborhood while considering an video/RDCost-based mode ranking. The video properties are also used to define block sizes and prediction modes orientation. To protect the multi-level fast MD algorithm from inserting excessive quality losses an early termination test is inserted between each prediction step. This algorithm defines QP-based thresholds for two distinct energy reduction strengths, the *relax* and *aggressive* strengths. By employing two operation modes it is possible to select the best energy vs. quality tradeoff for a given system state and video content. Moreover, multiple fast MD modes enable the integration of a energy-aware complexity adaptation control scheme. The multi-level fast MD algorithm evaluation, results, and comparison with related works, points to an average complexity reduction of 25% at the cost of 0.32dB quality loss and 10% bitrate increase, for *aggressive* mode, and 0.1dB quality loss and 3% bitrate increase, for *relax* mode.

This thesis work demonstrated that the coding properties and coding effort highly depends on the video content. Moreover, when considering embedded applications, the processing power is constrained by energy resources available in the embedded battery. From this observations, it is proposed an energy-aware complexity adaptation algorithm.

The goal is to jointly consider the video input characteristics and the battery state to sustain the highest possible video quality by selecting the appropriated MD algorithm and quality states. In case of battery discharging, further energy reduction is necessary leading to quality reduction. Thus, the complexity-adaptation algorithm delivers a graceful quality degradation by employing the binocular suppression theory knowledge. For binocular displaying, the Human Visual Systems (HVS) tends to perceive the highest quality view, so the proposed algorithms firstly drops the quality of odd views guaranteeing a high perceived quality while reducing energy consumption for encoding these odd views. Experimental results show the beneficial effect of the complexity adaptation for energy consumption and smooth quality variation along the time under battery charging and discharging scenarios.

The motion and disparity estimation consumes more than 90% of the total MVC encoding energy and represents the main target for energy reduction. In this work, a novel Fast ME/DE was detailed. It uses the motion and disparity vectors available in the 3D-Neighborhood to avoid, for multiple frames in the prediction structure, the complete motion/disparity search pattern. There are defined two classes for frames, key and non-key frames, where the key frames are encoded using off-the-shelf fast search patterns and the non-key employ our Fast ME/DE. According to the confidence, defined using image properties, on the vectors inferred from the neighborhood, each MB in the non-key frames select between fast mode or the ultra-fast mode. These modes test only 3 or 13 candidate blocks, respectively. The proposed Fast ME/DE algorithms is able to reduce 83% of the total encoding time at the cost of 0.116dB and 10% bitrate increase.

To compensate eventual losses posed by the energy-efficient algorithms, a video quality management based on our hierarchical rate control (HRC) algorithm was proposed. The HRC operates in two actuation levels, the frame level and the basic unit (BU) level, and features a coupled closed feedback loop. The frame-level RC employs a Model Predictive Controller (MPC) to predict the bitrate for future frames based on the bit allocation in the frames belonging to the 3D-Neighborhood. The multiple stimuli coming from temporal, disparity and phase neighboring frames compose the MPC input. The bitrate prediction is then used to define the optimal QP for that frame. The QP is further refined inside the frame by a Markov Decision Process (MDP)-based BU-level rate control. It considers Regions of Interest to prioritize hard-to-encode image regions. Reinforcement learning is used to update the MDP parameters. The HRC provides smooth bitrate and video quality variations along time and view axes, while respecting to bandwidth constraints and providing improved video quality. Compared to the fixed QP solution, the video quality was improved in 1.9dB (BjØntegaard). In comparison to the state-of-the-art, the bitrate prediction error is reduced in 0.83% in addition to 0.106dB PSNR increase or 4.5% BjØntegaard bitrate reduction.

In addition to the energy-efficient algorithms, the severe energy restrictions and performance requirements demanded by the MVC encoder require hardware dedicated acceleration able to employ sophisticated application-aware adaptive power management techniques. Three *energy-efficient hardware architectures* for motion and disparity estimation were proposed in order to provide multiple implementation options under distinct encoder design constraints. The proposed architectures provide throughput to encode, at real time, 4-view HD1080p video sequences.

The multi-level pipelined ME/DE hardware architecture featuring Fast ME/DE was jointly designed with the Fast ME/DE algorithm presented in this thesis. This dual-pipelined solution employs two parallel search control and dispatch units, one for the regular search and one for the fast algorithm itself, and three cache memories with distinct caching and fetching paradigms to minimize misses and avoid data retransmission. A novel processing scheduling was designed exploiting the multiple parallelism levels available in the MVC coding structure, view, frame, reference-frame and MB levels, to deal with data dependencies.

Merging the two pipelines, it was proposed a novel ME/DE architecture that incorporates a multi-bank video on-chip memory and the dynamic search window-based data prefetching technique for jointly reducing the on/off-chip memory energy consumption. A dynamically expanding search window is constructed at run time based on the neighborhood-extracted search map to reduce the off-chip memory accesses. Considering the multi-stage processing nature of advanced fast ME/DE schemes, the reduced-size multi-bank on-chip memory is partitioned in multiple sectors which can be power-gated depending upon the video properties while enabling fine-grained tuning for leakage current reduction.

The potential of memory-related energy savings motivates the proposal of a novel energy-efficient architecture featuring an elaborated application-aware dynamic power management scheme for the on-chip video memory. The memory organization (size, banks, sectors, etc.) is driven by an extensive analysis of memory-usage behavior for various 3D-video sequences. Considering the multiple power state model adopted, the application-aware power management scheme is employed to reduce the leakage energy of the on-chip memory. The knowledge of motion and disparity estimation algorithm in conjunction with video properties are considered to predict the memory requirements of each frame and refine to macroblock level. A cost function is evaluated to determine an appropriate sleep mode for each memory sector, while considering the wakeup overhead (latency and energy).

The architectural contribution presented in this thesis involves the architectures design, management schemes, complete RTL coding and ASIC synthesis down to physical layer using 65-nm fabrication technologies. From experimental results for multiple video sequences, the proposed architectures provide a dynamic energy reduction of 82-96% for the off-chip memory and up to 80% on-chip leakage energy reduction compared to state-of-the-art. From this contribution, it is possible to demonstrate the feasibility of performing motion and disparity estimation for up to 4-view HD1080p at 30fps with a power dissipation of 57mW running at 300MHz on an IC footprint with 102k gates.

The overall results and benchmarks demonstrate the energetic efficiency of the proposed algorithms and architectures in front of the state-of-the-art solutions. This proves our claim that for attending the 3D video coding requirements for embedded systems, it is required to jointly consider and optimize the coding algorithms and the underlying dedicated hardware architectures. Additionally, run-time adaptation is required to better predict the system behavior and react to changing video input, coding parameters and battery level scenarios. For that, deep MVC application knowledge coming from extensive analysis, such as the correlation available within the 3D-Neighborhood, must be employed.

7.1 Future Works

Beyond the contribution brought in this thesis work, there are multiple research topics related to 3D-video coding and video processing that were not addressed in this volume.

The algorithms and architectures here presented were centered in mode decision and motion and disparity estimation once these are the most energy-hungry coding units in the MVc encoder. Additionally, focusing on video quality issues the rate control was discussed. The MVC, however, brings a big set of other research challenges if embedded applications are considered. 3D videos pre- and post-processing also play key roles in the 3D-video system and present plenty of novel challenges. Finally, next generation 3D-video coding algorithms are under study for future standardization. The next 3D-video generation is expected to bring innovative tools and provide good perspective for future research opportunities in the 3D-multimedia field.

Remaining MVC challenges: Although the main challenges in terms of complexity and energy consumption are related to the MD and ME/DE blocks, attending to the MVC demands while respecting energy constraints presents challenges related to other MVC processing blocks. The entropy encoder, for instance, may become the bottleneck of the encoder system if no proper parallelization is employed. The block-level data dependencies in intra prediction also require research attention. Finding efficient solutions to deal with data dependencies and parallelization issues provide interesting research opportunities for future works.

3D video pre- and post-processing: Video encoding is one single stage in the 3D-video system. Between video capturing and video coding phases, there is a need for pre-processing such as geometrical calibration (for correcting the aligning of the multiple videos) and color correction (responsible for equalizing the brightness level and color gamut). After the transmission and decoding, the video is processed for displaying depending on the application and display technology. This post-processing phase includes color space mapping (in a system using color polarization), resolution scaling and viewpoint synthesis (generation of intermediate viewpoints for displaying). The pre- and post-processing implement complex and data-intensive algorithms (especially for viewpoint synthesis) that run concurrently with the video encoder/decoder and require real-time performance. Therefore, the embedded energy and hardware resources must be shared to attend both video coding and pre-/post-processing demands.

Next generation 3D video coding: The next generation for 3D video coding is currently referred as 3DV (3D Video)(ISO/IEC, 2009) and is based on the Video+Depth concept that defines distinct channels to transmit video and the depth maps. The 3DV is expected to be defined as an extension to the HEVC/H.265 (SULLIVAN e OHM, 2010). The 3DV tools will bring a completely new set of challenges boosting the research topics related to 3D multimedia. Moreover, the video coding standards lifetime is expected to reduce for future standard generations resulting in the simultaneous coexistence of multiple coding standards. Thus, there is a need to support multiple complex coding standards in the same device by employing flexible and adaptive solutions.

8 REFERENCES

ABBO, A. A. et al. Xetal-II: A 107 GOPS, 600 mW Massively Parallel Processor for Video Scene Analysis. **IEEE Journal of Solid-State Circuits**, 43, n. 1, January 2008. 192-201.

AGARWAL, K. et al. **Power Gating with Multiple Sleep Modes**. International Symposium on Quality Electronic Design. [S.l.]: [s.n.]. 2006. p. 633-637.

AGRAFIOTIS, D. et al. **Multiple Priority Region of Interest Coding with H.264**. IEEE International Conference on Image Processing. [S.l.]: [s.n.]. 2006. p. 53 -56.

AKIN, A.; SAYILAR, G.; HAMZAOGLU, I. **A Reconfigurable Hardware for One Bit Transform Based Multiple Reference Frame Motion Estimation**. Design, Automation & Test in Europe. Milão: EDAA. 2010. p. 393-398.

ARAPOSTATHIS, A.; KUMAR, R.; TANGIRALA, S. Controlled Markov chains with safety upper bound. **Automatic Control**, 48, n. 7, July 2003. 1230- 1234.

ARM LTD. ARM - The Architecture for the Digital World, 2012. Disponivel em: <http://www.arm.com/>.

ARSURA, E. et al. **Fast macroblock intra and inter modes selection for H.264/AVC**. International Conference on Multimedia and Expo (ICME). [S.l.]: [s.n.]. 2005. p. 378-381.

BARTO, A. G. Reinforcement learning control. **Current Opinion in Neurobiology**, 4, 1994. 888-893.

BAUER, L. et al. **RISPP:** rotating instruction set processing platform. 44th Design Automation Conference. San Diego, California: [s.n.]. 2007. p. 791-796.

BAUER, L. et al. **Run-time system for an extensible embedded processor with dynamic instruction set**. Design, automation and test in Europe. Munich, Germany: [s.n.]. 2008. p. 752-757.

BAUER, L.; SHAFIQUE, M.; HENKEL, J. **Run-time instruction set selection in a transmutable embedded processor**. 45th Design Automation Conference. [S.l.]: [s.n.]. 2008. p. 56-61.

BECK, A. C. S. et al. **Transparent Reconfigurable Acceleration for Heterogeneous Embedded Applications**. Design Automation Conference. [S.l.]: [s.n.]. 2008. p. 1208-1213.

BENNETT, K. Intel Core i7-3960X - Sandy Bridge E Processor Review. **HardOCP**, November 2011. Disponivel em: <http://hardocp.com/article/2011/11/14/intel_core_i73960x_sandy_bridge_e_processor_review/4>. Acesso em: 14 ago. 2012.

BEREKOVIC, M. et al. Mapping of nomadic multimedia applications on the ADRES reconfigurable array processor. **Microprocessors and Microsystems**, 33, n. 4, June 2009. 290-294.

BHASKARAN, V.; KONSTANTINIDES, K. **Image and Video Compression Standards:** Algorithms and Architectures. Boston: Kluwer Academic, 1999.

BLANCHE, P.-A. et al. Holographic three-dimensional telepresence using large-area photorefractive polymer. **Nature**, 468, November 2010. 80-83.

BLU-RAY DISC ASSOCIATION. White Paper Blu-ray Disc Read-Only Format, 2010. Disponivel em: <http://www.blu-raydisc.com/assets/Downloadablefile/BD-ROM_Audio_Visual_Application_Format_Specifications-18780.pdf>. Acesso em: 20 ago. 2012.

CADENCE DESIGN SYSTEMS, INC. Digital Implementation , 2012. Disponivel em: <http://www.cadence.com/products/di/Pages/default.aspx>. Acesso em: 25 jul. 2012.

CAO, Z. et al. Optimality and Improvement of Dynamic Voltage Scaling Algorithms for Multimedia Applications. **IEEE Transactions on Circuits and Systems I: Regular Papers**, 57, n. 3, March 2010. 681-690.

CHAN, C.-C.; TANG, C.-W. Coding statistics based fast mode decision for multi-view video coding. **Journal of Visual Communication and Image Representation**, Available online 18 January 2012. 2012.

CHANG, H.-C. et al. A Dynamic Quality-Adjustable H.264 Video Encoder for Power-Aware Video Applications. **IEEE Transactions on Circuits and Systems for Video Technology**, 12, December 2009. 1739 -1754.

CHANG, N. Y.-C. et al. Algorithm and Architecture of Disparity Estimation With Mini-Census Adaptive Support Weight. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, 20, n. 6, 2010. 792-805.

CHEN, C.-Y. et al. Level C+ Data Reuse Scheme for Motion Estimation With Corresponding Coding Orders. **IEEE Transactions on Circuits and Systems for Video Technology**, 16, n. 4, Abril 2006. 553-558.

CHEN, C.-Y. et al. Level C+ Data Reuse Scheme for Motion Estimation With Corresponding Coding Orders. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 16, n. 4, p. 553-558, Abril 2006.

CHEN, J. C.; CHIEN, S.-Y. CRISP: Coarse-Grained Reconfigurable Image Stream Processor for Digital Still Cameras and Camcorders. **IEEE Transactions on Circuits and Systems for Video Technology**, 18, n. 9, September 2008. 1223-1236.

CHEN, T.-C. et al. Fast Algorithm and Architecture Design of Low-Power Integer Motion Estimation for H.264/AVC. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, 17, n. 5, 2007. 568-577.

CHEN, T.-C. et al. Fast Algorithm and Architecture Design of Low-Power Integer Motion Estimation for H.264/AVC, v. 17, n. 5, p. 568-577, 2007.

CHEN, Y. et al. **Coding techniques in multiview video coding and joint multiview video model**. Picture Coding Symposium. Piscataway: IEEE. 2009. p. 313-316.

CHEN, Y. et al. **The Emerging MVC Standard for 3D Video Services**. 3DTV Conference. [S.l.]: [s.n.]. 2009. p. 1-13.

CHEN, Y.-H. et al. Algorithm and Architecture Design of Power-Oriented H.264/AVC Baseline Profile Encoder for Portable Devices. **IEEE Transactions on Circuits and Systems for Video Technology**, 19, n. 8, August 2009. 1118 -1128.

CHEN, Z.; ZHOU, P.; HE, Y. **Fast integer pel and fractional pel motion estimation for JVT - JVT-F017**. [S.l.]. 2002.

CHIEN, S.-Y. et al. An 8.6 mW 25 Mvertices/s 400-MFLOPS 800-MOPS 8.91 mm Multimedia Stream Processor Core for Mobile Applications. **IEEE Journal of Solid-State Circuits**, 43, n. 9, September 2008. 2025 - 2035.

CHIU, J.-C.; CHOU, Y.-L. Multi-streaming SIMD multimedia computing engine. **Microprocessors and Microsystems**, 34, n. 7-8, November 2010. 247-258.

CHUANG, T.-D. et al. **A 59.5mW scalable/multi-view video decoder chip for Quad/3D Full HDTV and video streaming applications**. IEEE International Conference on Solid-State Circuits (ISSCC). [S.l.]: [s.n.]. 2010. p. 330 -331.

CIRCUITS MULTI-PROJECTS. STMicroelectronics Deep Sub-Micron Processes, 2012. Disponivel em: <http://cmp.imag.fr/aboutus/slides/slides2007/04_KT_ST.pdf>. Acesso em: 12 jul. 2012.

CISCO. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011–2016, 2012. Disponivel em: <http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf>. Acesso em: 20 ago. 2012.

DE-FRUTOS-LÓPEZ et al. An improved fast mode decision algorithm for intraprediction in H.264/AVC video coding. **Signal Processing: Image Communication**, 25, n. 10, November 2010. 709-716.

DENG, Z.-P. et al. **A Fast View-Temporal Prediction Algorithm for Stereoscopic Video Coding**. International Congress on Image and Signal Processing (CISP). [S.l.]: [s.n.]. 2009. p. 1-5.

DÍAZ-HONRUBIA, A. J.; MARTÍNEZ, J. L.; CUENCA, P. **HEVC:** A Review, Trends and Challenges. Workshop on Multimedia Data Coding and Transmission. [S.l.]: [s.n.]. 2012.

DING, L.-F. et al. Content-Aware Prediction AlgorithmWith Inter-View Mode Decision for Multiview Video Coding. **IEEE Transactions on Multimedia**, 10, n. 8, 2008. 1553 - 1564.

DING, L.-F. et al. **Fast motion estimation with inter-view motion vector prediction for stereo and multiview video coding**. International Conference on Acoustics Speech and Signal Processing (ICASSP). [S.l.]: [s.n.]. 2008. p. 1373 - 1376.

DING, L.-F. et al. A 212 MPixels/s 4096 2160p Multiview Video Encoder Chip for 3D/Quad Full HDTV Applications. **IEEE Journal of Solid-State Circuits**, 45, n. 1, January 2010. 46-58.

DING, L.-F. et al. A 212 MPixels/s 4096 2160p Multiview Video Encoder Chip for 3D/Quad Full HDTV Applications, v. 45, n. 1, p. 46-58, 2010.

DODGSON, N. A. Autostereoscopic 3D Displays. **IEEE Computer**, 38, n. 8, 2005. 31-36.

DOLBY. Dolby 3D, 2012. Disponivel em: <http://www.dolby.com/us/en/consumer/technology/movie/dolby-3d.html>. Acesso em: 20 ago. 2012.

ERDAYANDı, K. JMVC Documentation. **JMVC - JVT-AD207**, 2009. Disponivel em: <http://students.sabanciuniv.edu/~kerdayandi/jmvc/index_jmvc.html>. Acesso em: 20 set. 2012.

FINCHELSTEIN, D. F.; SZE, V.; CHANDRAKASAN, A. P. Multicore Processing and Efficient On-Chip Caching for H.264 and Future Video Decoders. **IEEE Transactions on Circuits and Systems for Video Technology**, 19, n. 11, November 2009. 1704 - 1713.

FUJIFILM. FinePix REAL 3D W3 | FujiFilm Global, 2011. Disponivel em: <http://www.fujifilm.com/products/3d/camera/finepix_real3dw3/>. Acesso em: 20 ago. 2012.

FUJII, T. **Panel Discussion 1 (D1) - 3DTV/FTV**. [S.l.]: [s.n.]. 2010.

FUKANO, G. et al. **A 65nm 1Mb SRAM Macro with Dynamic Voltage Scaling in Dual Power Supply Scheme for Low Power SoCs**. [S.l.]: [s.n.]. 2008. p. 97-98.

FUKANO, G. et al. **A 65nm 1Mb SRAM Macro with Dynamic Voltage Scaling in Dual Power Supply Scheme for Low Power SoCs**. [S.l.]: [s.n.]. 2008. p. 97-98.

GARCÍA, C. E.; PRETT, D. M.; MORARI, M. Model predictive control: Theory and practice—A survey. **Automatica**, 25, n. 3, May 1989. 335-348.

GASSÉE, J.-L. Intel’s bold bet against ARM: visionary or myopic? **Monday Note**, 2010. Disponivel em: <http://www.mondaynote.com/2010/06/27/intel%E2%80%99s-bold-bet-against-arm-visionary-or-myopic/>. Acesso em: 02 set. 2012.

GHANBARI, M. The cross-search algorithm for motion estimation. **IEEE Transactions on Communications**, 38, n. 7, July 1990. 950-953.

GRECOS, C.; YANG, M. Y. Fast Inter Mode Prediction for P Slices in the H264 Video Coding Standard. **IEEE Trans. on Broadcadting**, 2005. 256- 263.

HAN, D.-H.; LEE, Y.-L. **Fast Mode Decision using Global Disparity Vector for Multiview Video Coding**. Future Generation Communication and Networking Symposia (FGCNS). [S.l.]: [s.n.]. 2008. p. 209–213.

HE, Z.; CHENG, W.; CHEN, X. Energy Minimization of Portable Video Communication Devices Based on Power-Rate-Distortion Optimization. **IEEE Transactions on Circuits and Systems for Video Technology**, 18, n. 5, May 2008. 596 -608.

HUANG, Y.-H.; OU, T.-S.; SHEN, H. **Fast H.264 Selective Intra Mode Decision for Inter-Frame Coding**. Picture Coding Symposium (PCS). [S.l.]: [s.n.]. 2009. p. 377–380.

HUANG, Y.-W. et al. Analysis and Complexity Reduction of Multiple Reference Frames Motion Estimation in H.264/AVC, v. 16, n. 4, 2006.

HUFF, H. R.; GILMER, D. C. **High Dielectric Constant Materials:** VLSI MOSFET Applications. New York: Springer, 2004.

IC INSIGHTS. IC Insights Raises Forecast for Tablets, Notebooks, Total PC Shipments. **Electronic Specifier**, 2012. Disponivel em: <http://www.electronicspecifier.com/Tech-News/IC-Insights-Raises-Forecast-Tablets-Notebooks-Total-PC-Shipments.asp>. Acesso em: 12 ago. 2012.

IMAX. IMAX3D, 2012. Disponivel em: <http://www.imax.com/about/imax-3d/>. Acesso em: 20 ago. 2012.

ISO/IEC. Vision on 3D Video, 2009. Disponivel em: <http://mpeg.chiariglione.org/visions/3dv/index.htm>. Acesso em: 20 ago. 2012.

ISO/IEC. **Commom Test Conditions for MVC - W12036**. Geneva, Switzerland. 2011.

ITU-T. **Subjective video quality assessment methods for multimedia applications - P.910**. [S.l.]. 1999.

JAVED, H. et al. **Low-Power Adaptive Pipelined MPSoCs for Multimedia:** An H.264 Video Encoder Case Study. Design Automation Conference. [S.l.]: [s.n.]. 2011. p. 1032-1037.

JEON, B. W.; LEE, J. Y. **Fast mode decision for H.264 - Document JVT-J033**. Waikoloa, Hawaii, USA. 2003.

JI, W. et al. Power Scalable Video Encoding Strategy Based on Game Theory. **ADVANCES IN MULTIMEDIA INFORMATION PROCESSING - PCM 2009**, 2009. 1237-1243.

JI, W. et al. ESVD: an integrated energy scalable framework for low-power video decoding systems. **EURASIP Journal on Wireless Communications and Networking**, 5, n. 13, January 2010. 5:1--5:13.

JIANG, M.; YI, X.; LING, N. **Improved frame-layer rate control for H.264 using MAD ratio**. International Symposium on Circuits and Systems. [S.l.]: [s.n.]. 2004.

JING, X.; CHAU, L. An efficient three-step search algorithm for Block motion estimation. **IEEE Transactions on Multimedia**, 6, n. 3, 2004. 435-438.

JING, X.; CHAU, L.-P. Fast Approach for H.264 Inter Mode Decision. **Electronic Letters**, 2004. 1050-1052.

JVT. **Draft ITU-T Rec. and final draft international standard of joint video specification**. [S.l.]. 2003.

JVT. **Joint Draft 8.0 on Multiview video coding - JVT-AB204**. [S.l.]. 2008.

JVT. **JMVC 6.0**. [S.l.]. 2009.

JVT. **Joint Multiview Vido Coding**. [S.l.]. 2009.

KAMAT, S. P. Energy management architecture for multimedia applications in battery powered devices. **IEEE Transactions on Consumer Electronics**, 55, n. 2, May 2009. 763 -767.

KAUFF, P. et al. Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability. **Signal Processing: Image Communication**, 22, 2007. 217-234.

KAY, R. Forbes. **Is The PC Dead?**, 2011. Disponivel em: <http://www.forbes.com/sites/rogerkay/2011/02/28/is-the-pc-dead/>. Acesso em: 05 set. 2012.

KHAILANY, B. K. et al. A Programmable 512 GOPS Stream Processor for Signal, Image, and Video Processing. **IEEE Journal of Solid-State Circuits**, 43, n. 1, January 2008. 202 -213.

KIM, B.-G.; CHO, C.-S. **A Fast Inter-Mode Decision Algorithm based on Macro-block Tracking for P Slices in the H.264/AVC Video Standard**. International Conference on Image Processing (ICIP). [S.l.]: [s.n.]. 2007. p. V-301–V-304.

KIM, C.; KUO, C.-C. J. Feature-Based Intra-/InterCoding Mode Selection for H.264/AVC. **IEEE Transactions on Circuits and Systems for Video Technology**, 17, n. 4, April 2007. 441–453.

KIM, D.-Y.; LEE, Y.-L. A fast intra prediction mode decision using DCT and quantization for H.264/AVC. **Signal Processing: Image Communication**, 26, n. 8-9, Oct 2011. 455-465.

KIM, Y.; KIM, J.; SOHN, K. Fast Disparity and Motion Estimation for Multi-view Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, 53, n. 2, 2007. 712-719.

KIM, Y.; KIM, J.; SOHN, K. Fast Disparity and Motion Estimation for Multi-view Video Coding, v. 53, n. 2, p. 712-719, 2007.

KO, H.; YOO, K.; SOHN, K. Fast mode-decision for H.264/AVC based on inter-frame correlations. **Signal Processing: Image Communication**, 24, n. 10, November 2009. 803-813.

KOLLIG, P.; OSBORNE, C.; HENRIKSSON, T. **Heterogeneous multi-core platform for consumer multimedia applications**. Design, Automation Test in Europe Conference. [S.l.]: [s.n.]. 2009. p. 1254 -1259.

KONDO, H. et al. Heterogeneous Multicore SoC With SiP for Secure Multimedia Applications. **IEEE Journal of Solid-State Circuits**, 44, n. 8, August 2009. 2251-2259.

KOO, H.-S.; JEON, Y.-J.; JEON, B.-M. **MVC Motion Skip Mode - Doc. JVT-W081**. [S.l.]. 2007.

KROLIKOSKI, S. Chipvision Design Systems. **Orinoco Saves Power**, 2004. Disponivel em: <http://www.eda.org/edps/edp04/submissions/presentationKrolikoski.pdf>. Acesso em: 15 nov. 2010.

KRÜGERA, J. et al. Image based 3DSurveillance for flexible Man-Robot-Cooperation. **CIRP Annals - Manufacturing Technology**, 54, n. 1, 2005. 19-22.

KUHN, P. **Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation**. Boston: Kluwer Academic Publishers, 1999.

KUME, H. Panasonic's New Li-Ion Batteries Use Si Anode for 30% Higher Capacity. **TechOn**, 2010. Disponivel em: <http://techon.nikkeibp.co.jp/article/HONSHI/20100223/180545/>. Acesso em: 20 ago. 2012.

KWON, D.-K.; SHEN, M.-Y.; KUO, C.-C. J. Rate Control for H.264 Video With Enhanced Rate and Distortion Models. **IEEE Transactions on Circuits and Systems for Video Technology**, 17, n. 5, May 2007. 517 -529.

LEE, P.-J.; LAI, Y.-C. **Vision perceptual based rate control algorithm for multi-view video coding**. International Conference on System Science and Engineering (ICSSE). [S.l.]: [s.n.]. 2011. p. 342 -345.

LEE, S.-Y.; SHIN, K.-M.; CHUNG, K.-D. **An Object-based Mode Decision Algorithm for Multi-view Video Coding**. International Symposium on Multimedia (ISM). [S.l.]: [s.n.]. 2008. p. 74 - 81.

LI, Z. G. et al. **Adaptive basic unit layer rate control for JVT - JVT-G012**. Thailand. 2003.

LIANG, Y.; AHMAD, I. Power and Distortion Optimization for Pervasive Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology**, 19, n. 10, October 2009. 1436-1447.

LIM, K. P. **Fast inter mode selection - Document JVT-I020**. [S.l.]. 2003.

LIN, J.-P.; TANG, A. C.-W. **A Fast Direction Predictior of Inter Frame Prediction for Multi-View video Coding**. IEEE International Symposium on Circuits and System. Piscataway: IEEE. 2009. p. 2598-2593.

LIN, Y.-K. et al. **A 242mW 10mm2 1080p H.264/AVC high profile encoder chip**. Design Automation Conference. [S.l.]: [s.n.]. 2008. p. 78-83.

LING, N. **Expectations and Challenges for Next Generation**. Conference on Industrial Electronics and Applicationsis. [S.l.]: [s.n.]. 2010. p. 2339-2344.

LIU, A. et al. Just Noticeable Difference for Images With Decomposition Model for Separating Edge and Textured Regions. **IEEE Transactions on Circuits and Systems for Video Technology**, 20, n. 11, November 2010. 1648 -1652.

LIU, X.; SHENOY, P.; CORNER, M. D. Chameleon: Applicationlevel power management. **IEEE Transactions on Mobile Computing**, 7, n. 8, August 2008. 995 - 1010.

LU, X. et al. **Fast mode decision and motion estimation for H.264 with a focus on MPEG- 2/H.264 transcoding**. International Conference on Circuits and Systems (ISCAS). [S.l.]: [s.n.]. 2005. p. 1246–1249.

MA, S.; GAO, W.; LU, Y. Rate-distortion analysis for H.264/AVC video coding and its application to rate control. **IEEE Transactions on Circuits and Systems for Video Technology**, 15, n. 12, December 2005. 1533 - 1544.

MARLOW, S.; NG, J.; MCARDLE, C. **Efficient motion estimation using multiple log searching and adaptive search windows**. International Conference on Image Processing and Its Applications. [S.l.]: [s.n.]. 1997. p. 214-218.

MCCANN, K. et al. Technical Evolution of the DTT Platform - An independent report by ZetaCast, commissioned by Ofcom, January 2012. Disponivel em: <http://stakeholders.ofcom.org.uk/binaries/consultations/uhf-strategy/zetacast.pdf>.

MENG, B. et al. **Efficient Intra-Prediction Mode Selection for 4x4 Blocks in H.264**. International Conference on Multimedia and Expo (ICME). [S.l.]: [s.n.]. 2003. p. III-521-III-524.

MENTOR GRAPHICS. ModelSim - Advanced Simulation and Debugging, 2012. Disponivel em: <http://model.com/>. Acesso em: 20 jul. 2012.

MERKLE, P. et al. Efficient Prediction Structures for Multiview Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology**, 17, n. 11, Novembro 2007. 1461-1473.

MERKLE, P. et al. **Stereo video compression for mobile 3D services**. 3DTV Conference. [S.l.]: [s.n.]. 2009. p. 1 -4.

MERRITT, L.; VANAM, R. **Improved Rate Control and Motion Estimation for H.264 Encoder**. IEEE International Conference on Image Processing. [S.l.]: [s.n.]. 2007. p. V-309-V-312.

MIANO, J. **Compressed Image File Formats:** Jpeg, Png, Gif, Xbm, Bmp. Boston: ACM Press, 1999.

MONDAL, S.; MEMIK, S. O. **Fine-grain leakage optimization in SRAM based FPGAs**. ACM Great Lakes symposium on VLSI. [S.l.]: [s.n.]. 2005. p. 238-243.

MORARI, M.; LEE, J. H. Model Predictive Control: Past, Present and Future. **Computers and Chemical Engineering**, 23, 1999. 667–682.

MULLER, K. et al. 3-D reconstruction of a dynamic environment with a fully calibrated background for traffic scenes, v. 15, n. 4, p. 538- 549, 2005.

NACCARI, M. et al. **LOW COMPLEXITY DEBLOCKING FILTER PERCEPTUAL OPTIMIZATION FOR THE HEVC CODEC**. International Conference on Image Processing. [S.l.]: [s.n.]. 2011. p. 737-740.

NINTENDO. Nintendo 3DS, 2011. Disponivel em: <http://www.nintendo.com/3ds>. Acesso em: 20 ago. 2012.

NVIDIA. Nvidia GeForce GX690, 2012. Disponivel em: <http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-690>. Acesso em: 20 ago. 2012.

NVIDIA. Tegra 3 Super Processors, 2012. Disponivel em: <http://www.nvidia.com/object/tegra-3-processor.html>. Acesso em: 20 ago. 2012.

NVIDIA CORP. Tegra 2 and Tegra 3 Super Processors, 2012. Disponivel em: <http://www.nvidia.com/object/tegra-3-processor.html>. Acesso em: 20 ago. 2012.

OH, K.-J.; LEE, J.; PARK, D.-S. **Multi-view video coding based on high efficiency video coding**. Pacific Rim conference on Advances in Image and Video Technology. [S.l.]: [s.n.]. 2011. p. 371-380.

OSTERMANN, J. et al. Video coding with H.264/AVC: tools, performance, and complexity, 1st Quarter 2004. 7 - 28.

OTERO, A. et al. **Run-time Scalable Systolic Coprocessors for Flexible**. International Conference on Field Programmable Logic and Applications (FPL). [S.l.]: [s.n.]. 2010. p. 70-76.

OU, T.-S.; HUANG, Y.-H.; CHEN, H. H. **Efficient MB and Prediction Mode Decisions for Intra Prediction of H.264 High Profile**. Picture Coding Symposium (PCS). [S.l.]: [s.n.]. 2009. p. 1-4.

OZBEK, N.; TEKALP, A. M.; TUNALI, E. T. **Rate Allocation Between Views In Scalable Stereo Video Coding using an Objective Stereo Video Quality Measure**. International Conference on Acoustics Speech and Signal Processing (ICASSP). [S.l.]: [s.n.]. 2007. p. 1045–1048.

PAN, F. et al. Fast mode decision algorithm for intraprediction in H.264/AVC video coding. **IEEE Trans. Circuits and Systems for Video Technology**, 15, n. 7, 2005. 813–822.

PANASONIC. Panasonic HDC-SDT750K, 2011. Disponivel em: <http://www2.panasonic.com/consumer-electronics/support/Cameras-Camcorders/Camcorders/3D-CAMCORDERS/model.HDC-SDT750K>. Acesso em: 20 ago. 2012.

PARK, I.; CAPSON, D. W. **Improved Inter Mode Decision based on Residue in H.264/AVC**. International Conference on Multimedia and Expo (ICME). [S.l.]: [s.n.]. 2008. p. 709–712.

PARK, J. S.; SONG, H. J. Selective Intra Prediction Mode Decision for H.264/AVC Encoders. **International Journal of Applied Science, Engineering and Technology**, 13, 2006. 214–218.

PARK, S.; SIM, D. **An efficienct rate-control algorithm for multi-view video coding**. IEEE International Symposium on Consumer Electronics (ISCE). [S.l.]: [s.n.]. 2009. p. 115 -118.

PAYÁ-VAYÁ, G. et al. **VLIW architecture optimization for an efficient computation of stereoscopic video applications**. International Conference on Green Circuits and Systems (ICGCS). [S.l.]: [s.n.]. 2010. p. 457 -462.

PEI, G. et al. FinFET design considerations based on 3-D simulation and analytical modeling. **IEEE Transactions on Electron Devices**, 49, n. 8, August 2002. 1411-1419.

PENG, Z. et al. Fast Macroblock Mode Selection Algorithm for Multiview Video Coding. **EURASIP Journal on Image and Video Processing**, 2008, 2008.

PENG, Z. et al. **Fast Mode Decision for Multiview Video Coding**. International Conference on Image Processing (ICIP). [S.l.]: [s.n.]. 2008. p. 1081 - 1084.

POURAZAD, M.; NASIOPOULOS, P.; WARD, R. **An Efficient Low Random-Access Delay Panorama-Based Multiview Video Coding Scheme**. IEEE Conference on Image Processing. Cairo: IEEE. 2009. p. 2945-2948.

QUALCOMM INC. Snapdragon S4 Processors: System on Chip Solutions for a New Mobile Age - White Paper, 2011. Disponivel em: <https://developer.qualcomm.com/download/snapdragon-s4-processors-system-on-chip-solutions-for-a-new-mobile-age.pdf>. Acesso em: 22 jul. 2012.

RAJAMANI, K. et al. **Application-Aware Power Management**. IEEE International Symposium on Workload Characterization. [S.l.]: [s.n.]. 2006. p. 39 -48.

REALD. RealD 3D, 2012. Disponivel em: <http://reald.com/>. Acesso em: 20 ago. 2012.

RESEARCH AND MARKETS. **3D TV Market and Future Forecast Worldwide (2010 - 2014)**. [S.l.]: [s.n.], 2010.

RICHARDSON, I. **The H. 264 advanced video compression standard**. [S.l.]: John Wliey and Sons, 2010.

ROY, S.; RANGANATHAN, N.; KATKOORI, S. State-Retentive Power Gating of Register Files in Multicore Processors Featuring Multithreaded In-Order Cores. **IEEE Transactions on Computers**, 60, n. 11, November 2011. 1547 -1560.

SALGADO, L.; NIETO, M. **Sequence Independent very fast Mode Decision Algorithm on H.264/AVC Baseline Profile**. International Conference on Image Processing (ICIP). [S.l.]: [s.n.]. 2006. p. 41-44.

SAMSUNG. Samsung Galaxy SIII, 2012. Disponivel em: <http://www.samsung.com/global/galaxys3/>. Acesso em: 12 ago. 2012.

SAMSUNG ELECTRONICS CO. LTDA. Samsung Exynos 4 Quad, 2012. Disponivel em: <http://www.samsung.com/global/business/semiconductor/minisite/Exynos/products4quad.html>. Acesso em: 12 ago. 2012.

SAPONARA, S.; FANUCCI, L. Data-adaptive motion estimation algorithm and VLSI architecture design for low-power video systems. **IEE Computers and Digital Techniques**, 151, n. 1, 2004. 51-59.

SAPONARA, S.; FANUCCI, L. Data-adaptive motion estimation algorithm and VLSI architecture design for low-power video systems, v. 151, n. 1, p. 51-59, 2004.

SHAFIQUE, M. et al. **Power-Aware Complexity-Scalable Multiview Video Coding for Mobile Devices**. 28th Picture Coding Symposium (PCS´10). [S.l.]: [s.n.]. 2010. p. 350-353.

SHAFIQUE, M.; BAUER, L.; HENKEL, J. **3-tier dynamically adaptive power-aware motion estimator for h.264/AVC video encoding**. International Symposium on Low Power Electronics and Design. [S.l.]: [s.n.]. 2008. p. 147-152.

SHAFIQUE, M.; L. BAUER, J.; HENKEL. **enBudget:** A Run-Time Adaptive Predictive Energy-Budgeting scheme for energy-aware Motion Estimation in H.264/MPEG-4 AVC video encoder. Design, Automation and Test in Europe (DATE). [S.l.]: [s.n.]. 2010.

SHAFIQUE, M.; MOLKENTHIN, B.; HENKEL, J. **An HVS-based Adaptive Computational Complexity Reduction Scheme for H.264/AVC Video Encoder using Prognostic Early Mode Exclusion**. IEEE Design, Automation and Test in Europe (DATE). [S.l.]: [s.n.]. 2010. p. 1713–1718.

SHARP. Lynx 3D SH-03C, 2011. Disponivel em: <http://www.sharp.co.jp/products/sh03c/index.html>. Acesso em: 20 ago. 2012.

SHEN, L. et al. **Fast Mode Decision for Multiview Video Coding**. International Conference on Image Processing (ICIP). [S.l.]: [s.n.]. 2009. p. 2953 - 2956.

SHEN, L. et al. Selective Disparity Estimation and Variable Size Motion Estimation Based on Motion Homogeneity for Multi-View Coding. **IEEE Transactions on Broadcasting**, 55, n. 4, December 2009. 761-766.

SHEN, L. et al. Early SKIP mode decision for MVC using inter-view correlation. **Signal Processing: Image Communication**, 25, 2010. 88–93.

SHEN, L. et al. View-Adaptive Motion Estimation and Disparity Estimation for Low Complexity Multiview Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, 20, n. 6, 2010. 925 - 930.

SHIM, H.; KYUNG, C.-M. Selective Search Area Reuse Algorithm for Low External Memory Access Motion Estimation. **IEEE Transactions on Circuits and Systems for Video Technology**, 19, n. 7, Julho 2009. 1044-1050.

SHIMPI, A. L. ARM's Mali-T658 GPU in 2013, Up to 10x Faster than Mali-400. **AnadTech**, 2011. Disponivel em: <http://www.anandtech.com/show/5077/arms-malit658-gpu-in-2013-up-to-10x-faster-than-mali400>. Acesso em: 20 ago. 2012.

SINGH, H. et al. Enhanced Leakage Reduction Techniques Using Intermediate Strength Power Gating. **IEEE Transactions on Very Large Scale Integration Systems**, 15, n. 11, November 2007. 1215 -1224.

SMOLIC, A. et al. Coding Algorithms for 3DTV - A Survey. **IEEE Transactions on Circuits and Systems for Video Technology**, 17, n. 11, Novembro 2007. 1606-1621.

SOCIAL TIMES. Social Times. **Cisco Predicts That 90% Of All Internet Traffic Will Be Video In The Next Three Years**, 2011. Disponivel em: <http://socialtimes.com/cisco-predicts-that-90-of-all-internet-traffic-will-be-video-in-the-next-three-years_b82819>. Acesso em: 05 set. 2012.

SOFTPEDIA. ARM Wants a Share Out of the Server and Desktop PC Market by 2015. **Softpedia**, 2010. Disponivel em: <http://news.softpedia.com/newsImage/ARM-Wants-a-Share-of-the-Server-and-Desktop-PC-Market-by-2015-5.png/>. Acesso em: 05 set. 2012.

SONY. HDR-TD10 - Full HD 3D Camcorder, 2011. Disponivel em: <http://www.sonystyle.com/webapp/wcs/stores/servlet/ProductDisplay?catalogId=10551&storeId=10151&langId=-1&productId=8198552921666294297>. Acesso em: 20 ago. 2012.

STELMACH, L. B.; TAM, J. W. Stereoscopicimage coding: Effect of disparate image-quality in left- and right-eyeviews. **Signal Processing: Image Communication**, 14, n. 1-2, November 1998. 111–117.

STELMACH, L. B.; TAM, W. J. Stereoscopic Image Coding: Effect of Disparate Image-Quality in Left- and Right-Eye Views, v. 14, p. 111-117, 1999.

STOYKOVA, E. et al. 3-D Time-Varying Scene Capture Technologies: A Survey. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, 17, n. 11, 2007. 1568-1586.

SU, Y.; VETRO, A.; SMOLIC, A. **Common Test Conditions for Multiview Video Coding - Doc. JVT-T207**. [S.l.]. 2006.

SULLIVAN, G. J. S.; OHM, J.-R. Recent developments in standardization of high efficiency video coding (HEVC). **SPIE Applications of Digital Image Processing XXXIII**, 7798, 2010.

SULLIVAN, G. J.; WIEGAND, T. Video Compression - From Concepts to the H.264/AVC Standard. **Proceedings of the IEEE**, 93, n. 1, 2005. 18 - 31.

SULLIVAN, G.; WIEGAND, T. Rate-Distortion Optimizatoin for Video Compression. **IEEE Signal Processing Magazine**, 15, 1998. 74-90.

SYNOPSYS, INC. IBM - 65NM, 2012. Disponivel em: <http://www.synopsys.com/dw/emllselector.php?f=IBM&g=65>. Acesso em: 24 jun. 2012.

TAN, T.; SULLIVAN, G.; WEDI, T. **Recommended Simulation Conditions for Coding Efficiency Experiments - VCEG-AA10**. Nice. 2005.

TANG, X.-L.; DAI, S.-K.; CAI, C.-H. **An analysis of TZSearch algorithm in JMVC**. International Conference on Green Circuits and Systems (ICGCS). [S.l.]: [s.n.]. 2010. p. 516 -520.

TANIMOTO, M. **FTV (Free Viewpoint Television) Creating Ray-Based Image Engineering**. International Conference on Image Processing. [S.l.]: [s.n.]. 2005. p. 25-28.

TATJEWSKI, P. Supervisory predictive control and on-line set-point optimization. **Journal International Journal of Applied Mathematics and Computer Science**, 20, n. 3, September 2010. 483-495.

TECH, G. et al. **Final report on coding algorithms for mobile 3DTV**. [S.l.]. 2010. http://sp.cs.tut.fi/mobile3dtv/results/tech/D2.6_Mobile3DTV_v1.0.pdf. Acesso em: 20 jul. 2012.

TEXAS INSTRUMENTS INC. OMAP™ Mobile Processors : OMAP™ 5 Platform, 2012. Disponivel em: <http://www.ti.com/general/docs/wtbu/wtbuproductcontent.tsp?templateId=6123&navigationId=12862&contentId=101230>. Acesso em: 20 ago. 2012.

THE DIGITAL ENTERTAINMENT GROUP. **3D White Paper**. [S.l.]. 2009.

TIAN, L. et al. **Analysis of quadratic R-D model in H.264/AVC video coding**. 17th IEEE International Conference on Image Processing (ICIP). [S.l.]: [s.n.]. 2010. p. 2853-2856.

TOURAPIS, A. M. **Enhanced predictive zonal search for single and multiple frame motion estimation**. Visual Communication and Image Processing Conference (VCIP). [S.l.]: [s.n.]. 2002.

TSAI, C.-Y. et al. **Low Power Cache Algorithm and Architecture Design for Fast Motion Estimation in H.264/AVC Encoder System**. International Conference on Acoustics Speech and Signal Processing (ICASSP). [S.l.]: [s.n.]. 2007. p. II-97 - II-100.

TSUNG, P.-K. et al. **Cache-Based Integer Motion/Disparity Estimation for Quad-HD H.264/AVC and HD Multiview Video Coding**. International Conference on Acoustics, Speech and Signal Processing. Taipei: IEEE. 2009. p. 2013-2016.

TUAN, T.; KAO, S.; TRIMBERGER, S. **A 90nm Low-Power FPGA for Battery-Powered Applications**. International symposium on Field programmable gate arrays (FPGA). [S.l.]: [s.n.]. 2006. p. 3-11.

VIMEO. Vimeo 3D, 2012. Disponivel em: <http://vimeo.com/channels/stereoscopy>.

VIZZOTTO, B. B. et al. **A Model Predictive Controller for Frame-Level Rate Control in Multiview Video Coding**. IEEE International Conference on Multimedia & Expo (ICME´12). [S.l.]: [s.n.]. 2012. p. 485-490.

WANG, S.-H.; TAI, S.-H.; CHIANG, T. A Low-Power and Bandwidth-Efficient Motion Estimation IP Core Design Using Binary Search. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, 19, n. 5, 2009. 760-765.

WANG, X. et al. **Fast Mode Decision for H.264 Video Encoder based on MB Motion Characteristic**. International Conference on Multimedia and Expo (ICME). [S.l.]: [s.n.]. 2007. p. 372–375.

WEI, Z.; NGAN, K. N.; LI, H. An efficient intra-mode selection algorithm for H.264 based on edge classification and rate-distortion estimation. **Signal Processing: Image Communication**, 23, n. 9, October 2008. 699-710.

WELCH, G. et al. **Remote 3D medical consultation**. [S.l.]: [s.n.]. 2005. p. 1026-1033.

WIEGAND, T. et al. Overview of the H.264/AVC Video Coding Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, 13, n. 7, Julho 2003. 560-576.

WILLNER, K. et al. **Mobile 3D Video Using MVC and N800 Internet Tablet**. 3DTV Conference. [S.l.]: [s.n.]. 2008.

WOO, J.-H. et al. A 195 mW/152 mW Mobile Multimedia SoC With Fully Programmable 3-D Graphics and MPEG4/H.264/JPEG. **IEEE Journal of Solid-State Circuits**, 43, n. 9, September 2008. 2047 -2056.

WU, C.-Y.; SU, P.-C. **A Region of Interest Rate-Control Scheme for Encoding Traffic Surveillance Videos**. International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP). [S.l.]: [s.n.]. 2009. p. 194 -197.

WU, D. et al. **Block Inter Mode Decision for Fast Encoding of H.264**. International Conference on Acoustics Speech and Signal Processing (ICASSP). [S.l.]: [s.n.]. 2004. p. iii - 181-184.

XILINX, INC. ISE Design Suite, 2012. Disponivel em: <http://www.xilinx.com/products/design-tools/ise-design-suite/index.htm>. Acesso em: 14 ago. 2012.

XU, L. et al. **Priority pyramid based bit allocation for multiview video coding**. IEEE Visual Communications and Image Processing (VCIP). [S.l.]: [s.n.]. 2011. p. 1 -4.

XU, X.; HE, Y. **Fast Disparity Motion Estimation in MVC Based on Range Prediction**. IEEE International Conference on Image Processing, 2008, San Diego. Piscataway: IEEE. 2008. p. 2000-2003.

YAMAOKA, M. A. S. Y. A. M. N. A. S. Y. et al. **A 300MHz 25 mu;A/Mb leakage on-chip SRAM module featuring process-variation immunity and low-leakage-active mode for mobile-phone application processor**. IEEE International Solid-State Circuits Conference. [S.l.]: [s.n.]. 2004. p. 494-542.

YAN, T. et al. **Frame-layer rate control algorithm for multi-view video coding**. ACM/SIGEVO Summit on Genetic and Evolutionary Computation. [S.l.]: [s.n.]. 2009. p. 1025-1028.

YAN, T. et al. **Rate Control Algorithm for Multi-View Video Coding Based on Correlation Analysis**. Symposium on Photonics and Optoelectronics. [S.l.]: [s.n.]. 2009. p. 1 -4.

YANG, J. **Multiview video coding based on rectified epipolar lines**. International Conference on Information, Communication and Signal Processing. [S.l.]: [s.n.]. 2009. p. 1-5.

YANG, S.; WOLF, W.; N.VIJAYKRISHNAN. Power and performance analysis of motion estimation based on hardware and software realizations. **IEEE Transactions on Computers**, 54, n. 6, 2005. 714-726.

YOUTUBE. YouTube - Broadcast Yourself, 2011. Disponivel em: <http://www.youtube.com/>. Acesso em: 20 ago. 2012.

YOUTUBE 3D. YouTube - 3D Channel, 2011. Disponivel em: <http://www.youtube.com/user/3D>. Acesso em: 20 ago. 2012.

YU, A. C. **Efficient Block-Size Selection Algorithm for Inter-Frame Coding in H.264/MPEG-4 AVC**. Internacional Conference on Acoustic, Speech and Signal Processing (ICASSP). [S.l.]: [s.n.]. 2004. p. III169–III172.

ZATT, B. et al. **Memory Hierarchy Targeting Bi-Predictive Motion Compensation for H.264/AVC Decoder**. IEEE Computer Society Annual Symposium on VLSI (ISVLSI). [S.l.]: [s.n.]. 2007. p. 445 - 446.

ZATT, B. et al. **A Multi-Level Dynamic Complexity Reduction Scheme for Multiview Video Coding using 3D-Neighborhood Correlation**. Design, Automation and Test in Europe (DATE). [S.l.]: [s.n.]. 2010.

ZATT, B. et al. **An Adaptive Early Skip Mode Decision Scheme for Multiview Video Coding**. Picture Coding Symposium (PCS). [S.l.]: [s.n.]. 2010. p. 42-45.

ZATT, B. et al. **A Low-Power Memory Architecture with Application-Aware Power Management for Motion & Disparity Estimation in Multiview Video Coding**. IEEE/ACM 29th International Conference on Computer-Aided Design (ICCAD´11). [S.l.]: [s.n.]. 2011. p. 40-47.

ZATT, B. et al. **A Multi-Level Dynamic Complexity Reduction Scheme for Multiview Video Coding**. IEEE 18th International Conference on Image Processing (ICIP´11). [S.l.]: [s.n.]. 2011. p. 761-764.

ZATT, B. et al. **Multi-Level Pipelined Parallel Hardware Architecture for High Throughput Motion and Disparity Estimation in Multiview Video Coding**. IEEE/ACM 14th Design Automation and Test in Europe Conference (DATE'11). [S.l.]: [s.n.]. 2011. p. 1448-1453.

ZATT, B. et al. **Run-Time Adaptive Energy-Aware Motion and Disparity Estimation in Multiview Video Coding**. ACM/IEEE/EDA 48th Design Automation Conference (DAC´11). [S.l.]: [s.n.]. 2011. p. 1026-1031.

ZENG, H.; MA, K.-K.; CAI, C. Fast Mode Decision for Multiview Video Coding Using Mode Correlation. **IEEE Transactions on Circuits and Systems for Video Technology**, 21, n. 11, November 2011. 1659-1666.

ZHANG, K. et al. SRAM design on 65-nm CMOS technology with dynamic sleep transistor for leakage reduction. **IEEE Journal of Solid-State Circuits**, April 2005. 895-901.

ZHANG, Y. et al. ASIP Approach for Multimedia Applications Based on a Scalable VLIW DSP Architecture. **Tsinghua Science and Technology**, 14, n. 1, February 2009. 126-132.

ZHOU, Y. et al. PID-Based Bit Allocation Strategy for H.264/AVC Rate Control. **IEEE Transactions on Circuits and Systems II: Express Briefs**, 58, n. 3, March 2011. 184 - 188.

ZHU, H.; LUICAN, I. I.; BALASA, F. **Memory Size Computation for Multimedia Processing Applications**. Asia and South Pacific Conference on Design Automation. [S.l.]: [s.n.]. 2006. p. 802-807.

ZONE, R. **Stereoscopic Cinema and the Origins of 3-D Film, 1838-1952**. [S.l.]: [s.n.], 2007. ISBN 0813124611.

APPENDIX A <JMVC SIMULATION ENVIRONMENT>

The JVT (Joint Video Team), formed from the cooperation between the ITU-T Study Group 16 (VCEG) and ISO/IEC Motion Picture Experts group (MPEG), responsible for the standardization of the H.264, SVC (Scalable Video Coding) and MVC provides software models used for algorithms experimentation and for standards prove of concept. The JMVC (Joint Model for MVC) (JVT, 2009), currently on version 8.5, is the reference software available for experimentation on the MVC standard. Along this work the JMVC software, coded using C++, was used and modified to implement the proposed algorithms. Initially, the version 6.0 was used followed by an upgraded to version 8.5. Considering the length and complexity of the software, a high-level overview of the interaction between the main encoder classes is presented here. Afterwards are shown the classes modified to enable our algorithms experimentation. For in deep details of the classes structure refer to JMVC documentation (ERDAYANDı, 2009).

B. 1 – JMVC Encoder Overview

The JMVC classes are hierarchically structured as shown in Figure A.1 (TECH, MÜLLER, *et al.*, 2010). The JMVC encodes each view at a time requiring as many calls as number of view to be encoded. The reference views are stored in temporary files. The class *H264AVCEncoderTest* represents the top encoder entity, it initializes the encoder, call the *CreateH264AVCEncoder* class to initialize the other coding classes. At this level, the *PicEncoder* is initialized and the frame-level loop is controlled. The *PicEncoder* loops over the slices inside each frame and reset the RDcost. The slice encoder controls the MB-level loop and set the reference frames for each slices. For MB encoding there are two main classes, the *MbEncoder* and the *MBCoder*. *MbEncoder* encapsulates all the prediction, transforms and entropy steps. It implements the mode decision by looping over and encoding all possible coding modes (in case of RDO-MD) and determining the minimum RDCost. At this point no MB coding data is written to the bitstream. Once the best mode is selected, the *SliceEncoder* calls the *MbCoder* to write the MB-level side information and residues to the bitstream.

Figure A.2 (TECH, MÜLLER, *et al.*, 2010) depicts the hierarchical call graph of methods inside the mode decision process implemented in *MbEncoder* class. Firstly, the SKIP and Direct modes are evaluated, along this thesis these modes are jointly referred as SKIP MBs. In the following, all inter-prediction block sizes are evaluated. For each partition size a call to the method *MotionEstimation::estimateBlockWithStart* (see Figure A.3 discussion) is performed. The same happens for the sub-partitions in case of 8x8 partitioning. *EstimateMb8x8Frext* is only called in case the FRExt flag in set. Finally, the intra-frame coding modes including PCM, intra4x4, intra8x8 (FRExt only) and intra16x16 are called. The *Estimate<mode>* methods call the complete coding loop for that specific mode including prediction, transforms, quantization, entropy encoding and reconstruction. It allows a precise definition of the minimum RDCost (λ) and an optimal best mode selection at the cost of elevated coding complexity. The *MbCoder* is called to entropy encode the best mode and write the data into the bitstream output buffer.

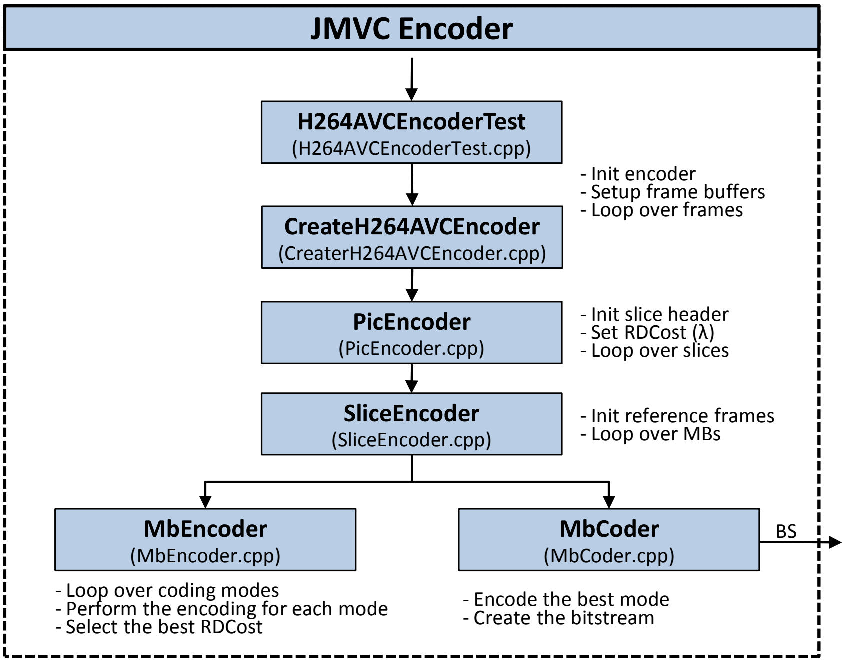


Figure A.1: JMVC Encoder High-Level Diagram

The motion and disparity estimation search itself is defined in the method *estimateBlockWithStart* and is composed of three basic steps. The ME/DE dataflow is represented by the arrows in Figure A.3. Once the *estimateBlockWithStart* is called, for instance in *EstimateMb16x16*, the search runs for each reference frame list (List 0 and List 1) and for an interactive B search mode that exploits both lists in an interactive fashion (in case the interactive B is active). At the software perspective, there is no distinction between ME and DE. List 0 and List 1 store both temporal and disparity reference frames. The search for a given reference frame firstly finds the best candidate block among the integer pixels (ME/DE Full Pel) and then refines the result considering half and quarter pixels (ME/DE Sub Pel). The search pattern depends on the search algorithm, JMVC implements TZ Search, Full Search, Spiral Search and Log Search. The goal is to find the candidate block that minimizes the Motion Cost (λ_{Motion}) in terms of SAD, SAD-YUV (considering chroma channels), SATD or SSE according to user defined coding parameters. The position of the best matching candidate block position defines the motion or disparity vector.

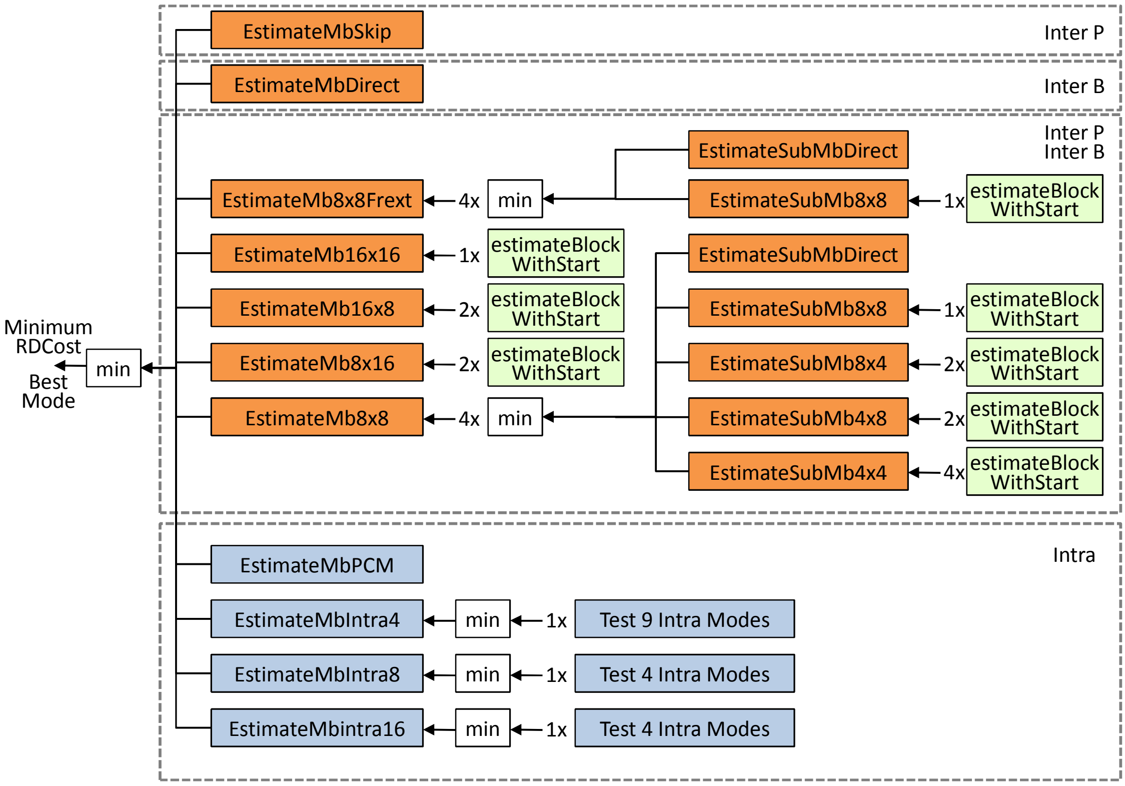


Figure A.2: Mode Decision hierarchy in JMVC

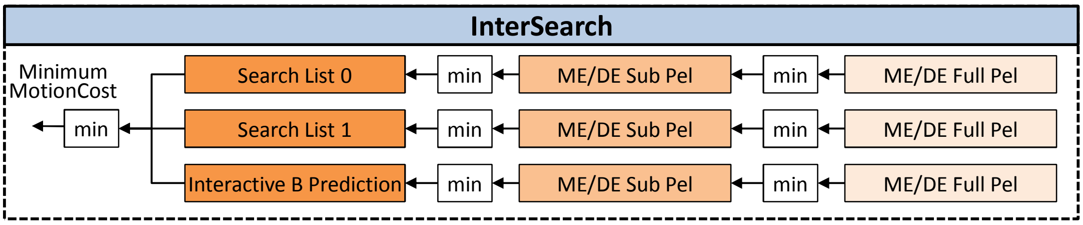


Figure A.3: Inter-frame search in JMVC

B. 2 – Modifications to the JMVC Encoder

B.2.1 – JMVC Encoder Tracing

In order to generate the statistics used for coding modes and motion/disparity vector some modifications were done in the original JMVC code. Te point selected for this tracing is inside the entropy encoder to guarantee that the extracted data is the same actually encoded and transmitted. The entropy encoder is declared as the virtual class *MbSymbolWriteIf* but the actual implementation is in *CabacWriter* and *UvlcWriter*, depending on the entropy encoder selected in the configuration file. The methods monitored are *skipFlag* that encodes the SKIP (and Direct) coded MBs and *mbMode* that encodes all other modes. Note, MB coding mode codes (*uiMbMode*) varies with the slice type as defined in tables 7-11, 7-12, 7-12, 7-13 and 7-14 of the MVC standard (JVT, 2008).

B.2.2 – Communication Channels in JMVC

Multiple algorithms proposed in this thesis employ the information from the 3D-Neighbohood. For that, there is a need to build communication channels between neighboring MBs in the special, temporal and disparity domains. In other words, it is required an infrastructure to send and receive data at MB level, at frame level (in same view) and at view level (frames in different views). Therefore, a hierarchical communication infrastructure was designed and implemented. Figure A.4 presents graphically the modified classes along with the new member data structures and communication methods.

The *MbDataAccess* already provides direct access to the left and upper neighbors (A, B, C and D in Figure A.4). This access was extended to the right and bottom neighbors (A*, B*, C* and D*) enabling access to data from all spatial neighboring MBs. For temporal neighboring MBs access the current MB data is sent to *SliceHeader* (using *Send*() methods) where a 2D array stores the information from the MBs belonging to the current slice. Once the slice is completely processed, the 2D array is sent to the *PicEncoder* class. *PicEncoder* maintains the data for the whole current GOP. For reading the data communication channel writes the requested data from *PicEncoder* to *SliceHeader* and finally to *MbDataAccess* (using *Receive*() methods). As far as the views a processed in distinct encoder calls there is a need to use external temporary files to transmit the disparity neighboring information. The current MB data is written in these files from *MbDataAccess* while the data from previous views is read in *PicEncoder*, as shown in Figure A.4.

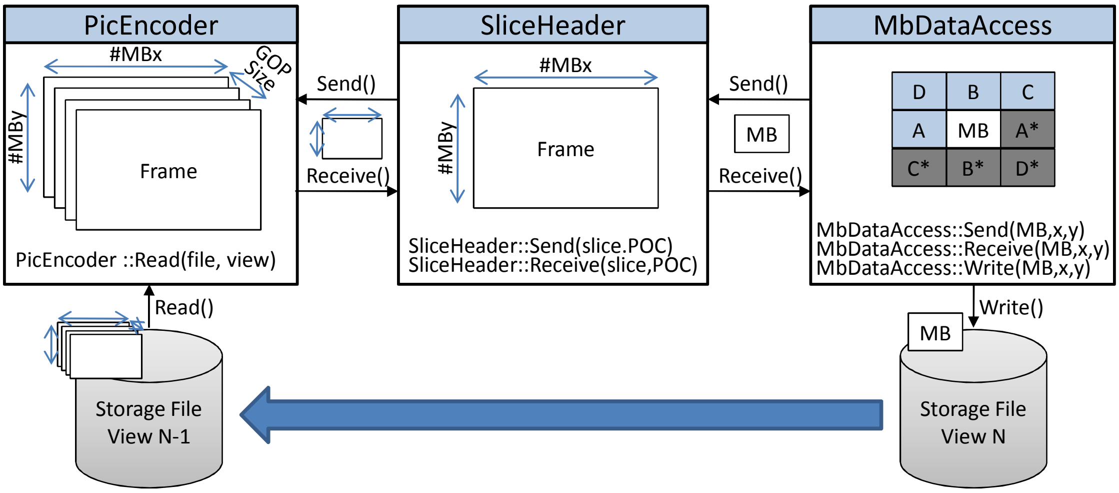

PicEncoder
#MBx
GOP Size
#MBy
Frame
PicEncoder ::Read(file, view)
Send()
Receive()
SliceHeader
#MBx
#MBy
Frame
SliceHeader::Send(slice.POC)
SliceHeader::Receive(slice,POC)
Send()
MB
Receive()
MbDataAccess
D
B
C
A
MB
A*
C*
B*
D*
MbDataAccess::Send(MB,x,y)
MbDataAccess::Receive(MB,x,y)
MbDataAccess::Write(MB,x,y)
Read()
Write()
MB
Storage File View N-1
Storage File View N

Figure A.4: Communication in JMVC

B.2.3 – Mode Decision Modification in JMVC

The mode decision is programmed in a very simple becoming easy to find and modify. MD is handle in *MbEncoder::encodeMacroblock*. To find the exact point search for the *xEstimateMb* methods responsible for calling the modes evaluation. Before this point are implemented the 3D-Neighborhood communication calls and the calculation required to take the fast decisions.

B.2.4 – ME/DE Modification in JMVC

The modification for fast ME/DE are inserted in two distinct classes. For modifications at higher level such as avoiding interactive B search, search direction and reference frames the modification are done in the *MbEncoder* by modifying the *xEstimateMb* methods. If the modifications are in the search step itself, *MotionEstimation* class is the right point for modification. *estimateBlockWithStart* method is responsible for fetching the image data, prediction SKIP vectors and calling the search methods (*xPelBlockSearch*, *xPelSpiralSearch*, *xPelLogSearch* and *xTZSearch*). By modifying these methods it is possible to reach low level modifications on the ME/DE search.

B.2.4 – Rate Control Modification in JMVC

The JMVC does not implement any rate control algorithms. Therefore, to implement the Hierarchical Rate Control (HRC) scheme one new class is created, the *RateControl*. Three files are used to better partition the RC hierarchy. File *RateCtlCore.cpp* describes the behavior of the whole HRC while *RateCtlMPC.cpp* and *RateCtlUB.cpp* are responsible for the calculations relative the MPC and MDP controllers. *RateCtl.h* file is used to define the MPC and MDP actuation parameters. The QP history is read from *CodingParameter* class and the generated bitrate is accessed via *BitWriteBuffer* and *BitCounter*. The QP defined for the next frames or BU are sent back to *CodingParameter*. Additional modifications were required in files MbCoder.cpp, CodingParameter.cpp, RateDistortion.cpp, ControlMngH264AVCEncoder.cpp, Multiview.cpp, and ControlMngH264AVCEncoder.h.

APPENDIX B <MEMORY ACCESS ANALYZER TOOL>

The MVC Viewer software is used as part of this work to plot and analyze the memory accesses that are required by the Motion and Disparity Estimation (ME/DE). The goal of this tool is to help the researchers in their projects in the visual and statistical analysis of the communication between the multiview video encoder and the reference samples memory. It provides a set of final statistics and several plots using the original input video.

The MVC Viewer was designed to be adapted to different encoder parameters. In a configuration file, the user should specify: (a) the number of views, (b) the GOP size, (c) the video resolution , (d) the original YUV video files path and, finally, (e) the memory tracing input files path. The tracing file is an intermediated way to communicate the video encoder output, like JM or x264, with the MVC Viewer tool. In this file, all memory accesses performed by ME/DE are listed.

This tool runs over the JVM (Java Virtual Machine) and provides a simple interface to the analysis. Figure B.1 presents the overview of the MVC Viewer main screen. The main parts are:

1. Encoding parameters: GOP Size, number of coded frames, number of coded views and video resolution (directly defined in the configuration files).
2. Tracing files path where all accessed regions of reference frames are listed.
3. Original YUV videos.
4. Program mode selection: the MVC Viewer has mainly two possible analysis tool: (a) current macroblocks based analysis and (b) reference frame based analysis.
5. Listbox with all memory access that will be plotted in the output.

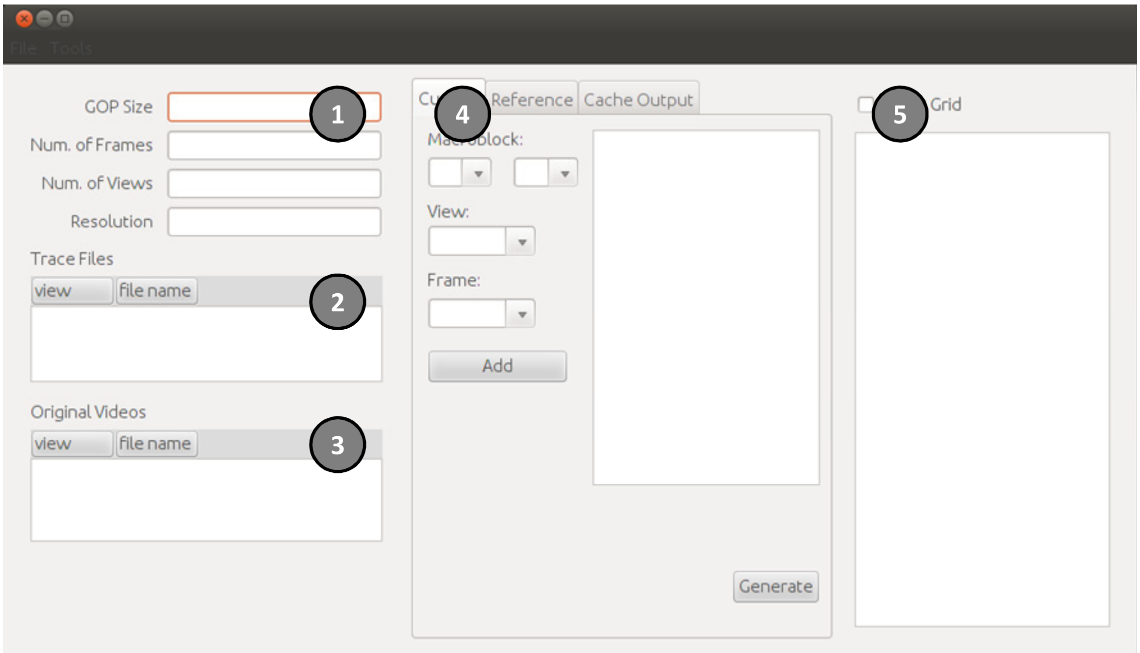


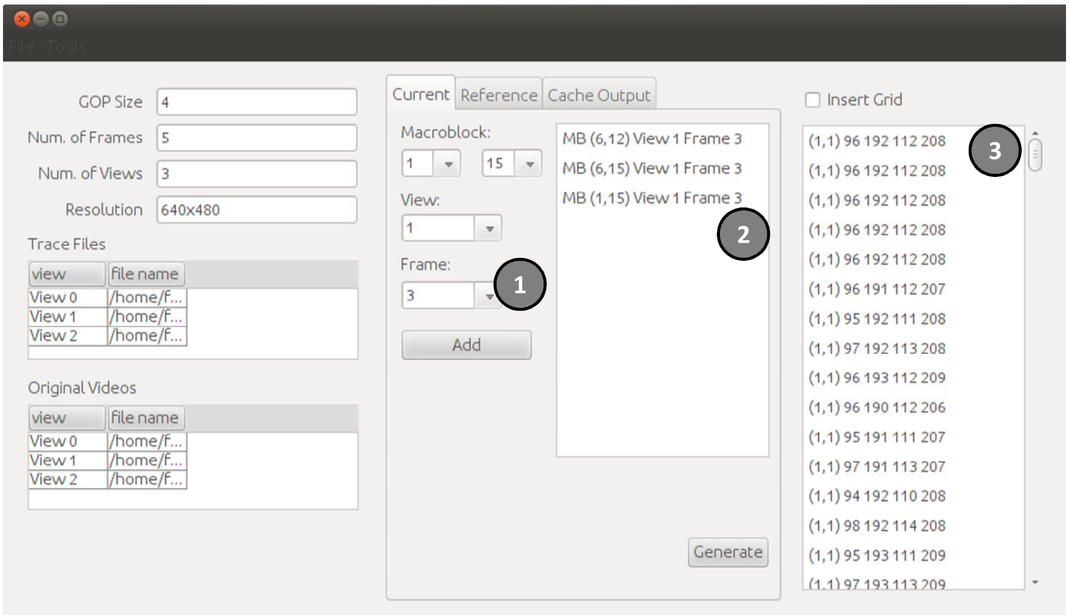
Figure B.1: MVC Viewer main screen

The two analyses that are allowed by the MVC Viewer tool will be explained in the next sections.

B. 1 - Current Macroblock-Based Analysis

In this analysis, the goal is to trace all accessed reference frame samples when the ME/DE is performed for one or more current macroblocks. The MVC massively uses multiple reference frames, then the MVC Viewer will generate several plots that will determine the accessed regions for each reference frame (temporal and disparity neighbors). The Figure B.2 shows a MVC Viewer screenshot when it is running this analysis. The main parts are:

1. Selection of the target macroblocks that will be traced.
2. List of all selected macroblocks.
3. List of all memory access caused by the ME/DE for the selected macroblocks.



GOP Size: 4

Num. of Frames: 5

Num. of Views: 3

Resolution: 640x480

Trace Files

view	file name
View 0	/home/f...
View 1	/home/f...
View 2	/home/f...

Original Videos

view	file name
View 0	/home/f...
View 1	/home/f...
View 2	/home/f...

Macroblock: 1 15

View: 1

Frame: 3

```
MB (6,12) View 1 Frame 3
MB (6,15) View 1 Frame 3
MB (1,15) View 1 Frame 3
```

□ Insert Grid

```
(1,1) 96 192 112 208
(1,1) 96 192 112 208
(1,1) 96 192 112 208
(1,1) 96 192 112 208
(1,1) 96 192 112 208
(1,1) 96 191 112 207
(1,1) 95 192 111 208
(1,1) 97 192 113 208
(1,1) 96 193 112 209
(1,1) 96 190 112 206
(1,1) 95 191 111 207
(1,1) 97 191 113 207
(1,1) 94 192 110 208
(1,1) 98 192 114 208
(1,1) 95 193 111 209
(1,1) 97 193 113 209
```

Figure B.2: Current Macroblock Based analysis screenshot

The Figure B.3 presents an output example for one macroblock that reflects in samples accesses in the four directions: past and future temporal reference frames, and right and left disparity reference frames.

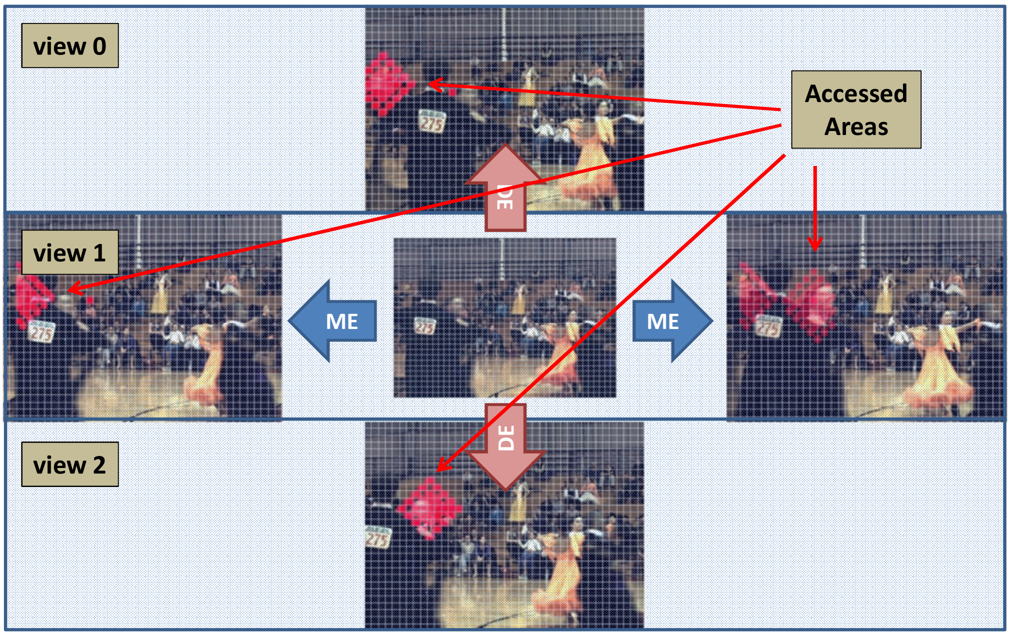
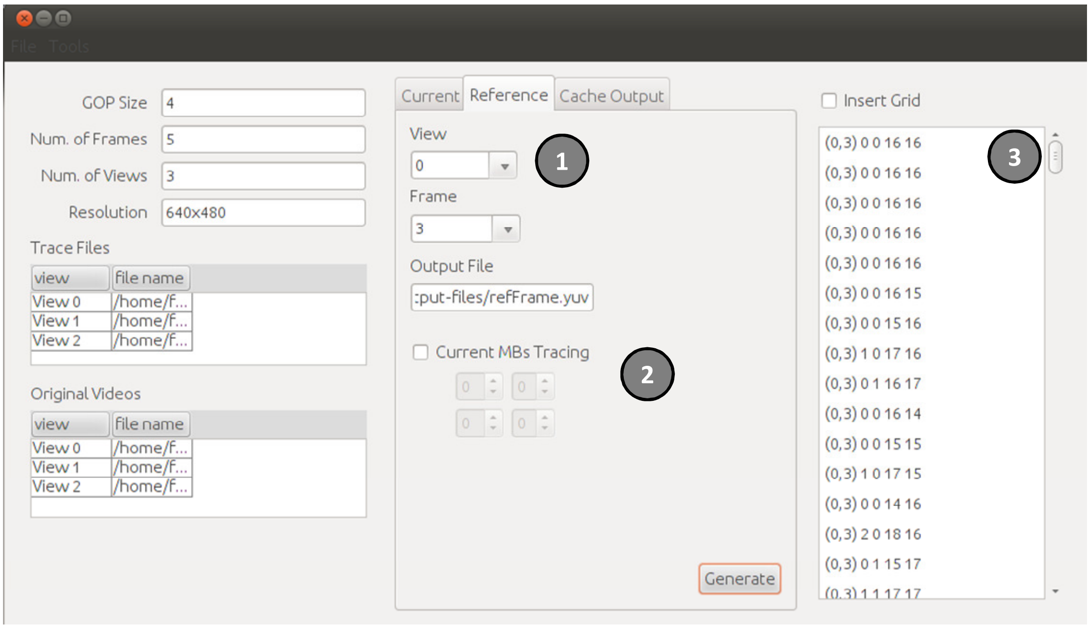


Figure B.3: Output Example: four prediction directions and their respective accessed areas.

B. 2 - Search Window-Based Analysis

This analysis selects one specific frame and traces all accesses performed by the ME/DE when the selected frame is used as reference. This way, it is possible to determine the most accessed regions of the frame. The knowledge about this behavior is important to define strategies to save memory bandwidth. Figure B.4 presents the MVC Viewer during this analysis, where the main parts are:

1. Reference frame selection: the user must define the frame identification (the view and frame positions) to be traced.
2. Current MBs Tracing option: the user has the possibility to delimit an area inside the reference frame to discover which are the current blocks processed by the ME/DE that cause the accesses.
3. List of all memory access caused by the ME/DE in the selected reference frame.



GOP Size: 4

Num. of Frames: 5

Num. of Views: 3

Resolution: 640x480

Trace Files

view	file name
View 0	/home/f...
View 1	/home/f...
View 2	/home/f...

Original Videos

view	file name
View 0	/home/f...
View 1	/home/f...
View 2	/home/f...

View: 0

Frame: 3

Output File: :put-files/refFrame.yuv

□ Current MBs Tracing

0 0
0 0

□ Insert Grid

```
(0,3) 0 0 16 16
(0,3) 0 0 16 16
(0,3) 0 0 16 16
(0,3) 0 0 16 16
(0,3) 0 0 16 16
(0,3) 0 0 16 15
(0,3) 0 0 15 16
(0,3) 1 0 17 16
(0,3) 0 1 16 17
(0,3) 0 0 16 14
(0,3) 0 0 15 15
(0,3) 1 0 17 15
(0,3) 0 0 14 16
(0,3) 2 0 18 16
(0,3) 0 1 15 17
(0,3) 1 1 17 17
```

Figure B.4: Current Macroblock Based analysis screenshot.

The Figure B.5 presents two different examples of the Reference Frame Based Analysis considering two search algorithms: (a) Full Search and TZ Search.

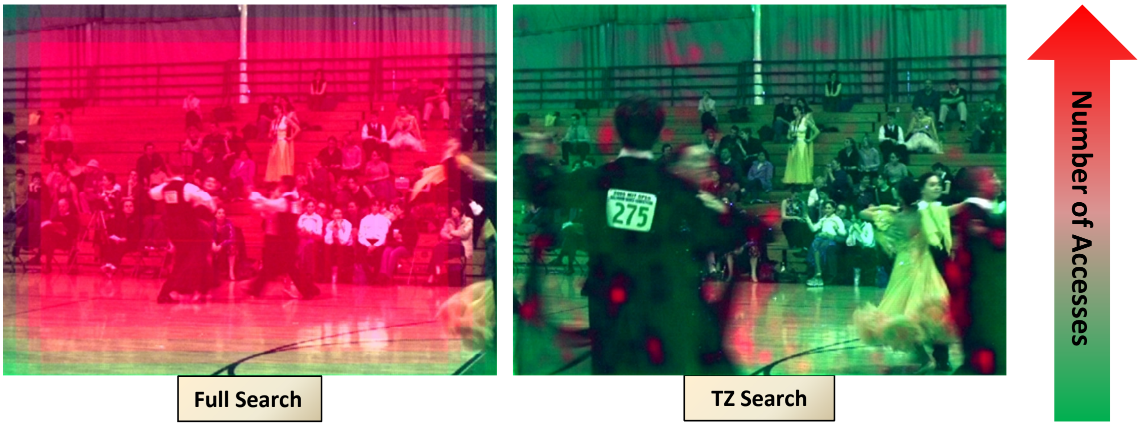
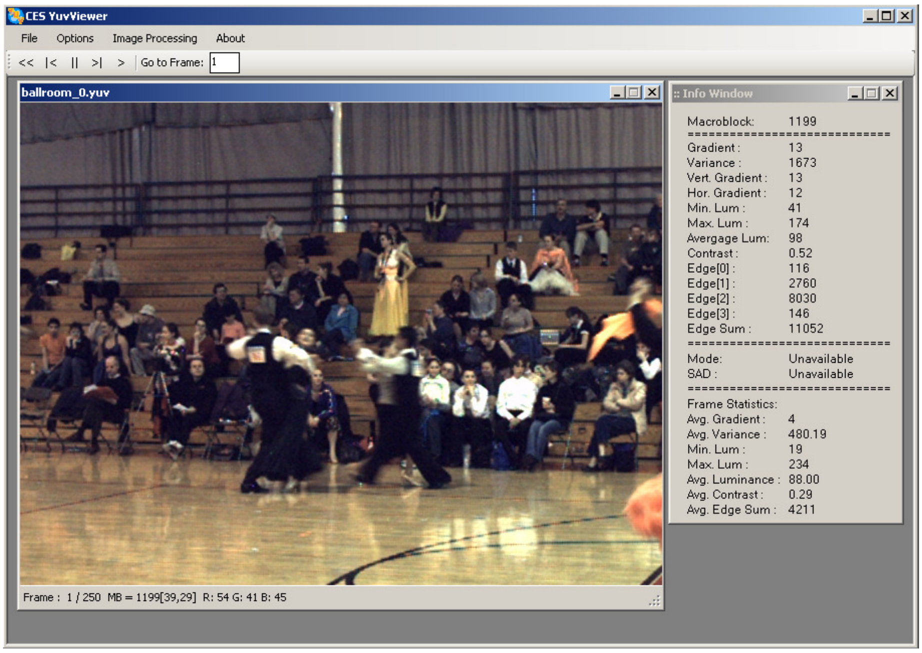


Figure B.5: Output Exapmle: reference frame access index considering two block matching algorithms: Full Search and TZ Search.

The Full Search has a regular access pattern where all samples inside the Search Window are fetched. On the other hand, the TZ Search has a heuristic behavior and the access index varies in according with the video properties (low/high motion/disparity). These two different cases are represented in the plots of the Figure B.5.

APPENDIX C <CES VIDEO ANALYZER TOOL>

The CES Video Analyzer tool was developed in house targeting the displaying and analyzing of video properties. It was described in C# programming language and features the Graphic User Interface presented in Figure C.1. The goal of the original tool is to support the decision making during novel coding algorithms design. The tool support diverse displaying modes including luminance only mode and applying MB grids. Also, the CES Video Analyzer implements image filters such as Sobel, Laplace, Kirsch and Prewitt filters besides of luminance, gradient and variance maps. An additional information window summarizes all image properties. Figure C.2 exemplifies the tool features presenting the original frame with the MB grid, the Sobel filtered image and the variance map.



```
Macroblock:       1199
=============================
Gradient :        13
Variance :        1673
Vert. Gradient :  13
Hor. Gradient :   12
Min. Lum :        41
Max. Lum :        174
Avergage Lum:     98
Contrast :        0.52
Edge[0] :         116
Edge[1] :         2760
Edge[2] :         8030
Edge[3] :         146
Edge Sum :        11052
=============================
Mode:             Unavailable
SAD :             Unavailable
=============================
Frame Statistics:
Avg. Gradient :   4
Avg. Variance :   480.19
Min. Lum :        19
Max. Lum :        234
Avg. Luminance :  88.00
Avg. Contrast :   0.29
Avg. Edge Sum :   4211
```

```
Frame : 1 / 250  MB = 1199[39,29]  R: 54 G: 41 B: 45
```

Figure C.1: CES Video Analyzer User Interface

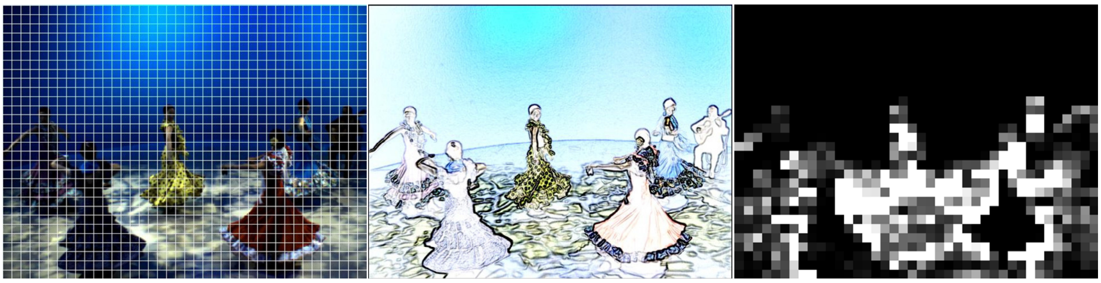


Figure C.2: CES Video Analyzer Features

To facilitate the development of the algorithms proposed in this volume, the CES Video Analyzer was extended to support and provide better visualization for MVC videos. Figure C.3 shows the visualization of a frame differentiating SKIP, inter and intra MBs. In Figure C.4 all MBs, including SKIPs, are classified in disparity estimation or motion estimation for different time instants.

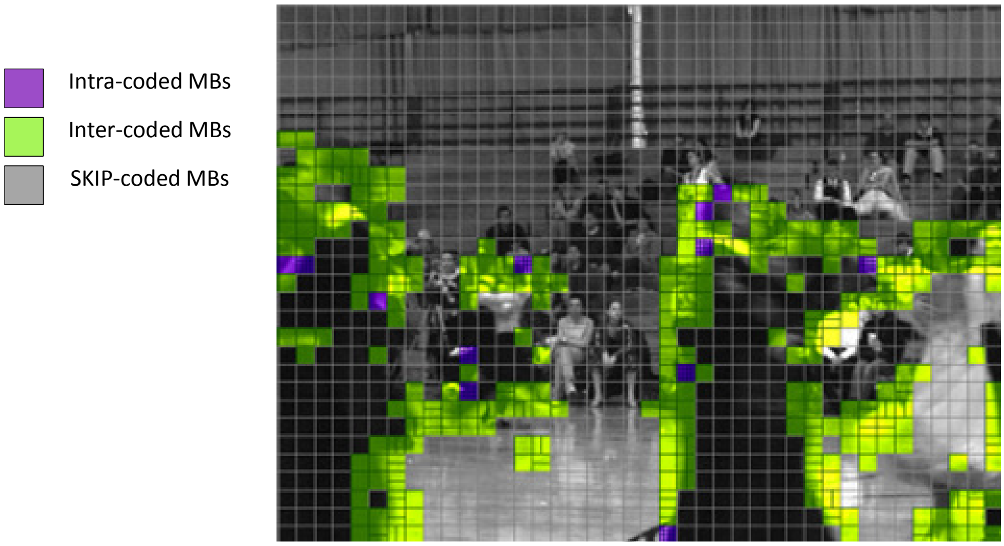


Figure C.3: Coding mode analysis using CES Video Analyzer

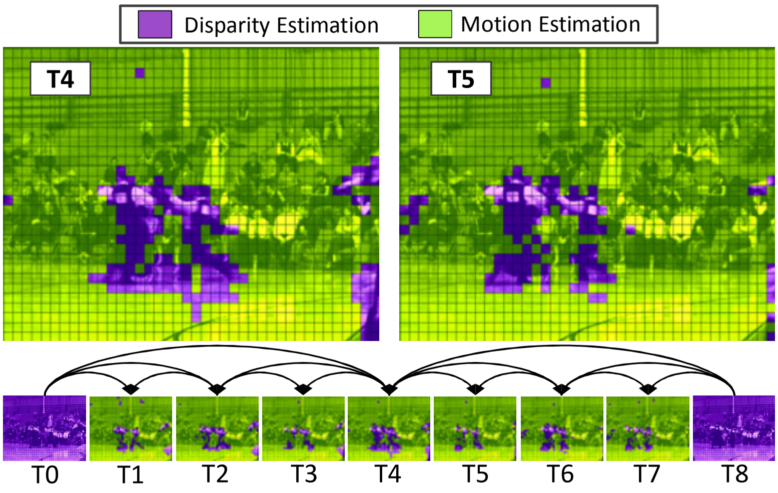


Figure C.4: ME/DE analysis using CES Video Analyzer

APPENDIX D <EXTENDED ABSTRACT: PORTUGUESE>

D.1 - Introdução

A busca do Mercado consumidor por tecnologias de multimídia imersivas aliada ao interesse da indústria de impulsionar o mercado de entretenimento levou a popularização dos vídeos e aplicações 3D além de dispositivos que processam tais vídeos. Embora o primeiro dispositivo 3D tenha sido desenvolvido em 1833 e a primeira exibição 3D tenha acontecido em 1915 (ZONE, 2007), esse formato apenas se tornou amplamente conhecido nos anos 1980 através da tecnologia IMAX (IMAX, 2012). A explosão de popularidade dos vídeos 3D, no entanto, aconteceu no final dos anos 2000 por meio da popularização dos cinemas 3D seguidos de televisores voltados par cinema em casa. Para melhor quantificar essa popularização, mais de 10% dos televisores vendidos nos EUA em 2011 eram capazes de reproduzir vídeos 3D (RESEARCH AND MARKETS, 2010). O último nicho a ser afetado pela popularização 3D é também responsável pelo maior crescimento da indústria de circuitos integrados depois dos computadores pessoais: os sistemas móveis embarcados. A venda de smartphones, tablets, càmeras portáteis, e outros dispositivos móveis já supera a venda de computadores pessoais (KAY, 2011) (IC INSIGHTS, 2012). Por exemplo, a expectativa de vendas para smartphones para 2013 é de 650 milhões enquanto computadores devem atingir a marca de 430 milhões (GASSÉE, 2010) de unidades vendidas no mesmo ano. Tal popularização leva a um grande aumento no volume de conteúdo 3D sendo gerado, codificado, armazenado, transmitido e exibido diariamente. De acordo com a CISCO (CISCO, 2012), videos digitais já representam 51% do tráfego da internet devendo atingir o patamar de 90% em 2014 (SOCIAL TIMES, 2011). Considere-se também o aumento no tráfego gerado por dispositivos móveis na internet previsto para crescer de 0,6 Exabytes/mês em 2011 para 10,8 Exabytes/mês em 2016 (CISCO, 2012).

Para cobrir a lacuna entre geração de conteúdo 3D e as capacidades de transmissão e armazenamento, é necessário codificar os vídeos 3D de forma eficiente reduzindo o volume de dados necessário para sua representação. O padrão de codificação para vídeos de múltiplas vistas, Multiview Video Coding (MVC), criado como uma extensão do H.264/AVC, representa a tecnologia mais avançada em codificação 3D. Baseado no paradigma de múltiplas vistas, assim como a grande maioria do conteúdo 3D atual, o MVC é capaz de reduzir o volume de dados de um vídeo em 20%-50% comparado ao H.264/AVC. Esse aumento de eficiência vem ao custo de aumento em complexidade computacional e consumo de energia na etapa de codificação. O aumento de consumo energético se deve ao grande número de unidades funcionais necessárias para processar tal volume de dados e do intenso tráfego de memória. Em um cenário dominado por sistemas móveis, o aumento no consumo de energia vai de encontro às limitações impostas pelas baterias de tais sistemas. Esse conflito de interesses entre eficiência de codificação e restrições energéticas carrega o principal desafio relacionado a codificação de vídeos 3D em sistemas embarcados: *desenvolver soluções algorítmicas e arquiteturais com eficiência energética para viabilizar a codificação de vídeos 3D de alta definição em tempo real e, ao mesmo tempo, manter alta qualidade sob severas restrições energéticas. O principal objetivo dessa tese é encontrar soluções para resolver esse desafio propondo algoritmos e arquiteturas de hardware inovadoras que possibilitem a codificação 3D em sistemas móveis.*

D.1.1 – Aplicações de Vídeos 3D

A utilização de vídeos 3D está associada a existência de aplicações multimídia que necessitem prover sensação de profundidade para aumentar a imersão dos telespectadores na cena. Aqui apresentaremos algumas da principais aplicações de vídeos 3D. Todas estas compartilham o conceito de utilizar múltiplas vistas capturas na mesma cena 3D. Para prover a sensação de profundidade, diferentes vistas são exibidas para cada um dos olhos do observador por meio de tecnologias como barreiras de paralaxe, lentes, polarização cromática, polarização direcional ou intercalação temporal (DODGSON, 2005).

- Vídeos Pessoais 3D: Popularizados por meio de dispositivos móveis capazes de processar vídeos 3D e de serviços de compartilhamento de vídeos 3D (YOUTUBE 3D, 2011) (VIMEO, 2012), o nicho dos vídeos pessoais representa a aplicação 3D com maior volume de conteúdo disponível. Dispondo de apenas um dispositivo móvel de gravação 3D cada usuário é capaz de criar e publicar seu próprio conteúdo.

- Televisão 3D (3DTV): a 3DTV é uma extensão da televisão 2D tradicional que proporciona também a percepção de profundidade ao observador (SMOLIC, MUELLER, *et al.*, 2007). Neste tipo de aplicação, duas ou mais vistas são decodificadas e exibidas simultaneamente de forma que cada observador veja duas vistas, uma para o olho direto e outra para o olho esquerdo. Os televisores mais simples são chamados estereoscópicos, eles exibem as duas vistas simultaneamente e requerem o uso de óculos especiais para efetuar a filtragem das vistas (óculos polarizados passivos ou de abertura e fechamento ativos). A evolução dos televisores estereoscópicos são os chamados autoestereoscópicos que eliminam a necessidade de óculos especiais. Os mais comumente encontrados são implementados por meio de barreiras de paralaxe ou lentes. Os televisores de múltiplas vistas são capazes de exibir maior número de vistas e aumentam a liberdade do observador pois suportam paralaxe de cabeça, ou seja, o conteúdo exibido se modifica quando o observador se desloca.

- Televisão com ponto de vista livre (FTV): Esta aplicação permite ao usuário selecionar o ponto de vista desejado para visualizar a cena 3D (POURAZAD, NASIOPOULOS e WARD, 2009). Além do realismo, a FTV proporciona interatividade para o usuário. A exibição pode ser feita utilizando televisores 2D ou televisores 3D (estéreo ou multi-vistas).

- Telepresença 3D: Viabiliza a comunicação e interação entre interlocutores remotos provendo aos mesmos a sensação de estarem no mesmo local fisicamente. A telepresença tem sido amplamente utilizada para video conferências, especificamente no meio corporativo, e na implementação dos chamados *home offices*. A evolução para conferências 3D (BLANCHE, BABLUMIAN, *et al.*, 2010) representa um grande avanço na qualidade da percepção e interação entre os conferencistas.

- Telemedicina 3D: A telemedicina (WELCH, SONNENWALD, *et al.*, 2005) foi criada para superar limitações físicas permitindo que médicos localizados remotamente sejam capazes de prestar consultas e efetuar cirurgias. Vídeos 3D levam a telemedicina a um novo patamar onde o médico especialista pode observar o espaço 3D com maior precisão e qualidade resultando em melhores diagnósticos e procedimentos cirúrgicos mais precisos. Esta aplicação tem grande importância no tratamento de pacientes que se encontram em locais remotos com carência de médicos especialistas.

- Vídeo Segurança 3D: Sistemas tradicionais de segurança por vídeo utilizam vídeos 2D para o monitoramento podendo levar a dificuldades quando a informação de profundidade se faz necessária. A utilização de vídeos 3D para segurança e monitoramento (KRÜGERA, NICKOLAYB, *et al.*, 2005) proporciona informação muito mais rica e detalhada com profundidade e angulação precisas de cada objeto na cena. Portanto, uma melhor descrição de possíveis criminosos e vítimas em uma cena é obtida pelo uso de vídeos 3D.

Entre estas aplicações, algumas não são voltadas para o uso e dispositivos móveis (por exemplo, segurança e telemedicina 3D) ou necessitam apenas decodificação em dispositivos móveis (3DTV e FTV). Para outras aplicações, no entanto, a habilidade de codificação 3D móvel é mandatória. Por exemplo, vídeos pessoais 3D demandam codificação de múltiplas vistas em tempo real e com alta eficiência energética. Telepresença 3D, quando rodando em dispositivos móveis, demanda além de tempo real e eficiência energética, baixa latência de codificação e decodificação. Ciente dos desafios envolvidos na implementação dessas aplicações em dispositivos móveis, este trabalho foca sua contribuição no codificador MVC voltado para dispositivos móveis.

D.1.2 – Requisitos e Tendências para Multimídia 3D

Embora o poder de processamento, principalmente para sistemas embarcados móveis, tem aumentado de forma acelerada, os requisitos de performance e energia das aplicações crescem ainda mais rapidamente devido ao aumento de resolução, taxa de quadros, precisão de amostragem e número de vistas, no caso de vídeos 3D. Em outras palavras, o volume de dados a ser processado em uma única sequência de vídeo tem aumentado em múltiplos eixos simultaneamente.

Figura D.1 relaciona o número de macroblocos (MB – unidade básica de codificação do MVC composta de 16x16 amostras) a serem processados por segundo com diferentes resoluções, taxas de quadros e número de vistas. Padrões de codificação anteriores, como o MPEG-2, foram amplamente utilizados para codificar vídeos de baixas-médias resoluções e taxas de quadros como CIF (352x288), VGA (640x480) e SDTV (768x576) a uma taxa de 15-30 fps (quadros por segundo). O H.264/AVC foca, principalmente, em resoluções altas como 720p (1240x720) e HD1080p a 30-60fps enquanto a próxima geração, o H.265/HEVC (High Efficiency Video Coding), tem como principal objetivo codificar vídeos de resoluções ultra elevadas incluindo QHD (3840x2160) e UHDTV (7680x4320) a 60-120 fps (MCCANN, MATTEI, *et al.*, 2012)(LING, 2010). Para quantificar esse crescimento, a relação entre os casos extremos representados nas Figura D.1a, CIF@15fps e QHD@60fps, é de 327x. Além disso, com o objetivo de promover maior qualidade, a largura de palavra usada para representar as amostras de vídeo tem aumentado de 8 bits para 14 bits, demandando operadores digitais mais largos e mais custosos em termos de *hardware*. Quando analisamos a complexidade e energia desprendidas o cenário se torna ainda pior uma vez que estas não crescem de forma linear com o volume de dados. O aumento de resolução, por exemplo, leva a um maior processamento por MB, maior tráfego de memória externa e maior memória on-chip relacionada a estimação de movimento, resultando em aumento energético. Somando, a evolução dos padrões de codificação contribui de forma severa com o aumento da complexidade e consumo de energia. Por exemplo, o codificador H.264/AVC é cerca de 10x mais complexo que o MPEG-4 (OSTERMANN, BORMANS, *et al.*, 2004), ao passo em que estima-se que o codificador HEVC aumente a complexidade computacional em 2-10x quando comparado ao codificador H.264 (DÍAZ-HONRUBIA, MARTÍNEZ e CUENCA, 2012).

Quando consideramos vídeos 3D, a tendência se torna ainda mais severa, como demonstrado na Figura D.1b. Além do aumento relacionado a uma única vista, o volume de dados tem um aumento diretamente proporcional ao número de vistas. Como o MVC implementa ferramentas de codificação adicionais (não existentes em padrões mono-vistas) a complexidade e consumo de energia crescem de forma não linear (crescimento superior ao crescimento linear) com relação ao número de vistas.

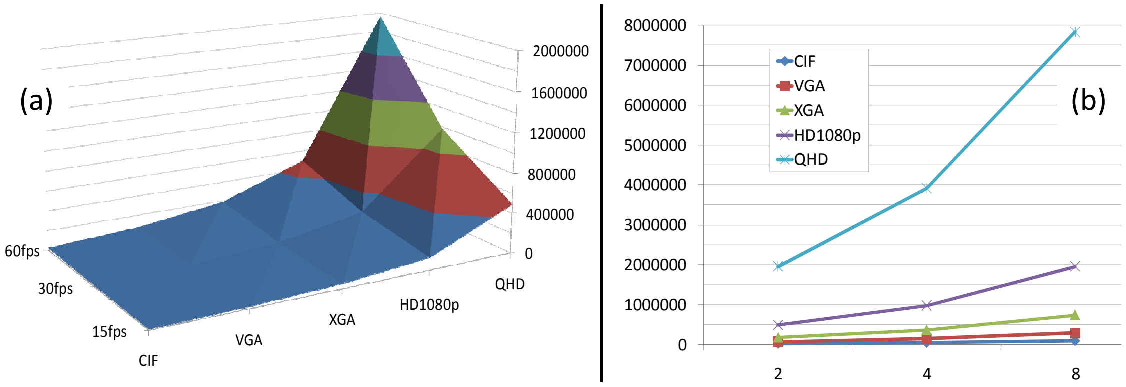


Figura D.1: Tendência de escala para vídeos 3D

D.1.3 – Revisão sobre Sistemas Multimídia Embarcados

A rápida evolução de sistemas multimídia embarcada tem sido impulsionada pela popularização dos chamados dispositivos *smart* (*smatphones*, *tablets* e outros dispositivos móveis capaz de processamento e comunicação de dados, áudio e vídeo). Grande progresso em termos de performance e eficiência energética foi feita pelos principais competidores do mercado de embarcados (ARM LTD., 2012) (NVIDIA, 2012) (QUALCOMM INC., 2011) (TEXAS INSTRUMENTS INC., 2012) (SAMSUNG ELECTRONICS CO. LTDA., 2012). O progresso, no entanto, não é suficiente para cobrir a lacuna entre os requisitos das aplicações multimídia e a evolução tecnológica dos circuitos integrados. A ARM, cujos processadores equipam 90% dos dispositivos embarcados atuais (SOFTPEDIA, 2010), prevê um aumento de performance na ordem de 10x para 2016 quando comparado as sistemas produzidos em 2009, conforme Figura D.2a. Restrições de energia relacionadas ao lento aumento de capacidade das baterias tem sido o fator limitante. De acordo com a Panasonic (KUME, 2010), a capacidade das baterias de Íons de Lítio aumenta em média 11% ao ano, como quantificado na Figura D.2b.

As altas performance e eficiência energética requisitadas pelas aplicações de vídeo 3D atuais não são atendidas por soluções de sistemas embarcados genéricas como processadores de propósito geral, GPUs e DSPs. Existe a necessidade de utilizar aceleradores de hardware dedicados para prover a performance necessária, manter o consumo energético em patamares toleráveis ao custo de perda em flexibilidade. Os sistemas em chip (SOCs) embarcados mais atuais já utilizam esta abordagem para processamento multimídia. Alguns exemplos são Qualcomm Snapdragon S4 (QUALCOMM INC., 2011), Nvidia Tegra 3 (NVIDIA CORP., 2012), Samsung Exynos 4 (SAMSUNG ELECTRONICS CO. LTDA., 2012) e Texas Instruments OMAP 5 (TEXAS INSTRUMENTS INC., 2012). O suporte de hardware, no entanto, precisa ser estendido para propiciar suporte ao processamento de vídeos 3D.

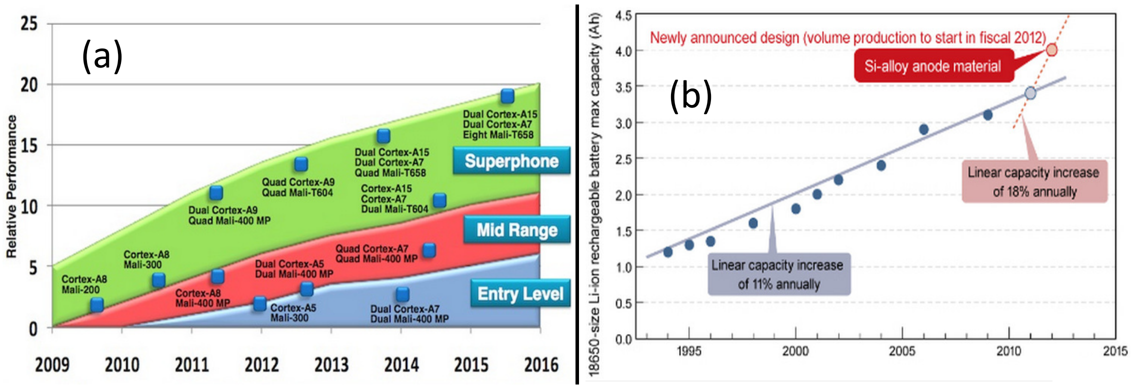


Figura D.2: (a) Tendência para performance em sistemas móveis (SHIMPI, 2011) e (b) crescimento da capacidade de baterias íons de Lítio (KUME, 2010)

D.1.4 – Dificuldades e Desafios

A alta demanda por processamento móvel 3D aliada a severas restrições de processamento e energia impõe sérios desafios aos pesquisadores e desenvolvedores que atua no ramo de sistemas embarcados móveis. Neste cenário, a utilização de aceleradores de hardware dedicados se faz mandatória. Dada a lacuna entre os requisitos de processamento multimídia 3D e a realidade dos sistemas embarcados atuais, existe ainda a necessidade otimizar a complexidade e o consumo de energia nos níveis de algoritmos e arquiteturas. Essas otimizações apenas são possíveis por meio de um profundo conhecimento da aplicação que permita otimizar conjuntamente os algoritmos e a arquitetura de hardware associada.

Além das diferentes configurações de codificação e estado da bateria, aplicações multimídia são sensíveis a variações no conteúdo de entrada que altera radicalmente o comportamento do sistema. Por exemplo, videos de alta movimentação demandam maior processamento e acesso à memória. Estas características fazem com que sejam necessárias mais unidades funcionais, maior memória on-chip e, consequentemente, um maior consumo de energia. Estas variações são detectada em tempo de execução, portanto, codificadores MVC eficientes energeticamente precisam ser adaptativos em tempo de execução e considerar características de algoritmos e vídeos de entrada. As técnicas de adaptação devem ser capazes de lidar com o compromisso entre eficiência energética e qualidade de vídeo e encontrar o ponto ótimo de operação para cada estado do sistema e vídeo de entrada.

Algoritmos para redução no consumo de energia podem levar a perdas na eficiência de codificação, ou seja, perda de qualidade para uma mesma taxa de bits. De forma a minimizar tais perdas, devem ser implementados mecanismos capazes de controlar perdas através da otimização na distribuição de bits entre vistas, quadros e macroblocos.

D.1.5 – Contribuições desta tese

O objetivo dessa tese é entender o comportamento dinâmico do codificador MVC sob a perspectiva energética e propor algoritmos e arquiteturas de hardware capazes de responder as demandas de performance e respeitar as restrições energéticas dos sistemas embarcados atuais. Nesta seção serão brevemente descritas as contribuições e inovações propostas ao longo desta tese.

D.1.5.1 Correlação na Vizinhança 3D

Os algoritmos e arquiteturas eficientes energeticamente propostos neste trabalho foram projetados com base em forte conhecimento dos algoritmos e comportamento dinâmico do codificador MVC. Ao longo deste trabalho, em muitos casos, o conhecimento da aplicação é estudado em termos da correlação dentro da vizinhança 3D. A vizinhança 3D é um espaço definido nesta tese que contem os macroblocos pertencentes as vizinhanças nos domínios espacial, temporal e de disparidade. Devido as redundâncias existentes nestas vizinhanças, a vizinhança 3D proporciona valiosa informação para predizer informações laterais de codificação, comportamento de algoritmos, padrões de acesso a memória, etc. Portanto, a análise *online* e *offline* da vizinhança 3D é utilizada para projetar e controlar algoritmos energeticamente eficientes, arquiteturas de hardware, hierarquias e dimensionamento de memória, etc.

D.1.5.2 Algoritmos Energeticamente Eficientes para MVC

Os algoritmos energeticamente eficientes propostos nesta tese estão concentrados em três blocos do codificador MVC: modo de decisão (MD), estimação de movimento e disparidade (ME/DE) e controle de taxa. A unidades de modo de decisão e ME/DE dominam o consumo de energia do codificador MVC. Em nossas soluções, MD e ME/DE buscam a redução energética por meio da redução da complexidade computacional. Estes interagem com nosso algoritmo de adaptação energética que modifica os parâmetros de codificação de acordo com o estado do sistema e estado da bateria. As perdas na eficiência de codificação impostas por tais algoritmos são minimizadas por meio de nosso algoritmo de controle de taxa hierárquico que otimiza a distribuição dos bits enquanto maximiza qualidade de vídeo e suaviza variações visuais nos eixos espacial, temporal e de disparidade.

- Modo de decisão rápido multi-nível: O modo de decisão proposto é um algoritmo composto por seis níveis de decisão incluindo um para detecção antecipada de MBs SKIP. O modo de decisão apresenta múltiplas intensidades de operação para controlar o compromisso entre qualidade e energia, considera informações da vizinhança 3D, classifica os MBs vizinhos e avalia propriedades do vídeo e do custo de codificação (RDCost).

- Adaptação de complexidade para minimização de energia: São definidos quatro estados de operação que implementam modos de decisão distintos. Estes modos podem ser alterados em tempo de execução de acordo com o estado do sistema e da bateria. Nosso algoritmo de adaptação utiliza codificação assimétrica de vistas para maximizar a qualidade percebida e prover uma degradação de qualidade suave mesmo em um cenário de bateria descarregando.

- Estimação de movimento e disparidade rápidas: A ME/DE rápida proposta neste trabalho se utiliza da correlação dos vetores de movimento e disparidade disponíveis na vizinhança 3D para evitar, completamente, o processamento da ME/DE para alguns quadros da estrutura de predição. Dependendo do nível de confiança nos MBs vizinhos o algoritmo seleciona entre os modos rápido e ultra-rápido.

- Controle de taxa hierárquico: Essa solução inovadora lança mão de dois níveis de atuação, nível de quadros e nível de unidades básicas. O primeiro utiliza um controlador de modelo preditivo (MPC) para estimar a taxa de bits e selecionar o QP (parâmetro de quantização) para um determinado quadro. Essa decisão é refinada no nível de unidades básicas que utiliza um processo de decisão de Markov (MDP) com reforço de aprendizagem (RL). Adicionalmente, o MDP utiliza o conceito de regiões de interesse (RoI) para distribuir os bits de acordo com as propriedades da imagem.

D.1.5.3 Arquiteturas de Hardware Energeticamente Eficientes para MVC

As arquiteturas energeticamente eficientes propostas neste trabalho tem como foco o processamento da estimação de movimento e disparidade, o bloco de codificação mais complexo e que demanda maior quantidade de energia. Três arquiteturas são são propostas focando no processamento, em tempo real, de 4 vistas HD1080p. Nas diferentes arquiteturas são explorados o algoritmos rápido para ME/DE proposto nesta tese bem como técnicas para reduzir a memória on-chip, reduzir o tráfego da memória externa e prover um gerenciamento dinâmico de potência eficiente. Algumas das inovações arquiteturais são apresentada abaixo.

- Memória de vídeo on-chip com múltiplos bancos: Essa proposta possibilita a implementação de uma memória de vídeo on-chip com número reduzido de bits e um gerenciamento de potência mais preciso (granularidade fina) propiciando redução de energia via diminuição da corrente de fuga (energia estática). A memória proposta opera como uma cache e dispões de múltiplos bancos para aumenta o paralelismo.

- Reuso de dados baseado em formação dinâmica de janela de busca: Macroblocos da vizinhança 3D previamente codificados são usados para prever o comportamento da busca feita pela ME/DE do macrobloco atual. Essa previsão de busca, o chamado de mapa de busca, é utilizado para controlar o esquema de reuso de dados que monta uma janela de busca, baseada no mapa de busca, de forma dinâmica. Essa técnica reduz o número de acessos a memória externa e o número de setores ativos da memoria de vídeo on-chip levando a redução energética nessas duas frentes.

- Gerenciamento de potência dinâmico baseado na aplicação: A proposta utiliza um complexo sistema para prever os requisitos de memória do codificador e controlar os estados de potência dos setores da memória on-chip. Essa decisão é definida em nível de quadros e refinada em nível de macroblocos. Os MBs da vizinhança 3D são utilizados como fonte de informação para tomada de decisões.

Nas seções que se seguem serão detalhados os algoritmos e arquiteturas energeticamente eficientes propostas nesta tese.

D.2 – Algoritmos Energeticamente Eficientes para MVC

D.2.1 –Modo de Decisão Rápido

Esta seção apresenta o algoritmo para modo de decisão rápido multi-níveis proposto nesta tese. Este modo de decisão é baseado na correlação existente na vizinhança 3D e nas propriedades dos vídeos de entrada. O fluxograma detalhado do modo de decisão multi-níveis está representado na Figura D.3. O algoritmo opera em seis níveis encadeados: (i) Classificação baseada em RDCost, (ii) predição SKIP antecipada, (iii) avaliação de modos de alta confiança, (iv) avaliação de modos de baixa confiança, (v) decisão baseada em propriedades de vídeo e (vi) modo de decisão baseado em tamanho e direção. Cada um dos níveis é executado e caso a predição não seja considerada boa, por meio de um teste de parada antecipada, o nível seguinte é executado. A condição de parada, bem como as demais limiares do algoritmo, são definidos por meio de um estudo estatístico offline na vizinhança 3D. Mais detalhes deste algoritmo estão disponíveis em (ZATT, SHAFIQUE, *et al.*, 2010) e (ZATT, SHAFIQUE, *et al.*, 2010).

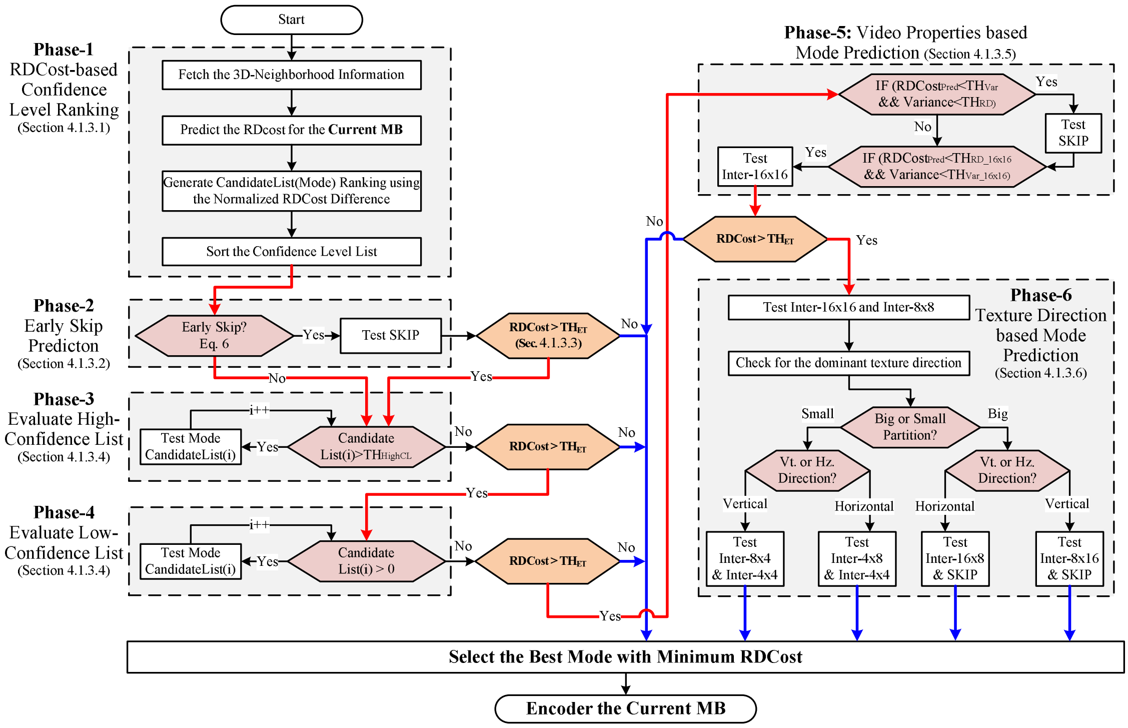

Start
Phase-1
RDCost-based Confidence Level Ranking
(Section 4.1.3.1)
Fetch the 3D-Neighborhood Information
Predict the RDcost for the Current MB
Generate CandidateList(Mode) Ranking using the Normalized RDCost Difference
Sort the Confidence Level List
Phase-2
Early Skip Predicton
(Section 4.1.3.2)
Early Skip? Eq. 6
Yes
Test SKIP
RDCost > TH_ET (Sec. 4.1.3.3)
No
Yes
No
Phase-3
Evaluate High-Confidence List
(Section 4.1.3.4)
i++
Test Mode CandidateList(i)
Yes
Candidate List(i)>TH_HighCL
No
RDCost > TH_ET
No
Yes
Phase-4
Evaluate Low-Confidence List
(Section 4.1.3.4)
i++
Test Mode CandidateList(i)
Yes
Candidate List(i) > 0
No
RDCost > TH_ET
No
Yes
Phase-5: Video Properties based Mode Prediction (Section 4.1.3.5)
IF (RDCost_Pred<TH_Var && Variance<TH_RD)
Yes
Test SKIP
No
IF (RDCost_Pred<TH_RD_16x16 && Variance<TH_Var_16x16)
Yes
Test Inter-16x16
No
RDCost > TH_ET
Yes
Phase-6
Texture Direction based Mode Prediction
(Section 4.1.3.6)
Test Inter-16x16 and Inter-8x8
Check for the dominant texture direction
Small
Big or Small Partition?
Big
Vt. or Hz. Direction?
Vertical
Horizontal
Vt. or Hz. Direction?
Horizontal
Vertical
Test Inter-8x4 & Inter-4x4
Test Inter-4x8 & Inter-4x4
Test Inter-16x8 & SKIP
Test Inter-8x16 & SKIP
Select the Best Mode with Minimum RDCost
Encoder the Current MB

Figura D.3: Modo de decisão rápido multi-níveis

D.2.2 –Adaptação de Complexidade

Além de um modo de decisão eficiente um algoritmo de adaptação de complexidade é importante para alterar dinamicamente a complexidade do codificador de acordo com o estado do sistema. Como o codificador MVC pode variar muito dependendo do nível da bateria, vídeo de entrada e cenário do sistema, nós propomos um algoritmo de adaptação de complexidade focado em controle energético. São definidas diferentes Classes de Qualidade-Complexidade (QCCs) de forma que cada uma opere em uma dada complexidade e uma dada qualidade de codificação. Além disso, codificação assimétrica é utilizada permitindo codificar vistas referentes a um olho utilizando qualidade inferior e, ainda assim, prover percepção de alta qualidade ao usuário. Abaixo são descritas as QCCs.

QCC 1: Testa os modos SKIP e Inter 16x16. Representa a classe de menor complexidade e qualidade inferior..

QCC 2: Em adição aos modos da QCC 1, testa os modos Intra 16x16, Inter 16x8, 8x16 e 8x8. Representa a classe intermediária em termos de qualidade e complexidade.

QCC 3: Esta classe é a mais complexa e de melhor qualidade de codificação. Testa todos modos das classes inferiores *QCC 1* e *QCC 2* mais Intra 4x4, Inter 8x4, 4x8 e 4x4.

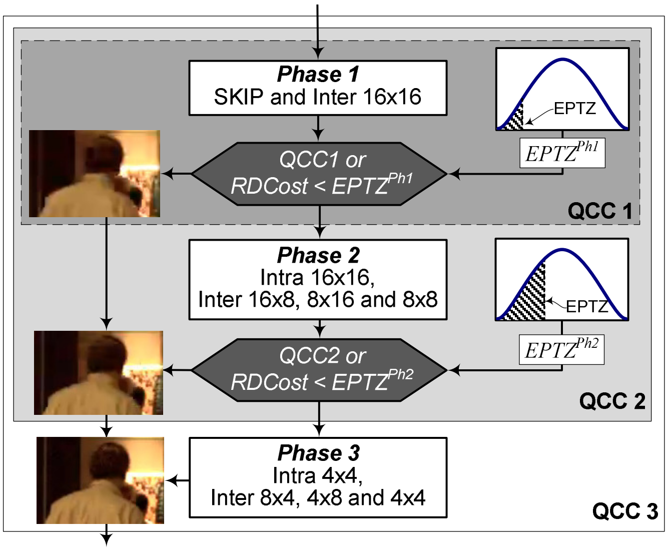


Figura D.4: Esquema de adaptação de complexidade

Figura D.4 apresenta um diagrama de alto nível do modo de decisão para as diferentes QCCs. O esquema é implementado em três níveis cascateados cujo fluxo é controlado pela QCC do MB codificado e por uma condição de término (EPTZ). A condição de término é definida estatisticamente de acordo com as propriedades de cada MB. O método de classificação dos MBs e a definição da EPTZ são apresentados em detalhes em (SHAFIQUE, ZATT, *et al.*, 2010).

D.2.3 –Estimação de Movimento e Disparidade Rápidas

O algoritmo rápido para ME/DE (ZATT, SHAFIQUE, *et al.*, 2011) explora a correlação existente no campo de movimento/disparidade dentro da vizinhança 3D. Para construir os campos demovimento/disparidade alguns quadros definidos como quadros chave são codificados de forma quase ótima utilizando o padrão de busca TZ. Desta forma limita-se a propagação de erro ao longo da sequência codificada. Os demais quadros, quadros não chave, utilizam nosso algoritmo rápido. Os quadros não chave utilizam vetores da vizinhança 3D para estimar o vetor de cada MB. Assim, o processo de busca ME/DE é totalmente evitado.

A Figura D.5 apresenta o diagrama de blocos do algoritmo rápido proposto para estimação de movimento e disparidade composto por três etapas: (i) Avaliação a nível de quadro; (ii) Avaliação e predição a nível de MB; (iii) Armazenamento de vetores de movimento e disparidade. O esquema proposto implementa dois modos de predição distinto: Rápido e Ultra-Rápido. O primeiro testa até 15 vetores encontrados na vizinhança 3D enquanto o segundo testa apenas 3 vetores candidatos. O modo Ultra-Rápido é utilizado apenas quando a grande maioria dos vetores da vizinhança 3D aponta para o mesmo vetor candidato.

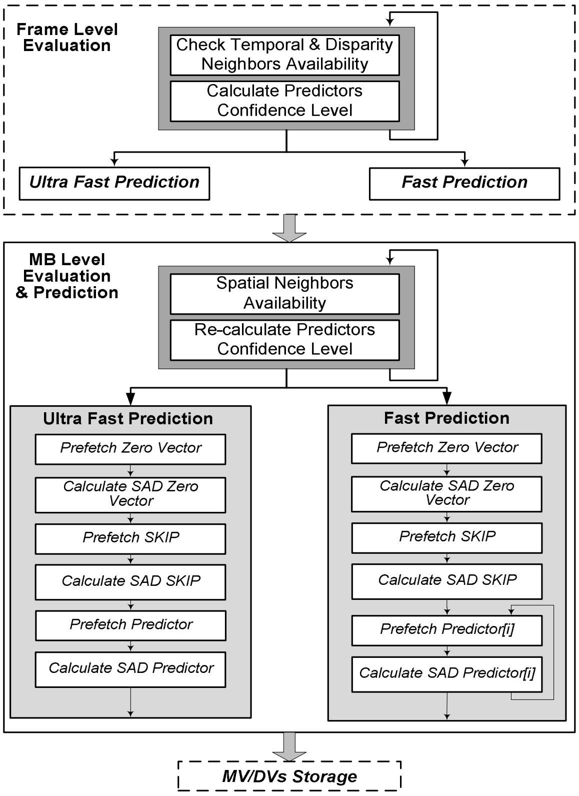


Figura D.5: Estimação de movimento e disparidade rápidas

D.2.4 –Controle de Taxa Hierárquico

O diagrama que representa o controle de taxa hierárquico (HRC) é apresentado na Figura D.6. O HRC é responsável por controlar o número de bits na saída do codificador, respeitando as restrições do canal de transmissão ou dispositivo de armazenamento, através do monitoramento do codificador MVC e atuando por meio de adaptação do QP. Ele foi desenvolvido em dois níveis de atuação: (i) nível de quadros para granularidade grossa e (ii) nível de unidades básicas para granularidade fina. Um controlador de modelo preditivo (MPC) para controle de taxa estima o volume de bits para cada quadro, analisando o comportamento do sistema, e atribui o QP mais adequado. Para isso, o MPC considera a vizinhança temporal, de disparidade e de fase entre grupos de quadros (GOP) contíguos. O QP definido para o quadro é repassado para o controlador a nível de unidades básicas que implementa um processo de decisão de Markov (MDP) que considera as características da imagem para alocar mais ou menos bits através de alteração do QP. O MDP opera sobre um mapa de regiões de interesse (RoI) definido como sendo o mapa de variância da imagem onde regiões de alta variância (de mais difícil predição) recebem maior cota de bits. Para atualizar os parâmetros do MDP é utilizado o mecanismo de reforço de aprendizagem (RL). Uma relimentação conjunta para o MDP e o MPC é feita através do RL e de um mecanismo observador garantindo a consistência entre os dois níveis de atuação. O observador usado no HRC lê, armazena e gerencia os dados do codificador MVC a serem realimentados ao controlador (número de bits alvo) além de variáveis de estado do sistema (QP, configuração do codificador, taxa de bits alvo, etc). Adicionalmente, uma unidade para extrair propriedades do vídeos é implementada para alimentar o HRC. Detalhes podem ser encontrados em (VIZZOTTO, ZATT, *et al.*, 2012).

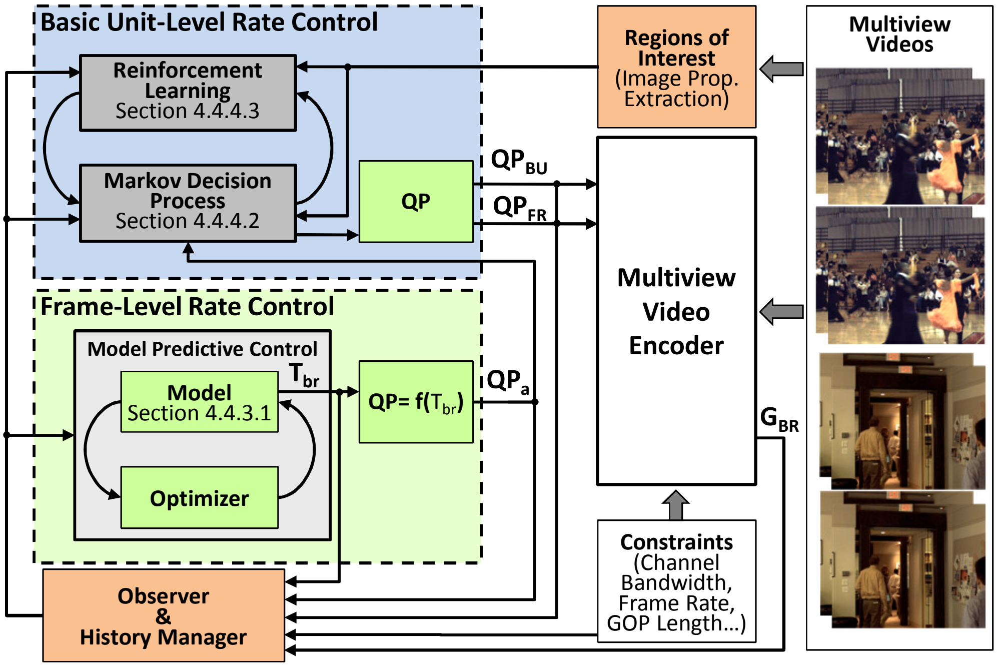

Basic Unit-Level Rate Control
Reinforcement Learning Section 4.4.4.3
Markov Decision Process Section 4.4.4.2
QP
QP_{BU}
QP_{FR}
Frame-Level Rate Control
Model Predictive Control
T_{br}
Model Section 4.4.3.1
Optimizer
$QP= f(T_{br})$
QP_a
Regions of Interest (Image Prop. Extraction)
Multiview Video Encoder
G_{BR}
Constraints (Channel Bandwidth, Frame Rate, GOP Length...)
Multiview Videos
Observer & History Manager

Figura D.6: Controle de Taxa Hierárquico

D.3 – Arquiteturas de Hardware Energeticamente Eficientes para MVC

A Figura D.7 apresenta uma arquitetura de hardware proposta que implementa a técnica de reuso baseada em formação dinâmica de janela de busca. Ela implementa, além da técnica de reuso citada, uma estrutura de memória de vídeo on-chip compostas por múltiplos bancos de memória SRAM capaz de controle de potência a nível de setores. Esta memória se comporta como uma memória cache. Para o processamento dos dados são utilizados 64 (4x4 pixels) elementos de processamento e árvores de SAD. Uma unidade de controle de busca para ME/DE micro-programável é utilizada gerenciar os padrões de busca desejados e fornecer informações para a unidade de gerenciamento de potência. A unidade de predição de mapa de busca estima o mapa de busca através da vizinhança 3D e controla a formação da janela de busca dinamicamente. Os dados que correspondem a janela de busca predita são buscados na memória externa e armazenados na memória de vídeo on-chip.

O agendamento do processamento projetado para nossa arquitetura de hardware é apresentado na Figura D.8 que inclui a etapa de computação de SAD e a etapa de obtenção de dados. Enquanto o SAD é calculado para um estágio do padrão de busca, os dados necessários para o estágio seguinte são buscados na memória externa. No entanto, em caso de um erro na predição da janela de busca, uma bolha é inseria no pipeline (veja A na Figura D.8). Caso o padrão de busca seja interrompido por uma condição de parada, os dados já lidos da memória externa são descartados.

Figura D.9 mostra o leiaute físico do circuito integrado dedicado que implementa uma de nossas arquiteturas para ME/DE proposta neste trabalho. O circuito foi completamente desenvolvido até o nível físico mas não foi fabricado. Maior nível de detalhes é encontrado em (ZATT, SHAFIQUE, *et al.*, 2011), (ZATT, SHAFIQUE, *et al.*, 2011) e(ZATT, SHAFIQUE, *et al.*, 2011).

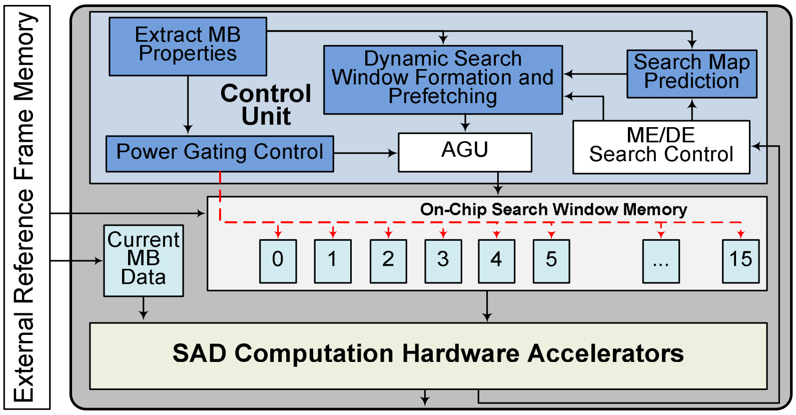

External Reference Frame Memory
Extract MB Properties
Control Unit
Dynamic Search Window Formation and Prefetching
Search Map Prediction
ME/DE Search Control
Power Gating Control
AGU
On-Chip Search Window Memory
0
1
2
3
4
5
...
15
Current MB Data
SAD Computation Hardware Accelerators

Figura D.7: Diagrama de blocos da arquitetura de ME/DE

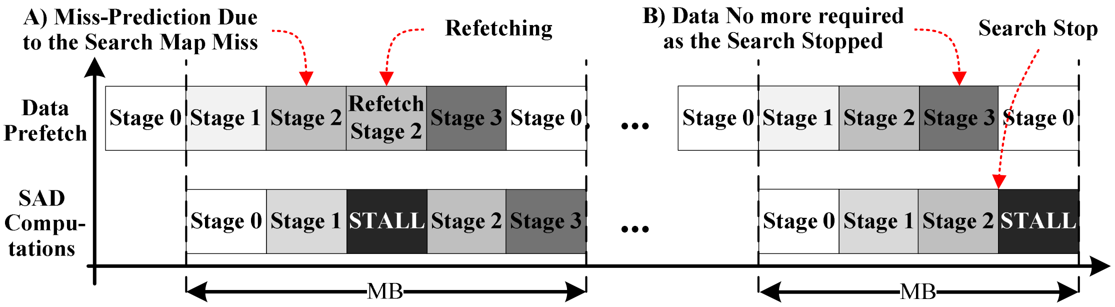
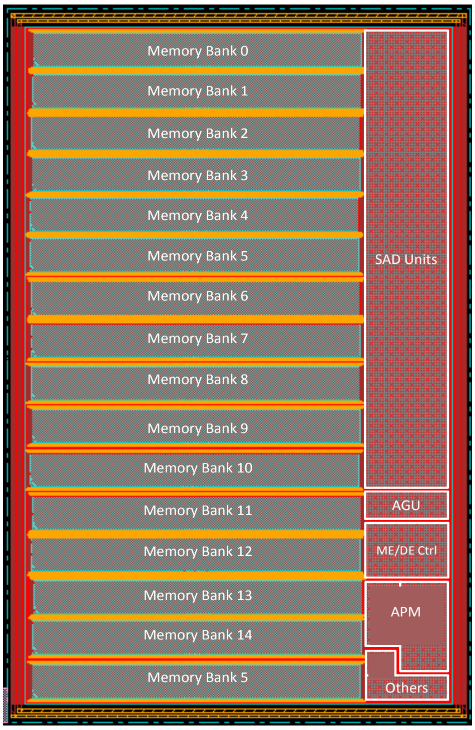


Figura D.8: Agendamento do pipeline de processamento ME/DE



SAD Units:	Sum of Absolute Differences Operators
ME/DE Ctrl:	Motion/Disparity Estimation Control
AGU:	Address Generation Unit
DPM:	Dynamic Power Management

Figura D.9: Mapeamento físico do chip que implementa a arquitetura pra ME/DE

D.4 – Conclusões e Trabalhos Futuros

A presente tese focou na redução do consumo de energia do Codificador de Vídeo Multivistas (MVC) para permitir a realização de codificação de vídeos 3D de alta definição em tempo-real em dispositivos portáteis com energia restrita. Para isto, técnicas eficientes em energia inovadoras foram propostas tanto no nível algorítmico como no nível arquitetural. A consideração conjunta de algoritmos e a arquitetura de hardware é o ponto chave para prover eficiência energética, como foi demonstrado nesta tese.

A forte correlação no domínio da vizinhança 3D, conceito definido nesta tese, foi base para o projeto da maioria dos algoritmos e esquemas de hardware adaptativos propostos. Um extenso estudo baseado na análise estatística que correlaciona informações laterais da codificação MVC (tais como modos de codificação, vetores de movimento/disparidade e RDCost) com as propriedades do vídeos foi conduzido para justificar a importância do entendimento da vizinhança 3D e para demonstrar seu potencial reduzir energia no codificador de vídeo MVC.

Um conjunto de *algoritmos eficientes em energia para MVC* compõe uma das maiores contribuições ao estado-da-arte propostas neste trabalho. Dois algoritmos para decisão rápida de modo são descritos focando na redução do consumo de energia através da redução de complexidade. A predição *Early SKIP* explora a alta ocorrência de MBs do tipo SKIP para acelerar o processo de codificação utilizando métodos estatísticos para definir se cada MB está na região de alta probabilidade de SKIP no sentido de evitar a avaliação de outros modos de codificação. O conceito de *early SKIP* é integrado a um algoritmo de decisão rápida multi-nível para reduzir ainda mais o consumo de energia. Ele elimina a avaliação dos modos de codificação mesmo no caso de um *early SKIP* não for detectado. Isto é feita através da análise dos modos de codificação disponíveis na vizinhanca 3D levando em consideração um ranking de modo baseado nas propriedades do video e RDCost. As propriedades do vídeo são usadas para definir tamanhos de bloco e orientações dos modos de predição.

Para evitar que o algoritmo de decisão rápida multi-nível diminua excessivamente a qualidade do vídeo, um teste de término antecipado foi inserido entre cada passo de predição. O algoritmo define limiares baseados no QP para diferentes forças de redução de energia chamadas de *relax* e *aggressive*. Empregando dois modos de operação é possível selecionar o melhor compromisso energia versus qualidade para um dado estado do sistema e conteúdo do vídeo. Além disto, estados múltiplos de MD possibilitam a integração de um esquema adapatativo de complexidade energeticamente eficiente. Avaliações, resultados e comparações com trabalhos relacionados apontaram uma redução de complexidade de 25% ao custo de perda de qualidade de 0,32dB e 10% de aumento no *bitrate* com o modo *aggressive* e perda de qualidade de 0,1dB e 3% de aumento no *bitrate* para o modo *relax*.

Esta tese demonstrou que as propriedades e o esforço de codificação dependem fortemente do conteúdo do vídeo. Além disto, se consideradas aplicações embarcadas, o poder de processamento está limitado aos recursos de energia disponíveis na bateria do sistema embarcado. A partir destas observações foi proposto um algoritmo de complexidade adaptativa eficiente em energia. O objetivo é considerar conjuntamente as características do vídeo de entrada e o estado da bateria para prover a máxima qualidade de vídeo através da seleção de algoritmos apropriados para MD e estados de qualidade. No caso de descarregamento da bateria, uma redução de energia adicional é necessária levando à redução de qualidade. Deste modo, o algoritmo de complexidade adaptativa reduz a qualidade através da aplicação do conhecimento da teoria de supressão binocular. Para exibição binocular, o Sistema Visual Humano tende a perceber a vista de mais alta qualidade, logo os algoritmos propostos tendem a reduzir qualidade das vistas ímpares garantindo uma alta qualidade perceptual final enquanto reduzem a energia para o processamento destas vistas ímpares. Resultados experimentais mostraram o efeito benéfico da adaptação da complexidade para o consumo de energia e uma variação suave de qualidade ao longo do tempo para cenários de carga e descarga da bateria.

A estimação de movimento e disparidade consomem mais de 90% da energia total de codificação MVC e representam o maior alvo para redução de energia. Nesta tese, um novo método de ME/DE rápida foi detalhado. Ele usa vetores de movimento e disparidade disponíveis na vizinhança-3D para evitar um padrão completo de busca de movimento/disparidade nos múltiplos quadros da estrutura de predição. Foram definidas duas classes para quadros-chave e quadros não-chave, no qual os quadros-chave são codificados utilizando técnicas conhecidas de busca de movimento/disparidade e para os quadros não-chave é utilizado nosso método rápido. De acordo com confiança (definida usando propriedades da imagem) nos vetores inferidos da vizinhança, cada MB dos quadros não-chave selecionam um modo entre rápido e ultra-rápido. Estes modos testam somente 3 ou 13 blocos candidatos, respectivamente. O algoritmo proposto é capaz de reduzir 83% do tempo de codificação total ao custo de redução em 0.116dB na qualidade e aumento de 10% no *bitrate*.

Para compensar perdas eventuais decorridas dos algoritmos eficientes em energia, um gerenciamento de qualidade baseado em nosso algoritmo de controle de taxa hierárquico (*hierarchical rate control*, HRC) foi proposto. O HRC opera em 2 níveis de atuação, o nível de quadro e o nível de unidade básica e caracteriza um laço fechado de realimentação. O controle de taxa no nível de quadro emprega um Modelo de Controle Preditivo (*Model Predictive Controller*, ou MPC) para predizer o *bitrate* para os quadros futuros baseado na alocação de bits dos quadros pertencentes à vizinhança 3D. Os múltiplos estímulos provenientes dos quadros vizinhos temporais, espaciais e de fase compõem a entrada MPC. A predição do *bitrate* é usada para definir o QP ótimo para o quadro. O QP é refinado internamente ao quadro um Controle de Taxa no Nível de unidade básica (BU) baseado em um Processo de Decisão de Markov (*Markov Decision Process*, MDP). Ele considera regiões de interesse para priorizar regiões da imagem difícies de codificar. *Reinforcement learning* é usado para atualizar os parametros do MDP. O HRC fornece variações suaves de bitrate e qualidade ao longo dos eixos de tempo e vistas, ao mesmo tempo respeitando restrições de largura de banda e aprimorando a qualidade do vídeo. Comparada à solução de QP fixo, a qualidade do vídeo foi aumentada em 1.9dB (método BjØntegaard). Comparado ao estado-da-arte, o erro de predição de bitrate foi reduzido para 0.83% com um aumento de qualidade de 0.106dB PSNR e 4.5% de redução de bitrate (método BjØntegaard).

Adicionalmente aos algoritmos eficientes em energia, as severas restrições de energia e os requisitos de desempenho do codificador de vídeo MVC requerem aceleração de hardware dedicado para possibilitar técnicas sofisticadas de gerenciamento de energia adaptativas à aplicação. Três *arquiteturas de hardware eficientes em energia* para estimação de movimento e disparidade foram propostas no sentido de proporcionar múltiplas opções de implementação para diferentes restrições de projeto do codificador. As arquiteturas propostas atingem *throughput* suficiente para codificar, em tempo-real, sequências de 4 vistas de vídeos HD1080p.

As arquiteturas de hardware ME/DE com *pipeline* multi-nível foi conjuntamente projetada com o algoritmo rápido de ME/DE apresentado nesta tese. A solução de pipeline dual emprega duas unidades de busca e despacho em paralelo, uma para a busca regular e a outra para o algoritmo rápido em si. Três memórias cache com paradigmas distintos de busca e carga foram projetadas para evitar faltas e evitar a retransmissão de dados. Um novo escalonamento para o processamento foi projetado explorando os múltiplos níveis de paralelismo disponíveis na estrutura de codificação MVC (nos níveis de vista, quadro, quadro de referência e MB) para lidar com as dependências de dados.

Fundindo os dois pipelines, foi proposta uma nova arquitetura para ME/DE que incorpora uma memória multi-banco interna ao chip para armazenamento de vídeo e uma técnica de pré-carga baseada em janela de busca dinâmica que conjuntamente reduzem o consumo de energia das memórias *on-chip* e *off-chip*. Uma janela de busca se expande dinamicamente em tempo de execução baseada no mapa de busca extraído da vizinhança para reduzir os acessos à memória *off-chip*. Considerando a natureza multi-estágio dos esquemas avançados de ME/DE rápida, uma memória *on-chip* multi-banco de tamanho reduzido é particionada em múltiplos setores que podem ser desligados (*power-gated*) dependendo das propriedades do vídeo além de empregar uma sintonia de grão-fino para redução da corrente de *leakage*.

O potencial de redução de energia da memória motivou a proposta de uma arquitetura inovadora energeticamente eficiente que contém um elaborado esquema de gerenciamento de potência dependente da aplicação para a memória de vídeo *on-chip*. A organização da memória (tamanho, bancos, setores, etc.) é realizada a partir de uma extensa análise do comportamento de uso da memória para várias sequências de vídeo 3D. Considerando o modelo de múltiplos estados de potência adotado, o esquema de gerenciamento de potência dependente da aplicação é adotado para reduzir a energia de *leakage* da memória *on-chip*. O conhecimento do algoritmo de estimação de movimento e disparidade em conjunto com as propriedades do vídeo é considerado para predizer os requisitos de memória para cada quadro e refinar para o nível de macrobloco. Uma função de custo é avaliada para determinar o estado de *sleep* apropriado para cada setor da memória, considerando o custo necessário para o *wakeup* (latência e energia).

A contribuição arquitetural apresentada nesta tese envolve projeto da arquitetura, esquemas de gerenciamento, codificação completa do RTL e síntese ASIC até o nível físico usando tecnologia de fabricação de 65-nm. Dos resultados experimentais para múltiplas sequências de vídeo, as arquiteturas propostas proporcionam uma redução de energia dinâmica de 82-96% para a memória *off-chip* e até 80% de redução de energia de *leakage* comparado com o estado-da-arte. Desta contribuição, é possível demonstrar a viabilidade de realizar estimação de movimento e disparidade para até 4 vistas de vídeos HD1080p a 30 quadros por segundo com uma dissipação de potência de 57mW executando a 300MHz para um consumo de 102k *gates* do circuito integrado.

Os resultados gerais e *benchmarks* demonstraram a eficiência energética dos algoritmos e arquiteturas propostos frente às soluções estado-da-arte. Isto prova nossa hipótese de que, para cumprir os requisitos de codificação de vídeo 3D para sistemas embarcados, é necessário considerar conjuntamente e otimizar os algoritmos de codificação e as arquiteturas de hardware dedicadas que os executam. Adicionalmente, adaptação em tempo de execução é necessária para melhor predizer o comportamento do sistema e reagir às mudanças no conteúdo do vídeo, parâmetros de codificação e cenários de nível da bateria. Para isto, um conhecimento profundo da aplicação MVC realizado de uma extensiva análise, como da correlação disponível da vizinhança-3D, deve ser empregado.

D.4.1 – Trabalhos Futuros

Além das contribuições feitas nesta tese, existem múltiplos tópicos de pesquisa relacionados à codificação e processamento de vídeo 3D que não foi abordado neste volume. Os algoritmos e arquiteturas aqui apresentados estão concentrados na estimação de movimento e disparidade uma vez que estes são as unidades de codificação que mais consomem energia no codificador MVC. Adicionalmente, questões de qualidade de vídeo foram discutidas na seção de controle de taxa. O MVC, contudo, traz um grande conjunto de outros desafios de pesquisa se aplicações embarcadas forem consideradas. O pré- e pós-processamento de vídeo 3D também cumpre um papel chave em sistemas de vídeo 3D e apresenta uma profusão de novos desafios. Finalmente, algoritmos de codificação de vídeo 3D de nova geração estão atualmente sendo estudados para nova padronização. Se espera que a nova geração de vídeos 3D traga ferramentas inovadoras e boas perspectivas para futuras oportunidades de pesquisa no campo multimídia 3D.

Desafios remanescentes em MVC: Apesar de os desafios principais em termos de complexidade e consumo de energia são relacionados aos blocos de MD e ME/DE, atender demandas do MVC e restrições de energia impostas traz novos desafios relacionados a outros blocos do codificador MVC. O codificador de entropia, por exemplo, pode se tornar um gargalo do sistema de codificação se nenhuma técnica apropriada de paralelização for aplicada. As dependências em nível de bloco da predição também possibilitam trabalhos de pesquisa. Encontrar soluções para lidar com as dependências de dados e os problemas de paralelização possibilitam oportunidades interessantes de pesquisa para trabalhos futuros.

Pré- e pós-processamento de videos 3D: codificação de vídeo 3D é um estágio único de um sistema de vídeo 3D. Entre a captura do vídeo e as fases de codificação, existe a necessidade de pré-processamento de vídeo como calibragem geométrica (para corrigir o alinhamento entre múltiplos vídeos) e correção de cor (responsável pela equalização do brilho e *gamut* de cores). Depois da transmissão e decodificação, o vídeo deve ser processado para exibição dependendo da aplicação e tecnologia do display. Esta fase de pós-processamento inclui mapeamento do espaço de cores (em um sistema usando polarização de cor), escalamento da resolução e síntese de pontos de vista (geração de pontos de vista intermediários para exibição). O pré- e pós-processamento implementa algoritmos complexos e intensivos em dados (especialmente para síntese de vistas) que executam concorrentemente com o codificador/decodificador de vídeo e requerem desempenho de tempo-real. Portanto, a energia e recursos de hardware devem ser compartilhados para atender as demandas tanto da codificação de vídeo como das etapas de pré- e pós-processamento.

Nova geração de codificação de vídeo 3D: a nova geração de codificação de vídeo 3D é atualmente referida como 3DV (3D Video) (ISO/IEC, 2009) e é baseada no conceito de Vídeo+Profundidade que define canais distintos para transmitir vídeo e mapas de profundidade. Espera-se que o 3DV seja definido como uma extensão do HEVC/H.265 (SULLIVAN e OHM, 2010). As ferramentas do 3DV irão prover um conjunto completamente novo de desafios que impulsionará a pesquisa relacionada a multimídia 3D. Além disso, o tempo de vida dos padrões de codificação futuros tende a reduzir exigindo que múltiplos padrões coexistam em um mesmo sistema. Portanto, um mesmo dispositivo multimídia embarcado deverá suportar múltiplos padrões sendo flexível e capaz de se adaptar aos diversos cenários possíveis.