

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**ALEXANDRE GASPARY HAUPT**

**DETECÇÃO DE MOVIMENTO, ACOMPANHAMENTO E  
EXTRAÇÃO DE INFORMAÇÕES DE OBJETOS MÓVEIS**

Porto Alegre

2004

**ALEXANDRE GASPARY HAUPT**

**DETECÇÃO DE MOVIMENTO, ACOMPANHAMENTO E  
EXTRAÇÃO DE INFORMAÇÕES DE OBJETOS MÓVEIS  
APLICAÇÃO EM CONTROLE DE TRÁFEGO**

Dissertação de mestrado apresentada ao  
Programa de Pós-Graduação em Engenharia Elétrica,  
da Universidade Federal do Rio Grande do Sul, como  
parte dos requisitos para a obtenção do título de Mestre  
em Engenharia Elétrica.

Área de concentração: Automação e  
Instrumentação Eletro-Eletrônica

**ORIENTADOR: Altamiro Amadeu Susin**

Porto Alegre

2004

ALEXANDRE GASPARY HAUPT

**DETECÇÃO DE MOVIMENTO, ACOMPANHAMENTO E  
EXTRAÇÃO DE INFORMAÇÕES DE OBJETOS MÓVEIS  
APLICAÇÃO EM CONTROLE DE TRÁFEGO**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_

Prof. Dr. Altamiro Amadeu Susin, UFRGS

Doutor pela INPG - Grenoble, FRANÇA.

Banca Examinadora:

Prof. Dr. Sérgio Bampi, UFRGS

Doutor pela Stanford University – Stanford, EUA

Prof. Dr. Renato Machado de Brito, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Adalberto Schuck Junior, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Coordenador do PPGEE: \_\_\_\_\_

Prof. Dr. Carlos Eduardo Pereira

Porto Alegre, dezembro de 2004.

## **DEDICATÓRIA**

Dedico este trabalho a minha mãe Ligia e minha Irmã Carla, pelo amor, força e compreensão durante as várias fases de nossa vida.

## **AGRADECIMENTOS**

Ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, pela oportunidade de realização de trabalhos em minha área de pesquisa.

Aos membros do PPGEE, em especial aos colegas do LaPSI pelo seu auxílio nas tarefas desenvolvidas durante a dissertação e apoio no desenvolvimento do trabalho.

A meus professores do PPGEE, meus modelos, que guiaram esta trajetória: Carlos Eduardo Pereira, Adalberto Schuck Jr., Alexandre S. Bazanella, Altamiro Amadeu Susin, Jorge Lisboa e Laschuk

Obrigado aos professores Renato Machado de Brito, Adalberto Schuck Jr. e Sérgio Bampi por aceitarem o convite para compor a banca examinadora desta dissertação.

Agradeço, especialmente, meus orientadores: o professor Altamiro Amadeu Susin, pelos mais de quatro anos de convivência e aprendizado e Letícia Vieira Guimarães que informalmente co-orientou esta dissertação.

Aos colegas do SENAI, onde destaco os professores Sergio Helegda, Almiro Luís da Rocha, Maria Berenice e Clarice Klein por estarem presentes em todas as horas.

Agradeço, de coração aos amigos Pablo Oliveira Correa, Pedro Spohr, Henrique Anunziato e Marcos Plaza pelo apoio.

## RESUMO

O tráfego de veículos, principalmente nas vias públicas de regiões densamente povoadas, tem sido objeto de preocupação por diversos aspectos: segurança no trânsito, supervisão da frota, controle de poluição, prevenção, registro de irregularidades, etc. Sistemas eletrônicos tradicionais são apontados como limitados pelas dificuldades apresentadas na instalação dos sensores e poucos recursos para extração de informações. Este trabalho propõe um sistema baseado no processamento digital de imagens em vídeo para detectar, acompanhar e extrair informações de um veículo em movimento, tais como: posição, velocidade e dimensões. Embora técnicas de pré-processamento sejam empregadas para minimizar problemas de iluminação e trepidação, informações obtidas nas etapas finais do processamento, tendem a distorcer em função destes fatores. Algoritmos baseados em análise probabilística no domínio espaço-temporal de imagens 2-D são propostos para segmentar o objeto mesmo na presença destes fatores. A fim de tornar o acompanhamento mais robusto a oclusões e pixels espúrios, é proposto um método que detecta e rejeita posições inconsistentes do objeto de acordo com a estimativa de sua próxima posição.

**Palavras-chave:** Processamento de imagens, detecção de movimento, segmentação, limiar, acompanhamento, controle de tráfego.

## **ABSTRACT**

The vehicles traffic, mainly in the public roads of densely populated regions, has been object of concern for many aspects: transit security, supervision of the fleet, pollution control, prevention, irregularities register, etc. Traditional electronic systems are pointed as limited for the difficulties presented by sensors installation and few resources for information extraction. This work presents a system based on video digital image processing to detecting, tracking and extracting information of one vehicle in movement, such as: position, speed and dimensions. However preprocessing techniques have been employed in order to resolve factors like illumination and vibration, information extracted on last processing stage, tend to be distorted by these factors. Algorithms for probabilistic analysis, based on space temporal domain of 2-D images, are proposed in order to segment the object even on these conditions. In order to become the tracking robust to occlusions and spurious pixels, a method is considered to detect and rejects inconsistent object positions in according to the next position estimation.

**Keywords: Image processing, motion detection, segmentation, threshold, tracking, traffic control.**

## SUMÁRIO

<b>LISTA DE ILUSTRAÇÕES</b> .....	10
<b>LISTA DE TABELAS</b> .....	12
<b>LISTA DE ABREVIATURAS</b> .....	13
<b>1 INTRODUÇÃO</b> .....	14
1.1 ESTADO DA ARTE .....	15
1.2 MOTIVAÇÃO .....	17
<b>2 PRINCÍPIOS DO PROCESSAMENTO DE IMAGENS</b> .....	18
2.1 REPRESENTAÇÃO DIGITAL DA IMAGEM .....	18
2.2 MODELOS MAIS USADOS PARA REPRESENTAÇÃO DE CORES .....	19
2.2.1 O formato RGB .....	20
2.3 OPERAÇÕES MORFOLÓGICAS .....	21
2.3.1 Dilatação Binária .....	22
2.3.2 Erosão Binária .....	23
2.3.3 Abertura e fechamento .....	24
2.4 RESTAURAÇÃO DE IMAGENS .....	26
2.5 DETERMINAÇÃO DO LIMIAR PARA BINARIZAÇÃO .....	30
2.6 DETECÇÃO DE BORDAS .....	31
2.6.1 Operadores Gradiente .....	32
2.6.2 Operadores Laplaciano .....	35
2.6.3 Método Canny .....	36
2.6.4 Transformada de Hough .....	38
<b>3 ARQUITETURA DO SISTEMA</b> .....	41
3.1 HARDWARE .....	41
3.2 SOFTWARE .....	41
<b>4 METODOLOGIA</b> .....	43
4.1 AQUISIÇÃO DAS IMAGENS .....	43
4.1.1 Ajuste de Parâmetros .....	44
4.2 PRÉ-PROCESSAMENTO .....	45
4.2.1 Conversão de RGB para Grayscale .....	46
4.2.2 Realce de Contraste .....	47
4.2.3 Suavização da imagem .....	48
4.3 SEGMENTAÇÃO E DETECÇÃO DE MOVIMENTO .....	49
4.4 BINARIZAÇÃO .....	51
4.4.1 Método Estatístico .....	52
4.4.2 Método Laplaciano .....	54
4.5 REMOÇÃO DE PIXELS DESCONECTADOS .....	58
4.6 DETERMINAÇÃO DO PONTO CENTRAL .....	59



<b>4.6.1 Método do Centróide</b> .....	59
<b>4.6.2 Método dos pontos máximos e mínimos</b> .....	60
<b>4.6.3 Comparação entre os métodos</b> .....	62
<b>4.7 EXTRAÇÃO DE INFORMAÇÕES</b> .....	64
<b>4.8 VERIFICAÇÃO E CORREÇÃO DA TRAJETÓRIA</b> .....	67
<b>5 RESULTADOS</b> .....	70
5.1 RESULTADOS COM IMAGENS ARTIFICIAIS .....	70
5.2 RESULTADOS COM IMAGENS EM RESOLUÇÃO 320X240 .....	71
5.3 RESULTADOS COM IMAGENS EM RESOLUÇÃO 640X480 .....	75
<b>6 CONCLUSÕES E TRABALHOS FUTUROS</b> .....	79
<b>REFERÊNCIAS</b> .....	81
<b>APÊNDICE A - Método pra determinar o tempo das rotinas</b> .....	83
<b>APÊNDICE B - Condições de aquisição das amostras</b> .....	84
<b>APÊNDICE C - Banco de amostras para calibração do sistema</b> .....	85
<b>APÊNDICE D - Biblioteca de funções filtro.h</b> .....	86
<b>APÊNDICE E - Bibliotecas de funções hali.h</b> .....	87
<b>APÊNDICE F - Resultados amostras 320 x 240</b> .....	88
<b>APÊNDICE G - Resultados amostras 640 x 480</b> .....	105
<b>APÊNDICE H - Tempos de execução das rotinas</b> .....	108
<b>ANEXO A - Frota de veículos por tipo no estado do RS</b> .....	109
<b>ANEXO B - Frota de veículos por tipo em Porto Alegre</b> .....	110
<b>ANEXO C - Descrição técnica – câmera Canon</b> .....	111

## LISTA DE ILUSTRAÇÕES

Figura 1 - Acidentes de trânsito com vítimas de 1997 a 2003 nas principais rodovias .....	14
Figura 2 - Convenção de eixos utilizada para representar uma imagem digital .....	18
Figura 3 - Cubo de cores do modelo RGB .....	20
Figura 4 - Dilatação e erosão de imagem binária .....	22
Figura 5 - Dilatação binária .....	23
Figura 6 - Erosão binária .....	24
Figura 7 - Operações morfológicas de abertura e fechamento .....	25
Figura 8 - Fechamento de falhas em contornos com operações morfológicas .....	26
Figura 9 - Diagrama da Convolução da máscara com a imagem .....	28
Figura 10 - Suavização da imagem .....	29
Figura 11 - Histogramas de nível de cinza .....	30
Figura 12 - Variação de intensidade luminosa e suas derivadas .....	32
Figura 13 - Operador Gradiente Sobel .....	34
Figura 14 - Resultados intermediários do método Canny .....	37
Figura 15 - Planos para representar uma reta .....	38
Figura 16 - Transformada de Hough .....	39
Figura 17 - Arquitetura do sistema .....	42
Figura 18 - Formação da Imagem em uma lente convergente .....	43
Figura 19 - Janela de calibração do sistema .....	45
Figura 20 - Conversão da imagem para Grayscale .....	46
Figura 21 - Operação de stretch do Histograma .....	47
Figura 22 - Histogramas antes e após stretch .....	48
Figura 23 - Máscara 3x3 com coeficientes genéricos – pesos .....	48
Figura 24 - Suavização da imagem .....	49
Figura 25 - Técnicas de segmentação do objeto .....	50
Figura 26 - Determinação do limiar de binarização via análise de histograma .....	52
Figura 27 - Determinação do limiar via análise estatística da distribuição de tons de cinza ....	53
Figura 28 - Imagens com limiar determinado pelo método estatístico .....	54
Figura 29 - Determinação automática do limiar de binarização através do operador Laplaciano .....	55
Figura 30 - Mapas obtidos com diferentes máscaras para implementar o operador laplaciano .....	56
Figura 31 - Resultados da determinação do limiar pelo método Laplaciano .....	57
Figura 32 - Operação morfológica de fechamento .....	58
Figura 33 - Centro do objeto determinado pelo método da centróide .....	59
Figura 34 - Máscaras para verificar conexão de pixels no método de máximos e mínimos .....	60
Figura 35 - Centro do objeto determinado através das coordenadas máximas e mínimas .....	62
Figura 36 - Comparação dos métodos de determinação das coordenadas centrais do objeto ...	63
Figura 37 - Determinação ds bordas do objeto considerando a região de interesse .....	64
Figura 38 - Projeções horizontal e vertical das bordas do objeto .....	65
Figura 39 - Janela principal do programa falcão .....	67
Figura 40 - Gráfico de atuação do filtro preditivo .....	68
Figura 41 - Correção da trajetória .....	69

<b>Figura 42 - Imagem artificial 1</b> .....	70
<b>Figura 43 - Imagem artificial 2</b> .....	71
<b>Figura 44 - Relação pixel x metros para amostras #1 a #18</b> .....	72
<b>Figura 45 - Erros nas medidas de comprimento obtidos para amostras em resolução 320x240</b> .....	73
<b>Figura 46 - Erros nas medidas de largura obtidos para amostras em resolução 320x240</b> .....	73
<b>Figura 47 - Erros nas medidas de velocidade obtidos para amostras em resolução 320x240</b> .....	75
<b>Figura 49 - Relação pixel x metros para amostras #19 a #23</b> .....	77
<b>Figura 50 - Erros nas medidas de comprimento obtidos para amostras em resolução 640x320</b> .....	77
<b>Figura 51 - Erros nas medidas de largura obtidos para amostras em resolução 640x320</b> .....	78
<b>Figura 52 - Erros nas medidas de velocidade obtidos para amostras em resolução 640x320</b> .....	78
<b>Figura 53 - <i>Tracking</i> veículo amostra #1</b> .....	88
<b>Figura 54 - <i>Tracking</i> veículo amostra #2</b> .....	89
<b>Figura 55 - <i>Tracking</i> veículo amostra #3</b> .....	90
<b>Figura 56 - <i>Tracking</i> veículo amostra #4</b> .....	91
<b>Figura 57 - <i>Tracking</i> veículo amostra #5</b> .....	92
<b>Figura 58 - <i>Tracking</i> veículo amostra #6</b> .....	93
<b>Figura 59 - <i>Tracking</i> veículo amostra #8</b> .....	94
<b>Figura 60 - <i>Tracking</i> veículo amostra #9</b> .....	95
<b>Figura 61 - <i>Tracking</i> veículo amostra #10</b> .....	96
<b>Figura 62 - <i>Tracking</i> veículo amostra #11</b> .....	97
<b>Figura 63 - <i>Tracking</i> veículo amostra #12</b> .....	98
<b>Figura 64 - <i>Tracking</i> veículo amostra #13</b> .....	99
<b>Figura 65 - <i>Tracking</i> veículo amostra #14</b> .....	100
<b>Figura 66 - <i>Tracking</i> veículo amostra #15</b> .....	101
<b>Figura 67 - <i>Tracking</i> veículo amostra #16</b> .....	102
<b>Figura 68 - <i>Tracking</i> veículo amostra #17</b> .....	103
<b>Figura 69 - <i>Tracking</i> veículo amostra #18</b> .....	104
<b>Figura 70 - Resultados processamento amostra #19</b> .....	105
<b>Figura 71 - Resultados processamento amostra #20</b> .....	105
<b>Figura 72 - Resultados processamento amostra #21</b> .....	106
<b>Figura 73 - Resultados processamento amostra #22</b> .....	106
<b>Figura 74 - Resultados processamento amostra #23</b> .....	107

## LISTA DE TABELAS

Tabela 1 - Máscaras para implementação do operador gradiente .....	34
Tabela 2 - Operadores discretos de Laplace .....	36
Tabela 3 - Número médio de pixels na imagem segmentada de acordo com classe do veículo ....	51
Tabela 4 - Resultados dos métodos para determinação do ponto central .....	62
Tabela 5 - Dimensões, velocidades e erros obtidos para cada amostra .....	71
Tabela 6 - Dimensões, velocidades e erros obtidos para cada amostra .....	72
Tabela 7 - Erros médios absolutos nas dimensões de acordo com a classe de iluminação ....	74
Tabela 8 - Dimensões, velocidades e erros obtidos para amostras com resolução VGA .....	76
Tabela 9 - Consumo de tempo das rotinas testadas para determinar do ponto central .....	83
Tabela 10 - Lista de amostras com características do veículo e condições de aquisição .....	84
Tabela 11 - Lista de amostras obtidas em ambiente controlado .....	85
Tabela 12 - Resultados das dimensões da amostra #1 .....	88
Tabela 13 - Resultados das dimensões da amostra #2 .....	89
Tabela 14 - Resultados das dimensões da amostra #3 .....	90
Tabela 15 - Resultados das dimensões da amostra #4 .....	91
Tabela 16 - Resultados das dimensões da amostra #5 .....	92
Tabela 17 - Resultados das dimensões da amostra #6 .....	93
Tabela 18 - Resultados das dimensões da amostra #8 .....	94
Tabela 19 - Resultados das dimensões da amostra #9 .....	95
Tabela 20 - Resultados das dimensões da amostra #10 .....	96
Tabela 21 - Resultados das dimensões da amostra #11 .....	97
Tabela 22 - Resultados das dimensões da amostra #12 .....	98
Tabela 23 - Resultados das dimensões da amostra #13 .....	99
Tabela 24 - Resultados das dimensões da amostra #14 .....	100
Tabela 25 - Resultados das dimensões da amostra #15 .....	101
Tabela 26 - Resultados das dimensões da amostra #16 .....	102
Tabela 27 - Resultados das dimensões da amostra #17 .....	103
Tabela 28 - Resultados das dimensões da amostra #18 .....	104
Tabela 29 - Tempo de consumo das rotinas .....	108
Tabela 30 - Quantidade de Veículos por tipo no estado do RS .....	109
Tabela 31 - Quantidade de Veículos por tipo em Porto Alegre .....	110

## LISTA DE ABREVIATURAS

AVI: Formato gravação de áudio e vídeo - Audio Video Interleave.

BMP: Formato de imagem sem compressão – BITMAP.

$d_H$ : Distância da câmera ao solo

HSI: Formato de cor que possui as componentes de matriz, saturação e intensidade.

IF: Frequência Instantânea.

$I_{TH}$ : Limiar de intensidade luminosa

JPEG: Formato de imagem. Joint Photographic Experts Group.

LaPSI: Laboratório de Processamento de Sinais e Imagem.

LiLi: Biblioteca de Funções em Linguagem C. LaPSI Image Library.

$N_{PMV}$ : Número de Pixels do menor veículo que deve ser detectado

$R_{es}$ : Resolução das imagens em pixels RGB: Formato de cor que possui as componentes vermelho-R, Verde-G e Azul-B.

ROI: Região de Interesse em uma imagem.

RS: Rio Grande do Sul – Estado do Brasil.

TF: Transformada de Fourier.

TH: Limiar Superior

TL: Limiar Inferior

USB: Porta de comunicação serial em computadores. Universal Serial Bus.

VGA: Resolução de Vídeo de 640 x 480 pixels

WAV: Formato de Som. Waveform Áudio.

YIQ: Formato de cor. Y corresponde a iluminância, I e Q são as duas componentes monocromáticas chamadas de fase e quadratura, respectivamente.

# 1 INTRODUÇÃO

Quanto mais cresce a densidade do tráfego nas rodovias, mais importante torna-se seu monitoramento visando diminuir o número de acidentes e facilitar a identificação de veículos em pedágios e postos policiais. O custo de um acidente pode ser bastante alto, tanto em termos humanos, quanto em termos das condições de tráfego na via. Os gráficos abaixo mostram o aumento do número de acidentes com vítimas nas rodovias dos territórios nacional, estadual e municipal.



**Figura 1 – Acidentes de trânsito com vítimas de 1997 a 2003 nas principais rodovias.**  
Fonte: Site do DETRAN: [www.detran.rs.com.br](http://www.detran.rs.com.br)

Recentes pesquisas na área de visão computacional podem fornecer novas alternativas para reduzir o grande número de acidentes, o que se traduz em benefícios nas áreas social e econômica. O custo decrescente de computadores e câmeras torna cada dia mais viável equipar estradas com um sistema de visão artificial capaz de detectar movimentos, seguir a trajetória do veículo e extrair informações, tais como sua velocidade, dimensões e densidade de tráfego. Muitos esforços têm sido realizados nos últimos anos para detectar e seguir objetos móveis (XIROUHAKIS, 2001), (FRAILE, 2001), (SULIVAN, 1996), (DJUROVIĆ, 2003), (GENTILE, 2004), utilizando técnicas baseadas em medidas espaciais de níveis de cinza.

Este trabalho envolve o estudo de diversas técnicas de processamento e análise de imagem que culminam no desenvolvimento de um sistema capaz de detectar movimento, acompanhar veículos estimando sua velocidade, posição e dimensões de um veículo a cada sequência de quadros. Os algoritmos propostos estão baseados em análise no domínio espaço de imagens bidimensionais adquiridas de topo em relação à rodovia. A fim de tornar o acompanhamento mais robusto a oclusões e pixels espúrios, é proposto um método que detecta e rejeita posições inconsistentes com a estimativa prévia da próxima posição do objeto.

As rotinas foram criadas em linguagem (PARKER, 1997), utilizando-se a ferramenta C-Builder e a biblioteca LiLi-LaPSI Image Library (UFRGS, 2004). Para validar a performance dos algoritmos foram analisadas imagens artificiais e reais em diferentes períodos do dia.

A revisão bibliográfica é apresentada no capítulo 2, enquanto o capítulo 3 trata dos requisitos do sistema, detalhando o hardware e o software necessários para implementação do trabalho. O capítulo 4 descreve a metodologia utilizada na pesquisa, incluindo a análise de etapas como: pré-processamento, detecção de movimento, segmentação, estimativa do limiar de binarização, eliminação de bits espúrios, acompanhamento, extração de informações e correção da trajetória.

Os resultados são apresentados no capítulo 5 com imagens adquiridas em situações reais de tráfego com amostras abrangendo diferentes períodos do dia.

## 1.1 ESTADO DA ARTE

De acordo com (FRAILE, 2001), o acompanhamento de objetos em 2D, com a câmera em perspectiva com o objeto, pode ser realizado desde que a câmera seja colocada suficientemente alta para evitar intersecções entre veículos e assim gerando oclusões indevidas. Os autores propõem uma aproximação da trajetória do veículo utilizando símbolos para marcá-la, estimando a velocidade e ângulo de cada um de seus segmentos. Uma vez obtida a seqüência de medidas, a trajetória pode ser classificada utilizando o modelo de Markov (HMM), porém este método só pode ser aplicado a um número reduzido de trajetórias, devido a seu alto peso computacional, além disso, o plano da pista deve ser conhecido.

Os autores Baoxin e Chellappa (2002); Baoxin e Chellappa (2001); Sukkivan e Baker (1996); Sullivan (1992), sistemas de acompanhamento para controle de trânsito utilizam medidas de tempo de seqüências 3D de veículos a partir de imagens monoculares. Neste método o movimento é detectado pela variação de pixels, acima de um limiar, entre uma cena e outra. No centro do movimento é ajustado um modelo em três dimensões na posição provável do objeto. O modelo é rotacionado e transladado sobre a imagem e para cada posição é gerado um *score* em função do grau de casamento entre a imagem e o modelo. A posição correta é aquela que produz o maior *score*. Quanto maior o *score*, melhor é a adequação do modelo à imagem. *Scores* acima de um limiar sugerem a presença de um objeto

móvel em uma dada posição da imagem. Conhecendo-se a posição do objeto a cada quadro e a amostragem da digitalização, é determinada a velocidade do objeto.

O problema deste método é que para o reconhecimento do objeto todos os modelos devem ser testados, gerando um alto peso computacional, uma vez que o casamento entre a imagem e o modelo é avaliado a cada posição. Neste método o problema não está em deduzir com precisão a presença do objeto, mas em fazer uma dedução inicial de sua posição para diminuir o peso computacional gasto na rotação e translação do modelo sobre a imagem.

Os autores Xirouhakis, Drosopoulos e Delopoulos (2001), propõem um sistema de acompanhamento óptico para estúdios virtuais com o objetivo de estimar com precisão o movimento da câmera em coordenadas 3D, no entanto é necessário um fundo com 2 tons de azul, com mapa disposto de forma conhecida.

O autor (DJUROVIĆ, 2003), propõe a estimativa de parâmetro de movimento de objetos em movimento em uma seqüência de vídeo através de uma representação tempo-freqüência. A transformada de Wigner é utilizada nesta representação. O método possui a vantagem de analisar múltiplos objetos, mas tem a desvantagem de produzir freqüência modulada quando ocorre variação de velocidade na seqüência de *frames* analisados. Isto implica na necessidade de um estimador de freqüência instantânea (IF) baseado na distribuição em tempo-freqüência.

Segundo (LEE, 2001), os procedimentos do processamento digital de imagens são, normalmente, feitos no domínio espaço onde as imagens são adquiridas e processadas com peso computacional mais baixo. Algoritmos para filtragens, convoluções, análise espectral, compressão e codificação de imagens são realizados no domínio freqüência com mais facilidade em termos computacionais, porém a transposição do domínio espaço para o domínio freqüência implica maior peso computacional.

A natureza linear da transformada de Fourier permite levar para o domínio freqüência somente às operações lineares. As operações não lineares do processamento digital de imagens não podem ser levadas diretamente ao domínio freqüência. De acordo com (AGAIAN; PANETTA, 2001), a vantagem de trabalhar com a imagem no domínio freqüência é a facilidade em manusear seus coeficientes com baixo custo computacional, sem dependência de informações espaciais.



## 1.2 MOTIVAÇÃO

O tráfego de veículos, principalmente nas vias públicas de regiões densamente povoadas tem sido objeto de preocupação por aspectos de segurança, supervisão da frota, controle de poluição, prevenção e registro de irregularidades, etc. Sistemas eletrônicos tradicionais são apontados como limitados pelas dificuldades apresentadas na instalação dos sensores e poucos recursos para extração de informações.

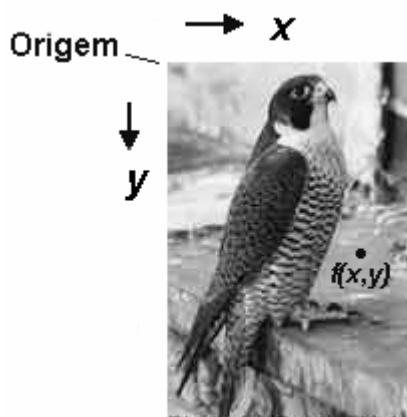
A área de processamento de imagens é nova, comparando-se com outros segmentos da engenharia elétrica, tem ainda muito a crescer e por isto oferece um grande desafio para os pesquisadores que se dedicam a seu estudo. A Universidade Federal do Rio Grande do Sul – UFRGS, através de seu núcleo de doutores, doutorandos, mestrados e pesquisadores tem desenvolvido, no Laboratório de processamento de Sinais e Imagem – LaPSI, alguns trabalhos que, pelos seus resultados e forma como são encaminhados, aguçam a curiosidade e estimulam o interesse de pessoas que o acompanham.

Buscando resolver o problema do controle de trânsito que preocupa a todos e observando as limitações dos sistemas tradicionais, surgiu a idéia de aplicar os conhecimentos adquiridos durante PPGEE, na área de processamento de imagens, que me identifiquei bastante, para desenvolver um sistema capaz de supervisionar o trânsito. A idéia inicial foi unir meu interesse em aprofundar conhecimentos relacionados à área de processamento de imagens a concepção de um sistema que poderia ser útil à sociedade.

## 2 PRINCÍPIOS DO PROCESSAMENTO DE IMAGEM

### 2.1 REPRESENTAÇÃO DIGITAL DA IMAGEM

De acordo com (GONZALEZ, 1993), uma imagem pode ser considerada uma função bidimensional  $f(x,y)$ , onde  $x$  e  $y$  são as coordenadas espaciais de linha e coluna, respectivamente e  $f$  é proporcional ao tom de cinza em uma imagem monocromática, ou é um vetor onde os componentes possuem informação de cor. O elemento básico de uma imagem digital é o pixel, que vem do inglês *picture element* ou elemento da figura. A figura 2 mostra a convenção espacial utilizada neste trabalho para representar a localização dos pixels em uma imagem digital.



**Figura 2 - Convenção de eixos utilizada para representar uma imagem digital**

Tanto a imagem digital monocromática como a colorida, possuem as coordenadas espaciais e informações de iluminação do pixel, discretizadas. Na imagem monocromática, a intensidade de iluminação do pixel pode variar entre os valores 0 e 255, onde zero representa ausência de iluminação e 255 o valor máximo de brilho no ponto.

De acordo com (CASTLEMAN, 1996), quando o nível de iluminação do ponto, em função de duas coordenadas espaciais torna-se função de três variáveis espectrais de cor, o processamento é classificado como processamento de imagens coloridas. A imagem digital colorida pode ser entendida como uma função tridimensional, escalar, de três coordenadas: duas espaciais e uma espectral. Para algumas aplicações pode ser mais conveniente tratar a imagem colorida como uma função bidimensional que possui duas coordenadas espaciais e uma de cor onde a informação de cor normalmente é fornecida a partir de três componentes com significado que varia de acordo com o modelo utilizado para representação de cores.

## 2.2 MODELOS MAIS USADOS PARA REPRESENTAÇÃO DE CORES

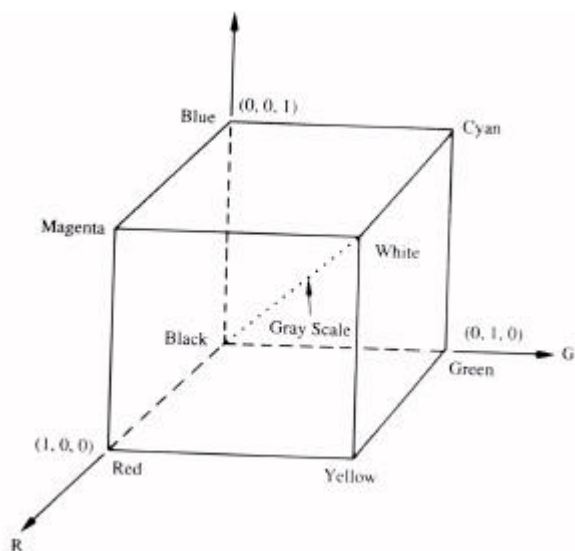
Para padronizar a representação de cores em uma imagem alguns modelos foram criados. Um modelo de cor é uma especificação em um sistema tridimensional de coordenadas e um sub espaço deste sistema onde cada cor é representada por um único ponto. Segundo (GONZALEZ, 1993), Muitos modelos de cor utilizados hoje foram criados objetivando hardware, tais como, monitores e impressoras, outros foram criados para aplicações onde a manipulação de cores é o foco, tal como criação de cores para animações gráficas. Os modelos de cores mais utilizados para processamento de imagens são: RGB, YIQ e HSI.

O modelo RGB (*Red, Green e Blue*), utilizado em monitores coloridos e modelos importados de vídeo câmeras, tem sua sigla oriunda das cores vermelho, verde e azul, respectivamente. O modelo YIQ é o padrão para transmissão de TV colorida aberta , onde Y corresponde a iluminância, I e Q são as duas componentes monocromáticas chamadas de fase e quadratura, respectivamente. Este modelo é uma decodificação do RGB para eficiência de transmissão e para manter a compatibilidade como padrão de televisão monocromática. De fato a componente Y fornece toda a informação de vídeo necessária para televisão monocromática. O modelo HSI tem sua sigla do inglês: *Hue, Saturation e Intensity* ou matriz, saturação e brilho (JAE, 1990). A matriz corresponde ao atributo que descreve a cor pura (amarelo, laranja ou vermelho), a saturação informa o quanto esta cor está diluída em branco e a intensidade, como diz o nome, informa a intensidade da cor. Este modelo é ideal para desenvolver algoritmos para processamento de imagens baseados em alguma propriedade relacionada à sensibilidade de cores do olho humano e isto se deve a dois fatos: o primeiro é que a componente de intensidade esta desacoplada da informação de cor da imagem e o segundo é que as componentes de matriz e saturação estão intimamente ligadas à forma como o olho humano percebe a cor.

### 2.2.1 O formato RGB

De acordo com (GONZALEZ, 1993), neste modelo as cores são representadas por seus componentes espectrais primários: o vermelho o verde e o azul. O subespaço de cores de interesse pode ser representado pelo cubo ilustrado na figura 3, onde este modelo é mostrado

com base em um sistema de coordenadas cartesianas. Os valores RGB estão em três vértices e as cores ciano, magenta e amarelo estão em outros três vértices do cubo.



**Figura 3 – Cubo de cores do modelo RGB.**

Fonte: (GONZALEZ, 1993, p.226).

O preto é representado na origem do cubo e o branco no vértice mais distante da origem. Neste modelo a escala de cinza se estende do preto ao branco ao longo da linha que une estes dois pontos. A percepção subjetiva dos valores de cinza é não linear e corresponde a curva conhecida por correção gama (KOYA, 2001).

No cubo RGB todos os valores de cores estão normalizados, assim o cubo mostrado na figura 3 é o cubo unitário, ou seja, todos os vetores R, G e B são assumidos na faixa  $[0,1]$ .

Imagens digitais no modelo de cores RGB consistem em três planos de imagem monocromática independentes, e sobrepostos, sendo um plano para cada cor primária. No monitor de vídeo estas cores são combinadas e produzem uma imagem de cores compostas. Cada uma das cores que compõem o formato RGB pode variar de 0 à 255, assim, cada pixel pode assumir um entre 16.777.216 ( $256 \times 256 \times 256$ ) valores possíveis de cor. Para cada pixel na posição  $(x,y)$  deve ser informado os valores da tríade de cores na forma: vermelho, verde, azul (KOYA,2001; CASTLEMAN, 1996).

## 2.3 OPERAÇÕES MORFOLÓGICAS

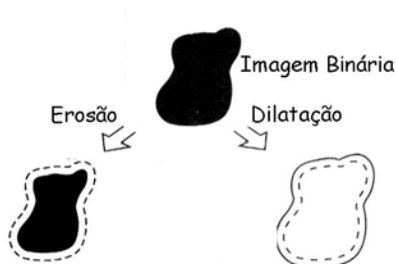
Para Castleman (1996), o objetivo da morfologia matemática é extrair informações relativas à geometria e topologia de um conjunto não conhecido que é a imagem, através de outro conjunto, totalmente definido: o elemento estruturante. Conhecendo-se a forma e o tamanho do elemento estruturante é possível determinar quanto e como ele está contido na imagem. O resultado desta interação vai depender do elemento estruturante e da imagem analisada.

Segundo Gonzalez (1993), a linguagem da morfologia matemática é a teoria dos conjuntos, onde os conjuntos representam a forma dos objetos em uma imagem. Nas imagens binárias os conjuntos em questão pertencem ao espaço bidimensional inteiro  $Z^2$  e cada elemento do conjunto é um vetor 2-D, cujas coordenadas  $(x,y)$  são, por convenção, as coordenadas dos pixels pretos na imagem. Imagens digitais em níveis de cinza podem ser representadas por conjuntos cujos componentes pertencem ao espaço  $Z^3$ . Neste caso, dois componentes de cada elemento do conjunto se referem às coordenadas espaciais do pixel e o terceiro corresponde à intensidade discreta do pixel.

De acordo com Koya (2001), na morfologia de imagens em tons de cinza, é necessário conhecer o valor do pixel mais escuro, do pixel mais claro e do pixel central, na vizinhança de cada pixel ou em parte da imagem original. O pixel resultante corresponde a uma combinação entre os pixels mais escuro, o mais claro e o central. O tamanho, a forma da vizinhança, as regiões de pesquisa e o algoritmo determinam completamente uma operação de morfologia cinza. Assim, as operações morfológicas mudam em função do tipo de imagem a ser processada. Conjuntos de maior ordem podem conter informações adicionais. Neste trabalho as operações morfológicas são aplicadas a imagens binárias e utilizadas no pós-processamento para realçar o objeto segmentado, eliminando ruídos do tipo espúrios ocasionados pela trepidação da câmera. As imagens foram analisadas considerando o objeto branco sobre fundo preto.

As operações morfológicas básicas são a erosão e a dilatação. Na primeira os pixels formando um determinado padrão são apagados da imagem e na segunda uma pequena área do conjunto de pixels é ajustada para uma determinada forma, de acordo com o objeto e o elemento estruturante (PARKER, 1997).

A figura 4 ilustra as operações de Dilatação e erosão.



**Figura 4 – Dilatação e erosão de imagem binária.**

Fonte: Castleman, (1997, p305).

### 2.3.1 Dilatação binária

Com os conjuntos  $A$  e  $B \subset Z^2$  e  $\emptyset$  representando o conjunto vazio, a dilatação de  $A$  por  $B$  é notada por  $A \hat{\oplus} B$  ou  $A \text{ dil } B$  é definida como:

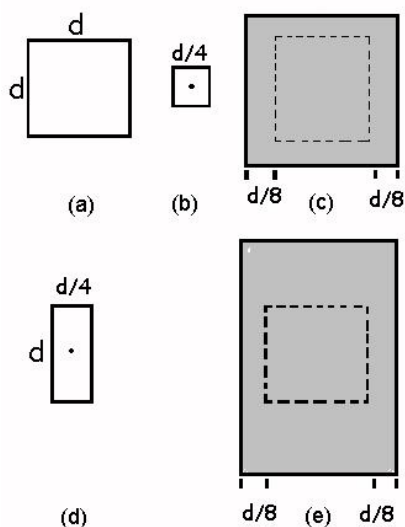
$$A \oplus B = \left\{ x \mid \hat{(B)}_x \cap A \neq \emptyset \right\} \quad (1)$$

Onde  $\hat{(B)}_x$  é o conjunto  $B$  refletido sobre a origem e após deslocado por  $x$ . Analisando desta forma, a dilatação de  $A$  por  $B$  é a reflexão de  $B$  sobre sua origem e este resultado é transladado por  $x$ . Assim a dilatação é definida como o conjunto de todos deslocamentos tais que  $B$  e  $A$  se sobreponham por pelo menos um elemento diferente de zero. A partir desta análise a equação 1 é re-escrita e mostrada na equação 2.

$$A \oplus B = \left\{ x \mid [(\hat{B})_x \cap A] \subseteq A \right\} \quad (2)$$

O conjunto  $B$  é chamado elemento estruturante nas operações morfológicas. A figura 5.a mostra um simples conjunto  $A$  e a 5.b mostra o elemento estruturante e sua reflexão. Como o conjunto  $B$  é simétrico, em relação à origem,  $B$  e seu complemento são iguais. As linhas tracejadas em 5.c mostram o conjunto original  $A$  para referência e a linha cheia mostra o limite além do qual qualquer deslocamento de  $B$  por  $x$  causaria a interseção de  $B$  e  $A$ . A figura 5.d mostra um elemento estruturante cuja finalidade é atingir maior dilatação vertical do que horizontal. A figura 5.e mostra a dilatação atingida com este elemento. A

dilatação, como o nome diz, dilata, expande o objeto, considerando o objeto branco no fundo preto.



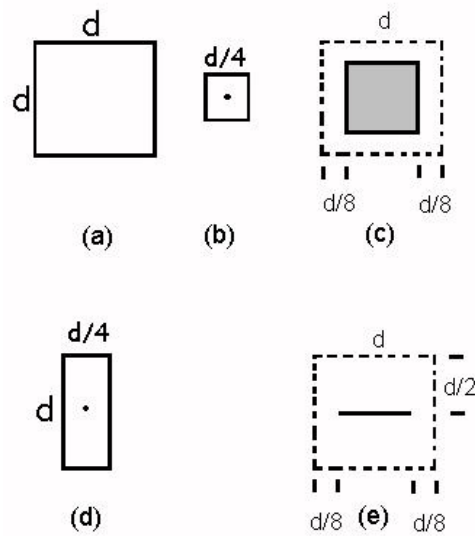
**Figura 5 – Dilatação binária.** a) Imagem original. b) Elemento estruturante com estrutura quadrada e sua reflexão. c) Área sombreada indica dilatação de  $A$  por  $B$ . d) Elemento estruturante de forma alongada. e) Dilatação de  $A$  utilizando o elemento estruturante alongado.  
 Fonte: Gonzalez (1993, p.522).

### 2.3.2 Erosão binária

Se a dilatação faz o objeto ficar maior, a erosão faz o contrário: diminui o tamanho do objeto. A erosão de  $A$  por  $B$  é notada por  $A \ominus B$  e definida como:

$$A \ominus B = \{x \mid (B)_x \subseteq A\} \quad (3)$$

Segundo Parker (1997), erosão é o conjunto de todos os pontos  $x$ , tal que,  $B$  transladados por  $x$  está contido em  $A$ . O elemento estruturante  $B$  desliza sobre a imagem  $A$  de modo que, posicionado e centrado em cada pixel  $x$  de  $A$ , haja uma comparação com a vizinhança de  $x$ . O resultado de uma erosão é um subconjunto da imagem original. Para explicitar melhor: qualquer pixel que não encontre o padrão formado pelos pixels pretos do elemento estruturante não pertencerá ao resultado. A figura 6 ilustra o processo de erosão. O conjunto  $A$  é mostrado tracejado no resultado, como referência. As linhas sólidas mostram o limite além do qual qualquer deslocamento adicional da origem de  $B$  faria com que o resultado não ficasse completamente contido em  $A$ . Em 6.c o resultado da erosão de 6.a pelo elemento estruturante quadrado em 6.b. A figura 6.d mostra um elemento estruturante alongado e em 6.e o resultado da erosão de 6.a pelo elemento estruturante de 6.d.



**Figura 6 – Erosão binária.** a) Imagem original. b) Elemento estruturante com estrutura quadrada e sua reflexão. c) Área sombreada indica erosão de A por B. d) Elemento estruturante de forma alongada. e) Erosão de A utilizando o elemento estruturante alongado. Fonte: Gonzalez (1993, p.523).

As operações de dilatação e erosão são duais uma em relação (GONZALEZ, 1993) à outra, no que diz respeito ao ajuste de complementação e reflexão que são feitos.

$$(A \ominus B)^c = A^c \oplus \hat{B} \quad (4)$$

### 2.3.3 Abertura e fechamento

Segundo (PARKER, 1996), a aplicação de uma erosão imediatamente seguida de uma dilatação, utilizando o mesmo elemento estruturante é definida como abertura. A abertura do conjunto A pelo elemento estruturante B é notada por  $A \circ B$  e definida como:

$$A \circ B = (A \ominus B) \oplus B \quad (5)$$

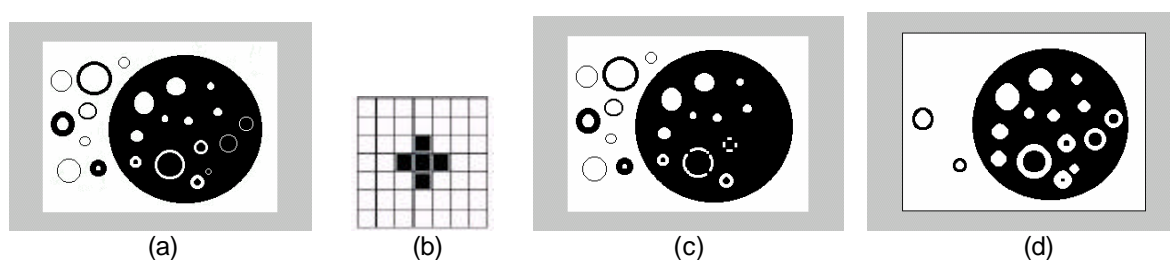
Em outras palavras, a abertura de A por B consiste na erosão de A por B e o resultado é dilatado por B. Já o fechamento do conjunto A pelo elemento estruturante B, notado por  $A \bullet B$ , é definido como:

$$A \bullet B = (A \oplus B) \ominus B \quad (6)$$



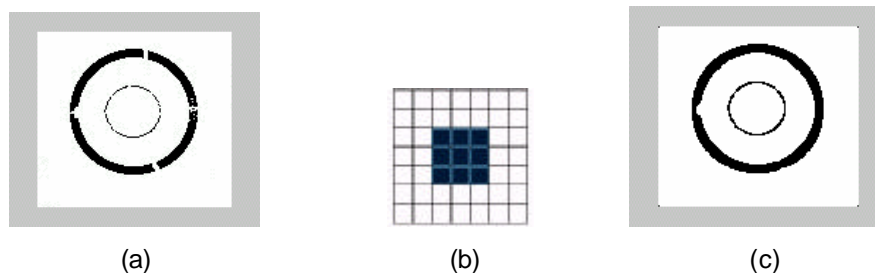
Assim, conforme mostra a equação 6, o fechamento consiste em dilatar  $A$  por  $B$  e erodir o resultado utilizando o mesmo elemento estruturante  $B$ . De acordo com (KOYA, 2001), objetos que sofrem erosão diminuem seu tamanho e objetos que sofrem dilatação aumentam seu tamanho. A propriedade da interatividade, das operações de abertura e fechamento, garante a preservação das características de forma e tamanho dos conjuntos.

Segundo (PARKER, 1996), a operação de abertura se caracteriza por suavizar o contorno de imagens, quebrar delgados objetos e eliminar pequenas protuberâncias. O fechamento também tende a suavizar secção de contorno, mas ao contrário da abertura, normalmente liga pequenas falhas do contorno. O fechamento também tem a função de eliminar pequenos buracos dentro do objeto. A figura 7 ilustra as operações de abertura e fechamento.



**Figura 7 – Operações morfológicas de abertura e fechamento. a) Imagem original. b) Elemento estruturante. c) Resultado da abertura. d) Resultado do fechamento.**

Círculos brancos sobre fundo preto cuja espessura da linha fosse menor que o elemento estruturante seriam eliminados na operação de abertura, já os que possuíam a espessura da linha apenas ligeiramente maior que o elemento estruturante tiveram seus contornos comprimidos em alguns pontos. Comparando-se as figuras 7.a e 7.d, observa-se que a operação de fechamento eliminou os círculos pretos sobre o fundo branco cuja espessura da borda era menor que do elemento estruturante e fechou as partes internas pretas de dois círculos cujo diâmetro também era menor que o elemento estruturante. Na figura 8 é mostrada uma aplicação para operação morfológica de fechamento: completar falhas de contornos, tornando-os fechados.



**Figura 8 – Fechamento de falhas em contornos com operações morfológicas**  
**a) Imagem original. b) Elemento estruturante. c) Resultado do fechamento.**

## 2.4 RESTAURAÇÃO DE IMAGENS

Capturar uma imagem exatamente como ela aparece no mundo é muito difícil, senão impossível. Existe ruído adicionado, que no caso de uma fotografia pode ser causado pelos grânulos da emulsão, pelo borrado devido à trepidação da câmera ou movimentação do objeto, por problemas de foco, defeito construtivos no sistema de lentes, pelo ruído de digitalização em imagens digitais, ou mesmo por problemas de iluminação (PARKER, 1997). A restauração é a arte de melhorar a qualidade de uma imagem baseando-se em algumas medidas absolutas e normalmente está relacionada com desfazer alguma distorção que foi imposta à imagem. A restauração, embora dificilmente deixe a imagem perfeita, permite melhoramentos em várias circunstâncias.

Segundo Castleman (1996), a filtragem está associada a operações matemáticas que permitem eliminar ou acentuar elementos construtivos em uma imagem. A filtragem pode ser no domínio espacial, no domínio frequência, linear ou não linear.

Uma imagem pode ser transposta para o domínio frequência. A transposição é um simples mapeamento de um conjunto de coordenadas para outro.

A transformada de Fourier (TF) permite decompor uma função contínua numa soma de funções periódicas. A TF converte coordenadas espaciais para e do domínio frequência. Qualquer curva ou superfície pode ser expressa como a soma de senos e co-senos (GONZALEZ, 1993; PARKER, 1997). O domínio frequência é chamado em função de que os dois parâmetros de uma curva senoidal são amplitude e frequência. O fato de uma imagem ser convertida para o domínio frequência implica que pode conter componentes de alta e baixa frequência, portanto pode ser filtrada. É possível construir filtros para remover ou realçar certas frequências e com isto pode-se obter muitas vezes o efeito restaurativo (JAE, 1990).

Considerando-se uma função  $I(x,y)$ , que representa os valores de intensidade luminosa do pixel nas coordenadas  $(x,y)$ , sua transformada de Fourier é definida na equação 7.

$$F\{I(x, y)\} = F(u, v) = \int I(x, y) e^{-j2\pi(ux,vy)} dx dy \quad (7)$$

A partir do momento em que as imagens estão decompostas no domínio espectral, pode-se, por exemplo, selecionar todas frequências inferiores a um limiar construindo-se, assim, um filtro passa-baixas. As operações de filtragem são executadas com menor tempo computacional que operações no domínio espaço, no entanto, a transposição de uma imagem do domínio espaço pra o domínio frequência é um processo que consome muito tempo computacional em computadores pessoais.

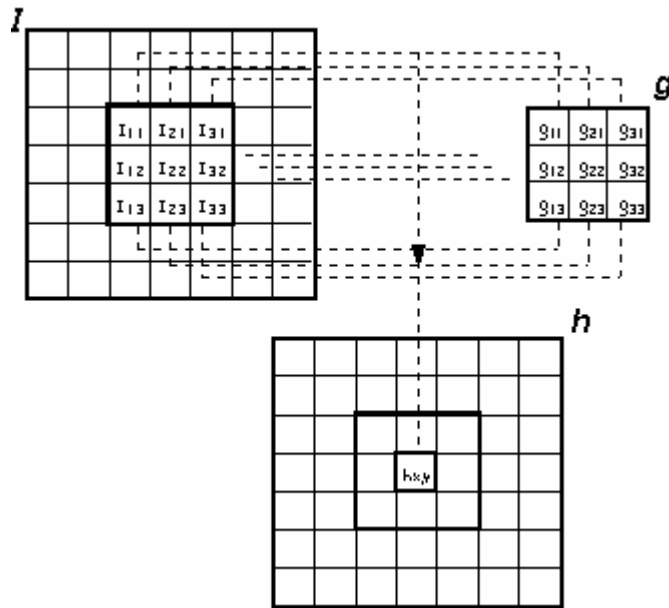
A filtragem linear no domínio espacial é feita pela convolução que teoricamente é o produto entre duas funções. A convolução é definida pela integral mostrada na equação 8.

$$h(x) = I * g = \int_{-\infty}^{+\infty} I(u)g(x-u)du \quad (8)$$

Considerando-se imagens bidimensionais discretas, considerando  $I$  a função que representa a imagem original,  $h$  a imagem filtrada e  $g$  a função filtro, a convolução da imagem com o filtro é descrita na equação 9.

$$h(x, y) = \sum_{u=0}^{a-1} \sum_{v=0}^{b-1} I(u, v)g(x-u, y-v) \quad (9)$$

A convolução é uma técnica que pode ser usada para efetuar várias funções em processamento de imagens, como suavização, deslocamento, detecção de bordas etc. A representação gráfica da convolução é mostrada na figura 9.



**Figura 9 – Diagrama da Convolução da máscara com a imagem.**

Onde  $I_{i,j}$  denota as observações da imagem,  $h_{i,j}$  denota os valores da máscara de convolução ou janela, mostrada como  $h$  na figura 9.

Os elementos desta matriz são multiplicados pelo valor do pixel correspondente nas vizinhanças da imagem original. Os resultados das multiplicações são somados e divididos pelo número de células da máscara, formando a imagem convolucionada resultante  $h$ .

A filtragem não linear é operacionalizada sem a utilização de um operador externo. Isto é realizado pela modificação de cada um dos pixels que compõem a imagem através de uma combinação ou estimativa estatística dos pixels vizinhos. Esta técnica de filtragem, também chamada suavização, consiste em substituir um pixel pela média, ou variância dos pixels da vizinhança. A suavização através da utilização de filtros não lineares como a filtragem pela gaussiana ou pela média dos pixels vizinhos tende a borrar as bordas porque as componentes de alta frequência da imagem são atenuadas. A suavização pela gaussiana é utilizada na fase de suavização do detector de bordas Canny. Uma aproximação alternativa, para suavizar uma imagem sem destruir suas bordas, é a mediana que é um filtro não linear muito utilizado. Por sua característica de eliminar ruídos impulsivos sem alterar significativamente o contorno da imagem. Isto vem do fato que o valor do ruído não interfere no cálculo dos pixels transformados. Este filtro tem a vantagem de não introduzir valores na imagem transformada, visto que o valor de saída é obtido a partir da mediana dos próprios valores de entrada. A filtragem consiste em selecionar um valor mediano dentre os valores de pixels da vizinhança. A mediana  $m$  de um conjunto de valores é tal que metade dos valores do conjunto são menores que  $m$  e a outra metade maiores que  $m$ . Supondo que os valores dos

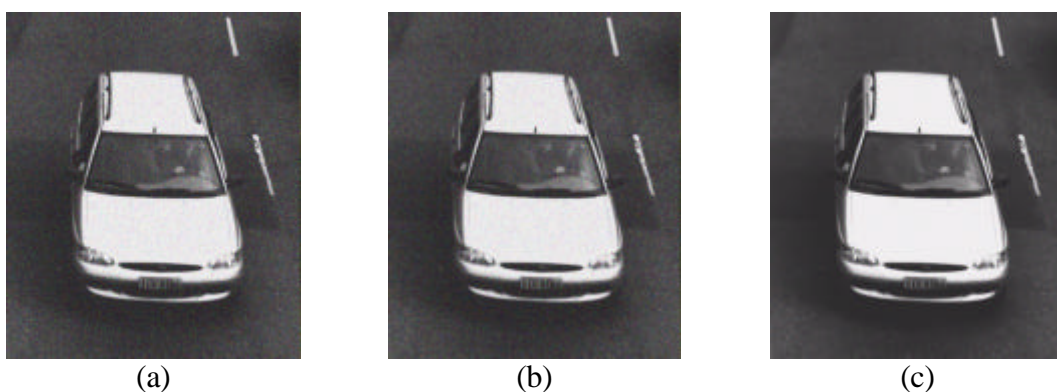
pixels em uma vizinhança 3x3 sejam:

(100, 25, 20, 15, 20, 20, 20, 20, 10).

Então, colocando-se os valores em ordem crescente tem-se:

(10, 15, 20, 20, 20, 20, 25, 100)

O valor mediano do conjunto é 20. Assim obtêm-se como resultado da filtragem a eliminação de pixels que têm valores muito afastados de seus vizinhos, ao mesmo tempo em que as bordas da imagem são preservadas. A figura 10 ilustra o resultado obtido com duas filtragens não lineares.



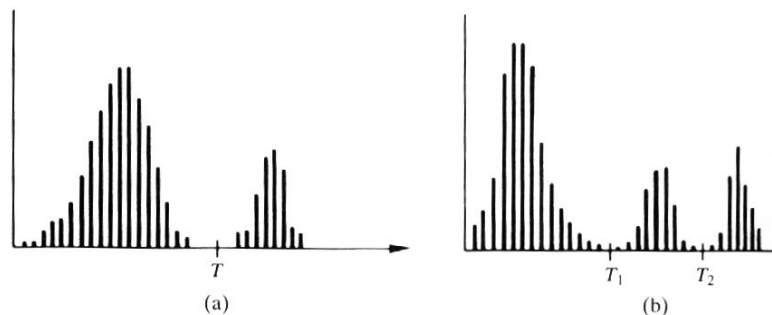
**Figura 10 – Suavização da imagem. a) imagem com adição de 30% de ruído aleatório. b) Imagem filtrada com a média dos pixels da região. c) Imagem filtrada pelo filtro mediana.**

Comparando-se os resultados obtidos nas figuras 10.b e 10.c, observa-se que a filtragem utilizando o filtro mediano é mais adequada para eliminação do ruído aleatório *salt-and-pepper*, pois elimina este tipo de ruído sem comprometer as bordas da imagem.

O filtro que utiliza a média dos pixels da vizinhança não consegue eliminar todo o ruído e suaviza a imagem, deteriorando levemente suas bordas. A filtragem utilizando a média é operacionalizada através da convolução da imagem com uma máscara 3x3 com todos coeficientes 1.

## 2.5 DETERMINAÇÃO DO LIMIAR PARA BINARIZAÇÃO

Segundo Gonzalez (1993), thresholding é um dos mais importantes métodos para segmentação de imagens. Supondo que o histograma de uma imagem, na escala de cinza, corresponda a  $f(x,y)$ , composta de objetos claros em um fundo escuro e que os pixels do objeto e do fundo tenham tons de cinza agrupados em dois modos dominantes, conforme mostrado na figura 11.a.



**Figura 11 – Histogramas de nível de cinza. a) Histograma com dois modos dominantes. b) Histograma com três modos dominantes**

Fonte: GONZALEZ, 1993, p 444

Uma maneira óbvia de extrair os objetos do fundo é selecionar um limiar (threshold),  $th$  que separe estes modos. Considerando o objeto branco sobre um fundo preto, qualquer ponto  $(x,y)$  para qual  $f(x,y) > th$  é chamado ponto do objeto e os demais pontos são chamados pontos do fundo. Se três, ou mais modos dominantes caracterizam o histograma da imagem, como por exemplo, dois tipos de objetos claros em um fundo escuro, as vezes é possível segmentar a imagem através de 2 níveis de limiar, conforme mostra a figura 11.b. Geralmente a segmentação com três, ou mais limiares é menos confiável do que a com um nível, pela dificuldade em estabelecer-se valores de limiar que isolem regiões de interesse. A iluminação que tem papel fundamental nas características do histograma da imagem resultante. Para (CASTLEMAN, 1996), a segmentação por limiar utiliza uma técnica orientada à região útil para imagens que contenham objetos sólidos colocados sobre um fundo contrastante. Este é um método, computacionalmente simples e nunca falha, para definir regiões distintas da imagem.

De acordo com (GONZALEZ, 1993), a escolha do limiar de uma imagem pode ser vista como uma operação que envolve testes com uma função de  $T$ , conforme mostra a equação 10.

$$T = T[x, y, p(x, y), f(x, y)] \quad (10)$$

Onde  $f(x,y)$  é o nível de cinza no ponto  $(x,y)$  e  $p(x,y)$  é uma propriedade local deste ponto, como por exemplo, a média de níveis de cinza dos vizinhos centrados em  $(x,y)$ . Quando  $th$  depende exclusivamente de  $f(x,y)$  a técnica de limiar é dita global. Se o limiar ( $th$ ) depende tanto de  $f(x,y)$  como de  $P(x,y)$  a técnica é chamada local. Além disso, quando o limiar que depende de coordenadas espaciais  $(x,y)$  é chamado dinâmico.

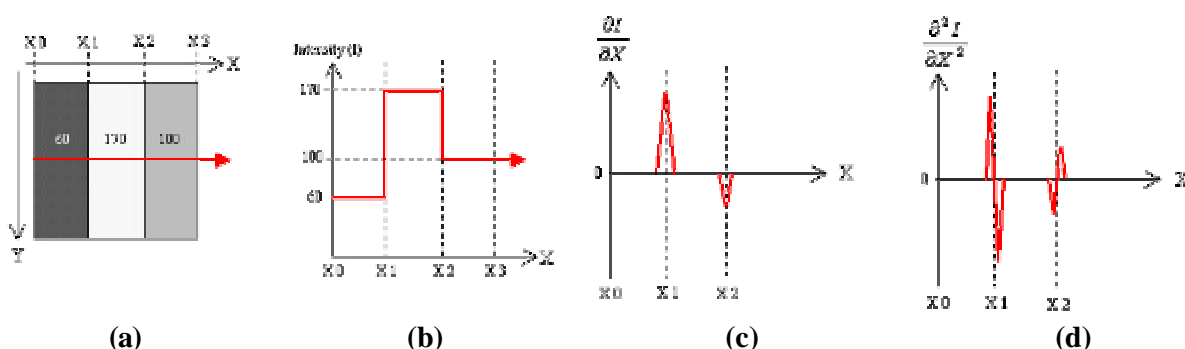
Para encontrar o limiar de erro mínimo, uma imagem é submetida à função de medida de densidade de probabilidade, média e desvio padrão de suas distribuições de objetos e fundo (PARKER, 1996). Este método, chamado estatístico, é adequado principalmente quando as distribuições de probabilidade estão sobrepostas, ou seja, possuem pixels com mesmo nível de cinza em comum.

Outra técnica é a chamada limiar ótimo considera sua escolha, na escala de cinza, através de técnicas de histograma e limiar adaptativo (CASTLEMAN, 1996). Há casos em que o nível de cinza do fundo não é constante e o contraste do objeto varia ao longo da imagem. Em tais casos o limiar funciona bem em algumas regiões e deixa a desejar em outras. Nestes casos é conveniente que se utilize um limiar que varie em função das coordenadas da imagem. Esta técnica de limiar é chamada limiar adaptativo. A escolha do limiar influencia a posição das bordas do objeto, assim, a área dos objetos segmentados é sensível a este valor. Outro método para determinação do limiar é o laplaciano, onde a transformada de Laplace da imagem é utilizada como mapa determinar os pontos que serão analisados para escolha do limiar.

## 2.6 DETECÇÃO DE BORDAS

Segundo (GONZALEZ, 1993), embora detecção de pontos e linhas seja um assunto que têm dispensado atenção, a detecção de bordas é de longe a aproximação mais comum para encontrar descontinuidades entre regiões de imagens em escala de cinza. A detecção de bordas é usada para delinear o contorno de objetos, ou formas, em uma imagem. Ao contrário do realce de bordas, que apenas torna as bordas mais visíveis, aqui é feita a localização de todas as bordas significantes.

Basicamente a idéia por trás da maioria das técnicas para de detecção de bordas é a computação do operador local derivativo em uma imagem que apresenta variações de intensidade luminosa, conforme mostra a figura 12. Os métodos mais comuns utilizados para detecção de bordas são Prewitt, Sobel e Laplace.



**Figura 12 – Variação de intensidade luminosa e suas derivadas. a) Imagem com variações de intensidade luminosa na escala de cinza. b) Gráfico de variação da intensidade luminosa. c) Derivada de 1ª ordem da variação de Intensidade luminosa: Gradiente. d) Derivada de 2ª ordem da variação de Intensidade luminosa: Laplaciano.**

### 2.6.1 Operadores Gradiente

Este método está baseado na aplicação do operador local derivativo. Aplicando-se um operador gradiente à imagem da figura 12.a resulta na imagem obtida em 12.c. A primeira derivada de uma imagem em tons de cinza é positiva na borda de subida, negativa na borda de descida e zero em áreas onde o nível de cinza é constante, portanto a magnitude da primeira derivada pode ser usada para detectar a presença de bordas na imagem.

A taxa de variação de intensidade luminosa de um pixel para outro, neste contexto, é o gradiente (JAE, 1990). Em termos matemáticos o gradiente é equivalente à diferenciação com respeito à posição, por isto é, também chamado, filtro derivativo.

O Operador **Sobel** usa a magnitude do gradiente para julgar a probabilidade de cada pixel fazer parte da borda. O operador gradiente é vetorial e pode ser representado, matematicamente como:

$$\nabla A(x, y) = \left( \frac{dA}{dx}, \frac{dA}{dy} \right) \quad (11)$$



Conforme pode ser observado na equação 11, a imagem é derivada em relação à  $x$  e também em relação à  $y$ , logo o operador é sensível a direção. Em sua implementação as máscaras serão apresentadas sempre em dupla: uma para implementar o operador em relação à  $x$  e outra para implementá-lo em relação à  $y$ .

A magnitude do vetor gradiente é o comprimento da hipotenusa do triângulo cujos lados são  $\nabla x$  e  $\nabla y$  e isto reflete a intensidade da borda em um dado pixel. A direção da borda, no mesmo pixel, é o ângulo que a hipotenusa faz com o eixo. Matematicamente a magnitude e a direção do vetor gradiente é dada por:

$$G_{mag} = \sqrt{\left(\frac{\partial A}{\partial x}\right)^2 + \left(\frac{\partial A}{\partial y}\right)^2} \quad (12)$$

$$G_{dir} = \tan^{-1} \left( \frac{\frac{\partial A}{\partial y}}{\frac{\partial A}{\partial x}} \right) \quad (13)$$

De acordo com (GONZALEZ, 1993), assim como a média dos pixels de uma região, que pode ser associada a uma integração, tende a borrar detalhes da imagem, a derivação tem efeito contrário: deixa a imagem com detalhes mais acentuados. O gradiente pode, também, ser obtido através de operações morfológicas.

Os operadores derivativos podem ser aplicados através de métodos no domínio espacial, realizados através da convolução de máscaras com a imagem de entrada, assim o valor do pixel nas coordenadas  $(x,y)$  na imagem realçada é resultado da operação de convolução da imagem de entrada com os pixels da máscara. A tabela 1 mostra os operadores gradiente mais comuns. A vizinhança pode ter qualquer forma, mas geralmente é utilizada a forma retangular.

**Tabela 1 - Máscaras para implementação do operador gradiente.**

	$H_1$	$H_2$
<b>Roberts</b>	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
<b>Prewitt</b>	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
<b>Sobel</b>	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

Fonte: GONZALEZ (1993).

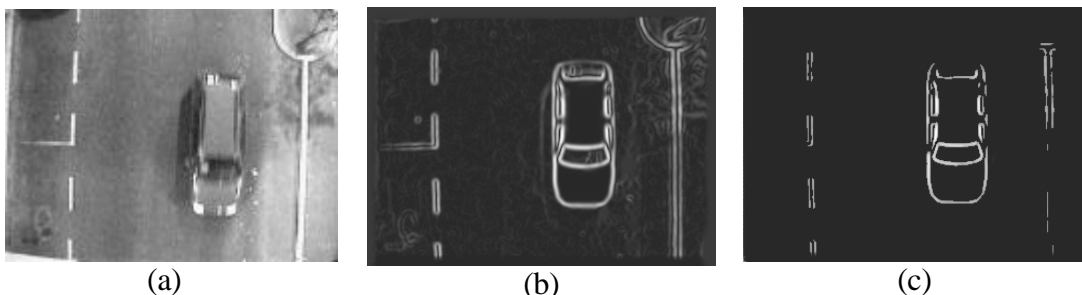
A soma dos coeficientes de todas as máscaras é zero, indicando resposta zero em áreas constantes, como é esperado de um operador derivativo. Um pixel é identificado como parte da borda se ultrapassar um limiar. As posições dos pontos de borda constituem o mapa de bordas  $e(x, y)$ , definido como:

$$e(x, y) = \begin{cases} 1, (x, y) \in I_g \\ 0, \text{senão} \end{cases} \quad (14)$$

onde

$$I_g \triangleq \{(x, y); g(x, y) > th\} \quad (15)$$

O limiar ( $th$ ) pode ser selecionado através da análise do histograma do gradiente da imagem de modo que pixels que apresentam maiores gradientes e estejam conectados sejam declarados como bordas. A figura 13 ilustra o gradiente e o mapa de bordas de uma imagem em tons de cinza.



**Figura 13 - Operador Gradiente Sobel. a) imagem original. b) Gradiente da imagem original. c) Mapa de bordas**

## 2.6.2 Operadores Laplaciano

Operadores derivativos de primeira ordem funcionam bem quando a variação dos níveis de cinza é abrupta, como a função degrau. À medida que esta região de transição se torna mais larga é mais vantajoso aplicar-se derivadas de segunda ordem. Um operador bastante encontrado na literatura é o Laplaciano, definido como:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (16)$$

O operador Laplaciano utiliza a derivada de segunda ordem (JAE, 1990), ou o gradiente do gradiente. A figura 12.d ilustra a aplicação do operador Laplaciano na imagem mostrada em 12.a. As bordas são indicadas no ponto de cruzamento por zero.

O operador Laplaciano pode ser implementado digitalmente de várias formas. A forma mais freqüente utiliza uma máscara 3X3 para convolução com a imagem de modo que:

$$\nabla^2 f = 4 * z_5 - (z_2 + z_4 + z_6 + z_8) \quad (17)$$

Os operadores Laplacianos geram um escalar, por isto utilizam apenas uma máscara. A tabela 2 mostra algumas máscaras para aplicar o operador Laplaciano. A condição para definir a máscara para o operador laplaciano é que a soma dos valores da máscara seja zero. As bordas de uma imagem operada com o Laplaciano são obtidas pela detecção de passagem por zero.

Segundo (GONZALEZ, 1993), como o laplaciano é um operador derivativo de 2ª ordem, tipicamente tem alta sensibilidade ao ruído, além disso, produzindo bordas duplas e não conseguindo detectar sua direção. Sua utilização para detectar os cruzamentos por zero é eficiente para detecção de bordas.

**Tabela 2- Operadores discretos de Laplace**

$$\text{a) } \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\text{b) } \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\text{c) } \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Fonte: Jae (1990).

### 2.6.3 Método Canny

De acordo com Parker (1996), um detector de bordas deve ter três características básicas:

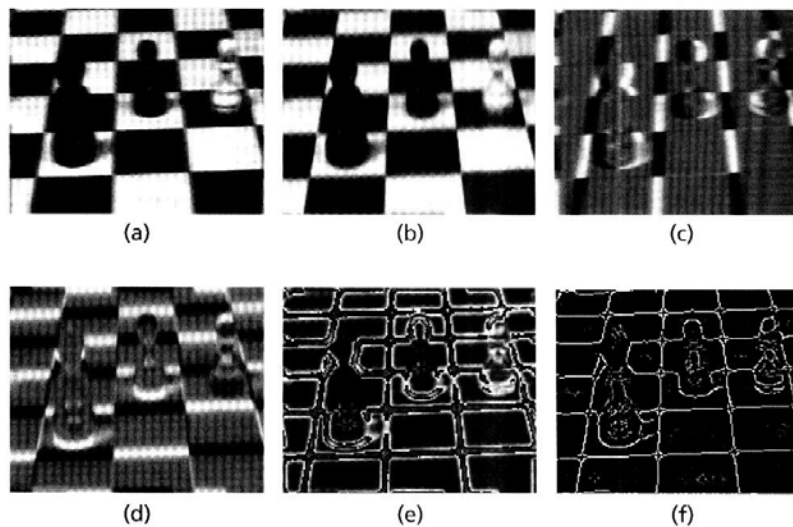
- a) responder apenas as bordas encontrando todos os pixels sem perder algum;
- b) fornecer a menor distância possível entre a borda encontrada e a borda real;
- c) não detectar múltiplas bordas onde existe apenas uma.

O detector de bordas Canny foi concebido como tendo um filtro de convolução que deve suavizar o ruído e localizar as bordas. O tamanho da máscara de convolução do filtro depende do desvio padrão determinado. Assim seu trabalho foi tentar encontrar um filtro que maximizasse o produto sinal/ruído *versus* localização. Segundo (BENSEBAA, 2000), o processamento de imagens usando o detector de bordas Canny é executado em quatro estágios: Suavização com o filtro gaussiano, computação do gradiente, supressão de não máximos e limiar. Embora a convolução bi-dimensional da imagem com a gaussiana seja muito pesada em termos computacionais, pode-se realizar a operação através em duas etapas com uma gaussiana de uma dimensão. O filtro da gaussiana e sua derivada, em uma dimensão, são mostrados nas equações 18 e 19, respectivamente.

$$G_s(x) = \frac{1}{\sqrt{2ps}} e^{\frac{-x^2}{2s^2}} \quad (18)$$

$$G'_s(x) = \left( -\frac{x}{s^2} \right) e^{\frac{-x^2}{2s^2}} \quad (19)$$

Assim a imagem de entrada é convolvida em ambas direções com  $G'_s(x)$  e  $G'_s(y)$ , resultando em uma imagem de saída com as bordas realçadas, mesmo na presença de ruído. O último passo consiste em remover os pixels que não são máximos locais. A supressão de pixels não máximos significa, que o pixel central que está sendo considerado, deve ter a magnitude do gradiente maior que seus vizinhos na direção do gradiente. A magnitude do gradiente é maior nos pixels das bordas. A figura 14 ilustra o processo Canny para detecção de bordas.



**Figura 14 – Resultados intermediários do método Canny. a) Componente x da convolução com a Gaussiana. b) Componente y da convolução com a Gaussiana. c) Componente x da imagem convolvida com a derivada da Gaussiana. d) Componente y da imagem convolvida com a derivada da Gaussiana. e) Imagem de magnitude resultante. f) Imagem após a supressão dos não máximos.**

Fonte: Parker (1997, p.28)

Na figura 14.e tem-se a magnitude sem a aplicação da supressão de máximos e a figura 14.f mostra como fica esta imagem após a aplicação da supressão de não máximos. Observa-se que as bordas duplas são eliminadas. A última imagem ainda contém tons na escala de cinza, sendo assim, é fortemente recomendado a aplicação de um limiar para tornar o resultado mais fiel.

O operador Canny utiliza dois limiares: um baixo  $T_L$  e outro alto  $T_H$ . Qualquer pixel na imagem cujo valor seja maior que  $T_H$  é aceito imediatamente como pixel da borda. Pixels conectados a este pixel de borda cujo valor é maior que  $T_L$  também são marcados como pixels da borda.

### 2.6.4 Transformada de Hough

Outra técnica relacionada à determinação de bordas do objeto, via processamento de imagens, consiste em conectar pontos se eles estão sobre uma curva com forma específica. Infinitas linhas passam por um ponto, mas todas satisfazem a equação da reta, com variação de valores de ângulo e termo independente, no domínio espaço. Quando ocorre a intersecção entre linhas, os pontos de intersecção significam que eles estão sob a mesma linha, portanto é possível encontrar bordas ou fronteiras retilíneas com esta técnica.

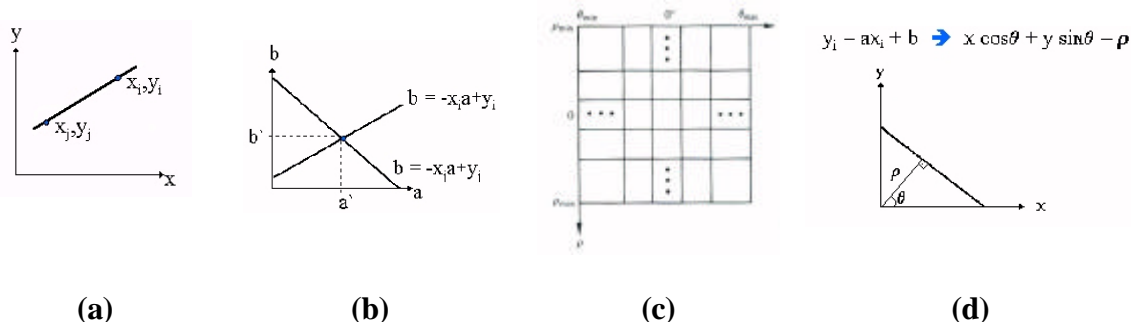
De acordo com Gonzalez (1993), Hough propôs em 1962 uma aproximação alternativa, conhecida como **transformada de Hough**, usada para detectar e ligar bordas. Considerando o ponto  $(x_i, y_i)$  e a equação geral da reta:

$$y_i = ax_i + b \tag{20}$$

O plano  $ab$ , também chamado parâmetro espaço, fornece a equação da reta para o par  $(x_i, y_i)$ . O problema em usar a equação 20 para representar as linhas é que ambos os parâmetros ângulo e termo independente se aproximam do infinito para retas verticais (CASTLEMAN, 1996), assim uma forma de contornar este problema é utilizar a representação *normal* de uma reta dado por:

$$r = x \cos \theta + y \sin \theta \tag{21}$$

Além disso, o segundo ponto  $(x_j, y_j)$  também tem a linha no parâmetro espaço associado com ele, e esta linha intercepta a linha associada com  $(x_i, y_i)$  em  $(a', b')$ , onde  $a'$  é o ângulo e  $b'$  o termo independente da linha contendo tanto o ponto  $(x_i, y_i)$  como  $(x_j, y_j)$  no plano  $xy$ . A figura 15 ilustra os planos  $xy$  e  $\rho\theta$ .



**Figura 15 – Planos para representar uma reta. a) Plano XY. b) Plano AB. c) Quantização do parâmetro plano para utilização na transformada de Hough. d) Plano r $\theta$ .**

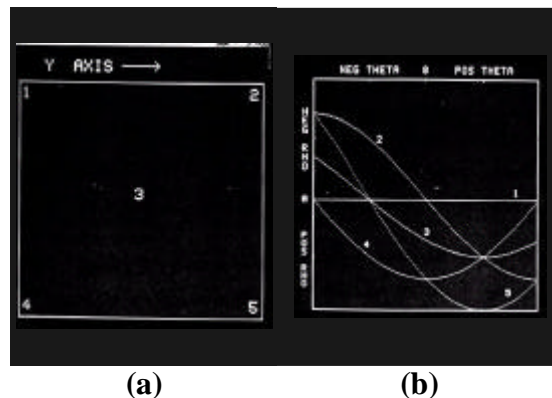
Fonte: Gonzalez (1993, p.434-436)

O atrativo computacional da transformada de Hough está na subdivisão do parâmetro espaço nas chamadas: células de acumulador, como é mostrado na figura 15.c, onde  $(a_{\max}, a_{\min})$  e  $(b_{\max}, b_{\min})$  são as faixas de valores esperados de ângulo e termo independente.

As células nas coordenadas  $(i,j)$ , com valor do acumulador  $A(i,j)$ , corresponde ao quadrado associado com o parâmetro coordenadas espaciais  $(a_i, b_i)$  (GONZALEZ, 1993). Inicialmente, estas células são ajustadas para zero. Então para todo ponto  $(x_k, y_k)$  no plano da imagem, deixa-se o parâmetro  $a$  igual a cada um dos valores de subdivisão permitido sobre o eixo  $a$  e calcula o  $b$  correspondente usando a equação  $b = -x_k a + y_k$ . Os valores dos  $b$ 's calculados são arredondados para abaixo com um valor permitido no eixo  $b$ . Os parâmetros  $\rho$  e  $\theta$  podem ser visualizados na figura 15.d. Agora, obtêm-se a Transformada de Hough no espaço de parâmetros  $\rho$  e  $\theta$  e cada ponto  $(x,y)$  será mapeado como uma função senoidal no espaço Hough, ou seja, um conjunto de pontos colineares no espaço real irá gerar um conjunto de senóides no espaço Hough. Conforme pode ser observado nas figuras 16.a e 16.b, as senóides terão um ponto de interseção que identifica a presença da reta que une estes pontos no espaço real.

Assim para o problema de conexão das bordas uma aproximação baseada na transformada de Hough consiste em:

- computar o gradiente da imagem;
- especificar as subdivisões no plano  $\rho\theta$ ;
- examinar a contagem das células acumuladoras para altas concentrações de pixels;
- examinar as relações entre os pixels vizinhos com célula escolhida.



**Figura 16– Transformada de Hough. a) Retas no plano AB. b) senóides no plano**  
 Fonte: Gonzalez (1992, p.434-436).

O conceito de continuidade, neste caso, está baseado em calcular a distância entre pixels desconexos, identificados durante do conjunto de pixels correspondentes a uma dada célula de acumulador. Um *gap* em qualquer ponto do contorno é significativo se a distância entre o ponto e seu vizinho mais próximo excede um certo limiar.



### 3 ARQUITETURA DO SISTEMA

Neste capítulo são descritos os equipamentos e aplicativos utilizados para aquisição, tratamento e processamento das imagens que culminaram no desenvolvimento do programa **Falcão**, destinado a monitorar o trânsito em rodovias.

#### 3.1 HARDWARE

O computador utilizado possui processador ATLON de frequência de clock de 1GHZ e memória RAM de 128 Mb. A coleta de dados é feita por uma câmera digital Canom, montada sobre a pista e posicionada de forma a capturar as imagens dos veículos de topo. As imagens são gravadas no formato AVI (Motion JPEG + WAV) e armazenadas na própria câmera em um cartão CF de 128 Mb, sendo posteriormente transferidas para o computador através de uma conexão USB. O formato das imagens é convertido para uma seqüência de imagens no formato BMP e armazenadas como amostras para posterior processamento no aplicativo Falcão, desenvolvido durante o PPGEE. As imagens são adquiridas com uma amostragem de 15 quadros por segundo, coloridas e com resolução de 320 x 240 e 640x480 pixels. No ANEXO A, encontram-se dados sobre a câmera digital e no ANEXO B dados da composição do banco de amostras.

#### 3.2 SOFTWARE

As rotinas para abertura, manipulação e salvamento de imagens foram elaboradas, inicialmente, no MATLAB 5.0 que serviu de plataforma para teste e comprovação de alguns resultados, porém não demonstrou eficiência computacional aceitável para trabalho com rotinas de processamento de imagens. As rotinas deste trabalho foram elaboradas para aplicações específicas em filtragem, segmentação, conversão e descrição das imagens em linguagem C para ambiente DOS e fazem parte da biblioteca falcao.h. Após serem testadas individualmente e em conjunto, as rotinas são gerenciadas pelo programa visual para Windows. – C BUILDER versão 3.0, que culminou na elaboração do programa Falcão.

A figura 17 ilustra a plataforma utilizada para desenvolvimento do programa falcão.

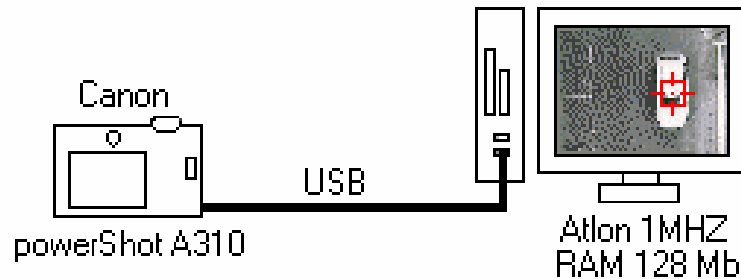


Figura 17– Arquitetura do sistema

Foram utilizadas, também, funções adicionais da biblioteca LiLi – LaPSI Image Library, desenvolvida no Laboratório de Processamento de Sinais e Imagens da UFRGS para abertura, manipulação e salvamento de arquivos. A biblioteca para manipulação de imagens, falcao.h, está disponível na biblioteca LiLi. Os seguintes aplicativos também foram utilizados para elaboração desta dissertação:

- a) Paint Shop Pró versão 5.0: manipulação e análise de imagens;
- b) Ezomn da Canon versão 4.2: transferência de arquivos da câmera para o computador;
- c) VirtualDub versão 1.6.0: conversão entre formatos de vídeos e descompactação para seqüência de imagens;
- d) Office 2000: edição de textos, tabelas e apresentações.

## 4 METODOLOGIA

Este capítulo descreve os métodos utilizados para detectar, acompanhar e extrair informações de objetos móveis. Os programas foram elaborados em linguagem C para ambiente Windows – C BUILDER. Foram utilizadas algumas funções, tais como abertura e salvamento de arquivos da biblioteca LiLi – LaPSI Image Library, desenvolvida no Laboratório de Processamento de Sinais e Imagens da UFRGS. A metodologia utilizada é descrita desde a etapa de aquisição das imagens e culmina na extração de informações do objeto, bem como no teste de consistência destas informações. As funções, desenvolvidas em linguagem C, específicas para este trabalho foram disponibilizadas para Biblioteca LiLi.

### 4.1 AQUISIÇÃO DAS IMAGENS

A coleta de dados é feita por uma câmera digital, montada sobre a pista e posicionada de forma a capturar as imagens dos veículos de topo. As imagens, gravadas na câmera digital, foram adquiridas com uma amostragem de 15 quadros por segundo, resolução de 320 x 240 pixels, e armazenadas no formato AVI (motion JPEG) para posterior descompactação, etapa que converte o arquivo de vídeo numa seqüência de imagens no formato BMP. A correspondência número de pixels x comprimento está intimamente ligada às características da câmera e de sua distância em relação ao solo. De acordo com (Tipler, 1998), as máquinas fotográficas utilizam lentes convergentes. Assim quando o objeto está a uma distância maior que o dobro da distância focal, em relação ao centro da lente, a imagem obtida é real, invertida e reduzida, que por sua vez sensibiliza o CCD - *Charge Couple Device*. A figura 18 ilustra o processo de formação da imagem na lente com objeto situado a distância maior que o dobro da distância focal da lente

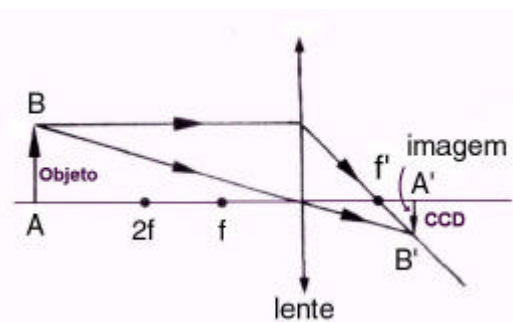


Figura 18 - Formação da Imagem em uma lente convergente.

A equação 22, conhecida como equação de Gauss, permite determinar  $p'$  conhecendo-se parâmetros da lente como a distância focal e sabendo-se a distância entre objeto e a câmera (TIPLER, 1998).

$$\frac{1}{f} = \frac{1}{p'} = \frac{1}{p} \quad (22)$$

O ganho linear transversal pode ser calculado a partir da equação 23, onde  $i$  é a altura da imagem,  $o$  a altura do objeto,  $p'$  a distância da imagem ao centro da lente e  $p$  a distância do objeto ao centro da lente.

$$A = \frac{i}{o} = \frac{-p'}{p} \quad (23)$$

Assim conhecendo-se a distância da câmera ao solo, o tamanho da imagem e a distância focal da câmera, calcula-se o tamanho do objeto. Neste trabalho a aquisição de imagens foi realizada de duas formas:

- Aquisição em baixa resolução - a câmera é posicionada a 13,27 m do solo. Uma marcação na cena medindo 3,30 m, permite determinar a relação comprimento x n° de pixels na imagem para posterior estimativa das dimensões do veículo.
- Aquisição em maior resolução - a câmera é posicionada a 27 m do solo. Uma marcação na cena medindo 4,50 m, permite determinar a relação *comprimento x num. pixels* na imagem para posterior estimativa das dimensões do veículo.

As imagens foram dispostas em um banco contendo 23 amostras, conforme APÊNDICE B.

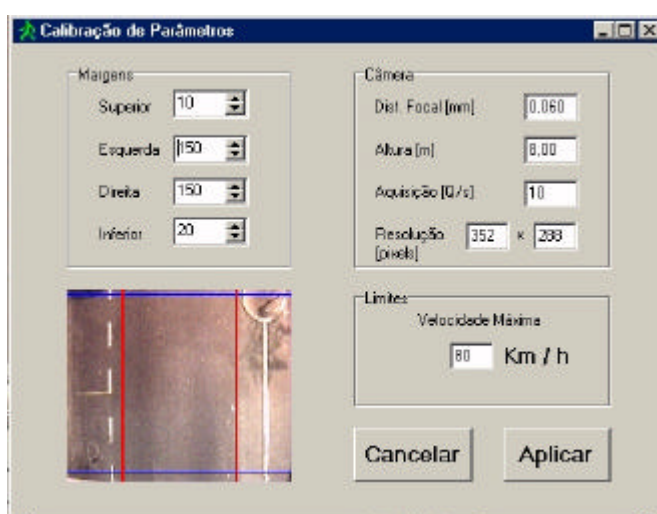
#### 4.1.1 Ajuste de parâmetros

Através desta interface, o usuário pode inserir e alterar dados para calibração do sistema, tais como: parâmetros intrínsecos da câmera, altura de instalação da câmera, velocidade máxima e ajustar a região de interesse.

O ajuste da região de interesse é feito através do ajuste das margens superior, esquerda, direita e inferior que podem ser movimentadas individualmente, de acordo com a

região de interesse. Assim, é possível eliminar bordas da imagem não necessárias as etapas subsequentes e que poderiam gerar a falsa detecção de movimento, como no caso do movimento da vegetação com vento. A interface que permite o ajuste individual das margens é mostrada na figura 19.

Para correta determinação das dimensões e velocidade do veículo, o sistema deve ser calibrado em função dos parâmetros: taxa de amostragem, comprimento da amostra e seu valor correspondente em número de pixels na imagem. Estes dados podem ser inseridos pelo usuário, via interface.



**Figura 19 - Janela de calibração do sistema**

## 4.2 PRÉ-PROCESSAMENTO

O tratamento da imagem antes da etapa de processamento é fundamental para evitar problemas nas etapas seguintes do processamento (GONZALEZ, 1993). Dentre os problemas que merecem atenção especial destacam-se trepidações da câmera, movimento da vegetação e variação da luz ambiente. Os exemplos citados interferem diretamente na etapa de segmentação do objeto. As técnicas de pré-processamento, descritas abaixo, têm por função reduzir os problemas que interferem na segmentação do objeto e diminuir o tempo de processamento de cada quadro.

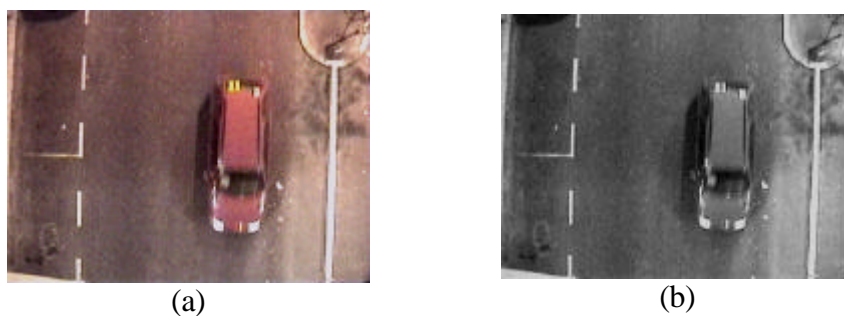
#### 4.2.1 Conversão de RGB para grayscale

Para diminuir o peso computacional do processamento, antes de qualquer operação, as imagens coloridas são convertidas do formato RGB para Grayscale (GONZALEZ, 1993). Todos os pontos da imagem colorida, ou pixels, são varridos da esquerda para direita e de cima para baixo. A cor de cada pixel na imagem colorida é formada por uma combinação de três diferentes graduações das cores: vermelho (R), verde (G) e azul (B). A intensidade de cada cor pode variar no intervalo de 0 até 255, onde 0 corresponde ao preto e 255 a cor em sua maior intensidade luminosa. No método utilizado, a intensidade de cada cor é lida e aplicada na equação 24 para obter o tom monocromático equivalente dos pixels da imagem resultante no formato Grayscale, assim um ponto colorido qualquer dentro do cubo pode ser convertido para grayscale.

$$r_{[i][j]} = (R_{[i][j]} + G_{[i][j]} + B_{[i][j]})/3 \quad (24)$$

Assim o novo ponto obtido a partir da equação 24, estará localizado sobre a linha diagonal do cubo mostrado na figura 3, que une os pontos preto e branco.

Onde [i] e [j] são, respectivamente, as coordenadas de linha e coluna de cada pixel. Conforme pode ser observado na figura 3, a média entre três cores quaisquer do espaço ortogonal RGB, limitado pelo cubo, cai sobre a linha diagonal que representa o tom de cinza que equivale em uma imagem monocromática. A figura. 20 ilustra o resultado obtido na aplicação do método.



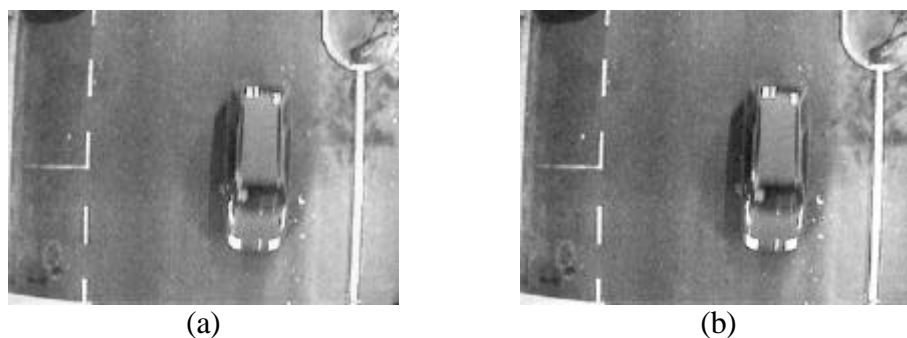
**Figura 20 - Conversão da imagem para Grayscale.**  
a)imagem no formato RGB. b)imagem Grayscale.

#### 4.2.2 Realce de contraste

Para corrigir problemas de contraste na imagem monocromática, causados por iluminação deficiente, excessiva ou mesmo calibração incorreta do obturador, o histograma da imagem é operado matematicamente. O método consiste em procurar, na imagem, os valores de menor  $I_{min}$  e maior intensidade luminosa  $I_{max}$ . Estes valores são aplicados na equação 25 para todas as coordenadas da imagem.

$$I_{stretch} = (A_{[i][j]} I_{min}) * \left( \frac{255}{I_{max} - I_{min}} \right) \quad (25)$$

Onde  $A_{[i][j]}$  é a intensidade luminosa do ponto de coordenadas  $(i,j)$  na imagem original e  $I_{stretch}$  é a nova intensidade luminosa do ponto de coordenadas  $(i,j)$ . A figura 21 mostra o resultado obtido na operação de *stretch* do histograma.



**Figura 21 - Operação de stretch do Histograma. a)imagem original. b)imagem produzida a partir do stretch do histograma.**

O método garante que as intensidades luminosas de todos os pixels da imagem ocuparão todo o espectro de 0 a 255, evitando que haja uma concentração de pontos em determinadas regiões do espectro que se traduziriam em imagens com baixo contraste. A figura 22 mostra os histogramas das imagens das figuras 21.a. e 21.b, respectivamente. Através da operação de *stretch* do histograma foi possível melhorar o contraste.

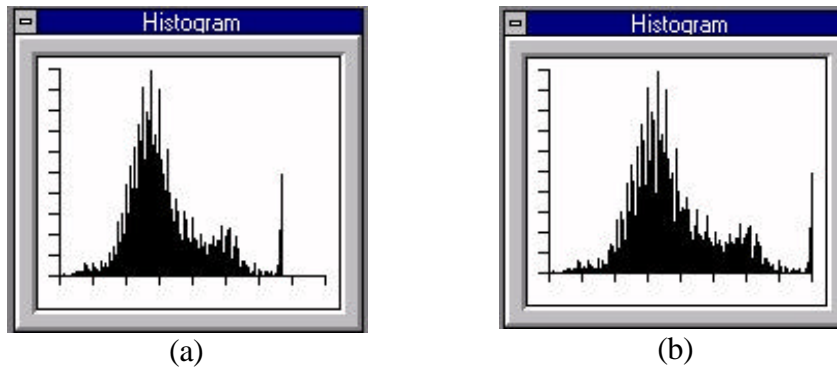


Figura 22 - Histogramas Antes e após o stretch. a) Histograma da imagem original mostrada em 21.a. b) Histograma processado que vai gerar a imagem mostrada em 21.b.

#### 4.2.3 Suavização da imagem

Os efeitos da trepidação da câmera, causados pela vibração do asfalto, ou mesmo pelo vento, são minimizados por uma filtragem não linear que consiste na convolução da máscara mostrada na figura 23 com a imagem a ser processada (PARKER, 1997) (JAE, 1990).

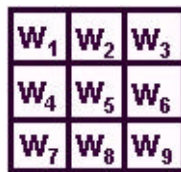


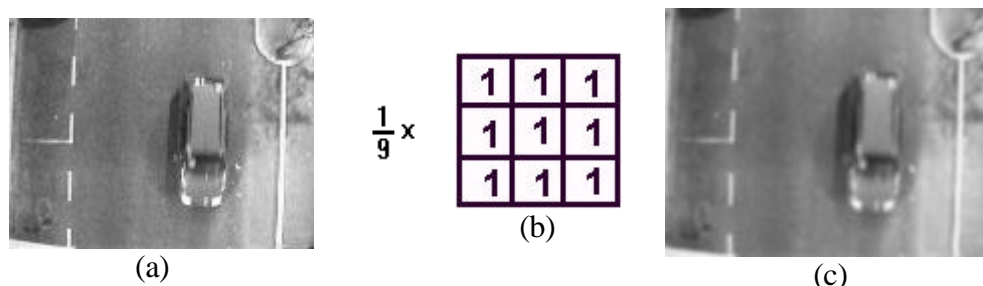
Figura 23 - Máscara 3x3 com coeficientes genéricos – pesos.

A função básica do filtro é suavizar a imagem, ou seja, diminuir a diferença entre os tons de cinza dos pixels da imagem que estão sob a máscara. Neste método a máscara percorre a imagem da direita para esquerda e de cima para baixo, deslocando-se de um em um pixel. A partir da equação 26, para cada posição da máscara é calculado o valor médio entre os tons de cinza da região da imagem original que está sob a máscara. No processo de suavização todos os pesos da máscara têm valor um e os tons de cinza na imagem, sob a máscara, são notados por  $z_1, z_2, \dots e z_9$ .

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 \quad (26)$$



A figura 24 mostra o resultado obtido da filtragem espacial para suavização de imagem grayscale utilizando-se máscara 3x3 com todos os pesos unitários. A suavização garante menor erro relacionado à trepidação da câmera no resultado obtido após a subtração.



**Figura 24 - Suavização da imagem. a)imagem original. b)máscara de convolução 3x3. c)imagem filtrada**

#### 4.3 SEGMENTAÇÃO E DETECÇÃO DE MOVIMENTO

Segmentar uma imagem significa identificar ou realçar determinadas características. Neste trabalho serão abordadas técnicas de segmentação aplicadas à segmentação de veículos em movimento. O objetivo da segmentação é separar o objeto em movimento do fundo.

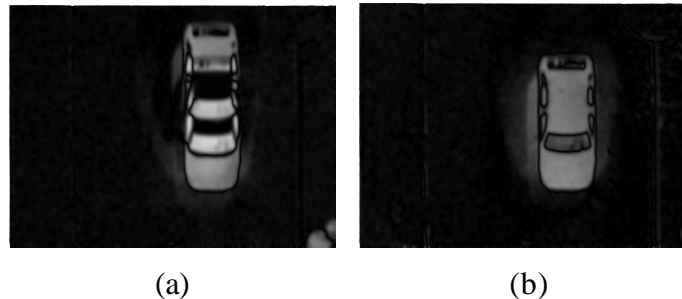
De acordo com (GONZALEZ, 1993), a segmentação é o primeiro e essencial passo em procedimentos automáticos para análise do conteúdo de uma imagem. Considerando uma seqüência de quadros contendo o objeto em movimento, a segmentação pode ser realizada, no domínio espaço através da diferença entre dois quadros tomados em no tempo  $t_1$  e  $t_2$  é definida por:

$$d_{ij}(x, y) = \begin{cases} 1 & \text{if } |f(x, y, t_1) - (f(x, y, t_j))| > q \\ 0 & \text{senão} \end{cases} \quad (27)$$

Assim  $d_{ij}(x, y)$  será 1 nas coordenadas espaciais (x,y) se a diferença entre as duas imagens for maior que um valor de limiar. Em análise dinâmica todos os pixels com  $d_{ij}(x, y)$  com valor 1 são considerados como resultado de um objeto em movimento. De acordo com esta concepção a segmentação no domínio espaço pode ser realizada de duas formas: através da comparação do quadro atual com o anterior e pela comparação do quadro atual com um quadro de referência ou fundo estático previamente armazenado. A comparação do quadro atual com o anterior é obtida através da subtração entre o quadro corrente e o quadro anterior. A formulação matemática aplicada a cada quadro é mostrada na equação 28.

$$\Delta I = |f(x, y, t) - f(x, y, t-1)| \quad (28)$$

Onde  $f(x, y, t)$  é o quadro corrente e  $t-1$  é o quadro anterior. Neste método as regiões onde existe movimento produzem grandes variações de intensidade. O movimento pode ser detectado através de um limiar de área cujos pixels ultrapassem um limiar de variação de intensidade luminosa. Neste método o contorno do objeto é alterado. Como os dois quadros utilizados na comparação estão sempre sendo atualizados, alterações no background não causarão falsa detecção de movimento. A figura 25.a ilustra o resultado obtido.



**Figura 25 – Técnicas de segmentação do objeto. a) comparação do quadro corrente com o anterior. b) comparação do quadro corrente com o fundo estático**

No segundo método de segmentação, que compara o quadro atual com o fundo estático, existe o problema da possível modificação do cenário, em função de obras, pinturas na pista, carros parados, etc. Esta alteração do cenário poderia ocasionar a falsa detecção de movimento. A imagem obtida pela comparação do quadro atual com o fundo é mostrada na figura 25.b. A imagem resultante é obtida a partir da aplicação da equação 29.

$$\Delta I = |f(x, y, t) - f(x, y)| \quad (29)$$

Onde  $f(x, y, t)$  é o quadro corrente e  $f(x, y)$  o fundo. Também, neste método, as regiões onde existe movimento produzem grandes variações de intensidade. O movimento pode ser detectado através de um limiar de área, ou pelo limiar de intensidade luminosa. Este método tem a característica de preservar as características de forma e medidas do objeto.

Os dois métodos de segmentação diferem, basicamente, no que diz respeito à preservação das características de forma e dimensões do objeto. Assim, como o segundo método preserva tais características, foi o escolhido para segmentar o objeto. A falsa detecção de movimento pode ser um problema: como o fundo é tomado como estático, sempre que neste houver alguma variação permanente no cenário ocorrerá falsa detecção de movimento. Para evitar a falsa detecção sempre que não for detectado movimento, a imagem de fundo é substituída pelo quadro atual. Para tornar a detecção de movimento mais consistente foram utilizados limiares de intensidade luminosa e área. É considerado movimento somente quando a imagem resultante da subtração contiver número de pixels igual ou maior que o número de pixels do menor veículo -  $N_{PMV}$  que deve ser detectado. São contados apenas pixels que possuem intensidade luminosa maior que um valor de limiar de intensidade -  $I_{TH}$ . Os valores de  $N_{PMV}$  e  $I_{TH}$  são definidos pelo usuário na interface de programa do *falcão*. Assim o número de pixels, ou limiar de área, que será entendido como movimento depende do menor objeto que se deseja detectar e varia em função da distância da câmera ao solo e da resolução das imagens. A tabela 3 ilustra o número aproximado de pixels para cada classe de veículo, considerando a distância da câmera ao solo -  $d_H$  e a resolução das imagens -  $R_{es}$ . A estimativa do número de pixels foi obtida pela média de contagem manual obtida em 3 imagens com classes conhecidas. O número de pixels do objeto é inversamente proporcional a distância da câmera ao solo e diretamente proporcional a resolução da imagem.

**Tabela 3 – Número médio de pixels na imagem segmentada de acordo com classe do veículo**

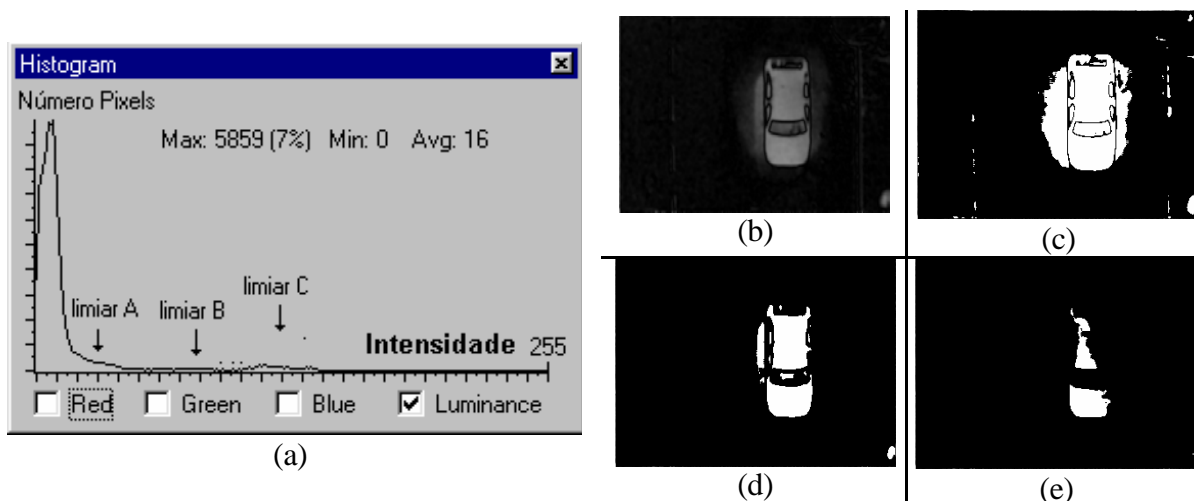
Classe	Num. aprox. pixels	
	$d_H = 13,46 \text{ m}$	$d_H = 27,00 \text{ m}$
Motocicleta	140	140
Sedan	660	660
utilitário	900	900
Micro ônibus / caminhão pequeno	1400	1400
Ônibus/ caminhão	+ 3500	+ 1350

#### 4.4 BINARIZAÇÃO

A imagem obtida após a subtração é escura, bastante próxima da cor preta e apresenta baixo contraste. Para melhorar a visibilidade do objeto segmentado, a imagem obtida na etapa de segmentação é binarizada utilizando-se um limiar. Deste limiar vai depender o sucesso em isolar o objeto do fundo eliminando ruídos, sombras e reflexões. O limiar pode ser obtido pela análise do histograma, mas isto não garante a repetibilidade para

outras imagens da seqüência.. Na figura 26 é mostrada a imagem e seu histograma, onde os limiares foram escolhidos via análise de histograma.

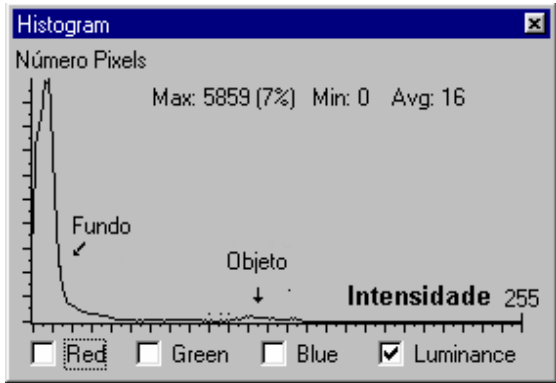
Esta etapa tem importância fundamental para as etapas seguintes do processamento para determinação dos descritores do objeto. O ideal é ter-se, ao final deste processo, o objeto isolado do fundo e livre de sombras, faróis e reflexos. Embora em algumas situações o problema da sombra, faróis e reflexos possa ser minimizado com iluminação artificial apropriada, o desafio é binarizar a imagem utilizando técnicas de processamento de imagem para isolar o objeto da melhor forma possível. Foram testados dois métodos para determinar o limiar para binarização: o estatístico e o Laplaciano.



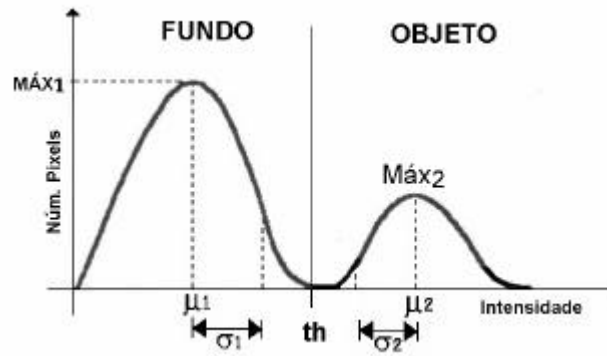
**Figura 26 – Determinação do limiar de binarização via análise de histograma. a) Histograma da imagem com indicação de limiares. b) imagem com baixo contraste. c) binarização com limiar A=20. d) binarização com limiar B=75. e) binarização com limiar C= 115.**

#### 4.4.1 Método estatístico

O desafio é encontrar um limiar para binarizar a imagem mostrada na figura 26.a, tal que separe o objeto de artefatos, sombras e reflexos com o mínimo de degradação no objeto. A análise do histograma, mostrado em 27.a, mostra que existe uma separação entre objeto e fundo. A idéia é determinar os picos de máxima do objeto e do fundo, bem como a média, variância e desvio padrão das distribuições do objeto e do fundo, aproximando as curvas à distribuição normal, conforme mostrado na figura 27.b.



(a)



(b)

**Figura 27 – Determinação do limiar via análise estatística da distribuição de tons de cinza.**

**a) Histograma da imagem a ser binarizada com definição entre objeto e fundo.**

**b) Aproximação da distribuição à distribuição normal.**

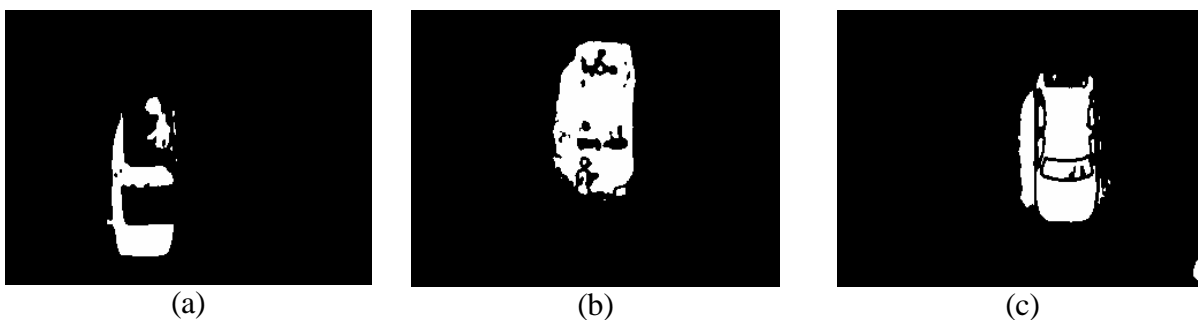
O algoritmo utilizado para determinação do limiar pelo método estatístico é:

- a) obtêm histograma da imagem;
- b) procura valor máximo do histograma  $\max_1$ ;
- c) o valor máximo é associado ao valor médio da distribuição do fundo  $m_1$ ;
- d) determina variância  $s^2$  e desvio padrão  $s$  fundo, através das equações 30 e 31;
- e) procura valor de máximo na distribuição do objeto  $\max_2$ , considerando que este está localizado a partir de  $m_1 + s$ ;
- f) associa o valor  $\max_2$  a média da distribuição do objeto  $m_2$ ;
- g) o limiar é determinado pela equação 32;

$$s^2 = \frac{1}{N} \sum_{i=1}^{l_{\max}} (f(i) - med)^2 \quad (30)$$

$$s = \sqrt{s^2} \quad (31)$$

$$th = \frac{(m_1 + s_1) + (m_2 - s_2)}{2} \quad (32)$$

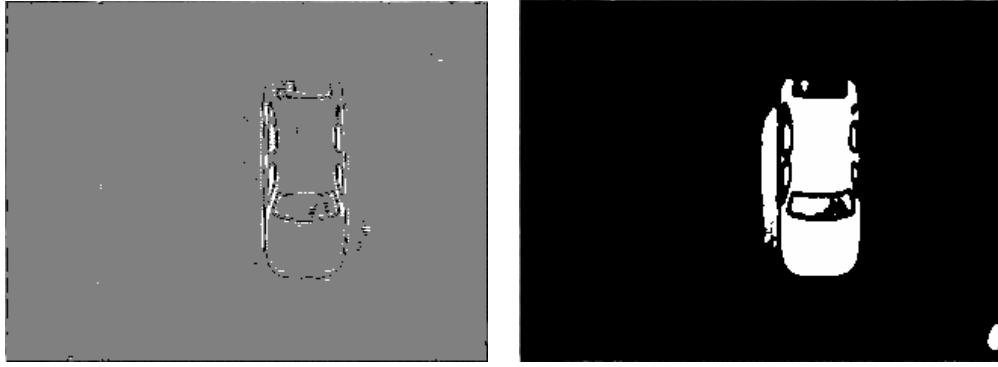


**Figura 28 – Imagens com limiar determinado pelo método estatístico. a) Amostra A, sol das 14h,  $th = 94$ . b) Amostra B, 21h, com iluminação artificial  $th = 66$ . c) Amostra C, 19h, sem iluminação artificial,  $th = 66$ .**

O método estatístico mostrou-se adequado para determinação automática do limiar de binarização. A figura 28 ilustra resultados do algoritmo aplicado a três imagens do banco. De acordo com a distribuição estatística dos tons de cinza, é determinado um limiar –  $th$ , tal que seja possível destacar a região de interesse do objeto móvel. Assim, embora a aplicação exclusiva do método, não possibilite binarizar a imagem eliminando as sombras, este mostrou-se adequado para determinar a região de interesse de diferentes imagens calculando o limiar automaticamente de acordo com as distribuições estatísticas do histograma.

#### **4.4.2 Método Laplaciano**

De acordo com o exposto no embasamento teórico, a transformada de Laplace de uma imagem fornece um mapa com valores positivos, negativos e zeros de acordo com as variações nos tons de cinza da imagem, sendo ideal para detecção de bordas. A figura 29 ilustra imagem obtida utilizando-se a transformada de Laplace, e a imagem binarizada com limiar obtido por Laplace. Na figura 29 os tons cinza representam os zeros, os pixels pretos os valores negativos e os brancos representam os valores positivos. A imagem obtida após a subtração do fundo apresenta valores na escala de cinza mais próximos ao branco (255) em coordenadas próximas ao objeto, e valores mais próximos ao preto (0) no *background*.



**Figura 29 – Determinação automática do limiar de binarização através do operador Laplaciano. a) mapa obtido pelo operador Laplaciano. b) imagem binarizada com limiar obtido automaticamente via análise da imagem em pontos delimitados pelo mapa -  $th=60$ .**

O método proposto consiste em determinar o limiar de binarização utilizando a imagem obtida pela transformada como mapa para localizar os tons de cinza que estão no exterior da borda do objeto e os tons de cinza que estão na borda interna do objeto. As equações 33 e 34 mostram como são obtidas as médias dos valores positivos e negativos próximos à borda.

$$M_P = \frac{\sum_{x=1}^{l_{\max}} \sum_{y=1}^{c_{\max}} f(x,y)}{N_P} \quad /L\{f(x,y)\} > 0 \quad (33)$$

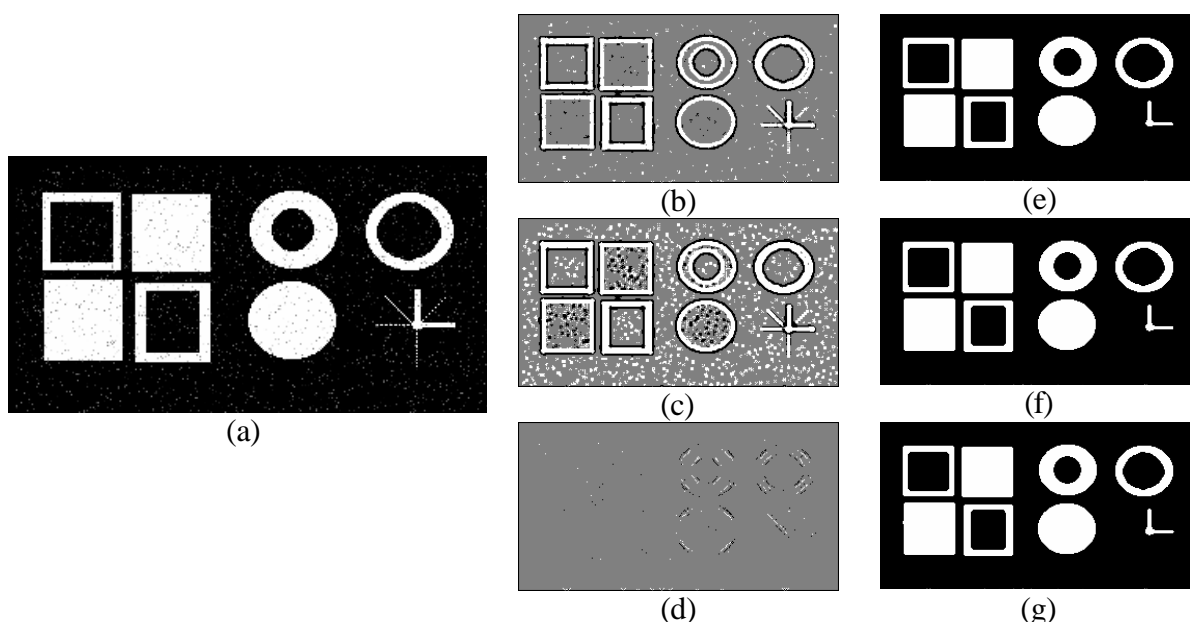
$$M_N = \frac{\sum_{x=1}^{l_{\max}} \sum_{y=1}^{c_{\max}} f(x,y)}{N_N} \quad /L\{f(x,y)\} < 0 \quad (34)$$

Onde  $N_P$ ,  $N_N$  são, respectivamente, o número de pixels positivos e negativos encontrados,  $L\{f(x,y)\}$  a transformada de Laplace,  $f(x,y)$  a função intensidade e  $(x,y)$  as coordenadas espaciais.

O limiar para binarização é obtido pela média das intensidades luminosas obtidas nas equações 33 e 34.

O operador Laplaciano é sensível mesmo a pequenas variações na intensidade de cinza, assim foi utilizado um filtro de suavização na etapa que o antecede. A escolha da máscara para operar o Laplaciano tem influência direta nas características da imagem serão

analisadas para determinação do limiar. A figura 30 ilustra a aplicação das máscaras mostrada na tabela 2 em uma figura de teste. A figura é submetida a 10% de ruído aleatório, porém é suavizada antes de ser operada.



**Figura 30 – Mapas obtidos com diferentes máscaras para implementar o operador laplaciano. a) Figura de teste submetida a 10% de ruído aleatório. a) Mapa laplace a partir de máscara Tabela 2.a aplicada a figura de teste,  $th = 127$ . b) Mapa laplace a partir de máscara Tabela 2.b aplicada a figura de teste,  $th = 115$ . c) Mapa laplace a partir de máscara Tabela 2.c aplicada a figura de teste,  $th = 108$ . (e)(f)(g) são os resultados obtidos binarizando a imagem (a) com o limiar determinado.**

Embora todas as figura tenham sido binarizadas de forma satisfatória, observa-se que os valores analisados na imagem original mudam de acordo com a máscara escolhida. A máscara da tabela 2.b é mais sensível ao ruído uma vez que considera os 8 vizinhos do ponto central igualmente. A máscara da tabela 2.a é menos sensível ao ruído, e leva em consideração a conexão  $N_4$  para compor o pixel na imagem filtrada. A máscara utilizada na tabela 2.c é mais adequada para realçar linhas inclinadas, já que detecta vértices. As linhas tracejadas e mais finas foram suprimidas na etapa de suavização. Assim, a máscara que mostrou melhor performance para análise pixels de acordo com a natureza das imagens adquiridas em rodovias é a máscara da tabela 2.a. A figura 31 ilustra o resultado da binarização obtido em diferentes imagens com o limiar determinado pelo método Laplaciano utilizando a máscara da tabela 2.a.





**Figura 31 – Resultados da determinação do limiar pelo método Laplaciano. a) Amostra A, sol das 14h,  $th = 75$ . b) Amostra B, 21h, com iluminação artificial,  $th = 63$ . c) Amostra C, 19h, sem iluminação artificial,  $th = 68$ .**

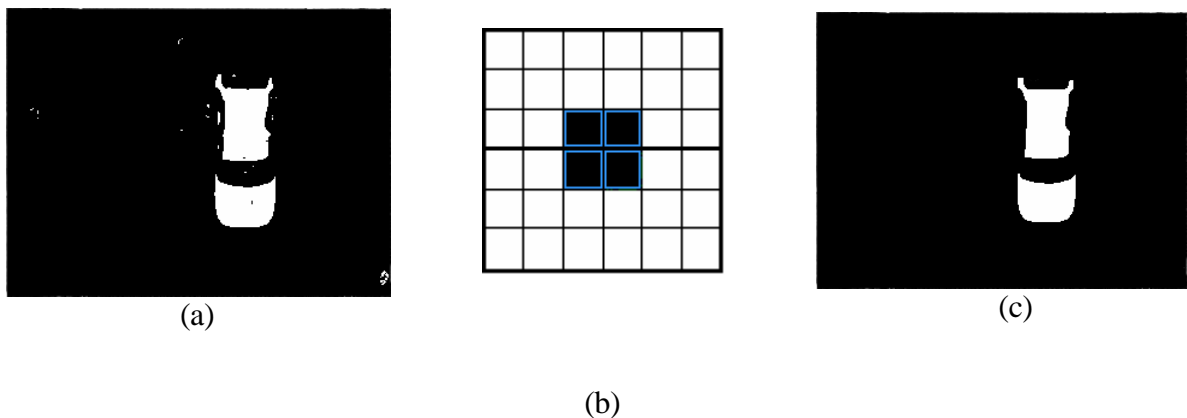
A análise das imagens mostra que o método para determinação automática do limiar por Laplace tem grande habilidade para isolar a região de interesse - ROI, isolando o objeto de artefatos do fundo, embora as sombras sejam segmentadas junto com o objeto.

O método para determinação do limiar pelo laplaciano mostrou-se mais robusto em relação ao método estatístico por analisar os pixels próximos a borda do objeto, além disso, pixels espúrios no histograma poderiam alterar a localização dos pontos máximos nas curvas de distribuição.

#### 4.5 REMOÇÃO DE PIXELS DESCONECTADOS

Visando remover pontos desconectados, que não pertencem ao objeto e que poderiam modificar a localização do ponto central do movimento nas etapas seguintes do processamento, foram utilizadas operações morfológicas<sup>1</sup>. Segundo (PARKER, 1996), a operação morfológica de abertura pode ser utilizada para remover ruído. Este processo remove pixels espúrios pretos sobre um fundo branco, mas não remove os brancos. Para remover os pixels brancos utilizou-se a operação morfológica de fechamento, conforme demonstrado no capítulo 2. De acordo com (KOYA, 2001), a operação morfológica de abertura foi utilizada com sucesso para desconectar células sangüíneas e reduzir o ruído, onde foi garantida a preservação das características de forma e tamanho do objeto das células.

A operação morfológica de abertura, conforme tratada por (GONZALEZ, 1993), geralmente suaviza o contorno, elimina pequenos pontos isolados e isola objetos conectados. O resultado obtido para remoção de ruídos em imagens binárias é mostrado na figura 32.



**Figura 32 – Operação morfológica de fechamento. a) imagem binarizada contendo pixels desconectados. b) elemento estruturante c) resultado da operação morfológica de fechamento**

---

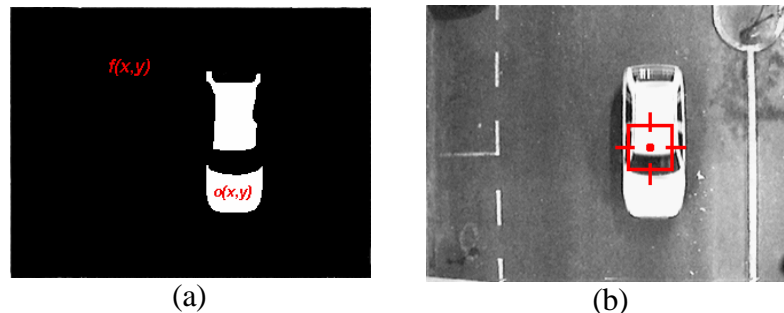
<sup>1</sup> Operações morfológicas de erosão e dilatação são utilizadas com o mesmo elemento estruturante para compor as operações morfológicas de abertura ou fechamento.

## 4.6 DETERMINAÇÃO DO PONTO CENTRAL DO OBJETO.

A determinação do ponto central do objeto tem papel fundamental no acompanhamento e estimativa de velocidade do objeto móvel. Fatores como o tipo de imagem de entrada, peso computacional do algoritmo e precisão desejada, vão definir o método mais adequado.

### 4.6.1 Método da Centróide

Consiste em determinar o centro de massa do objeto segmentado a partir da média das coordenadas de linha e coluna dos pixels brancos que correspondem ao objeto, portanto o método é aplicado a imagem já binarizada. Considerando  $I(x,y)$  a função da imagem que contém todos os pixels,  $O(x,y)$  a função que contém o objeto e  $f(x,y)$  a que contém o fundo. Na figura 33.a ilustra esta convenção para separação entre objeto e fundo. A figura 33.b mostra a janela plotada sobre as coordenadas centrais do objeto. As equações 35 e 36 mostram como podem ser determinadas as coordenadas centrais do objeto utilizando-se o método A- centróide.



**Figura 33 – Centro do objeto determinado pelo método da centróide. a) imagem binária. b)Centróide obtido da função lcentroide e marcado sobre a imagem de entrada grayscale.**

$$P_{YC} = \frac{1}{N} \sum_{(x,y) \in o(x,y)} y \quad (35)$$

$$P_{XC} = \frac{1}{N} \sum_{(x,y) \in o(x,y)} x \quad (36)$$

Onde  $N$  é o número de pixels do objeto e os pontos  $(x,y)$  considerados no cálculo, pertencem ao objeto. Os pixels brancos da imagem binarizada corresponde ao objeto e os pretos ao fundo. Esta técnica não é recomendada nos casos em que a imagem contenha pixels espúrios que não fazem parte do objeto, uma vez que, conforme sua distribuição, tendem a introduzir erro na determinação do ponto central. Este método demonstrou boa performance em termos de velocidade de processamento em utilizando imagens monocromáticas com resolução de 320 x 240 pixels.

#### 4.6.2 Método dos pontos máximos e mínimos

O ponto central da região de interesse – ROI é encontrado no centro da região onde estão localizados os pixels brancos. A técnica utilizada neste trabalho consiste em varrer a imagem na busca de pontos conectados. Primeiro varre-se a imagem verticalmente, de cima para baixo, em todas as colunas, encontrando para cada coluna, da região do objeto, o ponto mínimo de linha mínima -  $L_{min}$  e o ponto máximo de linha -  $L_{max}$ , se existirem. Os pontos  $L_{min}$  e  $L_{max}$  são validados pela verificação da conexão do ponto sob teste com os pixels vizinhos. A verificação de conexão evita que pixels espúrios sejam detectados, causando erro na determinação da região de interesse – ROI. As conexões testadas são mostradas na figura 34. De posse dos pontos  $L_{min}$  e  $L_{max}$ , encontra-se a linha central-  $P_{YC}$  a partir da equação 37.

$$P_{YC} = (L_{min} + L_{max}) / 2 \quad (37)$$

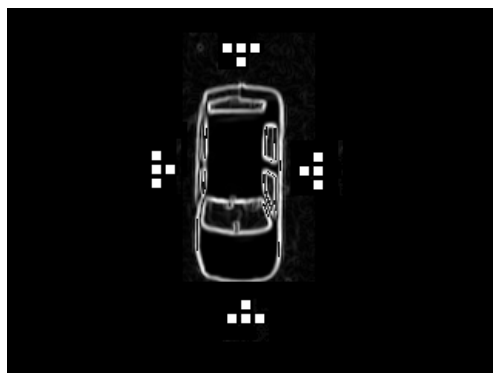


Figura 34 - Máscaras para verificar conexão de pixels no método de máximos e mínimos.

Para encontrar as coordenadas da coluna e a largura do objeto, em uma determinada linha, repete-se o algoritmo anterior, porém, agora, varrendo-se a imagem da esquerda para direita. A equação 38 é utilizada para determinar a coluna central -  $P_{XC}$ .

$$P_{XC} = (C_{\min} + C_{\max}) / 2 \quad (38)$$

As medidas finais do quadro em análise são obtidas a partir da média entre todas as larguras, comprimentos e coordenadas obtidas para um determinado quadro. A determinação das coordenadas centrais do objeto, a cada quadro, são efetuados através das equações 39 e 40.

$$P_{YCQ} = \frac{\sum_{i=1}^{i=n_c} P_{YC} [i]}{n_c} \quad (39)$$

$$P_{XCQ} = \frac{\sum_{i=1}^{i=n_l} P_{XC} [i]}{n_l} \quad (40)$$

Onde  $P_{YCQ}$ ,  $P_{XCQ}$ ,  $n_c$  e  $n_l$  são, respectivamente, o ponto central na coordenada  $y$ , o ponto central na coordenada  $x$ , o número de colunas e o número de linhas em que foram encontrados valores consistentes com a vizinhança, em um determinado quadro. Outras características do objeto podem ser determinadas a partir de valores já obtidos. O comprimento e largura do objeto são determinados pelas equações 41 e 42, respectivamente, com bases nos valores mínimos e máximos já apurados.

$$C = L_{\max} - L_{\min} \quad (41)$$

$$L = C_{\max} - C_{\min} \quad (42)$$

Sobre as coordenadas centrais do objeto, calculadas a cada quadro, é posicionada uma janela que facilita o acompanhamento visual de seu percurso. A figura 35 ilustra o processo de obtenção das coordenadas centrais e localização na imagem grayscale.

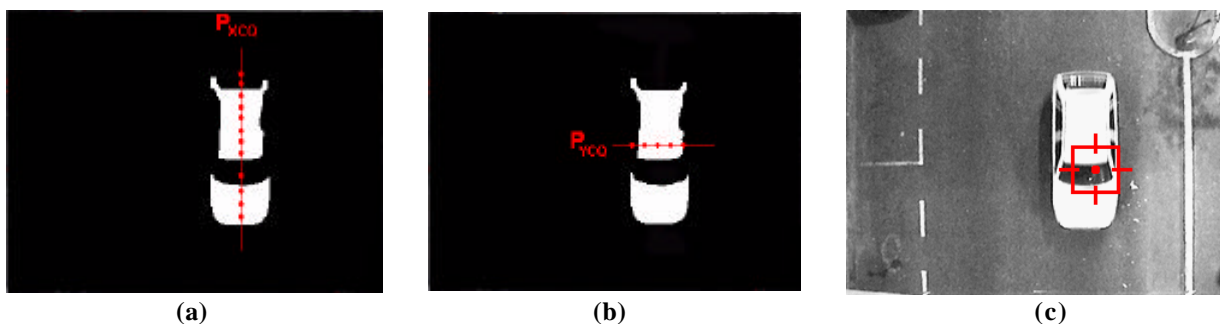


Figura 35 – Localização do centro do objeto. a) Obtenção da coluna central. b) Obtenção da linha central. c) Janela de acompanhamento plotada sobre a imagem grayscale

#### 4.6.3 Comparação entre os métodos

A tabela 4 mostra os resultados obtidos com a aplicação de dois métodos para determinação do ponto central, utilizando como imagem de teste uma circunferência com centro geométrico diferente do centro da imagem. As amostras são submetidas a diferentes percentuais de ruído aleatório.

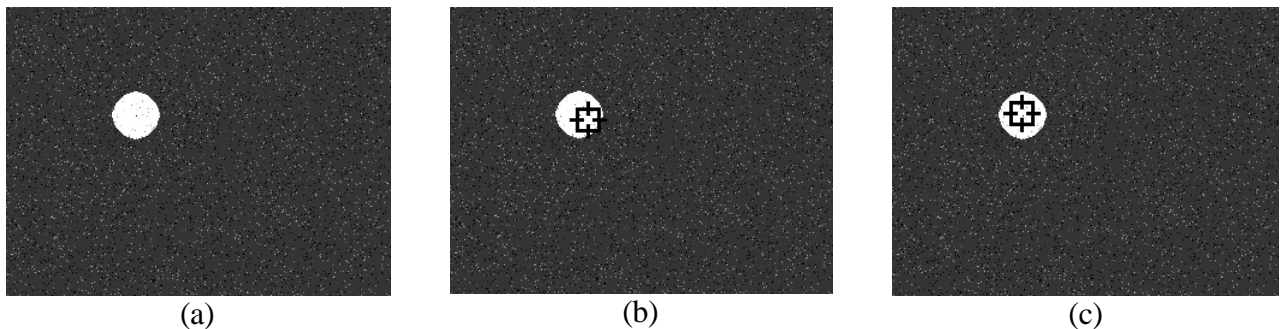
Tabela 4 – Resultados dos métodos para determinação do ponto central

Ruído aleatório (%)	Método A Centróide			Método B Pontos máx. e min.		
	Pxc	Pyc	Erro(%)	Pxc	Pyc	Erro(%)
0	107	89	7,1	107	89	7,1
3	111	91	16	107	89	7,1
5	113	92	27	107	89	11
8	115	94	40	107	89	7,1
10	115	95	40	107	89	11
15	121	99	69	107	89	7,1
20	127	101	105	107	89	7,1

A imagem utilizada para teste é mostrada na figura 36.a. Foi observado que quando o centro do círculo coincide com o centro geométrico da imagem o ruído aleatório não interfere de forma significativa na determinação do centro do objeto, produzindo um erro de 7% no método A e 0% no método B para um ruído aleatório de 30%. O ruído uniforme não produziu erro significativo da determinação do centro na imagem de teste. Mesmo quando a imagem é submetida a 30% de ruído, o erro ficou em torno de 0% . O erro é obtido pelo quociente da distância euclidiana entre dois pixels e o raio do círculo ( $r$ ), conforme mostra equação 43.

$$erro(\%) = \frac{\sqrt{(p_{xc} - p_{xc'})^2 + (p_{yc} - p_{yc'})^2}}{r} * 100 \quad (43)$$

Onde  $p_{xc}$  e  $p_{yc}$  são os valores obtidos pelos algoritmos e  $p_{xc}'$  e  $p_{yc}'$  são as coordenadas reais do centro do objeto na figura de teste, cujo centro está nas coordenadas  $p_{xc}'=108$  e  $p_{yc}'= 90$  e seu raio 20 pixel. A figura 36 ilustra o resultado obtido com os dois métodos.

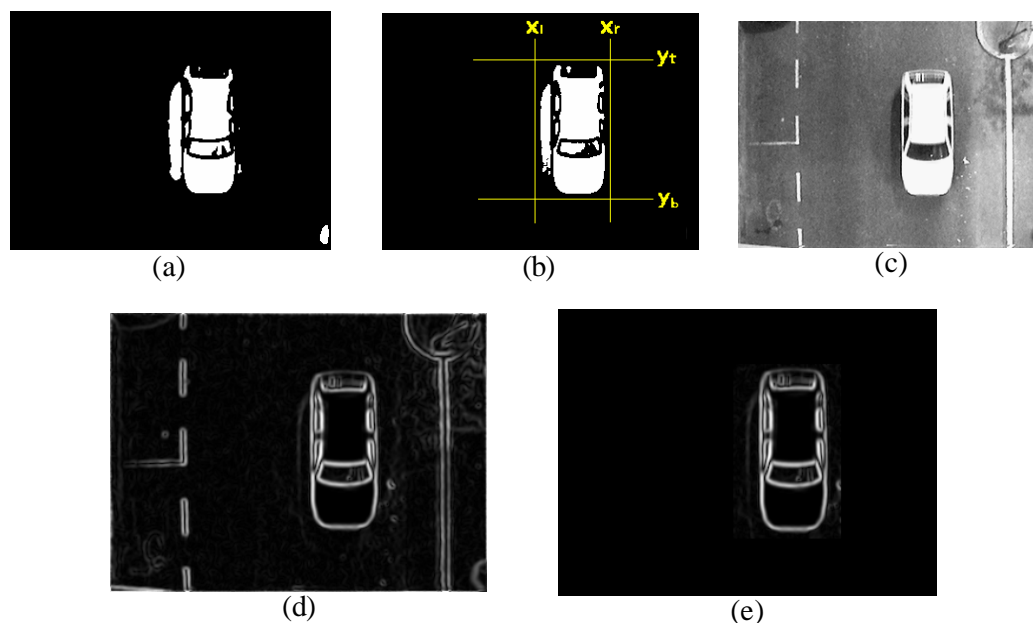


**Figura 36 - Comparação dos métodos de determinação das coordenadas centrais do objeto.**  
**a) Imagem de entrada com 10% de ruído aleatório. b) Resultado do método A – Centróide.**  
**c) Resultado do método B – Pontos máximos e mínimos.**

Assim a análise dos dois métodos mostrou que o método do centróide é mais sensível ao ruído aleatório quando as coordenadas centrais do objeto estão deslocadas em relação ao centro da imagem, no entanto este método tem menor peso computacional. O tempo e o método utilizado para medi-lo está demonstrado no APÊNDICE A para as 2 resoluções das imagens utilizadas. Assim considerando-se que a imagem a ser processada para o cálculo do centro do objeto é uma imagem binária que já foi filtrada pelo filtro mediana, suavizada e operada morfológicamente para eliminar pixels espúrios, pouco ruído aleatório existirá na imagem. Portanto, considerando-se, ainda, o menor peso computacional, optou-se pela utilização do primeiro método, do centróide, para localização das coordenadas centrais do objeto. Observa-se, também, que erros menores que 10% em relação as coordenadas centrais do objeto, não impossibilitam seu acompanhamento visual.

## EXTRAÇÃO DE INFORMAÇÕES

A imagem a ser analisada para extração das informações do objeto deve conter apenas seu contorno livre de artefatos. O método sugerido utiliza a imagem já obtida na etapa de binarização, para selecionar a região de interesse (ROI) que contém o contorno do objeto na imagem processada pelo operador derivativo. As bordas da imagem original, grayscale, são obtidas a partir do operador derivativo Sobel que se mostrou mais consistente, em relação a outros operadores derivativos como Roberts e Prewitt nas amostras testadas, conforme analisado no CAPÍTULO 2. A figura 37 ilustra as imagens obtidas ao final de cada operação utilizada para obter o contorno do objeto livre de artefatos.



**Figura 37 – Determinação ds bordas do objeto considerando a região de interesse. a) Imagem resultante da binarização através de limiar obtido automaticamente. b) Delimitação da região de interesse. c) imagem original grayscale. d) Detecção de bordas na imagem original. e) Seleção da região de interesse no mapa de bordas.**

A ROI é determinada a partir da figura 37.a, definindo-se as linhas e colunas que contornam o objeto segmentado. Para isto é testada a conexão dos pixels que determinarão seus limites nos moldes do método já aplicado para definir as coordenadas centrais da região no *método dos máximos e mínimos*. A imagem resultante, mostrada na figura 37.e, contém apenas as bordas do objeto acrescidas de alguns artefatos na região isolada - ROI, oriundos da aplicação do operador derivativo Sobel. A região externa a ROI é preenchida com pixels



pretos. Para obter as dimensões do objeto, todos os pixels da imagem obtida em 37.e são projetados para os eixos vertical e horizontal pela função *lprojeta*, conforme figura 38.

O rebatimento de todos os pontos é realizado pelo empilhamento dos pixels horizontal e verticalmente em um vetor de coluna e linha, respectivamente. A largura e comprimento são determinados sobre as projeções. Os picos que aparecem nas projeções indicam a presença de pixels retilíneos que podem indicar o contorno do objeto, porém como a região que contém o objeto em movimento está isolada, serão consideradas bordas as coordenadas extremas da projeção, que contém quantidade de pixels superiores a média. Este limiar de média foi utilizado para evitar a falsa detecção de borda em regiões de bordas internas ou onde aparecem artefatos. O limiar se mostrou consistente nas diversas amostras analisadas.

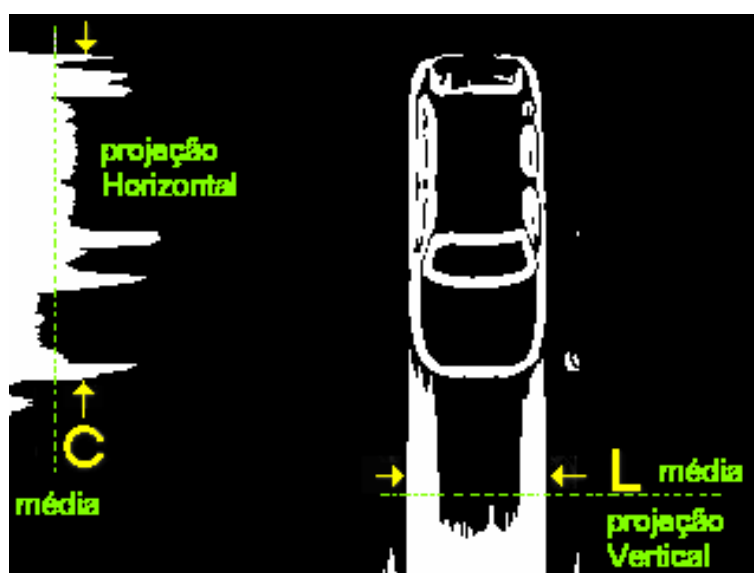


Figura 38 – Projeções horizontal e vertical das bordas do objeto.

As dimensões do objeto, exibidas no Falcão, são resultado da média de dimensões obtidas em quadros onde o objeto está próximo ao centro da tela. Isto assegura maior confiabilidade na medida e garante que o objeto está sendo medido em toda sua extensão.

A velocidade média do veículo é calculada entre os intervalos (80;160) de  $P_{YC}$ . O número de quadros dentro destes limites depende da velocidade do veículo. A taxa de amostragem da câmera é considerada uma base de tempo confiável, assim através da equação 44 determina-se o tempo gasto pelo veículo em função do número de quadros.

$$tempo = \frac{1}{taxa \text{ amostragem}} * num\_quadros \quad [s] \quad (44)$$

Onde *num\_quadros* é o número de quadros dentro do intervalo (80,160) de P<sub>YC</sub>.

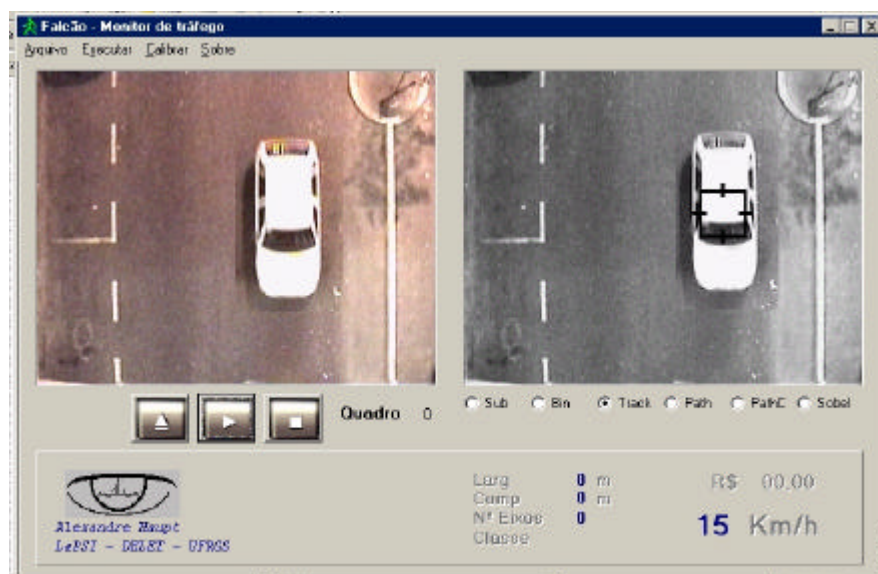
O valor padrão para taxa de amostragem no Falcão é 15 quadros por segundo, porém pode ser alterado pelo usuário na tela de calibração do sistema. Outros dados que devem ser informados nesta tela são o comprimento de *Dist\_calibra* e seu equivalente número de pixels na imagem. Estes valores são utilizados como base para transformar os valores obtidos em pixels para metros. A conversão das dimensões em pixels para metros é realizada pela equação 45.

$$distância = Npxls \frac{Dist\_calibra}{Npxls\_calibra} \quad [m] \quad (45)$$

Onde *Dist\_calibra* e *Npxls\_calibra* são os valores informados na tela de calibração e *Npxls* é o número de pixels que equivalem a distância que será convertida para metros. Com o tempo consumido em segundos (eq. 44) e da distância percorrida em metros (eq. 45), determina-se a velocidade do objeto. A equação 46 fornece a velocidade. A velocidade é posteriormente convertida para *Km/h* multiplicando-se o valor obtido na eq. 46 pela constante 3,6.

$$v = \frac{Dist}{tempo} \quad \left[ \frac{m}{s} \right] \quad (46)$$

A figura 39 ilustra tela principal do programa desenvolvido. A imagem real é mostrada a esquerda e a imagem processada a direita.



**Figura 39 - Janela principal do programa Falcão.**

O comprimento e a largura e velocidade do veículo são exibidos na interface do usuário do programa Falcão. Na tela principal o usuário define o tipo de imagem processada será exibida, em movimento, na tela da direita: subtração, binarizada, contorno ou trajetórias.

## VERIFICAÇÃO E CORREÇÃO DA TRAJETÓRIA

A função *path\_reduce\_shake* tem a finalidade de auxiliar o acompanhamento do objeto, mesmo em situações adversas como nos casos de oclusão temporária do objeto.

A correção da trajetória é realizada com os dados das últimas posições do objeto. Os dados da trajetória são armazenados em um buffer de dados do tipo FIFO – *first-in-first-out*. A partir destes dados, a média ponderada dos 3 últimos deslocamentos é determinada. A posição esperada do objeto, no quadro seguinte, é estimada somando-se média dos três últimos deslocamentos do objeto a sua posição anterior, conforme equação 46.

O erro é calculado em função do valor esperado e do valor obtido através da equação 47. A posição do objeto no quadro seguinte será uma média ponderada entre a posição estimada e a posição "real" do objeto, caso o erro percentual esteja entre o intervalo aberto de 5% e 20%. Dosando-se os pesos desta média é possível ajustar o nível de filtragem.

O filtro atua, basicamente, como um passa-baixas que corrige a trajetória vertical e horizontal do veículo. Quando o erro for maior que 20% a próxima posição do objeto será dada pelo valor estimado, conforme mostra equação 48. Assim é possível reduzir erros de tracking que ocorrem principalmente na oclusão temporária do objeto, trepidação da câmera e falhas na segmentação que não foram solucionadas em etapas anteriores do processamento.

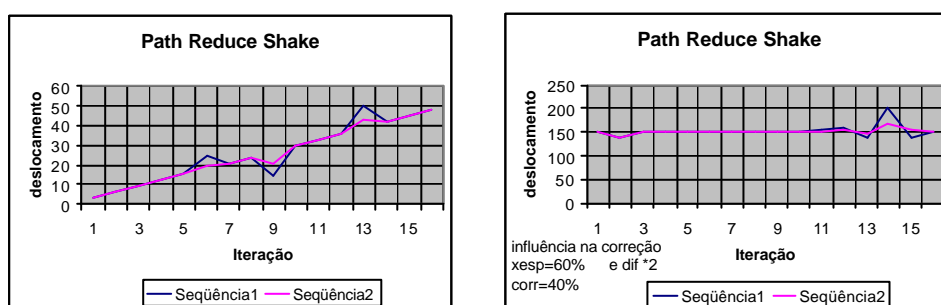
A figura 40 mostra a atuação do filtro em sinais de teste, onde a seqüência 1 é o sinal original e a seqüência 2 o sinal filtrado.

$$X_{esp} = X_{ant} + Média \quad (46)$$

$$Erro = X_{esp} - X \quad (47)$$

$$x \begin{cases} x_{esp} & erro > 20\% \\ x_{med} & 5\% \leq erro \leq 20\% \\ x & erro < 5\% \end{cases} \quad (48)$$

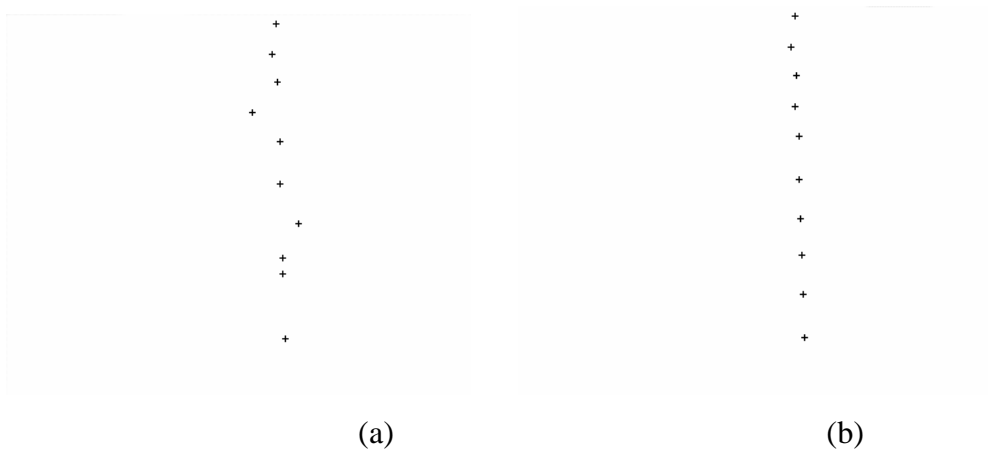
Onde  $X_{esp}$ ,  $X_{ant}$  e  $X$  são, respectivamente a previsão da próxima posição, a posição anterior e a posição determinada do veículo.



**Figura 40 – Gráfico de atuação do filtro preditivo.**

**a) sinal de teste com tendência crescente . b) sinal de teste com tendência constante**

Assim picos abruptos, que estão fora do intervalo previsto, são corrigidos em função do erro obtido. A figura 41 mostra a trajetória obtida sem o filtro e a trajetória corrigida.



**Figura 41 – Correção da trajetória. a) Trajetória sem correção do filtro. b) mesma trajetória com correção.**

O filtro para correção da trajetória é aplicado somente para corrigir as coordenadas em relação ao eixo x. Assim valores discrepantes, em relação ao deslocamento dos 3 últimos quadros são corrigidos. A correção da trajetória em relação ao eixo y não foi aplicada para que erros acumulados não alterassem a localização do objeto.

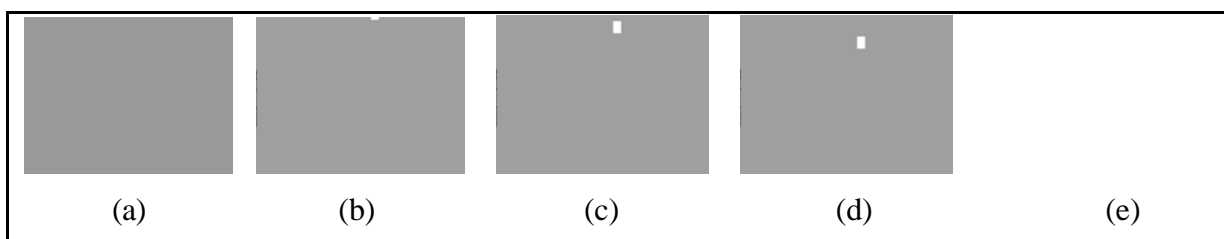
## 5 RESULTADOS

As imagens que compõem o banco de amostras utilizado nas análises estão relacionadas no apêndice B onde estão especificadas as condições de aquisição. São 23 vídeos no formato AVI decompostos quadro a quadro em imagens no formato BMP. As amostras são adquiridas em diferentes períodos do dia. Inicialmente foram testadas 2 imagens sintéticas com dimensões e deslocamento conhecidos e após testadas imagens reais. Para inferir na influência da resolução das imagens reais sobre os erros das medições, foram adquiridas imagens com duas resoluções: resolução de 320 x 280 pixels (amostras de #1 a #18) e resolução de 480 x 640 pixels (amostras de #19 a #23). Todas as imagens foram processadas pelo falcão que determinou a velocidade a dimensões do móvel para cada amostra.

Nas imagens com resolução de 320 x 240 pixels, as dimensões esperadas do veículo foram obtidas via contagem manual de pixels. Neste caso são analisados quadros em que o veículo está próximo ao centro da imagem. A velocidade esperada foi obtida via análise em vídeo: são determinadas as coordenadas da posição inicial e final do veículo (centróide), e o número de quadros entre estas duas posições. O falcão faz a medida de velocidade considerando o deslocamento do veículo em regiões próximas centro da imagem e o número de quadros para realizar este deslocamento e a taxa de amostragem da câmera. Os erros obtidos no processamento de imagens com diferentes resoluções e diferentes situações de iluminação são analisados.

### 5.1 RESULTADOS COM IMAGENS SINTÉTICAS

Para validação inicial dos algoritmos empregados no falcão, foram testadas 2 seqüências de imagens artificiais, em grayscale, criadas no Paint Shop Pró. As figuras 42 e 43 ilustram as imagens artificiais utilizadas para validação inicial dos algoritmos utilizados.



**Figura 42 – Imagem sintética 1. Quadro medindo 12x18 pixels e deslocamento de 23 pixels/quadro. a)quadro1. b)quadro 2. c)quadro 3. d)quadro 4. e)quadro 12**

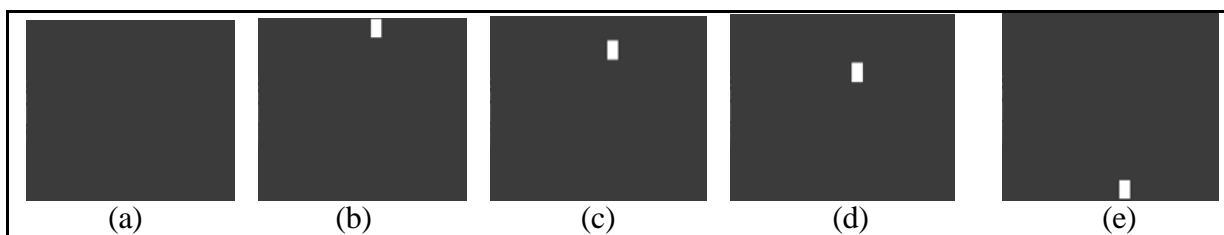


Figura 43 – Imagem sintética 2 com quadro medindo 16x24 pixels e deslocamento de 30 pixels/quadro. a)quadro1. b)quadro 2. c)quadro 3. d)quadro 4. e)quadro 12

A tabela 5 ilustra os resultados obtidos nas medidas das dimensões e velocidade de deslocamento dos objetos quando as imagens foram processadas pelo *falcão*.

Tabela 5 – Dimensões, velocidades e erros obtidos para cada amostra.

#	Num. quadros	Dim. Esp. (Pixel)		Veloc. Esperada (pixel/q)	Dim. Falcão (Pixel)		Veloc. Falcão (pixel/q)	Num. Quadros Analis.	Erro Larg (%)	Erro Comp (%)	Erro Vel. (%)
		L	C		L	C					
Art.1	13	12	18	23	12	18	23	4	0	0	0
Art.2	10	16	24	30	16	24	30	11	0	0	0

Os baixos erros obtidos nas medidas de dimensões e velocidade validam parcialmente os algoritmos utilizados pelo *falcão*. Em situações reais o trepidar da câmera, sobras projetadas no asfalto, faróis e transeuntes podem interferir na precisão das medidas.

## 5.2 RESULTADOS COM IMAGENS EM RESOLUÇÃO 320X240

As imagens do banco de amostras de #1 a #18, adquiridas em resolução de 320 x 280 pixels, e em diferentes situações de iluminação, conforme apêndice B, foram processadas pelo *falcão* para determinar as dimensões e velocidade dos veículos. A tabela 6 relaciona os valores esperados, os obtidos pelo *falcão* e os erros obtidos para cada amostra. Os dados coletados para cada amostra estão no apêndice F.

Utilizando-se relações entre medidas conhecidas de objetos em vídeo e seu número equivalente em pixels, foi possível exibir as medidas originalmente expressas em pixels, também em metros.

A figura 44 ilustra a relação *pixels/metro* utilizada para determinação das dimensões dos veículos em metros. Esta relação é válida para as amostras de #1 a #18, adquiridas com resolução de 320x240 e considerando a câmera instalada a 13,27 m de altura em relação ao solo.



Figura 44 –Relação pixel x metros para amostras #1 a #18

Assim de posse das medidas de largura e comprimento em pixels e conhecendo-se a relação *metros x pixels* determina-se as medidas das dimensões que são exibidas na interface falcão em metros. O erro das medidas é calculado em pixels.

A tabela 6 mostra os dados obtidos para dias nublados, ensolarados, na penumbra sem iluminação artificial e na penumbra com iluminação artificial.

Tabela 6 – Dimensões, velocidades e erros obtidos para cada amostra.

	(#)	Dim. Esp. (Pixel)		Veloc. Esp. (m/s)	Dim. Falcão (Pixel)		Veloc. Falcão (m/s)	Num. Quadros Analis.	Erro Larg (%)	Erro Comp (%)	Erro Vel. (%)
		L	C		L	C					
NUBLADO	1	36	105	11,08	38,5	101,5	12,53	4	-6,94	3,33	13,08
	2	17	51	11,31	14,5	49,5	9,41	4	14,71	2,94	-16,80
	3	40	104	10,00	36,5	108,7	7,32	6	8,75	-4,52	-26,80
	4	35	90	8,59	37,8	95,6	9,21	5	-8,00	-6,22	7,21
	5	40	90	9,90	39,0	91,2	9,63	5	2,50	-1,33	-2,72
SOL	6	52	115	16,67	63,0	149,0	12,10	3	21,15	-29,57	-27,41
	8	50	124	15,40	58,0	148,0	11,50	4	-16,00	-19,35	-25,32
	9	51	129	14,83	66,0	152,0	9,06	4	-29,41	-17,83	-38,91
	10	50	130	13,90	65,0	150,0	10,98	4	-30,00	-15,38	-21,01
PENUMBRA S/ ILUM. ARTIFICIAL	11	35	86	9,50	29,0	85,0	9,62	5	-17,00	1,16	1,20
	12	42	109	13,36	36,0	95,0	9,48	5	14,29	12,84	-29,04
	13	57	147	19,25	57,0	129,0	14,41	3	0,00	12,24	-25,14
	14	62	132	14,83	51,0	127,0	12,08	3	17,74	3,79	-18,54
NOITE C/ ILUMIN. ARTIFICIAL	15	54	125	15,98	54,0	107,0	13,25	3	0,00	14,40	-17,08
	16	27	67	15,20	16,0	53,0	12,08	3	40,74	20,90	-20,53
	17	51	132	17,19	57,0	121,0	9,20	5	-11,76	8,33	-46,48
	18	55	132	16,50	56,0	126,0	12,55	3	-1,82	4,55	-23,94



De acordo com análise dos dados da tabela 6, observa-se que as medidas sofrem dependência do tipo de iluminação. As medidas têm menor erro associado em dias nublados, quando a iluminação está difusa (amostras #1 a #5).

Sobre iluminação o sol inclinado (amostras #6 a #10), as sombras são interpretadas como parte do objeto a ser medido, introduzindo maior erro na medida.

No início da noite, na penumbra, (amostras #11 a #14), sem iluminação artificial, a região de interesse é subestimada em função do baixo contraste resultante da comparação do quadro analisado com o fundo, assim as medidas obtidas no processamento das imagens pelo *Falcão* são menores que as medidas estimadas. A noite, sob iluminação artificial das via públicas (amostra #15 a #18) o erro médio absoluto das dimensões é devido as sombras produzidas pela iluminação inadequada e pela projeção da luz dos faróis. As figuras 45 e 46 mostram, respectivamente, os erros obtidos nas medidas de comprimento e largura para cada amostra.

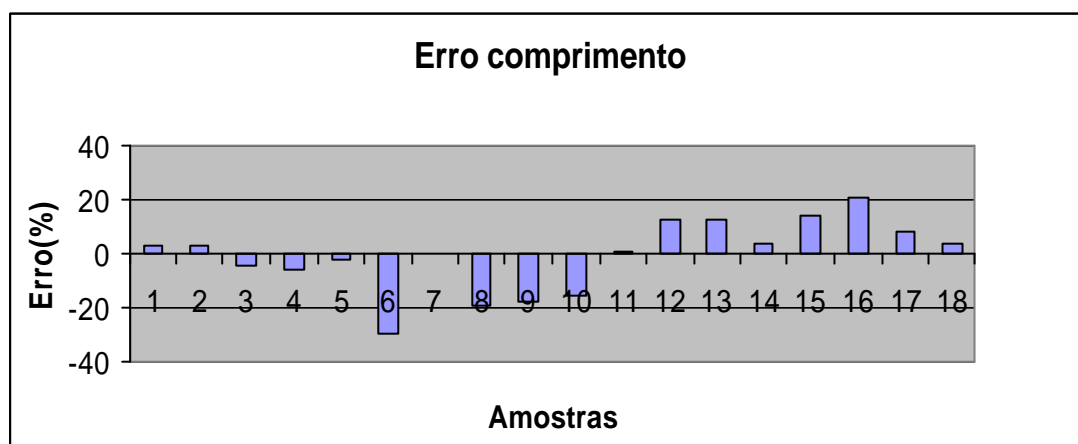


Figura 45 – Erros nas medidas de comprimento obtidos para amostras em resolução 320x240

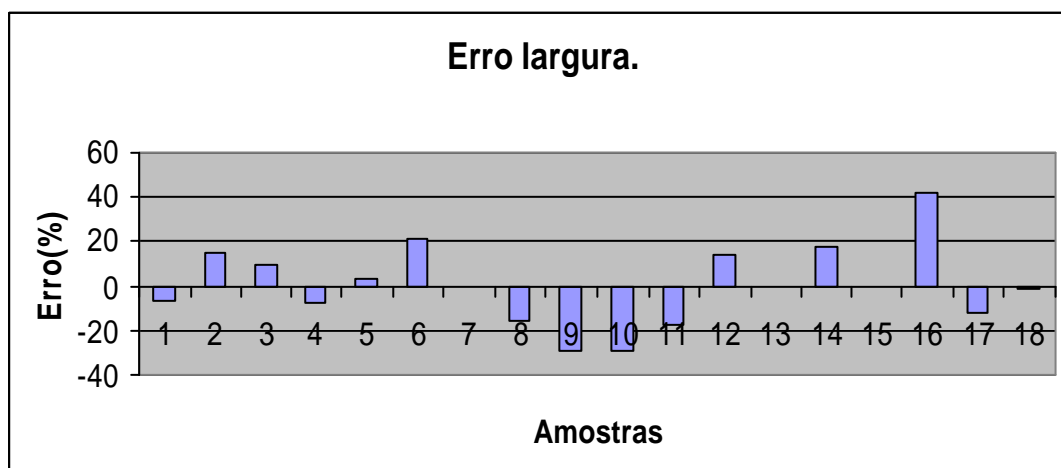


Figura 46 – Erros nas medidas de largura obtidos para amostras em resolução 320x240

A tabela 7 ilustra o erro médio absoluto das dimensões para cada classe de amostras agrupadas quanto ao tipo de iluminação.

**Tabela 7 – Erros médios absolutos nas dimensões de acordo com a classe de iluminação**

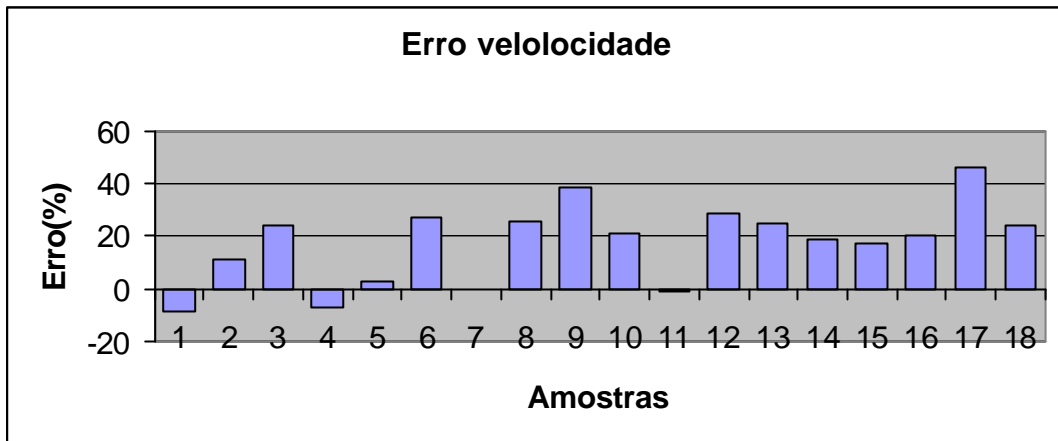
Iluminação	Erro médio absoluto (%)		
	Larg.	comp.	Vel.
Nublado	8,18	3,66	10,80
Ensolarado	24,14	20,53	28,16
Penumbra sem iluminação artificial	12,25	7,50	18,49
Noite com iluminação artificial	13,58	12,04	27,00

Assim observa-se que os resultados são mais consistentes em dias nublados e na penumbra que é atribuído a característica da iluminação difusa nestes períodos do dia: não ocorre projeção de sombras. A boa sensibilidade da câmera utilizada permite captar imagens na penumbra adequadas ao processamento, com menor erro que à noite sob luz dos faróis e iluminação pública artificial que muitas vezes projetando sombras. Na penumbra, embora o baixo erro nas medidas, a falta de iluminação também contribuído em boa parte para aumentar o percentual de erros na medida, uma vez que, neste caso, a ROI é subestimada fazendo com que as medidas obtidas pelo *Falcão* sejam menores que o valor esperado.

Os resultados do acompanhamento, *ou tracking*, se mostraram satisfatórios, alcançando 97,16 % de acerto nas imagens analisadas do banco de imagens, ou seja, apenas 10 quadros dos 351 analisados apresentaram erro visual de localização do objeto, conforme mostrado no apêndice F.

Neste caso considerou-se erro de acompanhamento quando o ponto central do objeto foi determinado com deslocamento maior que  $\frac{1}{4}$  das dimensões do objeto em pelo menos uma das coordenadas ( $P_{YC}$  ou  $P_{XC}$ ).

Conhecendo-se a taxa de amostragem da câmera, a distância percorrida e o número de quadros para uma dada distância percorrida determina-se a velocidade do móvel. em *pixels/s*, conforme equações 44, 45 e 46. A figura 47 apresenta os erros de velocidade determinados para cada amostra.



**Figura 47 – Erros nas medidas de velocidade obtidos para amostras em resolução 320x240**

Considerando-se a média aritmética dos erros absolutos, o erro médio percentual nas medidas realizadas em baixa resolução, foi:

<i>Largura</i> .....	<i>13,17 %</i>
<i>Comprimento</i> .....	<i>10,51 %</i>
<i>Velocidade</i> .....	<i>20,51 %</i>

A análise do gráfico da figura 45 mostra certa polarização dos erros nas medidas de velocidade. Nas 18 amostras analisadas 15 apresentaram velocidade menor que o valor esperado e em apenas 3 amostras o erro foi positivo. Sugere-se que o erro de 20,51% na velocidade obtida pelo falcão em relação ao valor esperado seja em função de dois aspectos:

- *O modo como a velocidade esperada foi obtida.*
- *A baixa resolução das imagens adquiridas.*

Assim, visando melhorar o grau de certeza das medidas, são propostos novos testes com amostras em maior resolução: 640x480 pixels e o conhecimento prévio das dimensões e velocidade do veículo objeto das medidas.

### 5.3 RESULTADOS COM IMAGENS EM RESOLUÇÃO 640X480

Para comprovar a influência da resolução da imagem no erro percentual obtido, imagens com resolução anterior foram analisadas. As cinco amostras foram adquiridas em dia ensolarado, com a iluminação do sol incidindo indiretamente sobre os veículos, conforme Apêndice B.

Para determinar a velocidade com maior grau de certeza, utilizou-se um velocímetro digital calibrado instalado no carro. O velocímetro digital foi calibrado comparando-se a velocidade exibida em seus displays com a velocidade medida externamente através de um cronômetro digital e dois feixes laser para iniciar e interromper a contagem. Assim, de posse do tempo gasto para o veículo percorrer a distância conhecida entre os dois feixes laser, determinou-se a velocidade do veículo. O velocímetro digital interno foi aferido tomando-se por base a velocidade medida externamente. Vários testes foram feitos até que o velocímetro fosse aferido.

A tabela 8 mostra o resultado da análise de 5 amostras com velocidades e dimensões dos objetos móveis conhecidas. As velocidades esperadas foram medidas pelo velocímetro digital aferido instalado no carro quando o veículo estava passado pela área em que as imagens estavam sendo adquiridas.

**Tabela 8 – Dimensões, velocidades e erros obtidos para amostras com resolução VGA.**

(#)	Dimensões Reais Manual (mm)		Dimensões Esperadas (Pixel)		Veloc. Esperada (Km/h)	Dimensões Falcão (Pixel)		Veloc. Falcão (Km/h)	Num. Quad. Analis.	Erro Larg (%)	Erro Comp (%)	Erro Vel. (%)
	L	C	L	C		L	C					
	19	1621	3916	43	101	11	39	93	13	13	-9,3	-7,9
20	1621	3916	43	101	20	39	94	19	8	-9,3	-6,9	5,0
21	1507	3877	40	100	25	36	102	22	7	-10,0	2,0	-12,0
22	1507	3877	40	100	29	35	99	25	6	-12,5	-1,0	-13,8
23	1507	3877	40	100	38	38	104	33	5	-5,0	4,0	-13,1

As amostras #19 e #20 são de um Carro Volkswagen- Gol ano 2004 e as amostras #21 a #23 são de um carro Chevette SL ano 1988.

A coluna número de quadros analisados da tabela 8 se refere ao número de quadros em que as medidas foram realizadas. Neste experimento utilizo-se os quadros compreendidos no intervalo [200;280] de  $P_{YC}$ .

A figura 49 ilustra a relação *pixels/metro* utilizada para determinação das dimensões dos veículos em metros. Esta relação é válida para as amostras de #19 a #23, adquiridas com resolução de 640x240 e considerando a câmera instalada a 27 m de altura em relação ao solo.

Considerando o maior número de pixels analisados nas imagens com resolução 640x480 pixels em relação as imagens de 320x240 pixels, o tempo total de execução das rotinas também aumenta. O apêndice G ilustra os tempos gastos na execução dos algoritmos, considerando a resolução VGA e computadores com clocks diferentes.



Figura 49 –Relação pixel x metros para amostras #19 a #23

As figuras 50, 51 e 52 mostram, respectivamente, os gráficos dos erros de comprimento, largura e velocidade, obtidos para as amostras #19, #20, #21, #22 e #23.

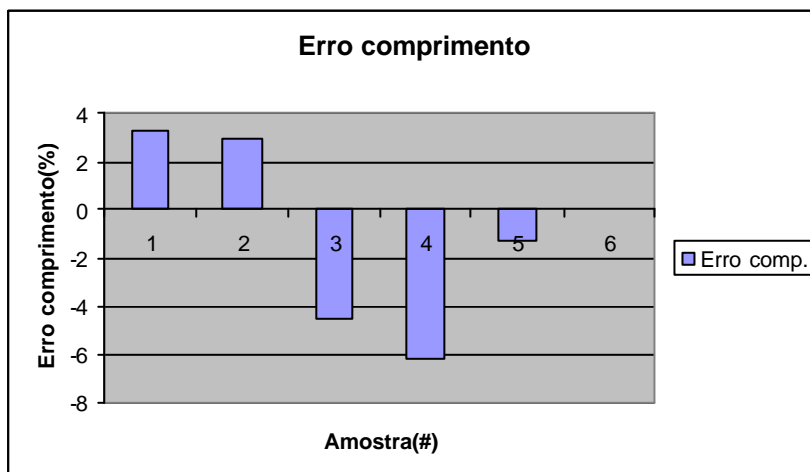


Figura 50 – Erros nas medidas de comprimento obtidos para amostras em resolução 640x320

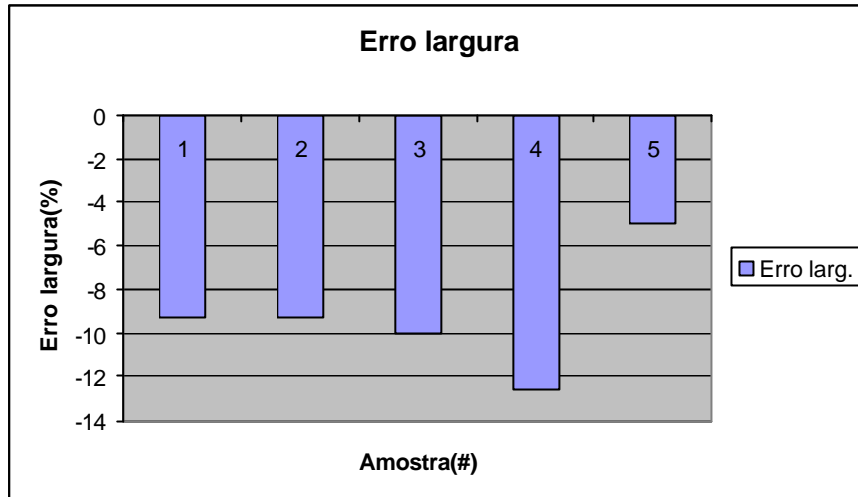


Figura 51 – Erros nas medidas de largura obtidos para amostras em resolução 640x320

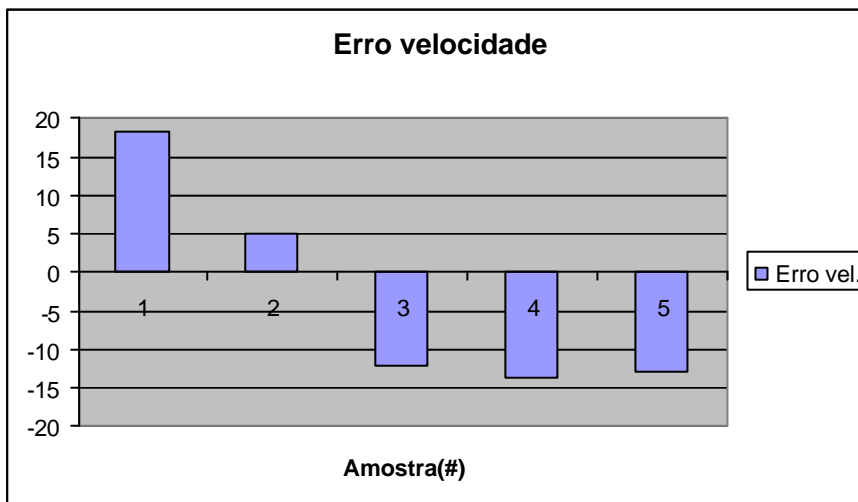


Figura 52 – Erros nas medidas de velocidade obtidos para amostras em resolução 640x320

O erro médio percentual nas medidas realizadas, considerando-se a média aritmética dos erros absolutos foi:

*Largura* ..... 9,22 %  
*Comprimento* ..... 4,36 %  
*Velocidade* ..... 12,42 %

## 6 CONCLUSÃO E TRABALHOS FUTUROS

Algoritmos baseados em análise estatística foram utilizados para detecção de movimento, estimativa de posição e dimensões, bem como a velocidade de veículos. A validação dos métodos empregados foi realizada através da análise de 23 vídeos, indicados no APÊNDICE B, adquiridos em situações reais de trânsito. O Falcão, realizou o *tracking* em 97,16 % das imagens analisadas, obteve erro médio de 11,84 % na determinação das dimensões dos veículos e 20,51 % de erro na velocidade para imagens coloridas com resolução de 320x240 pixels. Considerando que o erro poderia estar associado à baixa resolução das imagens adquiridas e que o padrão de comparação das medidas, determinado via análise das imagens em vídeo, poderia induzir a erro, foi proposta a análise de imagens em resolução de 640x480 pixels e com o padrão das medidas de largura, comprimento e velocidade, determinadas a partir do próprio objeto. Assim, o erro médio das dimensões analisadas caiu consideravelmente, passando para 11,84 % na determinação das dimensões dos veículos e 12,42 % de erro na determinação da velocidade.

A estimativa das dimensões mostrou-se foi sensível com relação às sombras projetadas no asfalto. Sensores externos poderiam ser utilizados para fornecer a direção da fonte de luz, auxiliando o processamento para reduzir a interferência das sombras. Os maiores índices de acerto foram obtidos em dias nublados quando não havia projeção de sombras. De uma forma geral as medidas obtidas pelo falcão, nas imagens com resolução de 640x480, foram menores que as medidas reais dos veículos (erro negativo). Isto pode ser observado quando o contorno do objeto é sobreposto ao objeto. Alguns componentes como pára-choques, cuja cor se confunde com a cor do asfalto, são eliminados no contorno. Como as medidas das dimensões que são exibidas, são a média das medidas realizadas em toda extensão do objeto, na média este valor tende a diminuir.

As amostras foram adquiridas com velocidade até 40 Km/h devido ao limite estabelecido na via, porém pretende-se que o sistema meça velocidades maiores. A precisão da medida de velocidade pode ser melhorada considerando maior extensão na área de medição e a correção do ângulo entre a câmera e o veículo, durante seu percurso, por regiões da imagem.

Os algoritmos para determinar o limiar de binarização e determinar o centróide do objeto foram analisados sob diferentes condições de ruído, uma vez que a câmera é submetida

a vibrações. A escolha dos algoritmos empregados foi baseada, também, no critério do menor peso computacional.

Embora utilizando o algoritmo para correção da trajetória, *o tracking* se mantenha mesmo com oclusões de alguns quadros, as informações de velocidade tendem a perder a fidelidade em casos de maiores períodos de oclusão.

Veículos de grande porte como caminhões e ônibus necessitam maior área de vídeo para ser analisada, implicando maior altura de instalação da câmera em relação ao solo ou um sistema de lentes adequado. A utilização da relação número de pixels, no vídeo, para uma medida de comprimento conhecido se mostrou adequada para determinar as dimensões reais do veículo.

Este trabalho proporcionou o contato com rotinas clássicas para o processamento de imagens, o desenvolvimento de outras, e principalmente, a vivência de campo onde questões teóricas são submetidas às interpéries da prática, como ruídos de segmentação (pixels espúrios), ruído térmico, trepidações da câmera, reflexos, sombras, transeuntes, vento, etc.

A implementação de um sistema de hardware dedicado é colocada como sugestão a fim de incrementar a performance computacional do sistema que poderá processar imagens com melhor resolução em menor tempo. A classificação dos veículos, via redes neurais, de acordo com classes relacionadas a seu tamanho, seria útil para automatizar sistemas para supervisão da frota, controle de poluição, prevenção e registro de irregularidades. O sistema de visão estéreo seria uma opção para estimativa de altura do veículo. Este dado além de contribuir para classificação do veículo pode ser utilizado para verificar se a altura do veículo é compatível com a altura permitida para cruzar túneis e pontes. Um sistema de marcação no vídeo para auxiliar o re-posicionamento nas imagens contribuiria fortemente para tornar o sistema robusto a trepidações e diminuir o erro das medidas.



## REFERÊNCIAS

AGAIAN, S. S.; PANETTA, K.; GRIGORYON, A. M. Transform-based Image Enhancement Algorithms with Performance Measure. **IEEE Transactions on Image Processing**, New York, v.10, n.3, p.367-382, mar. 2001.

AGUIAR, Pedro M. Q.; MOURA, José M. F. Maximum Likelihood Estimation of the template of a Rigid Moving Object. In: WORKSHOP ON ENERGY MINIMIZATION METHODS IN COMPUTER VISION AND PATTERN RECOGNITION, sept. 2001, **Proceedings eletrônico. . .**, France: Number 2134 in Lectures Notes in Computer Science, Springer-Verlag, 2001. Disponível em: <<http://www.isr.ist.utl.pt/~aguiar>>. Acesso em: 28 jun. 2004.

BAOXIN, Li; CHELLAPPA, Rama. A Generic Approach to Simultaneous Tracking and Verification in Video. **IEEE Transactions on Image Processing**, New York, v.11, n.5, p.530-544, may. 2002.

BAOXIN, Li; CHELLAPPA, Rama. Model-Based Temporal Object Verification Using Video. **IEEE Transactions on Image Processing**, New York, v.10, n.6, p.897-908, june. 2001.

BENSEBAA, K.; SUZIN, Amadeu. Image Analysis in Histological Sections: segmentation and qualification of tumor angiogenesis areas. **INTERNATIONAL CONFERENCE, CISST' 00**, 2000, [s.l.]. **Proceedings...**[s.l.]:[s.n.],2000, p.641-645.

CASTLEMAN, Kenneth R. **Digital Image Processing**, New Jersey: Prentice-Hall, 1996. 667 p. ISBN: 0-13-211467-4

DJUROVIĆ, Igor; STANKOVIĆ, Srdjan. Estimation of Time Varying Velocities of Moving Objects by Time-Frequency Representation. **IEEE Transactions on Image Processing**, New York, v.12, n.5, p.550-562, may. 2003.

FRAILE, R.; MAYBANK S. J. **Vehicle Trajectory Approximation and classification**. Berkshire: Department of Computer Science, The University of Reading Whiteknights, 2001, p.832-839.

GENTILE, Carnillo; CARNPS, Octavia; SZNAIER, Mario. Segmentation for Robust Tracking in the Presence of Severe Occlusion. **IEEE Transactions on Image Processing**, New York, v.13, n.2, p.166-178, feb. 2004.

GONZALEZ, Rafael C. ; WOODS, Richard E. **Digital Image Processing**, Boston, MA: Addison-Wesley, 1993. 716 p. ISBN: 0-201-50803-6.

KOYA, T. K. **Sistema Automático para Contagem das Células do Sangue**. 2001. 76 p. Tese (Mestrado em engenharia) – Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2001.

LEE, Zhenghong. Thresholding Implemented in the Frequency Domain. **IEEE Transactions on Image Processing**, New York, v.10, n.5, p.708-714, may. 2001.

LIM, Jae S. **Two Dimensional Signal and Processing**. Cambridge: Massachusetts Institute of Technology (MIT), Prentice Hall, 1990. 694p. ISBN: 0-13-935322-4.

PARKER, J. R. **Algorithms for image Processing and Computer Vision**. New York: John Wiley & Sons, inc, 1997, 417 p. ISBN: 0-471-14056-2.

PETERFREUND N. Robust Tracking of Position and Velocity with Kalman Snakes. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 21. n. 6. p. 565-569, jun. 1999.

RAKESH, R.R.; CHAUDHURI, P.; MURTHY, C.A. Thresholding in Edge Detection: a Statistical Approach. **IEEE Transactions on Image Processing**, New York, v.13, n.7, p.927-936, aug, 2004.

STEINBRUCH, A. ; WINTERLE, P. **Algebra Linear**. 2. ed. São Paulo: Mc-Graw-Hill, 1987, p 15-132. S835a.

SULLIVAN, G. D. Visual Interpretation of Known Objects in Constrained Scenes, **Phil. Trans. R.Soc.** London, p. 361-370, 1992.

SULLIVAN, G. D. et al. **Model-Based Vehicle Detection and Classification Using Orthographic Approximations**. In: Fisher, Robert B.; Trucco, Emanuele (editors). BRITISH MACHINE VISION CONFERENCE, 7. Scotland, University. **Proceedings . . .** Scotland: University of Edinburgh, 1996.

TIPLER, Paul A. **Physics for Scientists and Engineers**. 4. ed. New York: W.H. Freeman & Company, 1998. ISBN: 1572596155

UFRGS. **LILI – LaPSI Image Library**: Documentação Oficial. Versão 0.2.0, 2004. 36p. Disponível em: <<http://www.lapsi.eletr.ufrgs.br>>. Acesso em: 08 nov. 2004.

WIDROW, B. ; STEARNS, S. D. **Adaptative Signal Processing**. New Jersey: Prentice-Hall, inc, 1985, 474 p. ISBN: 0-13-004029-0

XIROUHAKIS, Y.S.; DROSOPOULOS A I. ; DELOPOULOS, A N. Efficient Optical Camera Tracking in virtual Sets. **IEEE Transactions on Image Processing**. New York, v. 10. n. 4. p.609-622, abr. 2001.

## APÊNDICE A - Método pra determinar o de tempo das rotinas.

O peso computacional é um critério que é levado em conta na avaliação de performance das rotinas elaboradas em linguagem C. Para medir o tempo de processamento das rotinas, foram utilizadas funções disponíveis na biblioteca `time.h` que faz parte dos compiladores Borland C. As avaliações foram realizadas em ambientes descritos no capítulo 3 deste trabalho. Abaixo é mostrado o trecho de código utilizado para avaliação das rotinas.

```
# include <time.h>
float    temp;
clock_t  start,end;
start = clock();    //início contagem de tempo

////////////////////
//////// rotina sob teste //////////
////////////////////

end = clock();    //fim contagem de tempo
temp = (double)(end-start)/CLOCKS_PER_SEC;
printf("A rotina foi executada em %.3f s.\n", temp);
```

Para melhorar a fidelidade da medida programas desnecessários devem estar fechados durante a avaliação e programas residentes desabilitados. O programa utilizado para determinação do tempo de processamento das rotinas apresenta variações, em torno da média, à cada nova amostragem. Tais variações se devem as formas de acesso do sistema operacional Windows que é multitarefa e trabalha com interrupções. Ainda sim, o temporizador incluso no C builder permite ter-se uma idéia geral do tempo consumido por cada rotina para comparar seus pesos computacionais. Para estimar o tempo com maior precisão foram realizadas 6 medidas para cada rotina. A tabela abaixo mostra os tempos de execução obtidos na execução de rotinas para determinação das coordenadas centrais do objeto, considerando imagens com resolução de 320x240 pixels.

**Tabela 9 – Consumo de tempo das rotinas testadas para determinar do ponto central**

Teste	Tempo (ms)	
	Método A-Centróide	Método B - Máx. Min.
1	5	10
2	4	9
3	6	16
4	6	15
5	7	18
6	7	12
Média	5,83 ms	13,33

## APÊNDICE B - Condições de aquisição das amostras

O banco de amostras é composto por vídeos adquiridos em baixa e alta resolução. As amostras de #1 a #18 foram adquiridas com resolução de 320x240 pixels, sob o viaduto Otávio Rocha, na Rua Duque de Caxias, com a câmera posicionada à 13,32 m do solo. Foram filmados veículos que transitaram na Av. Borges de Medeiros em diferentes horários do dia, com diferentes condições de iluminação. As amostras de #19 a #23 foram adquiridas na praça Dom Feliciano, com resolução de 640x480 pixels com a câmera posicionada a 26,4 m do solo. As aquisições foram realizadas de topo em relação à rodovia com auxílio de um tripé de 1,5 m de altura para fixar a câmera. Marcações no solo foram utilizadas como base de comparação para estabelecer a relação número de *pixels x dimensão* do objeto. As imagens foram adquiridas no formato MPEG e 24 bits de cores e posteriormente convertidas para seqüência de imagens no formato BMP.

**Tabela 10 – Lista de amostras com características do veículo e condições de aquisição**

Amostra (#)	Condição Iluminação	Veículo	Cor	Dimensões (Pixel)		Hora Aquisição	Data Aquisição	Núm. Quadros
				L	C			
1	Nublado	sedan	Azul Metal.	36	105	16h	25/11/2004	19
2	Nublado	Moto <sup>2</sup>	Branco	17	51	16h	25/11/2004	15
3	Nublado	sedan	Prata	40	104	16h	25/11/2004	26
4	Nublado	Clio	Azul Marin.	35	90	16h	25/11/2004	23
5	Nublado	Classe A	Branco	40	90	16h	25/11/2004	23
6	Sol	Gol Antigo	Verm. Taxi	52	115	13h	31/07/2004	20
7	Sol	Moto	Preto	30	66	14h	31/07/2004	16
8	Sol	Monza	Azul Metal	50	124	15h	31/07/2004	19
9	Sol	Kadet	Vinho	51	129	16h	31/07/2004	21
10	Sol	Sedan	Branco	50	130	17h	31/07/2004	22
11	Penumbra	Gol 84	Verm. Taxi	35	86	19h	20/11/2004	22
12	Penumbra	Vectra	Prata	42	109	20h	20/11/2004	19
13	Noite <sup>3</sup>	Gol	Branco	57	147	20h	20/11/2004	16
14	Noite	Hatch	Preto	62	132	20h	20/11/2004	20
15	Noite <sup>4</sup>	Parati	Verm. Taxi	54	125	22h	05/08/2004	18
16	Noite	Moto	Branco	27	67	22h	05/08/2004	16
17	Noite	Hatch	Vinho	51	132	22h	05/08/2004	16
18	Noite	Hatch	Branco	55	132	22h	05/08/2004	20
19	Sol / sombra	Gol	Branco	43	101	14h	25/02/2005	91
20	Sol / sombra	Gol	Branco	43	101	14h	25/02/2005	50
21	Sol / sombra	Chevette	Azul metálico	40	100	15h	25/02/2005	58
22	Sol / sombra	Chevette	Azul metálico	40	100	16h	25/02/2005	41
23	Sol / sombra	Chevette	Azul metálico	40	100	16h	25/02/2005	39

<sup>2</sup> A cor das motos é pouco visível na vista de topo.

<sup>3</sup> Nas amostras 13 e 14 os faróis dos veículos estão desligados. Iluminação pública ligada

<sup>4</sup> As amostras 15, 16, 17 e 18 foram adquiridas com a iluminação pública e os faróis dos veículos ligados.

## APÊNDICE C Banco de amostras para calibração do sistema

As amostras para calibração do Falcão foram adquiridas em laboratório com condições. A altura da câmera ao solo: 2,32 m; distância percorrida de 1,50 m, conforme amostra; iluminação incandescente direta. Aquisição foi realizada de topo em relação ao solo com auxílio de um tripé para fixação da câmera Canon Power Shot A310.

As marcações no chão servem como referência para determinar nas imagens processadas a relação *número de pixels x dimensão*. A amostragem 15 quadros por segundo da câmera é confiável para determinação do tempo, assim com a análise das imagens em vídeo determina-se o número de quadros entre o início do percurso e seu término e portanto o tempo gasto na trajetória. As medidas de tempo foram realizadas com o móvel em velocidade constante dentro do percurso de interesse. A tabela 11 mostra os dados obtidos.

**Tabela 11 – Lista de amostras obtidas em ambiente controlado.**

<b>Amostra</b>	<b>Num. quadros do percurso</b>	<b>Núm. total de quadros</b>	<b>Tempo (s)</b>	<b>Velocidade (m/s)</b>
A	22	20	1,46	102,74
B	23	19	1,53	98,03
C	23	20	1,53	98,03
D	24	20	1,60	93,75

## APÊNDICE D - Biblioteca de funções filtro.h

/\* Funções desenvolvidas para filtrar imagens

Filtro Library

Alexandre Gaspary Haupt - haupt@cpovo.net

Criada em 01/03/2004 Atualizada em 01/12/2004

LaPSI - Laboratório de Procesamento de Sinais e Imagem

DELET - Departamento de Engenharia Elétrica

UFRGS - Universidade Federal do Rio Grande do Sul

C:\Arquivos de programas\Borland\lili\lili\include  
\*/

#ifndef \_FILTRO\_H

#define \_FILTRO\_H

#include "lili.h"

int lLimiar\_estatistico(limage\* image1);

limage\* lret\_cut(limage\* src, int x1, int y1, int x2, int y2);

limage\* lconvert\_gray(limage\* imageCor);

int lLimiar(limage\* src);

void lcentroideA(limage\* imagem1);

void lcentroideB(limage\* imagem1);

limage\* lstrech\_gray(limage\* imagem1);

limage\* lbinariza(limage\* imagem1, int th1);

limage\* lspatial\_filter\_gray(limage\* imagem1, int mask);

limage\* lLimiar\_laplace\_gray(limage\* imagem1);

limage\* lprewitt(limage\* imagem1);

limage\* lsobel(limage\* imagem1);

limage\* legde\_link(limage\* imagem1);

limage\* linverte(limage\* imagem1);

limage\* lseleciona\_road(limage\* imagem1, long int x1c);

#endif // \_FILTRO\_H

## APÊNDICE E - Bibliotecas de funções hali.h

/\* Funções desenvolvidas para manipular imagens

HALI - Haupt Library  
Alexandre Gaspary Haupt - haupt@cpovo.net  
Criada em 01/03/2004 Atualizada em 01/12/2004  
LaPSI - Laboratório de Procesamento de Sinais e Imagem  
DELET - Departamento de Engenharia Elétrica  
UFRGS - Universidade Federal do Rio Grande do Sul

C:\Arquivos de programas\Borland\lili\lili\include  
\*/

```

                                #ifndef _HALI_H
                                #define _HALI_H
                                    #include "lili.h"
                                    #include "hali.h"

                                int findmov(limage* imagem1, int m);
                                limage* lsubtrai(limage* imagem1, limage* imagem2);
                                limage* logicaE(limage* imagem1, limage* imagem2);
                                    limage* l copia(limage* imagem1,int type);
                                long int reduce_shake_horizontal(int cont,long int numero);
                                long int reduce_shake_vertical(int cont,long int numero);
                                    limage* plot_pos(limage* imagem1);
                                    float speed(int m);
                                    int find_dimensao(int m);
                                void RGB_to_HSI(limage* im_RGB,long int centrox,long int centroy);
                                limage* lrotaciona(limage* imagem1,int graus);
                                limage* ldraw_window(limage* imagem1,long int xc, long int yc,int lado );
                                limage* ltranslada(limage* imagem1,long int x1c, long int y1c, long int x2c, long int y2c);
                                limage* lcopy_paste_window(limage* imagem1,long int x1c, long int y1c, long int x2c, long int y2c);
                                limage* lzoom(limage* imagem1,long int x, long int y, int esc);
                                limage* lcopy_window(limage* imagem1,long int xc, long int yc);
                                limage* lseleciona_roi(limage* imagem1,long int x1c, long int y1c);
                                limage* lseleciona_road(limage* imagem1);

                                #endif // _HALI_H
```

Neste capítulo são analisados os resultados obtidos no processamento das amostras tabeladas no APÊNDICE B. São apresentados os resultados obtidos no processamento de cada seqüência de imagens pelo Falcão.

**APÊNDICE F – Resultado processamento amostras 320 x 240 pixels**

**RESULTADOS OBTIDOS COM AMOSTRA #1**

**Tabela 12– Resultados das dimensões da amostra #1**

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
11	78	39	111
12	100	39	110
13	128	38	102
14	149	38	83
<b>Resultados</b>	Desloc.	Largura média	Comprimento médio
	72 pxls	38,5 pxls	101,5 pxls

Deslocamento ..... 72 pxls      → 3,346 m

Tempo ..... 4 quadros      → 0,266 s

Velocidade ..... 12,53 m/s      → 45 Km/h

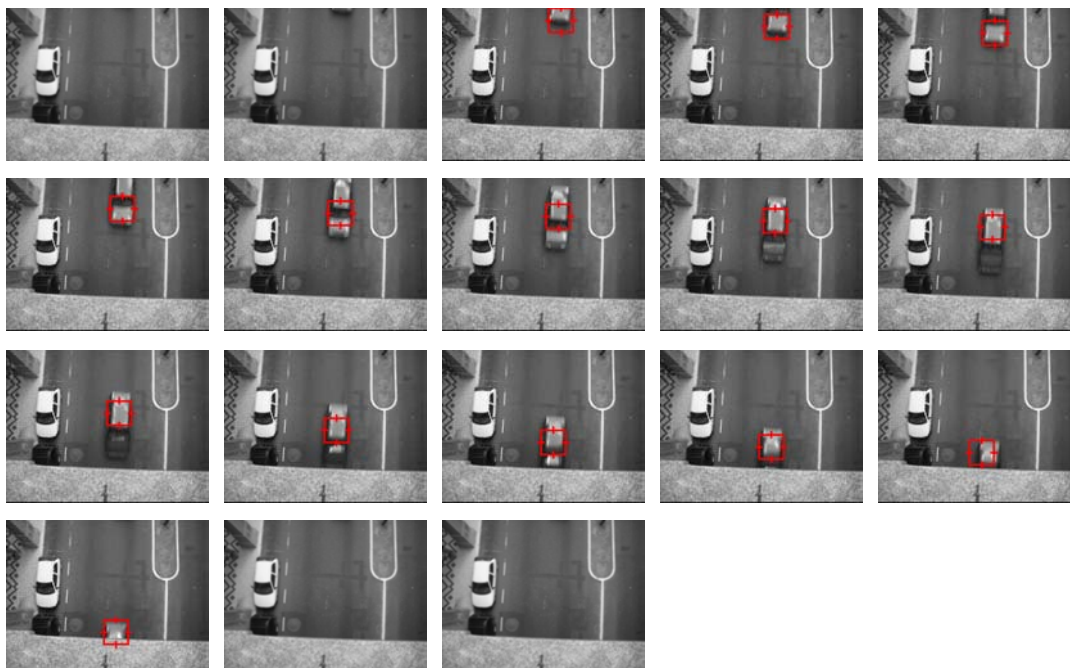


Figura 53 –Tracking veículo amostra #1.



## RESULTADOS OBTIDOS COM AMOSTRA #2

Tabela 13 – Resultados das dimensões da amostra #2

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
9	88	11	51
10	107	16	51
11	126	16	50
12	141	15	47

Resultados	Desloc.	Largura média	Comprimento médio
	54 pxls	14,5 pxls	49,5 pxls

Deslocamento ..... 54 pxls      → 2,509 m

Tempo ..... 4 quadros      → 0,266 s

Velocidade ..... 9,41 m/s      → 34 Km/h

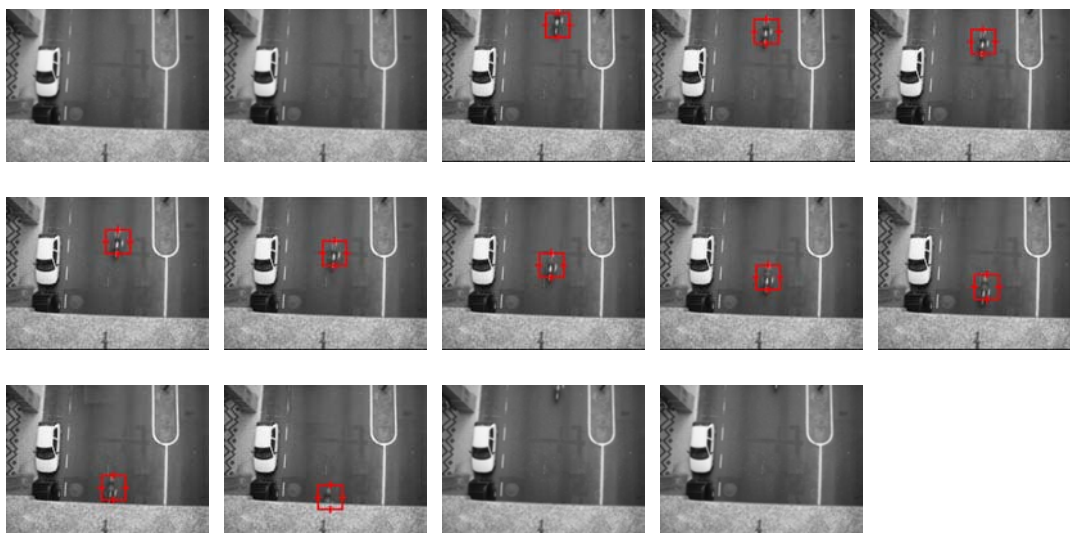


Figura 54 –Tracking veículo amostra #2.

## RESULTADOS OBTIDOS COM AMOSTRA #3

Tabela 14 – Resultados das dimensões da amostra #3

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
13	83	36	110
14	100	37	110
15	113	36	106
16	124	37	105
17	136	36	89
18	145	37	66

Resultados	Desloc.	Largura média	Comprimento médio
	63 pxls	36,5 pxls	108,7 pxls

Deslocamento ..... 63 pxls      → 2,93 m

Tempo ..... 6 quadros      → 0,4 s

Velocidade ..... 7,32 m/s      → 26 Km/h

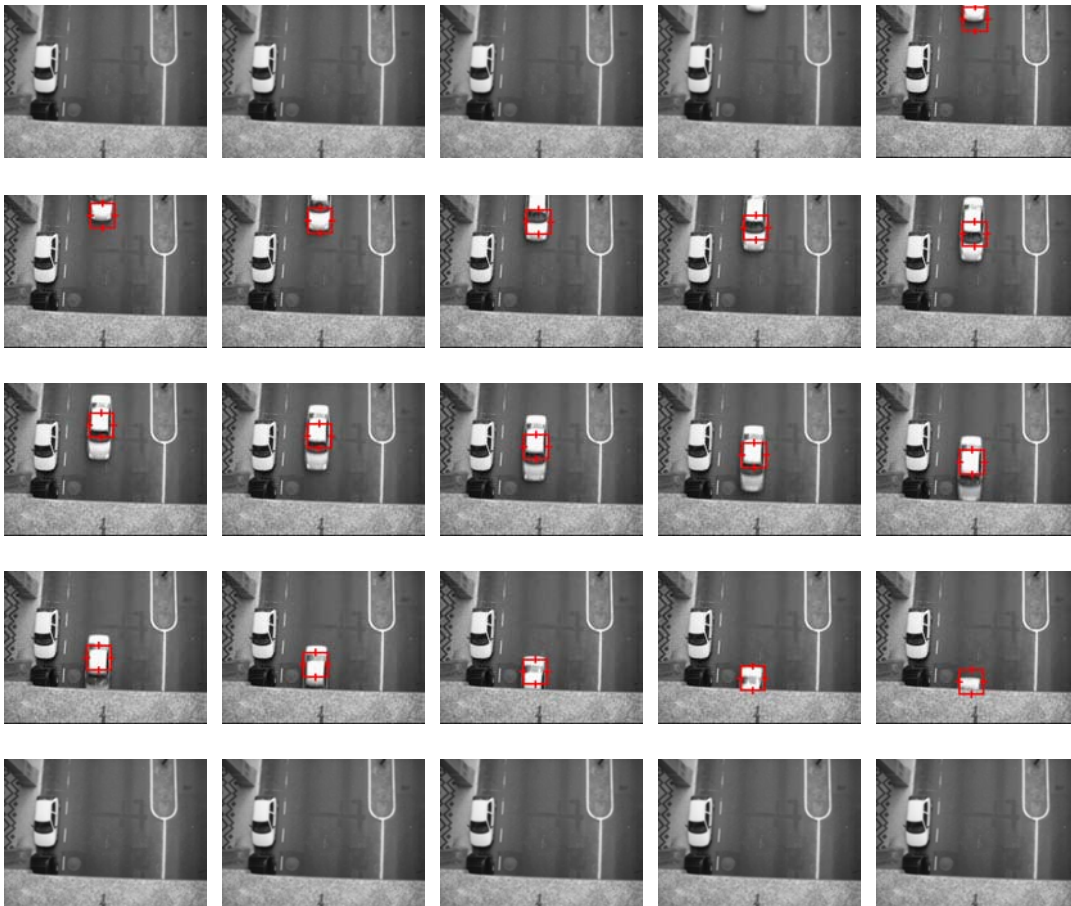


Figura 55 –Tracking veículo amostra #3.

## RESULTADOS OBTIDOS COM AMOSTRA #4

Tabela 15 – Resultados das dimensões da amostra #4

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
12	73	37	93
13	88	38	94
14	104	38	94
15	119	38	95
16	138	38	102

Resultados	Desloc.	Largura média	Comprimento médio
	66 pxls	37,8 pxls	95,6 pxls

Deslocamento ..... 66 pxls      → 3,07 m  
 Tempo ..... 5 quadros      → 0,333 s  
 Velocidade ..... 9,21 m/s      → 33 Km/h

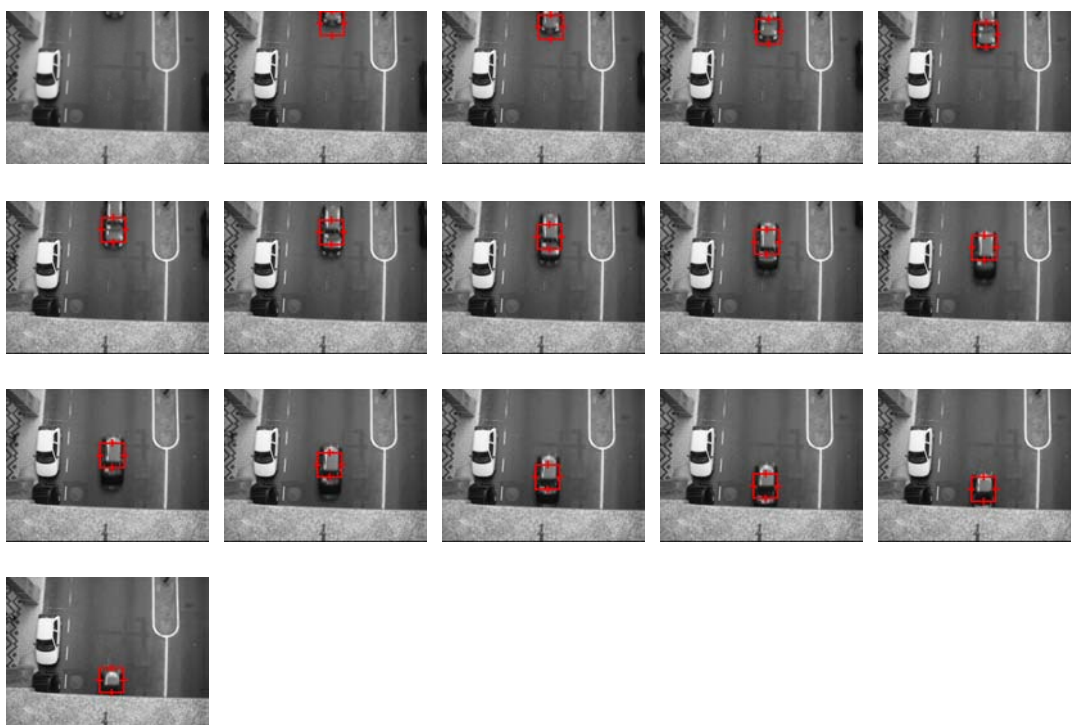


Figura 56 –Tracking veículo amostra #4.

## RESULTADOS OBTIDOS COM AMOSTRA #5

Tabela 16 – Resultado das dimensões da amostra #5

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
11	75	38	92
12	92	39	93
13	109	39	92
14	125	40	88
15	143	39	91

Resultados	Desloc.	Largura média	Comprimento médio
	69 pxls	39 pxls	91,2 pxls

Deslocamento ..... 68 pxls      → 3,21 m  
 Tempo ..... 5 quadros      → 0,333 s  
 Velocidade ..... 9,63 m/s      → 35 Km/h

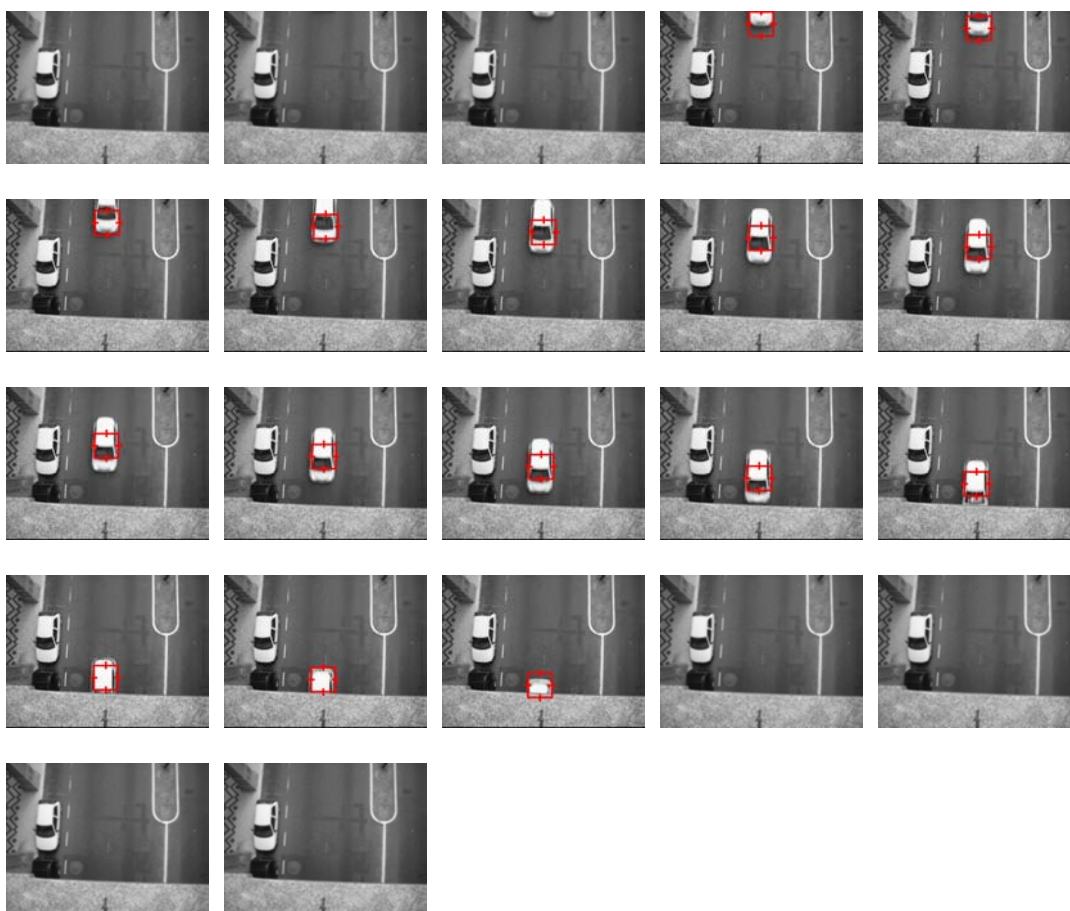


Figura 57 –Tracking veículo amostra #5.

## RESULTADOS OBTIDOS COM AMOSTRA #6

Tabela 17 – Resultados das dimensões da amostra #6

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
9	90	62	144
10	123	65	149
11	141	63	154

Resultados	Desloc.	Largura média	Comprimento médio
	52 pxls	63,0 pxls	149,0 pxls

Deslocamento ..... 52 pxls      → 2,42 m

Tempo ..... 3 quadros      → 0,200 s

Velocidade ..... 12,1 m/s      → 43 Km/h

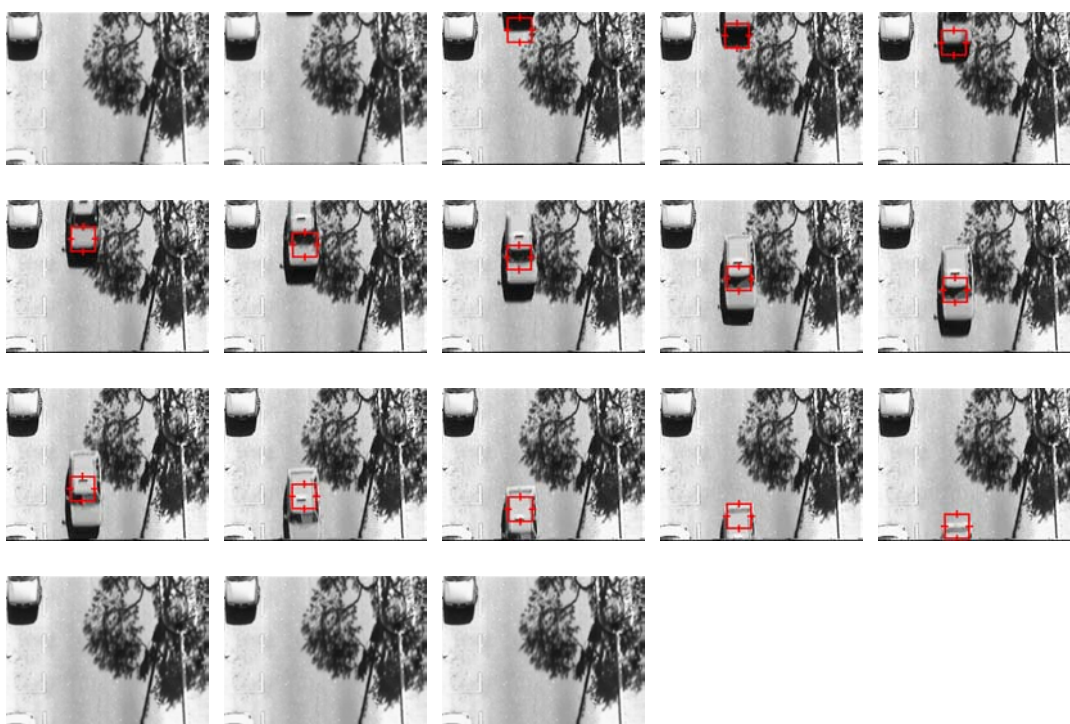


Figura 58 –Tracking veículo amostra #6.

## RESULTADOS OBTIDOS COM AMOSTRA #8

Tabela 18 – Resultados das dimensões da amostra #8

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
8	82	61	151
9	102	58	157
10	130	57	157
11	147	58	130

Resultados	Desloc.	Largura média	Comprimento médio
	66 pxls	58,0 pxls	148,0 pxls

Deslocamento ..... 66 pxls      → 3,07 m  
 Tempo ..... 4 quadros      → 0,266 s  
 Velocidade ..... 11,51 m/s      → 41 Km/h

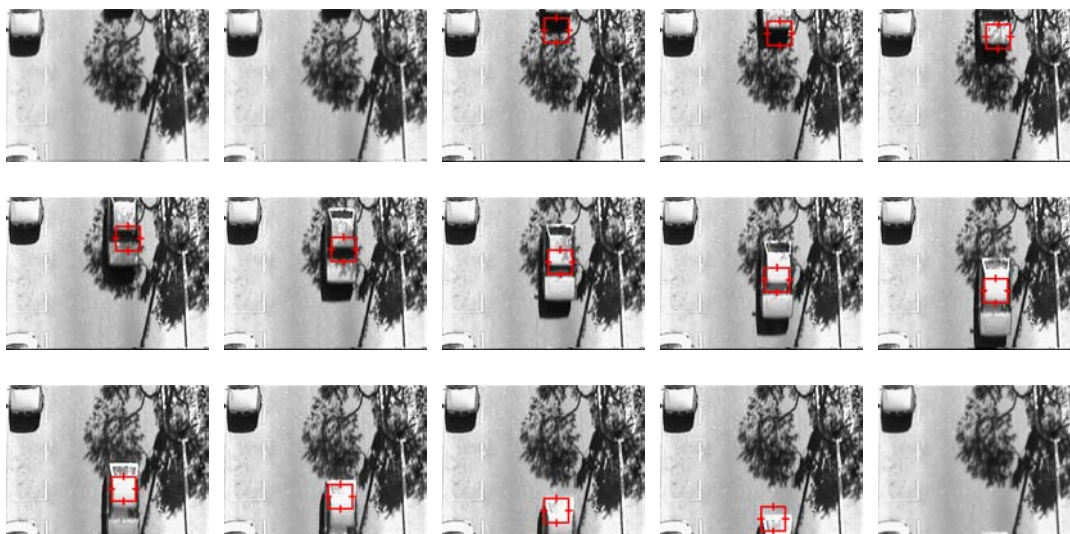


Figura 59 –Tracking veículo amostra #8.



## RESULTADOS OBTIDOS COM AMOSTRA #9

Tabela 19– Resultados das dimensões da amostra #9

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
10	80	64	157
11	110	67	160
12	134	68	139
13	131	65	139

Resultados	Desloc.	Largura média	Comprimento médio
	52 pxls	66,0 pxls	152,0 pxls

Deslocamento ..... 55 pxls      → 2,56 m  
 Tempo ..... 3 quadros      → 0,333 s  
 Velocidade ..... 12,08 m/s      → 46 Km/h

A Linha destacada da tabela indica erro na determinação da posição da centróide. Este erro ocorre em função de vibrações da câmera e movimentação da vegetação. A estimativa de velocidade é também alterada em função disto.

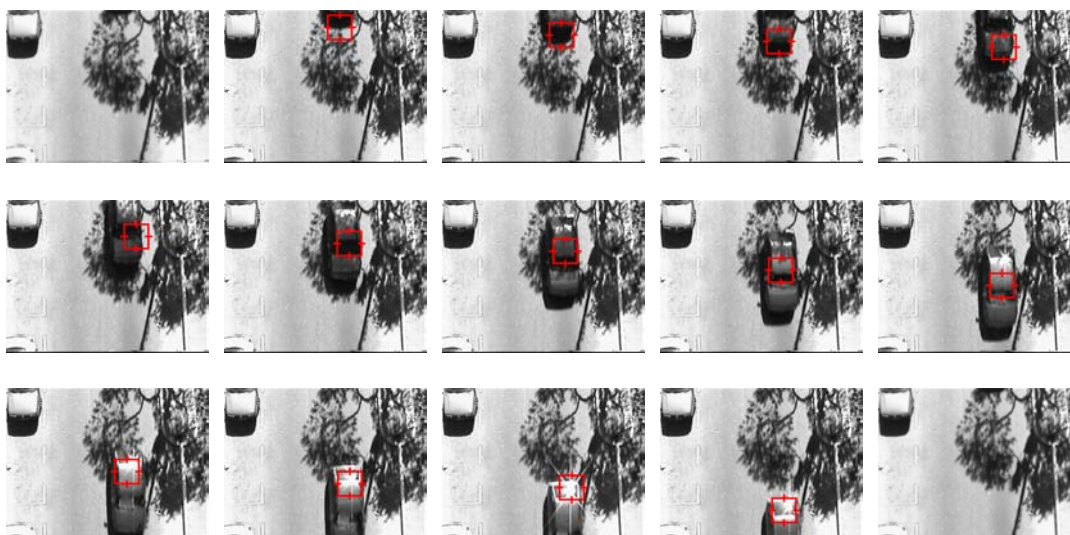


Figura 60 –Tracking veículo amostra #9.

## RESULTADOS OBTIDOS COM AMOSTRA #10

Tabela 20– Resultados das dimensões da amostra #10

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
9	81	63	144
10	101	63	158
11	125	67	160
12	143	67	140

Resultados	Desloc.	Largura média	Comprimento médio
	63 pxls	65,0 pxls	150,0 pxls

Deslocamento ..... 63 pxls      → 2,93 m  
 Tempo ..... 4 quadros      → 0,2667 s  
 Velocidade ..... 10,98 m/s      → 40 Km/h

O sol projeta sombra lateralmente e a frente do objeto que se confunde com o próprio objeto na imagem binarizada e no diagrama de bordas.

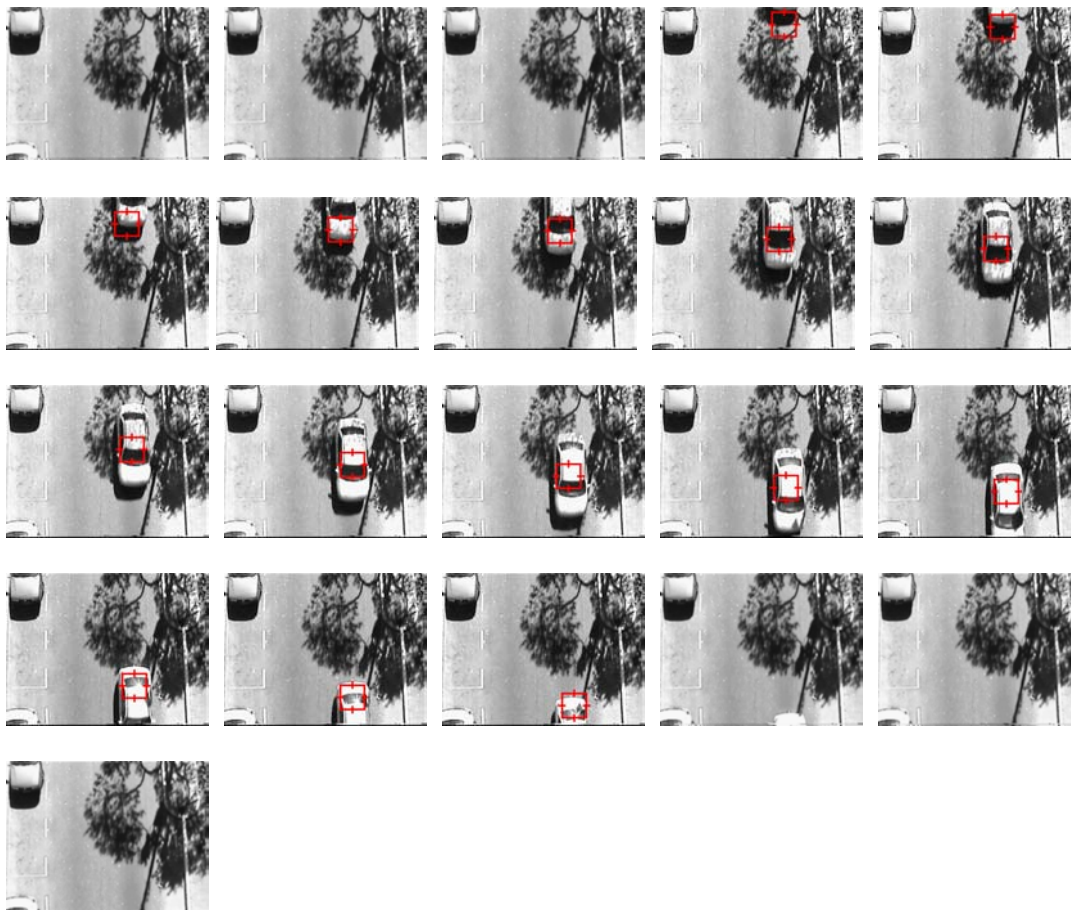


Figura 61 –Tracking veículo amostra #10.



## RESULTADOS OBTIDOS COM AMOSTRA #11

Tabela 21– Resultados das dimensões da amostra #11

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
12	79	30	84
13	92	30	85
14	110	30	87
15	127	29	89
16	147	30	83

Resultados	Desloc.	Largura média	Comprimento médio
	69 pxls	29,0 pxls	85,0 pxls

Deslocamento ..... 69 pxls      → 3,21 m  
 Tempo ..... 5 quadros      → 0,3333 s  
 Velocidade ..... 9,62 m/s      → 35 Km/h

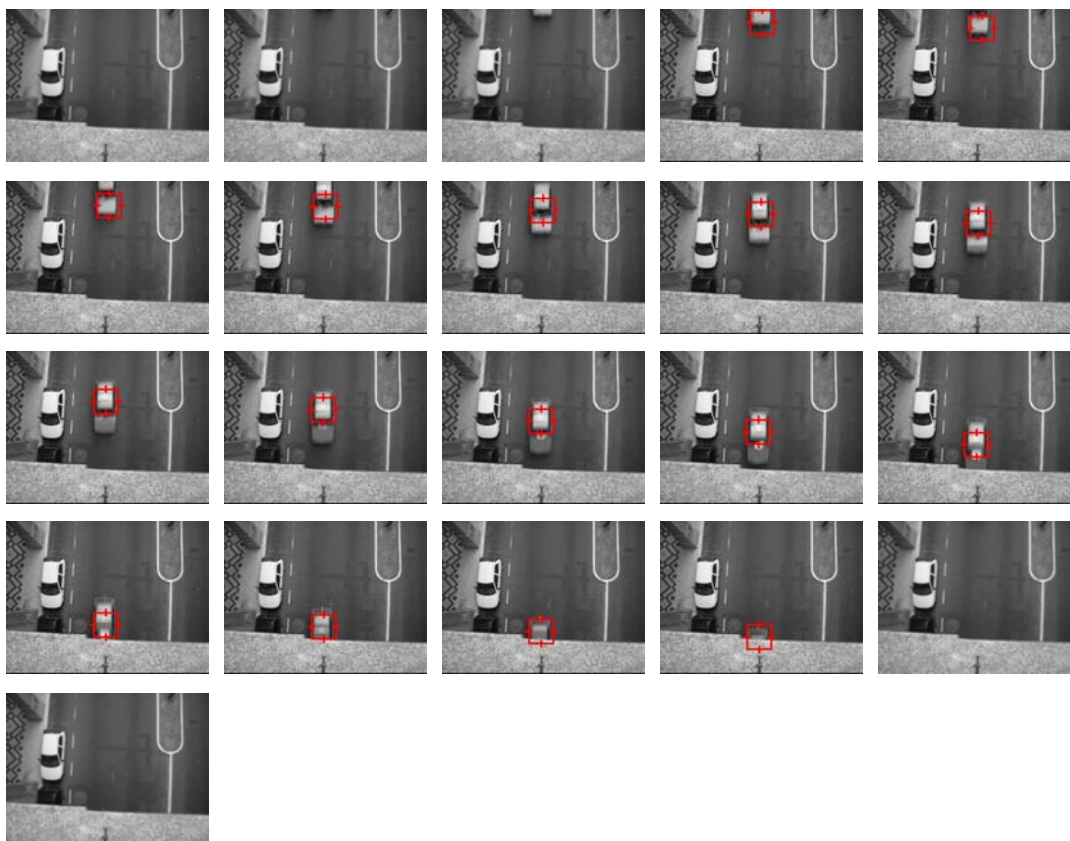


Figura 62 –Tracking veículo amostra #11.

## RESULTADOS OBTIDOS COM AMOSTRA #12

Tabela 22– Resultados das dimensões da amostra #12

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
9	78	39	107
10	102	39	107
11	125	38	106
12	136	36	89
13	145	30	67

Resultados	Desloc.	Largura média	Comprimento médio
	68 pxls	36,0 pxls	95,0 pxls

Deslocamento ..... 68 pxls      → 3,16 m  
 Tempo ..... 5 quadros      → 0,3333 s  
 Velocidade ..... 9,48 m/s      → 34 Km/h

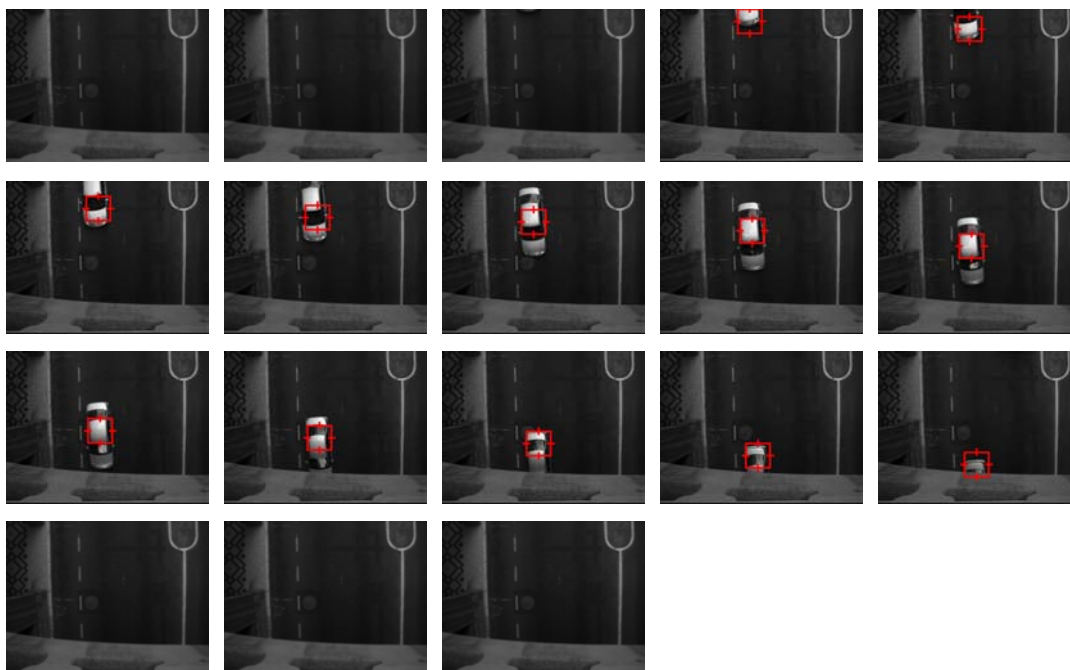


Figura 63 –Tracking veículo amostra #12.

## RESULTADOS OBTIDOS COM AMOSTRA #13

Tabela 23 – Resultados das dimensões da amostra #13

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
8	82	57	126
9	113	57	128
10	143	57	133

Resultados	Desloc.	Largura média	Comprimento médio
	62 pxls	57,0 pxls	129,0 pxls

Deslocamento ..... 62 pxls      → 2,88 m  
 Tempo ..... 3 quadros      → 0,2000 s  
 Velocidade ..... 14,41 m/s      → 52 Km/h



Figura 64 –Tracking veículo amostra #13.

## RESULTADOS OBTIDOS COM AMOSTRA #14

Tabela 24– Resultados das dimensões da amostra #14

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
8	73	52	120
9	95	51	133
10	124	50	128

Resultados	Desloc.	Largura média	Comprimento médio
	52 pxls	51,0 pxls	127,0 pxls

Deslocamento ..... 52 pxls      → 2,42 m  
 Tempo ..... 3 quadros      → 0,2000 s  
 Velocidade ..... 12,08 m/s      → 44 Km/h

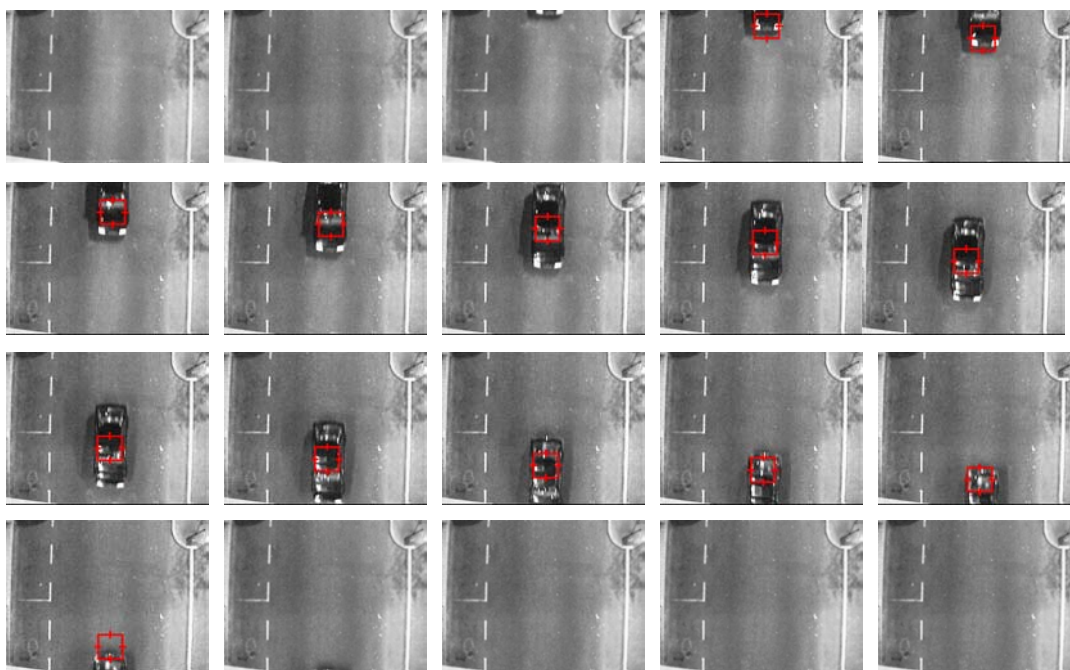


Figura 65 –Tracking veículo amostra #14.

## RESULTADOS OBTIDOS COM AMOSTRA #15

Tabela 25– Resultados das dimensões da amostra #15

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
8	89	62	111
9	118	51	102
10	145	51	109

Resultados	Desloc.	Largura média	Comprimento médio
	57 pxls	54,0 pxls	107,0 pxls

Deslocamento ..... 57 pxls      → 2,65 m  
 Tempo ..... 3 quadros      → 0,2000 s  
 Velocidade ..... 13,25 m/s      → 48 Km/h

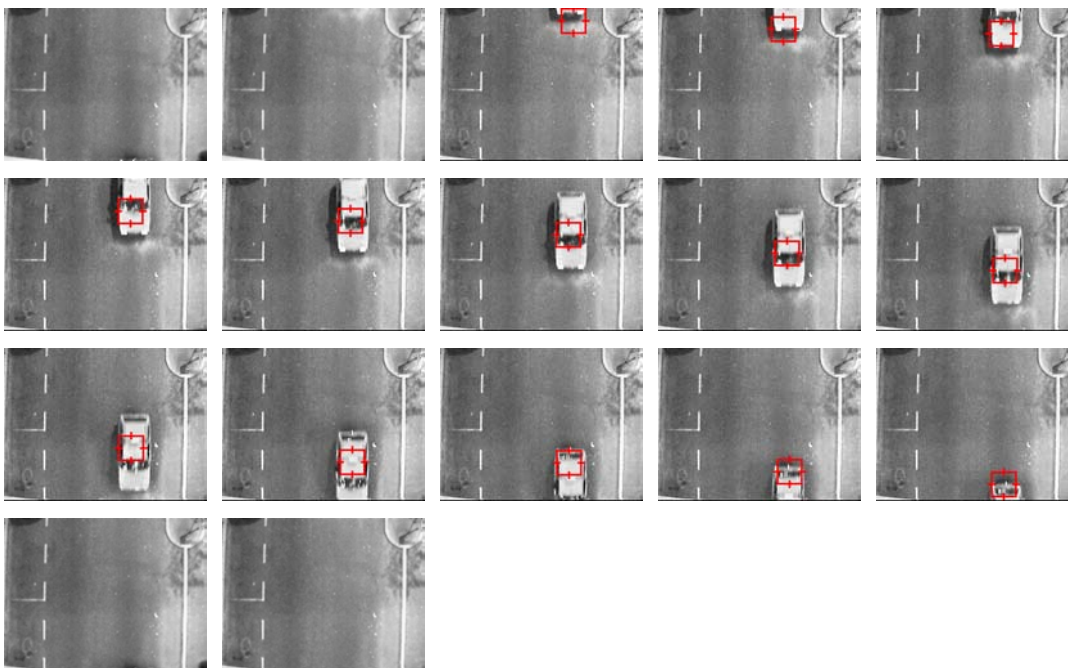


Figura 66 –Tracking veículo amostra #15.

## RESULTADOS OBTIDOS COM AMOSTRA #16

Tabela 26– Resultados das dimensões da amostra #16

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
9	97	17	64
10	124	17	45
11	148	15	51

Resultados	Desloc.	Largura média	Comprimento médio
	52 pxls	16,0 pxls	53,0 pxls

Deslocamento ..... 52 pxls      → 2,42 m  
 Tempo ..... 3 quadros      → 0,2000 s  
 Velocidade ..... 12,08 m/s      → 44 Km/h

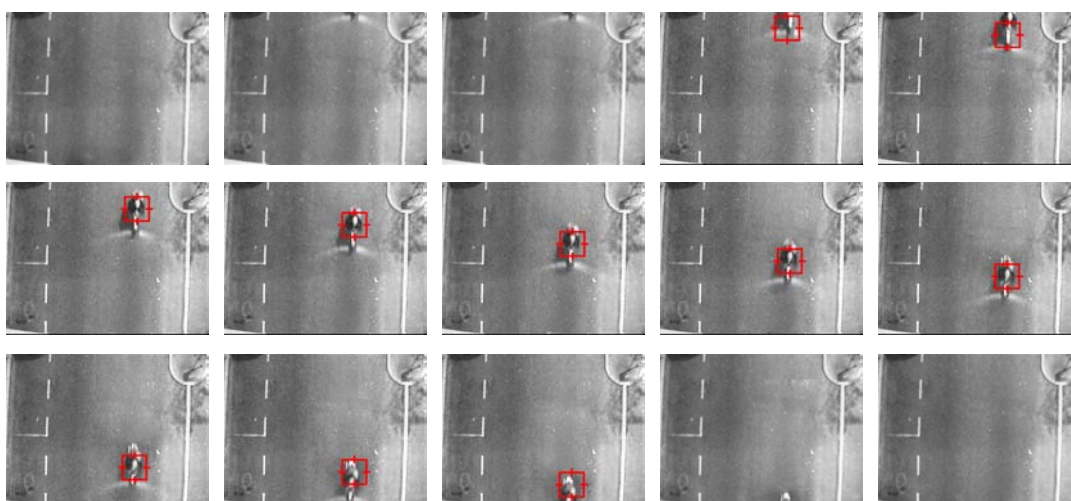


Figura 67 –Tracking veículo amostra #16.

## RESULTADOS OBTIDOS COM AMOSTRA #17

Tabela 27– Resultados das dimensões da amostra #17

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
6	79	65	107
7	83	65	135
8	116	53	136
9	145	54	134
10	144	51	93

Resultados	Desloc.	Largura média	Comprimento médio
	66 pxls	57,0 pxls	121,0 pxls

Deslocamento ..... 66 pxls      → 3,07 m  
 Tempo ..... 5 quadros      → 0,3333 s  
 Velocidade ..... 9,20 m/s      → 33 Km/h



Figura 68 –Tracking veículo amostra #17.



## RESULTADOS OBTIDOS COM AMOSTRA #18

Tabela 28– Resultados das dimensões da amostra #18

Quadro	P <sub>YC</sub> (pixel)	Largura (pixel)	Comprimento (pixel)
9	80	56	125
10	107	57	127
11	133	57	126

Resultados	Desloc.	Largura média	Comprimento médio
	54 pxls	56,0 pxls	126,0 pxls

Deslocamento ..... 54 pxls      → 2,51 m  
 Tempo ..... 3 quadros      → 0,2000 s  
 Velocidade ..... 12,55 m/s      → 45 Km/h

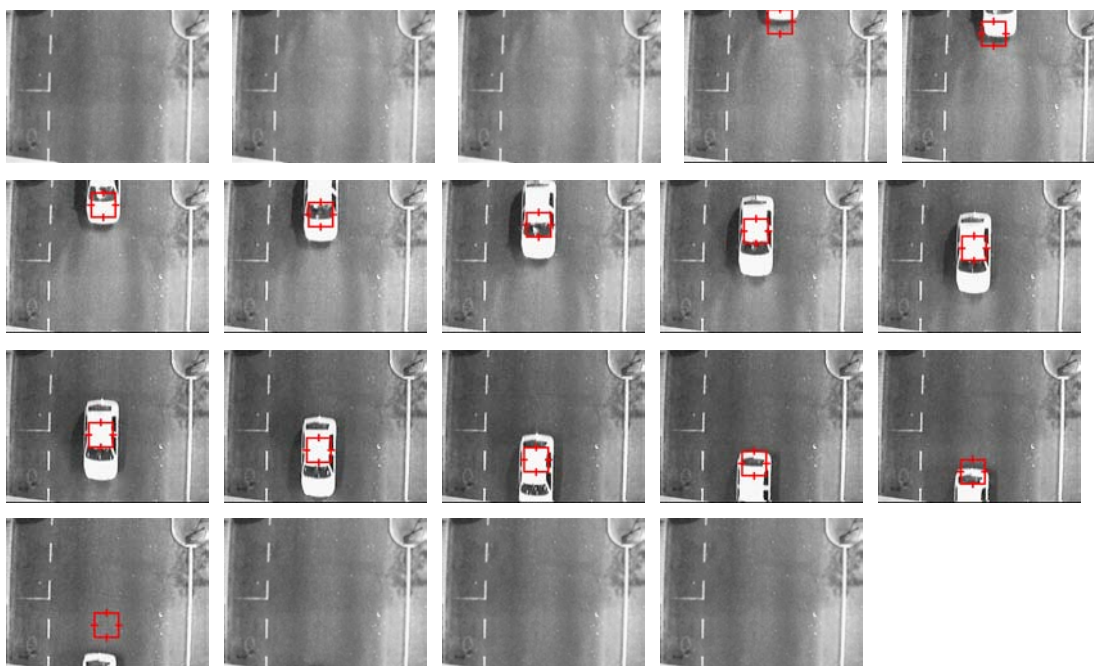
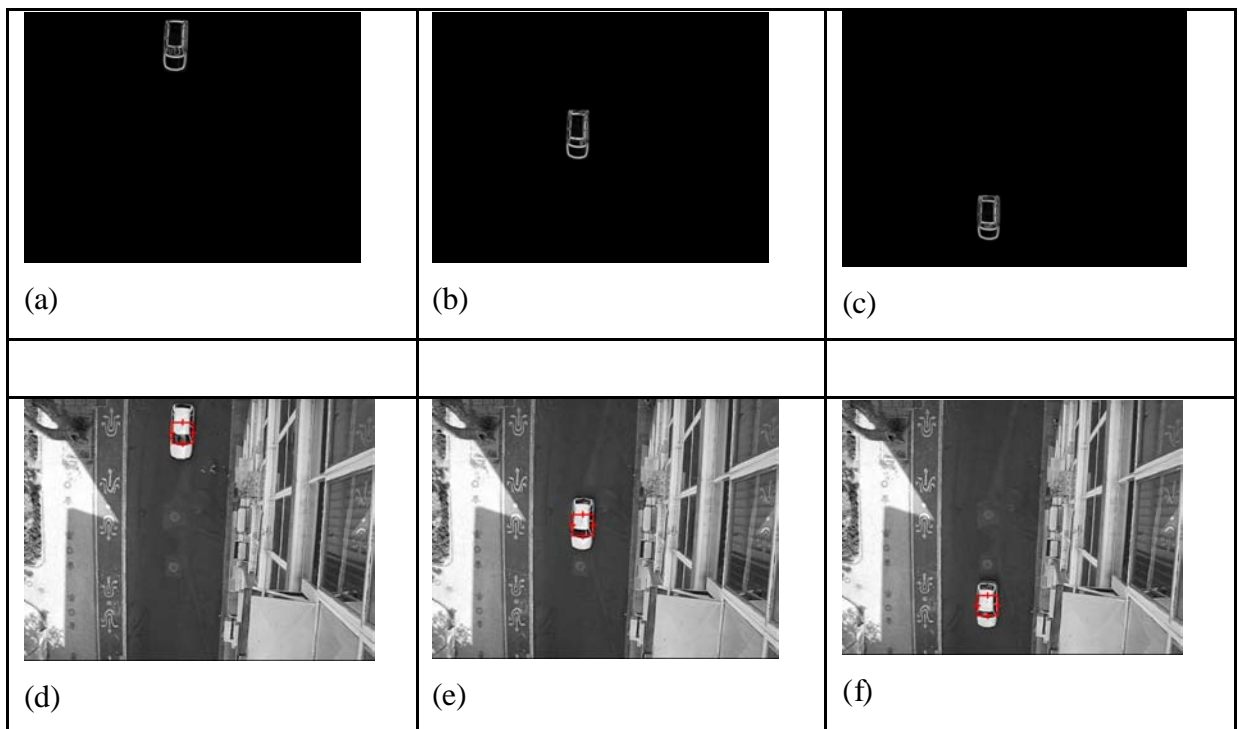


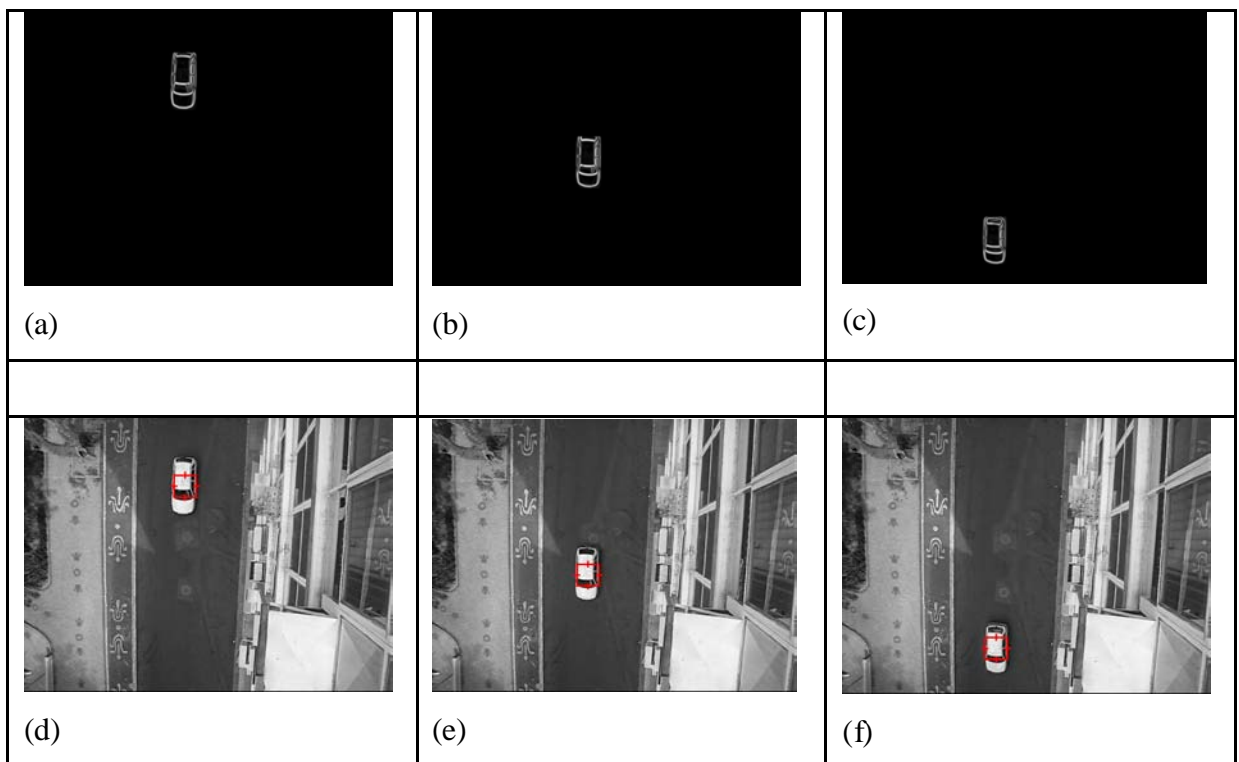
Figura 69 –Tracking veículo amostra #18.



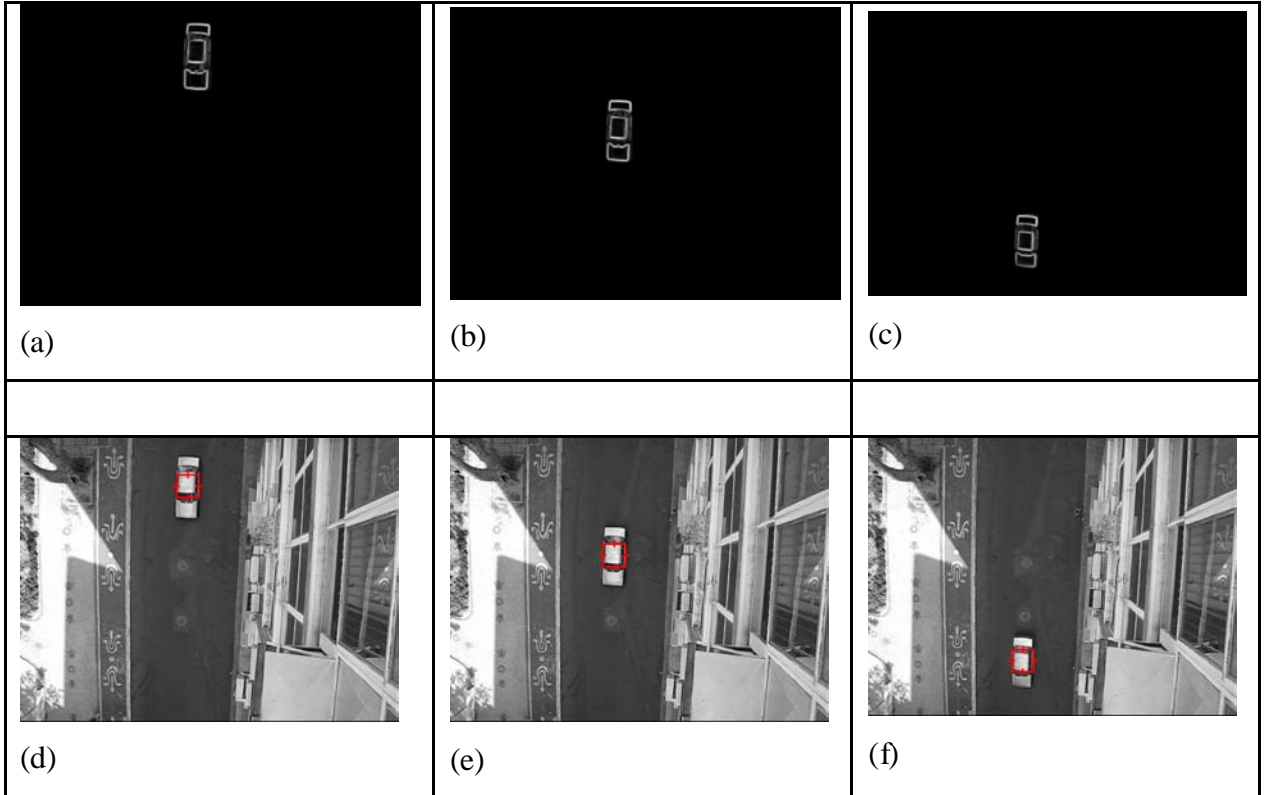
**APÊNDICE G –Tela de Resultados das amostras de 640 X 480 PIXELS**



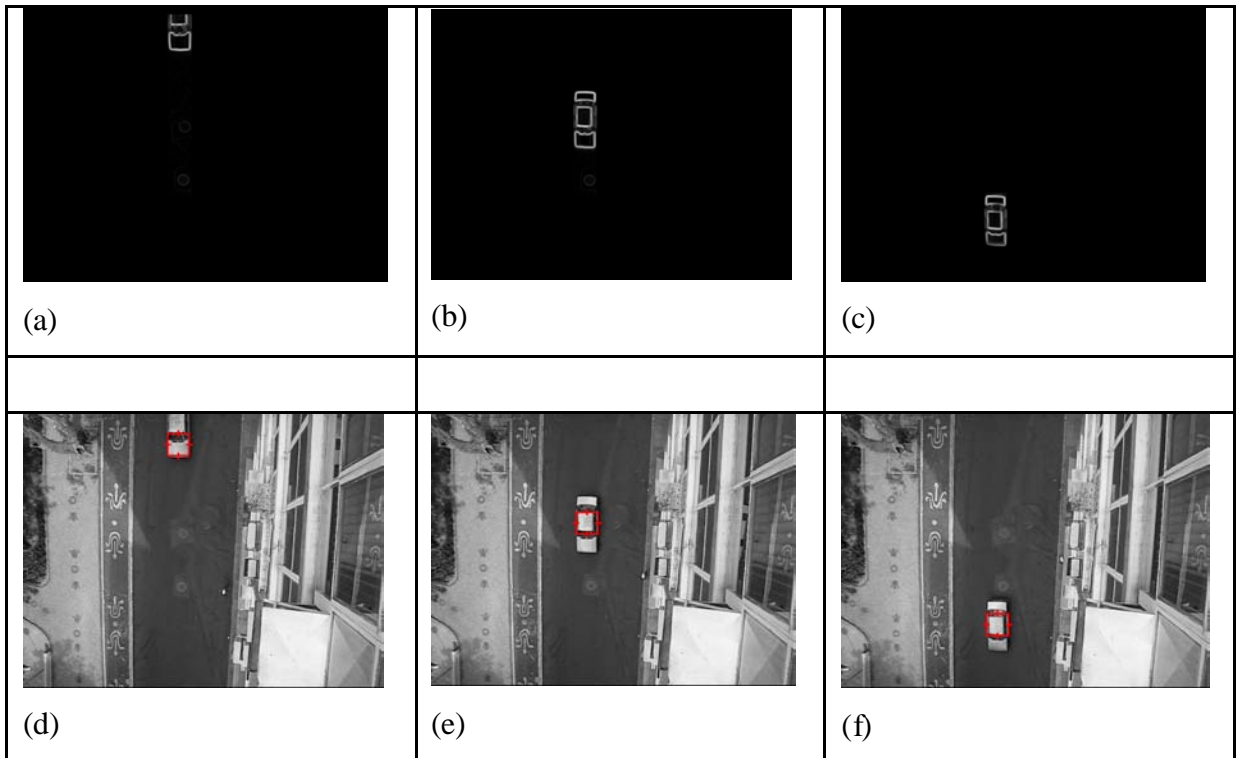
**Figura 70 –Resultados processamento amostra #19. a) gradiente quadros #24. b) gradiente quadro #51. c) gradiente quadro #75. d) tracking quadros #24 e) tracking quadro #51. f) tracking quadro #75.**



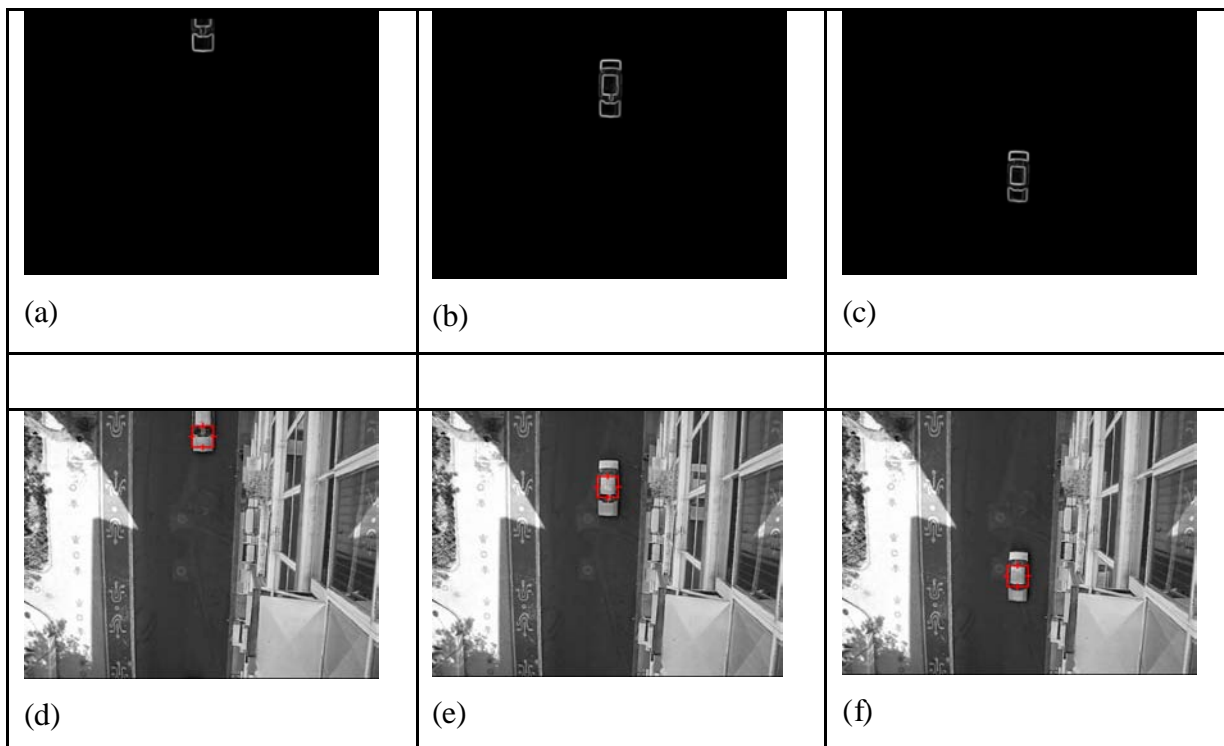
**Figura 71 –Resultados processamento amostra #20. a) gradiente quadros #19. b) gradiente quadro #33. c) gradiente quadro #47. d) tracking quadros #19. e) tracking quadro #33. f) tracking quadro #47.**



**Figura 72 –Resultados processamento amostra #21. a) gradiente quadros #13. b) gradiente quadro #23. c) gradiente quadro #41. d) tracking quadros #13. e) tracking quadro #23. f) tracking quadro #41.**



**Figura 73 –Resultados processamento amostra #22. a) gradiente quadros #8. b) gradiente quadro #19. c) gradiente quadro #31. d) tracking quadros #8. e) tracking quadro #19. f) tracking quadro #31.**



**Figura 74 –Resultados processamento amostra #23. a) gradiente quadros #7.  
b) gradiente quadro #13. c) gradiente quadro #22. d) tracking quadros #7.  
e) tracking quadro #13. f) tracking quadro #22.**

## APÊNDICE H – Tempos de execução das rotinas

A tabela 39 mostra o consumo de tempo para processamento das amostras considerando diferentes frequências de clock.

**Tabela 29 – Tempo de consumo das rotinas**

Amostra #	Tempo (s)	
	CLK=1 MHZ	CLK=2,4 MHZ
19	162	70
20	97	41
21	89	39
22	73	29
23	70	28

## ANEXO A - Frota de veículos por tipo no estado do RS

### Frota de veículos por tipo

Mês: 10/2004 Município: TODOS DO RIO GRANDE DO SUL



**Tabela 30 – Quantidade de Veículos por tipo no estado do RS**

<b>Tipo</b>	<b>Quantidade</b>
Automóvel	2.099.144
Caminhonete	74.492
Caminhão	144.081
Caminhão Trator	31.382
Camioneta	220.368
Ciclomotor	5.121
Microônibus	9.524
Motocicleta	426.158
Motoneta	55.996
Quadriciclo	47
Reboque	67.920
Sem Tipo	1.426
Semi-Reboque	44.025
Side Car	92
Trator de esteiras	83
Trator de rodas	2.309
Trator misto	196
Triciclo	221
Utilitário	1.353
Ônibus	25.120
<b>Total</b>	<b>3.209.058</b>

fonte: www.detran.org.br

## ANEXO B - Frota de veículos por tipo em Porto Alegre

### Frota de veículos por tipo

Mês: 10/2004 Município: PORTO ALEGRE



**Tabela 31 – Quantidade de Veículos por tipo em Porto Alegre**

<b>Tipo</b>	<b>Quantidade</b>
Ônibus	3.633
Reboque	11.323
Caminhão	11.894
Motoneta	3.879
Sem Tipo	349
Side Car	4
Triciclo	39
Automóvel	402.701
Camioneta	33.713
Ciclomotor	625
Utilitário	295
Caminhonete	12.388
Microônibus	1.661
Motocicleta	37.705
Quadriciclo	19
Semi- Reboque	2.642
Trator misto	28
Caminhão Trator	1.464
Trator de rodas	560
Trator de esteiras	10
<b>Total</b>	<b>524.932</b>

fonte: [www.detran.org.br](http://www.detran.org.br)

## ANEXO C - Descrição técnica da câmera Canon



### Specifications

---

<b>[Camera Effective Pixels]</b>	Approx. 3.2 million
<b>[Image Sensor]</b>	1/2.7-inch CCD (Total number of pixels : Approx. 3.3 million)
<b>[Lens]</b>	5.0 mm (35 mm film equivalent : 33 mm), f/3.6
<b>[Digital Zoom]</b>	Up to approx. 5.1x
<b>[Optical Viewfinder]</b>	Inverted Galilean viewfinder
<b>[LCD Monitor]</b>	1.5-inch. low-temperature polycrystalline silicon TFT color LCD, approx. 78,000 pixels
<b>[AF System]</b>	TTL autofocus Focusing frame : 5-point AiAF / 1-point AF Focus lock available
<b>[Shooting Distance (from the front of the lens)]</b>	Normal AF : 20 cm (7.9 in.) - infinity Macro AF : 5 - 20 cm (2.0 in. - 7.9 ft.)
<b>[Shutter]</b>	Mechanical shutter + electronic shutter
<b>[Shutter Speeds]</b>	15 - 1/2000 sec. * 1 - 1/6 sec. shutter speeds are available in manual or stitch assist mode if the flash is off and in slow synchro mode. * Slow shutter speeds of 1/6 sec. or slower operate with noise reduction.
<b>[Light Metering System]</b>	Evaluative, Center-weighted average or Spot
<b>[Exposure Control System]</b>	Program AE
<b>[Exposure Compensation]</b>	+/- 2 stops in 1/3-stop increments
<b>[Sensitivity]</b>	Auto, ISO 50 / 100 / 200 / 400 equivalent
<b>[White Balance]</b>	TTL auto, pre-set (available settings : Daylight, Cloudy, Tungsten, Fluorescent, Fluorescent H) or Custom
<b>[Flash]</b>	Auto, Red-eye Reduction, On, Off, Slow Synchro
<b>[Flash Range]</b>	20 cm - 2.0 m (7.9 in. - 6.7 ft.) (When sensitivity is set to ISO Auto)
<b>[Shooting Modes]</b>	Auto / Manual / Stitch assist / Movie
<b>[Continuous Shooting]</b>	Approx. 2.2 shots/sec. (Large/Fine mode, LCD monitor off)
<b>[Self-timer]</b>	Activates shutter after an approx. 10-sec. / approx. 2-sec. Delay
<b>[PC-controlled Shooting]</b>	Available (USB connection only. Exclusive software program is included in the camera kit.)
<b>[Recording Media]</b>	CompactFlash (CF) card (Type I)
<b>[File Format]</b>	Design rule for Camera File System and DPOF compliant
<b>[Image Recording Format]</b>	Still images : JPEG (Exif 2.2) *1 Movies : AVI (Image data : Motion JPEG; Audio data : WAVE [monaural])
<b>[Compression]</b>	Superfine, fine, Normal

<b>[Number of Recording Pixels]</b>	<p>&lt;Still images&gt;          Large : 2048 x 1536 pixels          Medium 1 : 1600 x 1200 pixels          Medium 2 : 1024 x 768 pixels          Small : 640 x 480 pixels</p> <p>&lt;Movies&gt;          640 x 480 pixels (Approx. 30 sec.) *2          320 x 240 pixels (Approx. 3 min.)          160 x 120 pixels (Approx. 3 min.)          15 frames/sec.</p> <p>The data in parentheses refer to the maximum movie length per shooting.</p>
<b>[Playback Modes]</b>	Single (histogram displayable), Index (9 thumbnail images), Magnified (approx. 10x (max.) in LCD monitor), Auto play or Sound memos (up to 60 sec.)
<b>[Direct Print]</b>	Canon Direct Print / Bubble Jet Direct / PictBridge compatible
<b>[Display Languages]</b>	12 languages available for menus and messages (English, German, French, Dutch, Danish, Finnish, Italian, Norwegian, Swedish, Spanish, Chinese and Japanese)
<b>[My Camera Settings]</b>	<p>The start-up image, start-up sound, shutter sound, operation sound and self-timer sound can be customized using the following methods :</p> <ol style="list-style-type: none"> <li>1. Using the images and sounds recorded on a camera.</li> <li>2. Using downloaded data from your computer using the supplied software.</li> </ol>
<b>[Interface]</b>	USB (mini-B, PTP [Picture Transfer Protocol])
<b>[Power Source]</b>	<ol style="list-style-type: none"> <li>1. Two AA-size alkaline batteries (included in the camera kit)</li> <li>2. Two AA-size rechargeable NiMH batteries (sold separately)</li> <li>3. A C Adapter Kit ACK800 (sold separately)</li> </ol>
<b>[Operating temperatures]</b>	0 - 40 °C (32 - 104 °F)
<b>[Operating Humidity]</b>	10 - 90 %
<b>[Dimensions]</b>	110.0 x 58.0 x 36.6 mm (4.3 x 2.3 x 1.4 in.) (excluding protrusions)
<b>[Weight]</b>	Approx. 175 g (6.2 oz.) (excluding batteries and CF Card)

\*1 This digital camera supports Exif 2.2 (also called "Exif Print"). Exif Print is standard for enhancing the communication between digital cameras and printers. By connecting to an Exif Print-compliant printer, the camera's image data at the time of shooting is used and optimized, yielding extremely high quality prints.

\*2 Recording time obtained when using the FC-32MH CF card (included with the camera) or the separately sold FC-256MH or FC-512 MSH CF card (not sold in some regions).