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

HENRIQUE WEBER

Técnica para Interação com Mãos em Superfícies Planares Utilizando uma Câmera RGB-D

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Claudio Rosito Jung

Porto Alegre
2016

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Weber, Henrique

Técnica para Interação com Mãos em Superfícies Planares Utilizando uma Câmera RGB-D / Henrique Weber. – Porto Alegre: PPGC da UFRGS, 2016.

63 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2016. Orientador: Claudio Rosito Jung.

1. Visão computacional. 2. Interação humano-computador. 3. Reconhecimento de gestos. 4. Imagens RGB-D. I. Jung, Claudio Rosito. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos Alexandre Netto
Vice-Reitor: Prof. Rui Vicente Oppermann
Pró-Reitor de Pós-Graduação: Prof. Vladimir Pinheiro do Nascimento
Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb
Coordenador do PPGC: Prof. Luigi Carro
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“I want the world, chico, and everything in it.”

— Tony Montana

AGRADECIMENTOS

À minha família, especialmente meu pai Martinho, minha mãe Veronica e minha irmã Marthyna, pelo apoio incondicional nesta e em todas as etapas da minha vida. Jamais teria chegado tão longe se não fosse por vocês. Muito obrigado!

À minha noiva Verônica pelo carinho e companheirismo nestes anos todos.

Aos meus amigos pelos bons momentos e aos colegas de laboratório pelas discussões construtivas.

Ao professor orientador Claudio Jung pelas sugestões, revisões e incentivo ao longo destes três anos.

À *Hewlett Packard* R&D Brasil pelo financiamento parcial desta pesquisa.

À Universidade Federal do Rio Grande do Sul e ao Instituto de Informática, pela excelência de ensino, sua infraestrutura e professores.

Ao PPGC pela oportunidade e ao CNPQ pelo financiamento da bolsa de pesquisa.

RESUMO

Sistemas de Interação Humano-Computador baseados em toque são uma tecnologia disseminada em *tablets*, *smartphones* e *notebooks*. Trata-se de um grande avanço que aumenta a facilidade de comunicação e, ao mesmo tempo, diminui a necessidade de interfaces como *mouse* e teclado. Entretanto, a superfície de interação utilizada por esses sistemas normalmente é equipada com sensores para a captação dos movimentos realizados pelo usuário, o que impossibilita transformar uma superfície planar qualquer (uma mesa, por exemplo) em uma superfície de interação. Por outro lado, a popularização de sensores de profundidade a partir do lançamento do *Microsoft Kinect* propiciou o desenvolvimento de sistemas que adotam objetos do dia a dia como superfícies de interação. Nesta dissertação é proposta uma interface natural para interação com superfícies planares utilizando uma câmera RGB-D em posição descendente. Inicialmente, o plano de interação é localizado na nuvem de pontos 3D através de uma variação do algoritmo RANSAC com coerência temporal. Objetos acima do plano são segmentados a partir da transformada *watershed* baseada em uma função de energia que combina cor, profundidade e informação de confiança. A cor de pele é utilizada para isolar as mãos, e os dedos que interagem com o plano são identificados por um novo processo de esqueletonização 2D. Finalmente, as pontas dos dedos são rastreadas com o uso do algoritmo Húngaro, e o filtro de Kalman é usado para produzir trajetórias mais suaves. Para demonstrar a utilidade da técnica, foi desenvolvido um protótipo que permite ao usuário desenhar em uma superfície de forma natural e intuitiva.

Palavras-chave: Visão computacional. Interação humano-computador. Reconhecimento de gestos. Imagens RGB-D.

A Technique for Hand Interaction with Planar Surfaces Using an RGB-D Camera

ABSTRACT

Touch-based Human-Computer Interfaces (HCIs) are a widespread technology present in tablets, smartphones, and notebooks. This is a breakthrough which increases the ease of communication and at the same time reduces the need for interfaces such as mouse and keyboard. However, the interaction surface used by these systems is usually equipped with sensors to capture the movements made by the user, making it impossible to substitute this surface by any other such as a table, for example. On the other hand, the progress of commercial 3D depth sensing technologies in the past five years, having as a keystone Microsoft's Kinect sensor, has increased the interest in 3D hand gesture recognition using depth data. In this dissertation, we present a natural Human-Computer Interface (HCI) for interaction with planar surfaces using a top-down RGB-D camera. Initially, the interaction plane is located in the 3D point cloud by using a variation of RANSAC with temporal coherence. Off-plane objects are segmented using the watershed transform based on an energy function that combines color, depth and confidence information. Skin color information is used to isolate the hand(s), and a novel 2D skeletonization process identifies the interaction fingers. Finally, the fingertips are tracked using the Hungarian algorithm, and a Kalman filter is applied to produce smoother trajectories. To demonstrate the usefulness of the technique, we also developed a prototype in which the user can draw on the surface using lines and sprays in a natural way.

Keywords: Computer Vision, Human-Computer Interaction, Gesture Recognition, RGB-D Cameras.

LISTA DE ABREVIATURAS E SIGLAS

3D	Três Dimensões
2D	Duas Dimensões
VC	Visão Computacional
ToF	Time-of-Flight
RGB-D	Red, Green, Blue and Depth
IR	Infrared
IHC	Interação Humano-Computador
LED	Light-emiting Diode
RDF	Random Decision Forests
OBB	Oriented Bounding Box
SDK	Software Development Kit

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

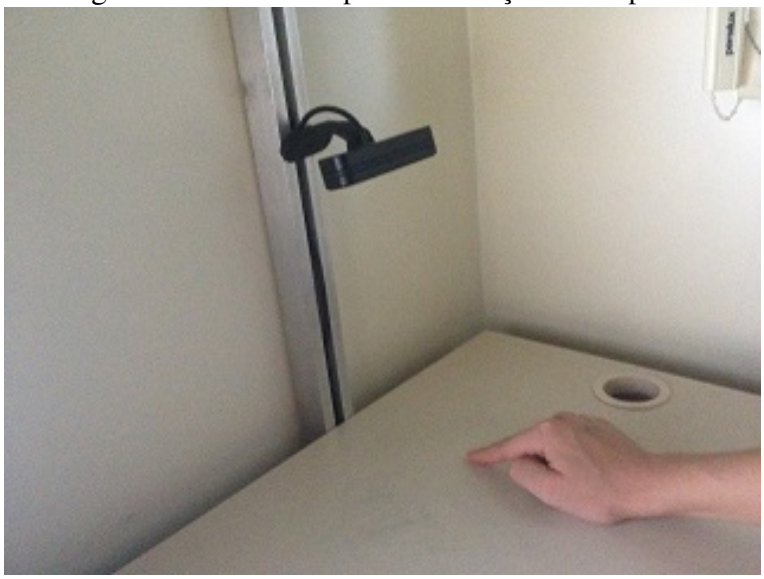
Apesar de periféricos como *mouse* e teclado terem sido (e ainda serem) as principais Interfaces Humano-Computador (IHC) para computadores de mesa, interfaces baseadas em toque conquistaram recentemente um vasto segmento de dispositivos eletrônicos. São *laptops*, *tablets* e *smartphones* que, não raro, apresentam o toque na tela ou no *touchpad* como principal meio de interação. Esta nova realidade proporcionou uma série de vantagens que vão desde redução do tamanho do dispositivo até a transmissão de comandos de forma mais intuitiva. Além disso, a popularização de IHCs baseadas em toque tem também encorajado o desenvolvimento de sistemas interativos que transformam superfícies do dia a dia em planos de interação.

Mais recentemente, houve uma série de propostas que fazem uso de câmeras que rastreiam os dedos para realizar a detecção de toques em superfícies quaisquer. Algumas delas utilizam projetores orientados juntamente com uma câmera a fim de localizar a ponta do dedo que está interagindo com o plano a partir do uso de casamento de padrões (KJELDSEN et al., 2002) (LETESSIER; BÉRARD, 2004) ou usando códigos de luz estruturada embarcada (DAI; CHUNG, 2012). Outros empregam duas câmeras para detectar a mão em profundidades específicas (WILSON, 2004) (WREN; IVANOV, 2001). Porém, tais abordagens costumam possuir um custo elevado ou ambientes complexos para sua utilização.

Por outro lado, o progresso de tecnologias de sensores de profundidade 3D nos últimos cinco anos, tendo como marco o sensor *Kinect* da *Microsoft* (ZHANG, 2012), tem aumentado o interesse em reconhecimento de gestos 3D da mão utilizando-se dados sobre a profundidade da cena. Em geral, essas abordagens são utilizadas em cinco domínios de aplicação: reconhecimento de linguagens de sinais, manipulação virtual, assistência diária, verificação da palma da mão, jogos eletrônicos e interação entre humanos e robôs (CHENG; YANG; LIU, 2015).

Na maioria dos casos, o sistema é construído partindo do pressuposto de que o usuário está posicionado de frente para a câmera e com as mãos para a frente, de forma a permitir a execução dos gestos. Neste caso, um simples limiar de profundidade geralmente é suficiente para isolar as mãos dos demais componentes da cena (SUAREZ; MURPHY, 2012). Porém, quando se trata de interfaces baseadas em toques monitoradas por uma câmera RGB-D, a mão/dedo pode estar arbitrariamente próxima do plano ou até mesmo encostando nele, o que torna a segmentação da mão e a detecção da interação um problema mais complexo.

Figura 1.1 – Cenário típico de interação com o plano.



Fonte: Compilado pelo autor.

1.1 Motivação

O recente avanço na tecnologia de sensores de profundidade ou RGB-D de baixo custo, como *Microsoft Kinect*, *Leap Motion* e *Softkinetic DS325*, gerou um aumento no interesse em sistemas de reconhecimento de gestos com as mãos no espaço 3D utilizando dados de profundidade. A quantidade de soluções para esta problemática é crescente, sendo motivada por um mercado de profissionais e usuários comuns de dispositivos eletrônicos que buscam uma forma de interação mais expressiva com dispositivos eletrônicos, de modo a capturar comandos mais complexos e de uma forma menos intrusiva do que é possível atualmente com periféricos como *touchpad, mouse* e teclado.

Os sistemas que fazem uso destes dispositivos normalmente exigem que o usuário fique de frente para a câmera e que suas mãos sejam o objeto mais próximo a ela (LIANG; YUAN; THALMANN, 2014) (WANG; LIU; CHAN, 2015). Entretanto, o uso intenso de interfaces gestuais como essas, onde não há a possibilidade de o usuário apoiar o braço para descanso, pode resultar no efeito conhecido como “braço de gorila”, o qual provoca fadiga e até mesmo movimentos involuntários do braço, desestimulando seu uso por períodos prolongados (BORING; JURMU; BUTZ, 2009). Adicionalmente, a exigência da mão isolada dos demais elementos da cena faz com que essas técnicas falhem caso a mão interaja com objetos/superfícies (SUPANCIC III et al., 2015).

Em contrapartida, o uso de uma superfície planar qualquer como interface de interação não apresenta requisitos quanto à postura do usuário. Porém, verificou-se que os trabalhos com

esta proposta apresentam uma série de precondições que limitam sua utilização a cenários específicos. Algumas destas técnicas operam apenas em cenários estáticos (KJELDSEN et al., 2002) (LETESSIER; BÉRARD, 2004). Outras pesquisas recentes requerem que o plano de interação possua propriedades especiais que o distingam claramente das mãos, que vão desde a cor até características retroreflexivas (MALIK; LASZLO, 2004) (KIM et al., 2014). Há também técnicas que dispensam determinadas especificações para o plano, porém não são capazes de diferenciar as mãos de outros objetos presentes na cena (DAI; CHUNG, 2014) ou apenas detectam dedos a uma determinada distância (WILSON, 2004).

1.2 Objetivo

O objetivo deste trabalho consiste em propor uma nova técnica que permita a interação de um usuário com uma superfície planar através de toques usando uma câmera RGB-D. Essa câmera deve estar posicionada acima do plano e direcionada a ele, não devendo haver qualquer predefinição a respeito da distância mínima e máxima entre ambos exceto aquela definida pelo fabricante da câmera. Acima do plano, pode haver zero ou mais mãos, bem como objetos variados, com a condição de que a superfície continue visível para a câmera. O usuário deve ser dispensado do uso de quaisquer equipamentos especiais, como luvas ou pulseiras, para o correto funcionamento da técnica. Assim, espera-se que este trabalho proporcione as informações necessárias para a criação de interfaces ainda mais intuitivas e menos intrusivas do que aquelas oferecidas atualmente por periféricos como o *touchpad, mouse* e teclado.

1.3 Objetivos específicos

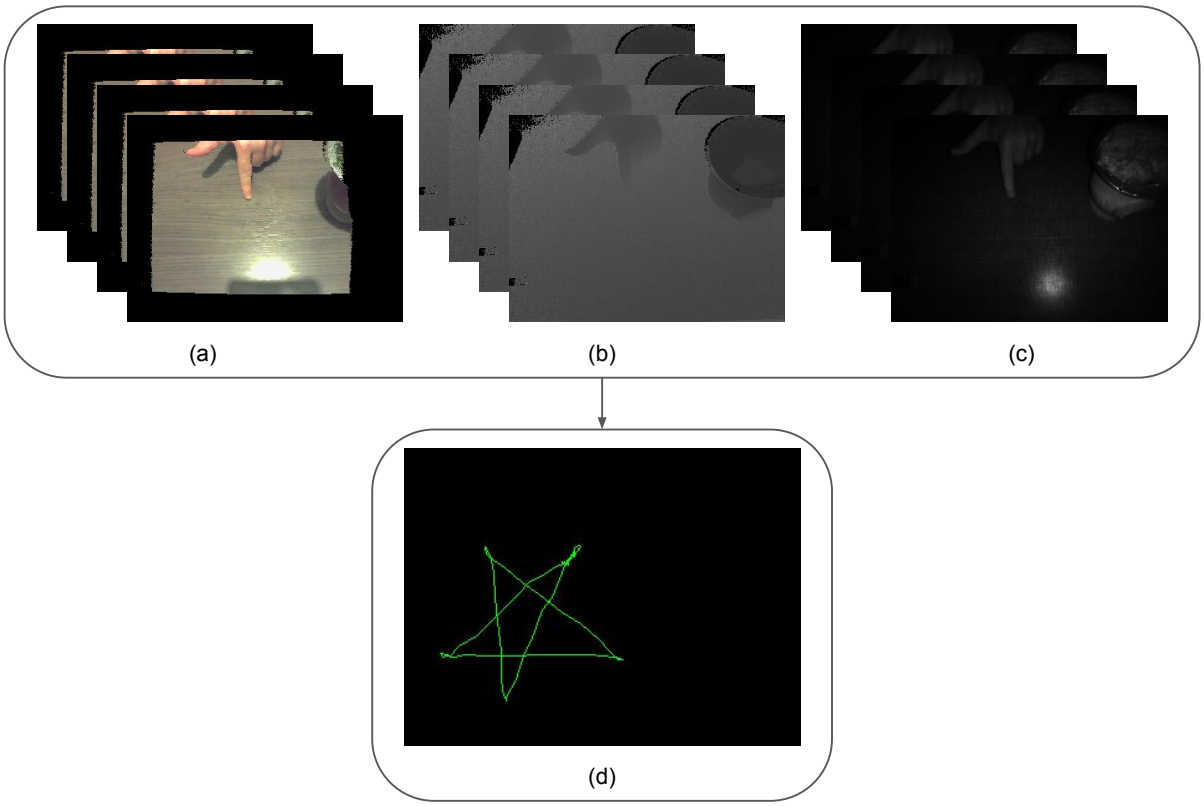
Para atingir o objetivo geral, os seguintes objetivos parciais foram traçados:

- Desenvolver uma técnica para segmentar a mão do plano de interação.
- Desenvolver uma técnica para identificar o(s) dedo(s) a partir da mão segmentada.
- Desenvolver um protótipo que detecte interações de toque com o(s) dedo(s) e o plano.

1.4 Visão geral

Este trabalho visa construir um sistema que seja capaz de capturar toques e localização da(s) mão(s) de um usuário sobre um plano de interação e transformar estes dados em comandos para o dispositivo em uso. O algoritmo proposto recebe como entrada a imagem no espaço de cores RGB, as coordenadas 3D de cada ponto e a respectiva confiança de cada um deles de acordo com a câmera, retornando a coordenada dos dedos que participam da interação pretendida pelo usuário. A técnica consiste em cinco etapas distintas. Na primeira, é realizada a detecção da equação do plano que representa a superfície de interação. Na segunda fase, é feita a segmentação dos objetos sobre o plano. Em seguida, as mãos presentes na cena são identificadas dentre esses objetos. Na quarta fase, é feita a localização 3D da ponta dos dedos visíveis na cena. Por fim, a quinta etapa é responsável pela detecção dos dedos que tocam o plano. O *pipeline* simplificado do trabalho se encontra na Figura 1.2.

Figura 1.2 – *Pipeline* da abordagem. (a) Dada uma série de imagens RGB, (b) a nuvem de pontos que corresponde à cena e (c) a confiança desses pontos, (d) retorna a interação final pretendida pelo usuário.



Fonte: Compilado pelo autor.

O trabalho está organizado em cinco capítulos. O capítulo a seguir apresenta uma seleção de trabalhos relacionados à interação de mãos com superfícies planares e reconhecimento de

gestos. No Capítulo 3, a técnica proposta é detalhada em cinco etapas. Resultados experimentais demonstrando a utilidade da técnica, bem como uma avaliação a respeito da precisão da mesma, são expostos no Capítulo 4. Conclusões sobre o sistema e possíveis melhoramentos para trabalhos futuros são apresentados no Capítulo 5.

2 TRABALHOS RELACIONADOS

A área de Interfaces de Interação Humano-Computador baseadas em mãos possui uma ativa comunidade de pesquisa e há uma série de revisões a respeito na literatura (RAUTARAY; AGRAWAL, 2015) (PISHARADY; SAERBECK, 2015). Mais especificamente, o desenvolvimento de técnicas que utilizam câmeras de topo para detecção de toques em superfícies planas tem sido documentado em diversos trabalhos nos últimos 25 anos (WELLNER, 1991) (KRUEGER, 1991), mas o uso de informação de profundidade (especialmente vinda de câmeras RGB-D) se deu mais recentemente.

Há diversas propostas para o reconhecimento de gestos/poses de mãos utilizando-se informação de profundidade, os quais são descritos em pesquisas recentes (CHENG; YANG; LIU, 2015) (SUAREZ; MURPHY, 2012). Mais precisamente, este Capítulo irá abordar trabalhos que focam na etapa de segmentação de mãos utilizando câmeras RGB-D e também em propostas que desenvolvam sistemas para a interação com superfícies planares a partir do uso de informação de profundidade.

2.1 Segmentação da Mão

Uma das maiores vantagens do uso de câmeras de profundidade sobre câmeras de cor se encontra na etapa de segmentação da mão. Em aplicações onde se espera que o usuário esteja de frente para a câmera e com as mãos para a frente, é muito comum utilizar um limiar de profundidade para que seja possível isolar as mesmas do restante da cena. Apesar de ser utilizada em diversos trabalhos (WANG; LIU; CHAN, 2015) (LIANG; YUAN; THALMANN, 2014) (BERGH; GOOL, 2011), ela não produz bons resultados no cenário proposto nessa dissertação por impossibilitar que a mão entre em contato com quaisquer outros elementos da cena (incluindo o plano de interação).

Em contrapartida, há uma série de técnicas que lidam com a mão interagindo com superfícies planares. Uma delas chama-se *Visual Touchpad* (MALIK; LASZLO, 2004), a qual é constituída por duas câmeras posicionadas acima do plano de interação que, por sua vez, possibilitam a captura da posição 3D das pontas dos dedos do usuário na superfície e acima dela. Assim, o sistema é capaz de detectar tanto toques no plano como gestos com a mão pairando no ar.

Para que a técnica funcione corretamente, é necessário posicionar um papel com um retângulo negro no centro, o qual é cercado por uma borda branca, no local onde se deseja realizar

as interações. É computada então a homografia que define o mapeamento plano-projetivo entre o retângulo negro e a tela do dispositivo de interação. Para realizar a segmentação das mãos do usuário é feita uma operação de subtração de fundo, o qual corresponde ao retângulo negro. Posteriormente, é utilizada a técnica de *flood-fill* nos objetos do *foreground*, e assume-se que os dois maiores componentes conexos acima de um determinado limiar correspondem às mãos.

Já a ponta dos dedos é detectada através da busca por picos na borda dos *blobs* (conjuntos conexos de *pixels*) candidatos à mão. Esses pontos na borda, por sua vez, são caracterizados por vetores, e caso o ângulo entre o vetor de um ponto de borda k até $k + n$ e $k - n$ seja menor do que um determinado limiar (e desde que os pontos não façam parte de um vale, como o existente entre dedos), é determinado que esses pontos fazem parte da ponta de um dedo. Já o ângulo desse dedo é dado pela linha que vai do ponto médio localizado entre os pontos $k + n$ e $k - n$ até o ponto k. O cenário de uso, bem como a visão de uma das câmeras e resultado da identificação dos dedos pode ser visto na Figura 2.1. A desvantagem desta técnica encontra-se na necessidade de uso do retângulo negro e na calibração das câmeras, o que reduz sua facilidade de uso e limita a superfície de interação àquela região.

Figura 2.1 – Detecção de dedos do sistema *Visual Touchpad*. (a) Exemplo de configuração. (b) Visão retificada de uma das câmeras. (c) Dedos identificados.

(a) (b) (c)

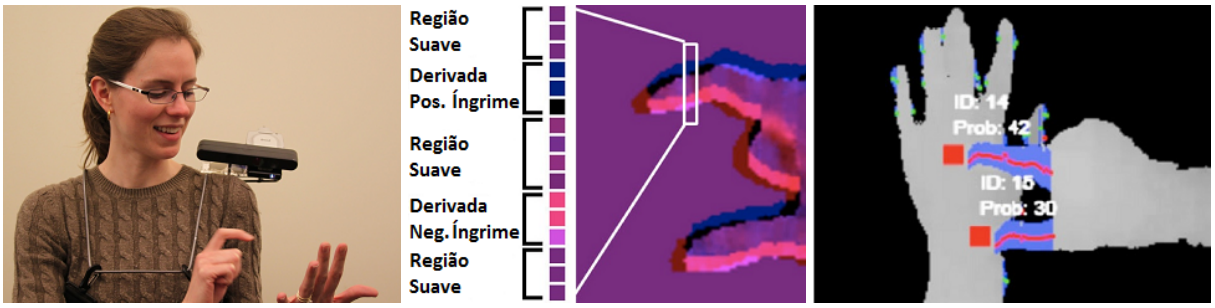
Fonte: (MALIK; LASZLO, 2004).

Em um outro trabalho (HARRISON; BENKO; WILSON, 2011), foi apresentado o *Omni-Touch*, o qual é um sistema vestível de projeção e cálculo de profundidade. Acoplado ao ombro do usuário, ele permite que o mesmo utilize as mãos, braços ou pernas como superfícies interativas. O primeiro passo da técnica consiste em segmentar os dedos do usuário. Para isso, é computada a derivada da profundidade da cena tanto no eixo X como no Y com o uso de uma janela deslizante de tamanho fixo. Em seguida, são feitas iterações na imagem à procura de fatias verticais de objetos com aspecto cilíndrico. Mais especificamente, para que uma fatia de

pixels seja candidata, ela precisa apresentar uma derivada positiva íngreme, seguida por uma região de relativa suavidade (aproximadamente plana), e finalmente uma derivada negativa íngreme. Essa ordem deve ser respeitada para que não se aceitem regiões côncavas. Além disso, essas fatias devem possuir largura entre 5 e 25mm.

Uma vez que todas as fatias candidatas a dedo foram identificadas, é feito o agrupamento das mesmas de forma gulosa, gerando caminhos contíguos. Caminhos maiores ou menores do que o esperado para um dedo são descartados. Por fim, para definir a ponta de cada dedo, partiu-se do pressuposto de que os usuários são destros e, portanto, a extremidade mais à esquerda do caminho é considerada a ponta do dedo. A Figura 2.2 mostra etapas intermediárias e o resultado final desse procedimento.

Figura 2.2 – Ambiente de uso e segmentação da mão com o sistema *OmniTouch*. (a) Exemplo de uso do sistema. (b) Análise da fatia de um candidato a dedo. (c) Caminho contíguo de dois dedos detectados.



(a) (b) (c)

Fonte: (HARRISON; BENKO; WILSON, 2011).

Já uma outra proposta, chamada de *MirageTable* (BENKO; JOTA; WILSON, 2012), é capaz de proporcionar visão perspectiva estereoscópica 3D para um único usuário, além de permitir que o mesmo interaja com objetos virtuais projetados no plano. Para isso, é utilizada uma câmera RGB-D, um projetor e uma tela curva. O sistema é calibrado de tal forma que tanto a posição da câmera RGB-D quanto do projetor sejam conhecidos no sistema de coordenadas de mundo. Para segmentar a mão presente na cena, é feita inicialmente a captura da geometria da superfície de interação. Essa informação é utilizada como base para o fundo do cenário, facilitando a segmentação de novos objetos ou partes do corpo do usuário através da segmentação de fundo.

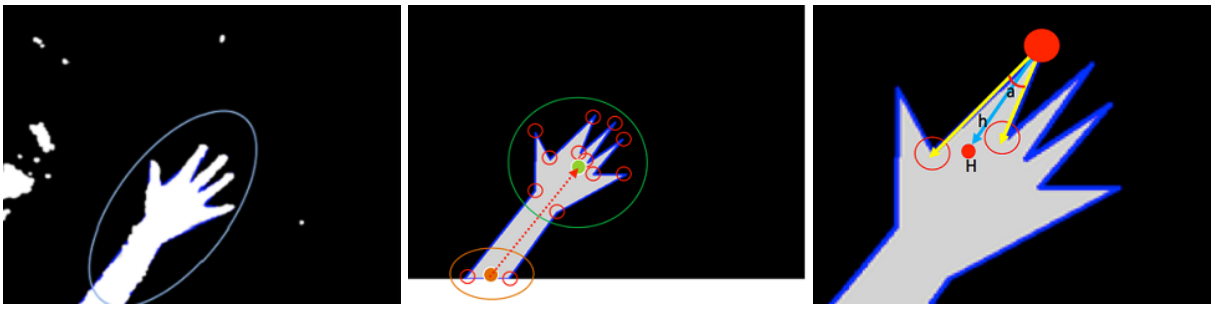
Também em 2012, foi apresentado o *framework dSensingNI* (*Depth Sensing Natural Interaction*) (KLOMPMAKER; NEBE; FAST, 2012). De acordo com o autor, o sistema suporta toques múltiplos e interação tangível com objetos arbitrários. A técnica utiliza imagens de profundidade e provê rastreamento dos dedos dos usuários, permitindo também pegar, agrupar e empilhar objetos presentes na cena.

Para reconhecer objetos e mãos acima do plano, a técnica utiliza remoção dinâmica de fundo baseada em duas imagens distintas (A e B). Assim que o sistema começa a operar, a primeira imagem capturada é armazenada tanto em A como em B. Enquanto A é utilizada para detectar objetos físicos, B serve para a detecção de braços, dedos e toque na superfície. Os *pixels* da imagem B são atualizados sempre que uma nova imagem é gerada, porém apenas onde não havia braços, mãos ou dedos no quadro anterior. Desta forma, é possível reconhecer os diferentes componentes envolvidos na interação através da diferença de profundidade entre a imagem atual e a imagem B. Já a imagem A é atualizada apenas quando não há braços, dedos ou palmas na cena atual.

Em seguida, as áreas cujo tamanho excedam um limiar mínimo e que não estejam completamente cercadas por *pixels* de fundo (ou seja, aparentam se projetar das bordas para o interior da imagem) são analisadas. A forma dessas áreas é então simplificada a simples polígonos 2D. Os vértices de cada polígono, por sua vez, são agrupados de acordo com a proximidade uns com os outros, o que possibilita a identificação da palma e do fim do braço. O centro de ambos agrupamentos, juntamente com a informação de profundidade de cada um, é utilizado para se definir o vetor que indica a direção e sentido do braço correspondente.

Já para identificar os diferentes dedos da mão, todos os vértices pertencentes à palma da mesma são avaliados. Três consecutivos vértices são analisados através do ângulo por eles criado. Se esse ângulo for menor do que um determinado valor, é calculada a sua bissetriz h. Caso o extremo H da semirreta h estiver dentro da região da palma do polígono, então os três pontos representam um dedo. Estas etapas estão ilustradas na Figura 2.3.

Figura 2.3 – *Framework* para interação com objetos arbitrários *dSensingNI*. (a) Imagem binária da profundidade. (b) Clusterização dos vértices. (c) Análise dos dedos pela linha da bissetriz.



(a) (b) (c)

Fonte: (KLOMPMAKER; NEBE; FAST, 2012).

Mais recentemente, outra proposta capaz de capturar a silhueta de mãos e outros objetos acima de superfícies foi proposta. Chamado de *RetroDepth* (KIM et al., 2014), o sistema é composto por duas câmeras de infravermelho, LEDs difusos infravermelhos e um material re-

troreflexivo. De acordo com os autores, a proposta possibilita cenários interativos que mesclam gestos com as mãos acima da superfície, toques e pressão no plano.

O ambiente de uso do sistema consiste em duas câmeras posicionadas acima do plano com os LEDs infravermelho circulando suas lentes, as quais são calibradas *offline*. Por sua vez, a superfície de interação compreende as regiões cobertas pelo material retroreflexivo, as quais são identificadas pela extração de contornos de quaisquer silhuetas brilhantes observadas pelas câmeras. Em seguida, é feita a extração de contornos no interior das regiões de interação (gerando, por exemplo, a silhueta da mão na Figura 2.4(c)). Um classificador RDF (*Random Decision Forests*) é treinado utilizando o contorno 1D da silhueta de mãos para reconhecer oito estados correspondentes à versão esquerda e direita de quatro poses distintas da mão. A tarefa do RDF consiste em classificar o formato da mão levando em consideração seu contorno, bem como localizar e identificar a ponta dos dedos tendo como base uma métrica de curvatura. A Figura 2.4 mostra partes desta etapa de segmentação da mão.

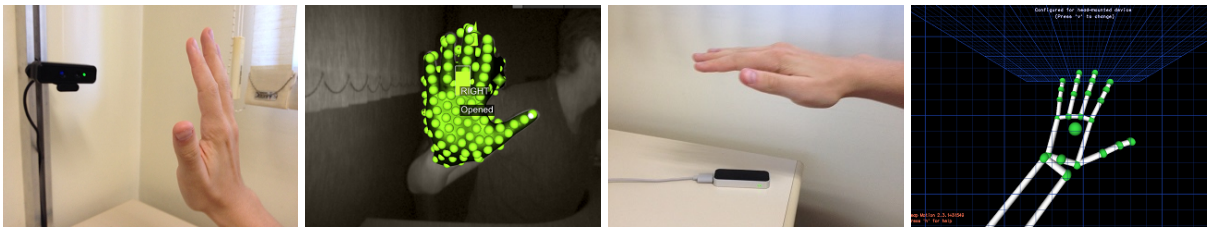
Figura 2.4 – Exemplos de uso e etapas intermediárias do sistema *RetroDepth*. (a) Exemplo de uso. (b) Superfícies retroreflexivas. (c) Imagem gerada pela câmera de infravermelho.

(a) (b) (c)

Fonte: (KIM et al., 2014).

Há também soluções comerciais que se propõem a segmentar mãos para interações com outros dispositivos. O *middleware iiSu* (Softkinetic, 2016b) é um deles, sendo capaz de gerar o rastreamento de até duas mãos em uma série de gestos predefinidos. Há também o sensor de profundidade *Leap Motion* (Leap Motion, Inc., 2016a), o qual suporta movimentos de mãos e dedos como comandos do usuário efetuados acima do equipamento. A Figura 2.5 mostra o ambiente típico de uso de cada um deles e o resultado da detecção. Ambas as técnicas, porém, partem do pressuposto de que a mão está isolada dos demais elementos da cena e, portanto, não são apropriadas para o uso no cenário proposto nesta dissertação.

Figura 2.5 – Produtos comerciais para interação com mãos de uma forma natural. (a) Ambiente de uso do *middleware iiSu* e (b) resultado da detecção de interação do mesmo. (c) Ambiente de uso do *Leap Motion* e (d) resultado da detecção de interação do mesmo.



(a) (b) (c) (d)

Fonte: Compilado pelo autor.

2.2 Interação com Superfícies Planares

Outra etapa importante em técnicas de interação com superfícies consiste em detectar os toques efetuados pelo usuário, levando em conta a localização do toque, quantos (e quais) dedos foram utilizados e qual a intenção do usuário. As variações entre elas vão desde as restrições para o ambiente de uso até a definição do que pode provocar um evento de toque (o dedo ou qualquer objeto, por exemplo).

Para completar esta etapa, o sistema *Visual Touchpad* anteriormente citado faz uso da informação de disparidade provida pelas duas câmeras. Neste caso, para um dedo encostado no plano, a posição (x, y) dada por cada uma das câmeras será a mesma, porém será diferente caso o dedo esteja acima do plano (como pode ser visto na Figura 2.6(c)), uma vez que a homografia provê apenas o mapeamento planar. Exemplos de imagens geradas pelas câmeras para obtenção da altura do dedo acima do plano podem ser vistos na Figura 2.6.

Figura 2.6 – Detecção de interação com superfícies no sistema *Visual Touchpad*. (a) Visão retificada da câmera 1. (b) Visão retificada da câmera 2. (c) Sobreposição das imagens indicando que o dedo não encosta no plano.

(a) (b) (c)

Fonte: (MALIK; LASZLO, 2004).

Já outra técnica, chamada de *TouchLight* (WILSON, 2004), é composta por duas câmeras posicionadas atrás de um plano semitransparente (o qual fica de frente para o usuário), um emissor de luz de infravermelho e um projetor. A imagem resultante mostra objetos que estão no plano, fazendo com que essa superfície possa se tornar um dispositivo sensível ao toque.

Para evitar interferência da luz vinda do ambiente (bem como do projetor que compõe o sistema), a técnica proposta pelos autores atua somente no domínio infravermelho, o qual é produzido pelo emissor IR. A detecção do toque de objetos em geral e de mãos com o plano é calculada relacionando disparidade binocular com a profundidade do objeto em coordenadas de mundo. A fim de reduzir o custo computacional normalmente envolvido com algoritmos *stereo*, é determinado apenas o que está localizado em um plano 3D em particular (a superfície de interação).

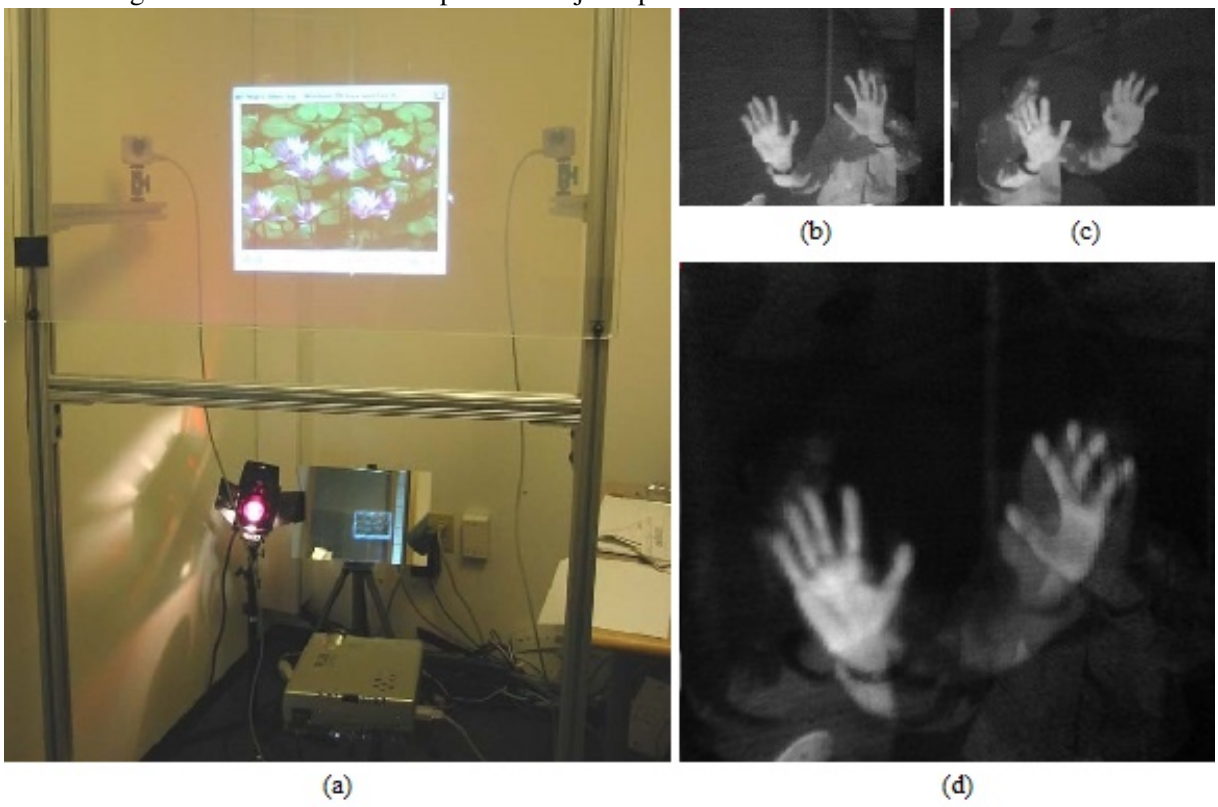
Para se demonstrar a utilidade da técnica, foi implementado um simulador de *mouse*. Neste caso, o maior objeto em contato com o plano determina a posição do *mouse*, enquanto uma região no canto esquerdo inferior da superfície de interação funciona como um botão do mesmo: quando o usuário posiciona sua mão na região, é gerada uma quantidade de *pixels* acima do limiar de toque, e um evento de clique é gerado. Por fim, é acoplado um microfone junto ao plano, de forma a ser possível ouvir toques na superfície. Essa informação, apesar de não indicar a localização do toque, pode complementar a etapa de identificação de objetos (especialmente dedos) interagindo com o sistema.

A Figura 2.7 demonstra o ambiente de uso e as etapas intermediárias do sistema *TouchLight*. De acordo com o autor, apesar dos testes iniciais terem sido realizados em uma tela vertical, também é possível utilizá-lo na posição horizontal. Entretanto, persiste a desvantagem relacionada ao elevado número de componentes para montar o sistema, tornando seu uso limitado a ambientes que acomodem todos os equipamentos.

O sistema de detecção de eventos de toques no plano do sistema *OmniTouch*, por sua vez, inicialmente computa o ponto médio do caminho de um dedo. A partir dele, é aplicado o algoritmo de *flood fill*, porém com a particularidade de que o mesmo pode percorrer todas as direções exceto à direita. Uma tolerância de 13mm é utilizada para determinar se um *pixel* vizinho deve ser ocupado. Caso o algoritmo de preenchimento pare no próprio dedo, considera-se que o mesmo não está tocando o plano. Porém, se mais do que 2000 *pixels* forem preenchidos, a técnica assume que o dedo encosta na superfície.

Para detectar a superfície de interação, é feito um cálculo de componentes conexos no mapa de profundidade. Superfícies menores do que o tamanho da mão são descartadas. Há também a possibilidade de o próprio usuário definir manualmente a localização e tamanho da superfície

Figura 2.7 – Sistema de interação com superfícies *TouchLight*. (a) Protótipo exibindo uma projeção. (b-c) Imagens de entrada do sistema com correção da perspectiva. (d) Imagem obtida multiplicando-se ambas imagens de entrada exibindo apenas os objetos próximos à tela.



Fonte: (WILSON, 2004).

com a qual deseja interagir. A Figura 2.8 mostra algumas das etapas do sistema, bem como o ambiente de utilização.

Como pode ser observado, o sistema *OmniTouch* apresenta uma grande flexibilidade em relação à superfície de interação. Porém, isso vem ao custo de determinadas restrições infligidas sobre o usuário, como a exigência de uso apenas da mão direita e a impossibilidade de haver objetos com aspecto cilíndrico (como uma caneta) presentes na cena, pois os mesmos podem ser confundidos com dedos.

Em 2010, Wilson (WILSON, 2010) apresentou um dos trabalhos pioneiros a explorar a factibilidade do uso de apenas uma câmera RGB-D para interação com planos. De acordo com o autor, tal sistema é incapaz de proporcionar uma precisão igual ou superior ao de uma tela sensível ao toque, e um dos motivos é a incapacidade de detectar o toque de um dedo no plano quando o mesmo for ocluído pela mão, por exemplo. Apesar disso, o autor encoraja seu uso em função das demais características do sistema, como a flexibilidade de usar uma superfície não instrumentada e a possibilidade de detecção de toque acima do plano.

Para estudar as vantagens e as limitações do uso de uma câmera RGB-D, o autor propôs um método simples de detecção de toque. A técnica assume, inicialmente, possuir um modelo

Figura 2.8 – Etapas de detecção da interação do sistema *OmniTouch*. (a) Resultado do *flood filling* para um dedo acima do plano. (b) Resultado do *flood filling* para um dedo tocando o plano. (c) Usuário interagindo com os botões projetados.

(a) (b) (c)

Fonte: (HARRISON; BENKO; WILSON, 2011).

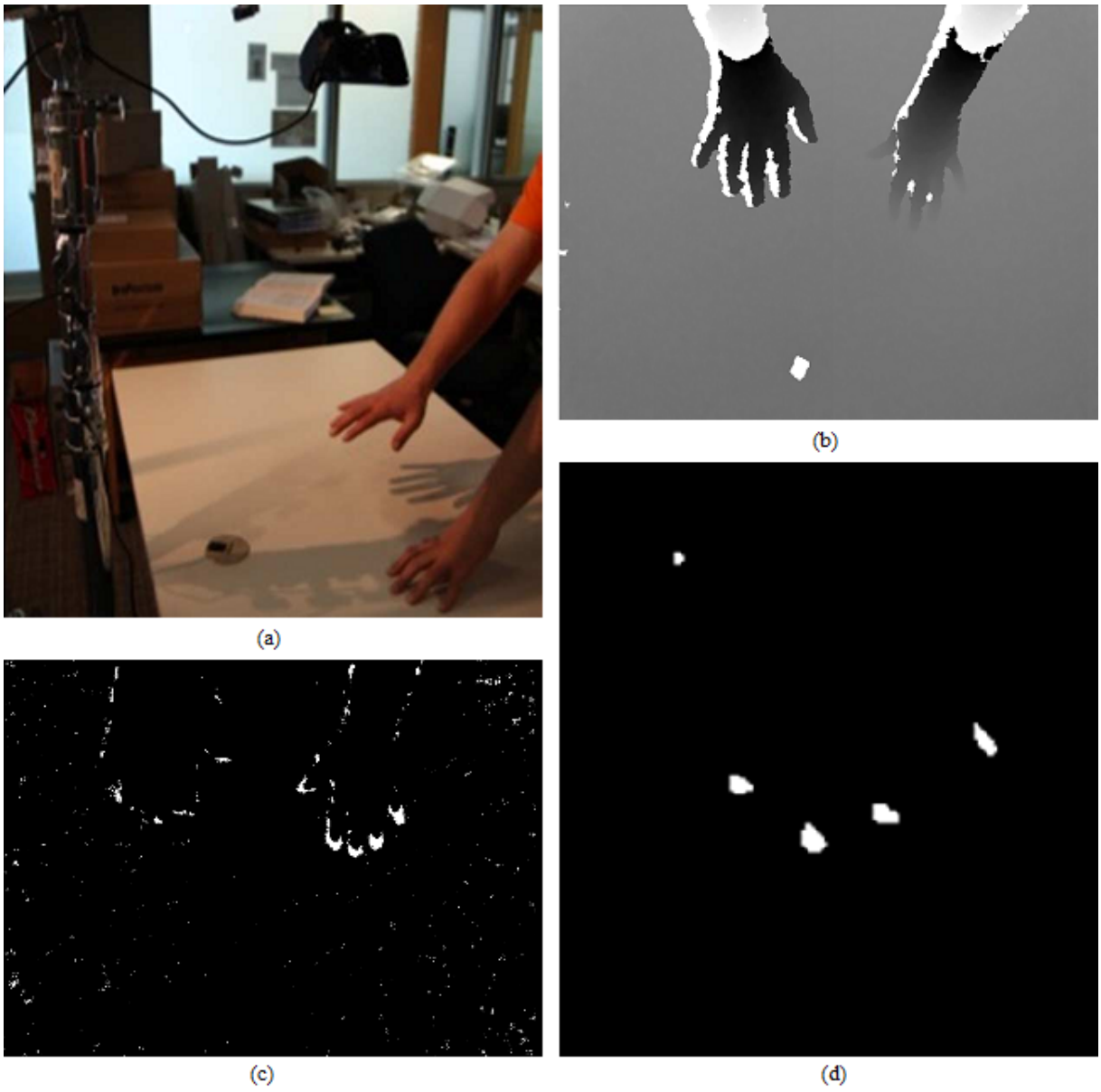
do plano para que seja possível detectar objetos acima dele. Para obter esse modelo, o autor propõe o uso de três limiares para cada *pixel* da imagem, chamados de $d_{superfície}$, d_{max} e d_{min}. Para o cálculo de $d_{superfície}$, é feita uma "fotografia" da profundidade da cena, onde o plano encontra-se vazio. O valor de d_{max}, por sua vez, é calculado percorrendo-se o histograma dos dados brutos da profundidade, a partir de $d_{superfície}$ em direção a valores maiores, até o primeiro valor do histograma maior do que um determinado limiar. Já d_{min} é baseado na espessura típica de um dedo, dada por τ, de forma que $d_{min} = d_{max} - \tau$.

Em seguida é formada uma imagem binária, onde o *pixel* da posição (x, y) é considerado parte de um objeto em contato com o plano se $d_{max} > d_{x,y} > d_{min}$. Ruídos na imagem binária são removidos com um filtro do tipo *boxcar*, enquanto pontos de contato distintos são identificados através da análise de componentes conexos. A Figura 2.9 demonstra o ambiente de uso, bem como as etapas intermediárias. É possível perceber que a técnica assume que haja apenas mãos acima do plano, e não são exploradas as consequências advindas do uso da câmera em ângulos variados com relação ao plano, por exemplo, o que poderia comprometer os resultados do sistema.

No sistema *MirageTable*, a simulação de interação entre as mãos do usuário e objetos virtuais é feita a partir da simulação de colisão com computação gráfica. Para isso, a imagem de profundidade é segmentada de acordo com os objetos presentes na cena. Em seguida, cada objeto é subdividido em fragmentos de 2cm, e a cada um deles é atribuído uma esfera de raio 1cm. Essa esfera é posicionada na exata localização da cena 3D onde o fragmento correspondente se encontrava, permitindo a simulação física de colisões entre objetos reais e virtuais inseridos na cena.

A Figura 2.10 apresenta o sistema e os passos intermediários do mesmo. Apesar da técnica

Figura 2.9 – Sistema de interação com plano utilizando uma câmera RGB-D. (a) Cenário experimental. (b) Visão da câmera de profundidade. (c) Imagem do toque. (d) Resultado final das regiões de toque depois da filtragem.

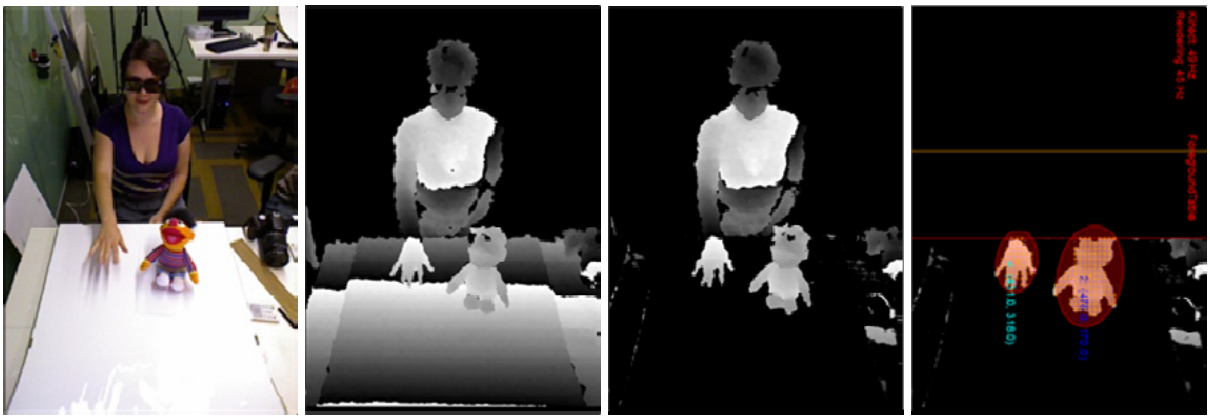


Fonte: (WILSON, 2010).

de remoção de fundo apresentar resultados com grande precisão, seu uso limita o sistema a cenários estáticos, impossibilitando-o de operar em objetos com movimento frequente (como uma câmera RGB-D localizada na parte superior de um *notebook*, por exemplo).

No *framework dSensingNI*, a detecção do plano de interação inicia pela detecção de todas as áreas que estão inteiramente cercadas por *pixels* de fundo. Em seguida, as características das superfícies tangíveis (como orientação, comprimento e largura) são extraídas a partir do cálculo de uma *oriented bounding box* (OBB). Já a detecção do toque é feita a partir da comparação

Figura 2.10 – Sistema de interação com plano *MirageTable*. (a) Imagem RGB. (b) Visão da câmera de profundidade. (c) Objetos em primeiro plano. (d) Esferas projetadas sobre os objetos rastreados.



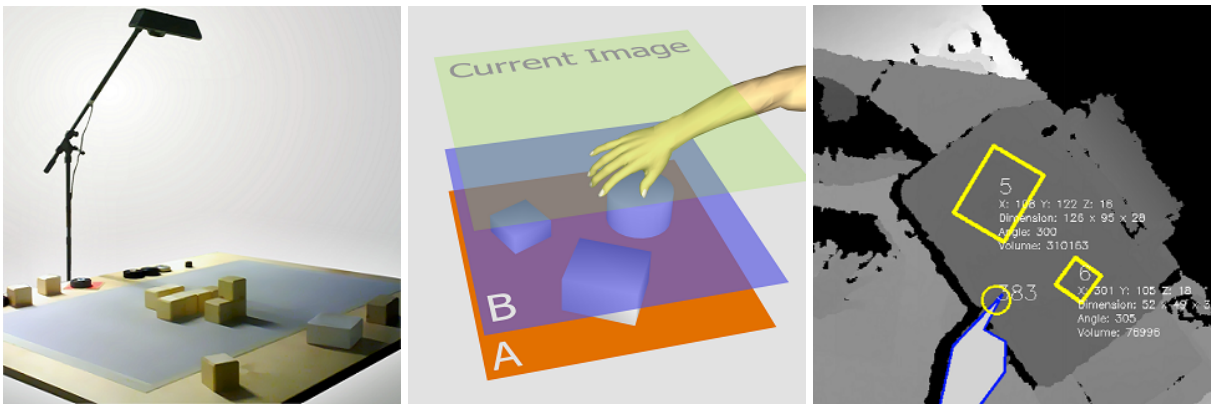
(a) (b) (c) (d)

Fonte: (BENKO; JOTA; WILSON, 2012).

dos valores de profundidade da ponta dos dedos e da cena de fundo para a mesma posição no espaço, e caso a diferença entre eles seja menor do que um determinado limiar, é considerado que houve toque.

A Figura 2.11 mostra a detecção dessas superfícies, bem como o cenário de uso e a detecção de dedos. Embora a técnica apresente uma série de possibilidades de interação com resultados interessantes, a necessidade, por parte do usuário, de remover os braços da cena para que o plano de interação seja novamente calculado pode comprometer a usabilidade do sistema.

Figura 2.11 – *Framework* para interação com objetos arbitrários *dSensingNI*. (a) Ambiente de uso. (b) Imagens de fundo. (c) Imagem de profundidade analisada pelo *dSensingNI*.



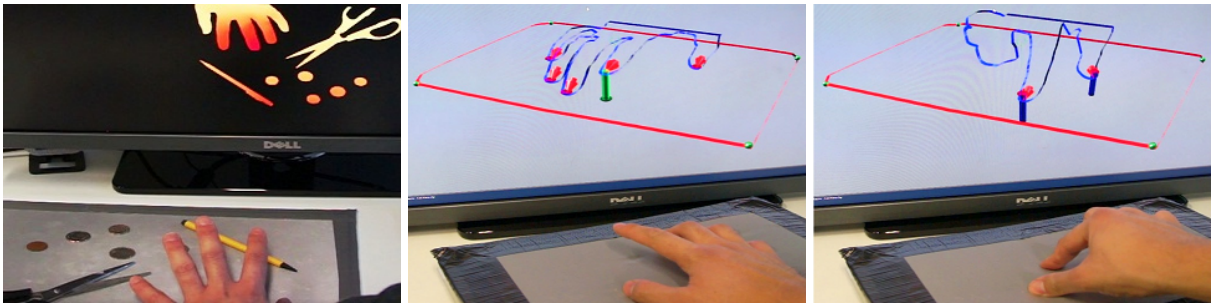
(a) (b) (c)

Fonte: (KLOMPMAKER; NEBE; FAST, 2012).

Já o método de detecção da interação com a superfície proposto no sistema *RetroDepth* inicia pela estimativa da localização do plano levando em conta os contornos da superfície retroreflexiva, tornando possível a identificação de qualquer intersecção com as extremidades (de

mãos ou canetas) previamente reconhecidos pelo sistema. A Figura 2.12 exemplifica cenários de uso, juntamente com resultados da detecção. Embora apresente resultados com uma precisão bastante elevada (mesmo quando o algoritmo foi adaptado para funcionar com câmeras de profundidade de baixo custo como *Kinect*), a necessidade de utilização de superfícies retroreflexivas reduz a facilidade de uso do sistema, além de impossibilitar a interação com objetos cuja superfície seja retroreflexiva, pois os mesmos serão considerados como sendo parte da superfície de interação.

Figura 2.12 – Detecção da interação com o plano no sistema *RetroDepth*. (a) Estimativa da profundidade. (b) Distinção entre plano (contorno vermelho) e objetos de interação (contornos azuis). (c) Detecção da interação no ar (linha verde), toque (vermelho) e pressão (azul).



(a) (b) (c)

Fonte: (KIM et al., 2014).

2.3 Discussão

Nesse capítulo foram revisados trabalhos de detecção de gestos sobre superfícies utilizando informação de profundidade julgados relevantes para revisão bibliográfica. Em geral, percebe-se a tendência pela implementação de técnicas de remoção de fundo, o que exige um cenário relativamente estático. Há também trabalhos que acabam instrumentando a superfície de interação, enquanto outros utilizam uma série de equipamentos auxiliares para facilitar a identificação de toques no plano. Por outro lado, existem propostas que fizeram uso de apenas uma câmera RGB-D, mas mesmo essas continuam apresentando ao menos uma das restrições citadas acima. Em contrapartida, este trabalho propõe uma solução que também faz uso de apenas uma câmera RGB-D porém permitindo uma montagem flexível da mesma (desde que posicionada acima do plano) e sem exigir qualquer instrumentação da superfície, além de admitir objetos não-planares na cena. As suas etapas de processamento são abordadas a seguir.

3 TÉCNICA PROPOSTA

Neste capítulo, é apresentada a principal contribuição desta dissertação: uma técnica de detecção da interação de mãos com superfícies planares utilizando-se uma câmera RGB-D. Dada uma imagem RGB, as coordenadas 3D de cada ponto e o mapa de confiança destas coordenadas (provido pela câmera), busca-se retornar a posição 3D dos dedos que estejam participando da interação pretendida pelo usuário. Assim, o usuário pode definir trajetórias ao longo do plano para criar desenhos, enviar comandos ao computador através de "cliques" no plano de forma semelhante a já feita em *mouses* e *touchpads*, bem como realizar gestos acima do plano de interação.

O capítulo está organizado em cinco seções. A primeira descreve os passos para a detecção da equação do plano que representa a superfície de interação presente na cena. A segunda trata da segmentação dos objetos que se encontram sobre esta superfície, enquanto a terceira seção expõe a forma com que as mãos são detectadas dentre estes objetos. A quarta seção detalha como as pontas dos dedos que participam da interação são detectadas, e a quinta trata do reconhecimento do gesto em si. A Figura 3.1 apresenta o *pipeline* geral da técnica.

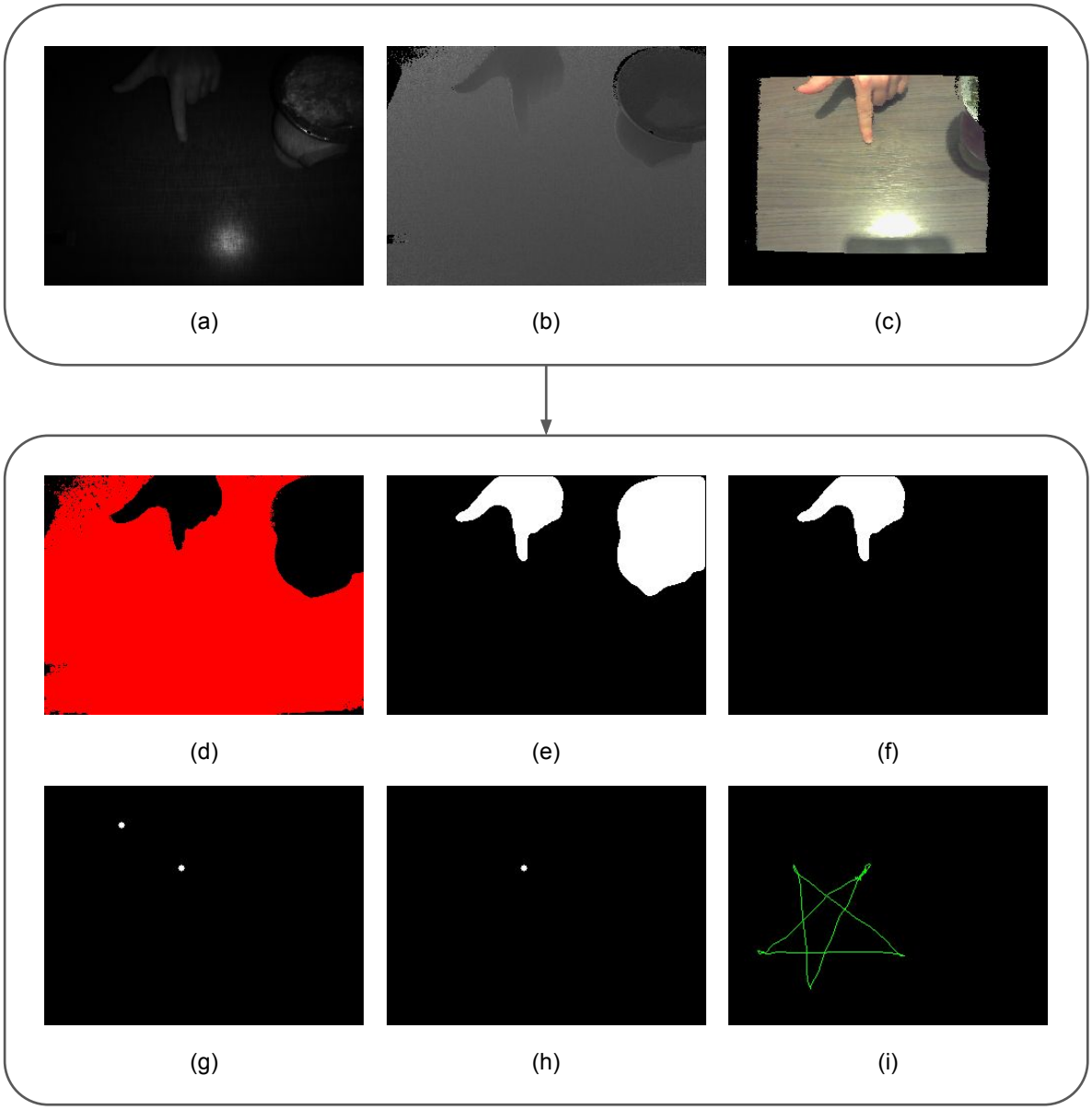
3.1 Detecção do Plano de Interação

Esta seção descreve a sequência de passos proposta para a detecção do plano de interação presente na cena. Tal detecção é feita sem qualquer procedimento de remoção de fundo ou conhecimento prévio a respeito da localização da câmera. Espera-se, porém, que o plano seja visível pela câmera RGB-D exceto quando ocluso pelos objetos acima dele (incluindo a mão). Ao final do processamento deste módulo, é gerada uma equação que representa a superfície de interação. Uma versão simplificada desta técnica foi publicada anteriormente (WEBER; JUNG; GELB, 2015).

3.1.1 Remoção de ruído

Como imagens de profundidade produzidas por câmeras RGB-D de consumo costumam ser ruidosas (KHOSHELHAM; ELBERINK, 2012), a primeira parte do algoritmo consiste na atenuação desse ruído. Apesar da existência de uma enorme variedade de abordagens para filtragem na literatura, constatou-se que um simples (porém efetivo e rápido) filtro da média

Figura 3.1 – *Pipeline* do método proposto. (a) Mapa de confiança. (b) Profundidade e localização 3D. (c) UVMap. (d) Detecção do plano. (e) Segmentação dos objetos. (f) Detecção da mão. (g) Localização da ponta dos dedos. (h) Detecção do toque. (i) Resultado da interação.



Fonte: Compilado pelo autor.

adaptativo mostrou-se capaz de auxiliar na identificação dos principais objetos no escopo deste trabalho: o plano de interação e as mãos.

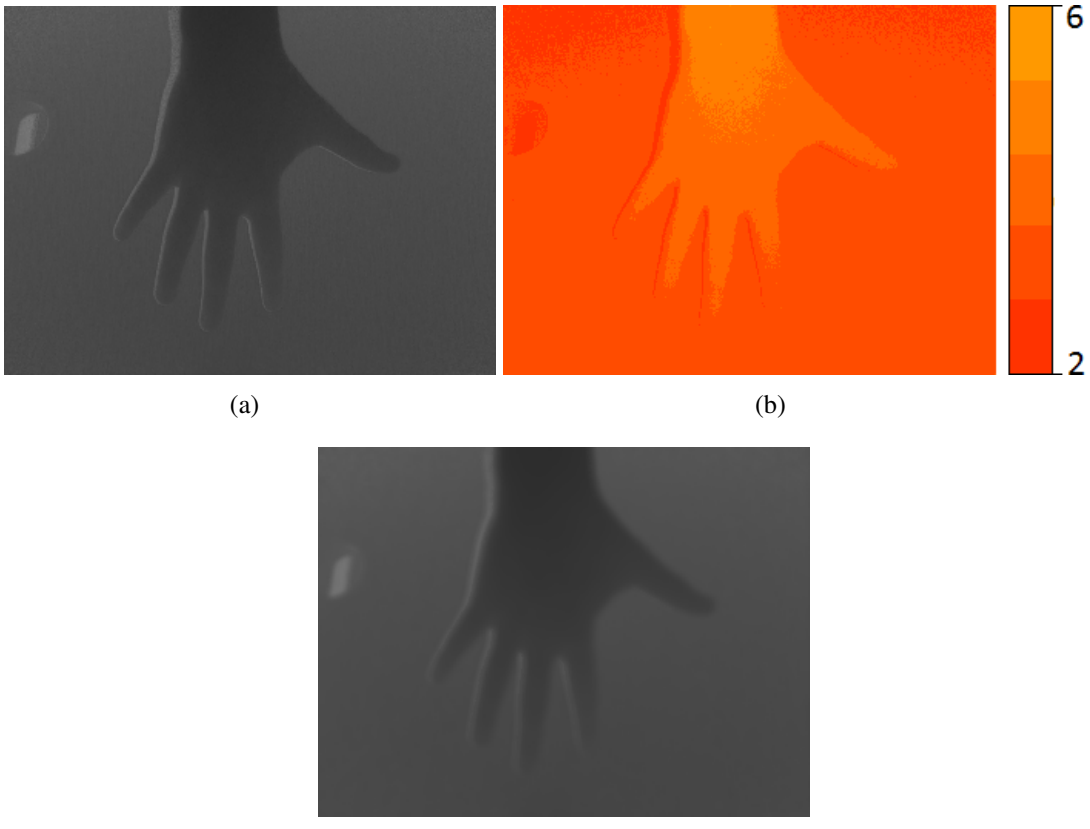
O filtro da média pode ser implementado em tempo linear utilizando-se imagens integrais para qualquer tamanho de janela (VIOLA; JONES, 2001). Porém, a seleção da janela representa o compromisso entre remoção de ruído e borramento de bordas: janelas maiores removem mais ruído e produzem mais borramento, e o contrário ocorre para janelas de tamanho menor. Em geral, esse compromisso é baseado na distância do objeto para a câmera: objetos próximos aparecem maiores, e consequentemente demandam janelas maiores. Dado que a aparência de

um objeto é inversamente proporcional à sua distância para a câmera em um modelo de projeção em perspectiva, adota-se neste trabalho um raio r_i para a janela de filtragem em cada ponto 3D i da imagem, o qual é dado por

$$r_i = \frac{r_b}{z_i}, \tag{3.1}$$

onde r_b é o *raio base* e z_i é a profundidade do ponto i. A Figura 3.2 ilustra o processo de filtragem. A imagem de profundidade original é mostrada na Figura 3.2(a), a imagem que representa as janelas de tamanho adaptativo na Figura 3.2(b) (onde, quanto maior a intensidade do tom de cinza, maior o raio) e o mapa de profundidade filtrado na Figura 3.2(c). Nesses resultados, a cor codifica o tamanho da janela de filtragem para cada ponto da imagem utilizando o mapa de cores "outono", onde o vermelho denota raios menores.

Figura 3.2 – Resultado da aplicação do filtro da média com janela de tamanho adaptativo sobre o mapa de profundidade. (a) Mapa de profundidade original. (b) Janelas de tamanho adaptativo. (c) Mapa de profundidade filtrado.



(a) (b)

(c)

Fonte: Compilado pelo autor.

3.1.2 Cálculo da Equação do Plano de Interação

Uma vez que o mapa de profundidade encontra-se filtrado, o próximo passo consiste em determinar o plano de interação $b_1x + b_2y + b_3z = b_4$. Como o plano não passa pela origem, temos $b_4 \neq 0$, e então pode-se dividir toda equação por b_4. Dessa forma, a equação resultante é $\boldsymbol{a}^T\boldsymbol{x} = 1$, onde $\boldsymbol{a} = (a_1, a_2, a_3)^T$ é o vetor normal ao plano e $\boldsymbol{x} \in \mathbb{R}^3$ é um ponto 3D. Assume-se que o plano ocupa a totalidade da imagem gerada pela câmera, com exceção da oclusão causada por objetos sobre ele. Para localizar a superfície de interação, utiliza-se uma versão do algoritmo *Random sample consensus*, ou RANSAC (FISCHLER; BOLLES, 1981) com coerência temporal, no qual o resultado da iteração anterior é utilizado como palpite inicial no quadro atual.

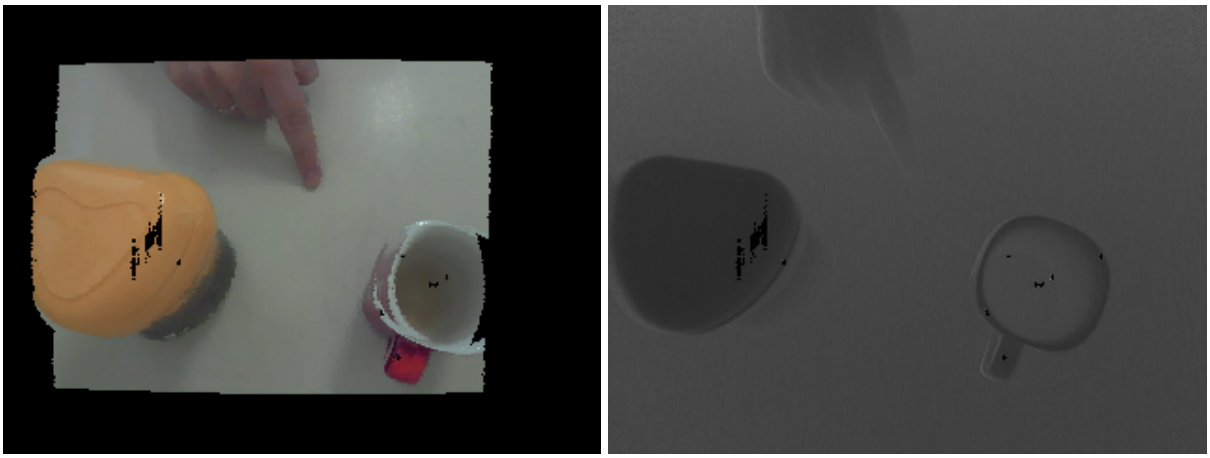
No quadro inicial, escolhem-se três pontos 3D de forma aleatória e é feito o cálculo do vetor normal $\boldsymbol{a}$ do plano correspondente. Em seguida, é calculada a distância

$$d(\boldsymbol{x}) = \frac{|\boldsymbol{a}^T\boldsymbol{x} - 1|}{\|\boldsymbol{a}\|} \tag{3.2}$$

a partir de todos os pontos $\boldsymbol{x}$ da nuvem de pontos para o plano. Para que esta equação do plano seja validada, isto é, considerada a equação que representa o plano de interação presente na cena, espera-se que ao menos uma fração p_{in} dos pontos 3D esteja a uma distância de, no máximo, d_{in} deste plano. Caso isso ocorra, é feito o refinamento da equação do plano a partir do uso de todos os pontos válidos (*inliers*) com o Método dos Mínimos Quadrados. Caso contrário, outro conjunto de três pontos é selecionado, de forma análoga ao fluxo tradicional do algoritmo RANSAC.

Para os demais quadros da sequência, é realizado procedimento análogo, porém tendo a equação do plano do quadro anterior como palpite inicial. Em ambientes onde a câmera encontra-se estática, este palpite gerado no quadro anterior tende a ser novamente validado, fazendo com que o algoritmo RANSAC convirja já na primeira iteração. Caso a câmera (ou o plano) se mova, esta abordagem ainda é capaz de capturar o plano, porém apenas depois de algumas iterações. A respeito dos parâmetros, optou-se por $p_{in} = 0.2$ de acordo com os experimentos realizados, e $d_{in} = 1cm$ a fim de atenuar o erro residual presente no mapa de profundidade. O resultado da aplicação desta técnica pode ser visto na Figura 3.3.

Figura 3.3 – Resultado da detecção do plano de interação. (a) Imagem RGB. (b) Mapa de profundidade. (c) Detecção do plano de interação (em vermelho) usando RANSAC.



(a) (b)

(c)

Fonte: Compilado pelo autor.

3.2 Segmentação dos Objetos

Esta seção trata da sequência de passos necessária para a correta segmentação dos objetos que se encontram acima da superfície de interação, a qual é descrita pela equação obtida na etapa anterior. No final do processamento deste módulo, é gerado um conjunto de *blobs*, onde cada um representa um objeto distinto presente na cena.

Dada a equação do plano, é possível estimar os pontos acima do mesmo avaliando-se a Equação 3.2 e definindo-se um limiar de aderência. Para cenários simples, esta abordagem, seguida por marcação a partir do uso de componentes conexos, poderia indicar objetos individuais acima do plano. Porém, em situações onde a mão encontra-se muito próxima ou completamente encostada na mesa, o procedimento acima descrito pode erroneamente marcar parte da

mão (e particularmente os dedos) como sendo integrante do plano de interação. Mesmo uma abordagem morfológica baseada em componentes conexos é incapaz de distinguir objetos nesta situação.

Este problema é abordado como sendo de segmentação de imagens através do uso da transformada *watershed* (MEYER, 1992). Nessa versão da transformada, são necessários marcadores para o fundo da cena e para os objetos de interesse. Além disso, é necessária uma função de energia que representa a fronteira dos objetos de interesse. Esses dois aspectos são abordados a seguir.

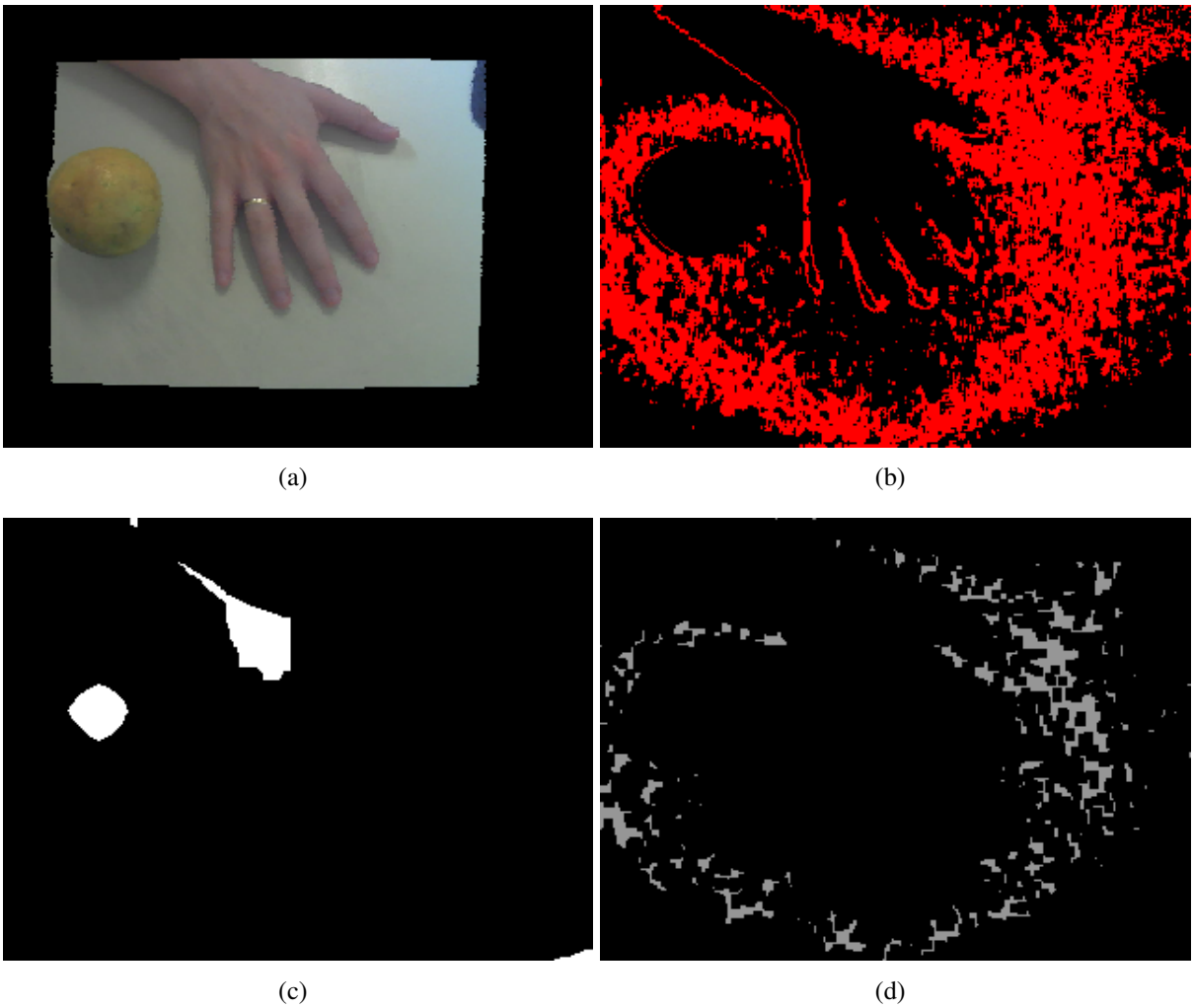
3.2.1 Definição dos Marcadores

Inicialmente, um conjunto de marcadores para o fundo é selecionado baseado na Equação 3.2, e em seguida pontos para os quais a distância para o plano for menor do que um limiar d_{bg} são selecionados. Apesar de ser possível dar o mesmo valor de d_{in} para d_{bg}, decidiu-se por utilizar um valor mais conservador com o propósito de se reduzir a incidência de objetos não-planares sendo incorretamente classificados como parte do plano, especialmente a ponta de dedos que encostam na superfície de interação (os quais são essenciais para a captura de eventos de toque). Mais precisamente, optou-se por $d_{bg} = 0.25cm$.

A fim de minimizar a seleção de falsos marcadores para o fundo gerados por esta abordagem que se encontram sobre objetos acima do plano, é aplicada a operação de erosão morfológica, de forma que apenas regiões com suficiente suporte da vizinhança sejam marcados. Processo similar é realizado para se obter os marcadores de objetos, porém utilizando pontos para os quais a distância ao plano (Equação 3.2) é maior do que um certo limiar $d_{fg} = d_{in} = 1$cm. Assim, evita-se a seleção de pontos pertencentes ao plano para a criação dos marcadores de objetos.

A Figura 3.4 ilustra o processo de seleção de marcadores para o plano e objetos acima desse plano. A imagem de entrada é mostrada na Figura 3.4(a), e os pontos planares detectados com d_{bg} são exibidos na Figura 3.4(b). Figuras 3.4(c) e 3.4(d) mostram os marcadores extraídos para os objetos acima do plano e para o plano propriamente dito, respectivamente.

Figura 3.4 – Ilustração dos marcadores para a segmentação baseada na transformada *watershed*. (a) Imagem RGB. (b) RANSAC com limiar conservador. (c) Marcadores para objetos acima do plano. (d) Marcadores para o plano.



(a) (b)

(c) (d)

Fonte: Compilado pelo autor.

3.2.2 Definição da Função de Energia

Outro fator crucial para a segmentação baseada em *watershed* é a definição da função de energia, a qual deve ter um valor alto na borda dos objetos de interesse e um valor baixo nas demais regiões da imagem. Para objetos longe do plano de interação, o simples uso da magnitude do gradiente da distância de um ponto para a superfície (dado pela Equação 3.2) geralmente apresenta resultados satisfatórios. Porém, esta abordagem normalmente falha quando os objetos estão muito próximos do plano, fazendo com que seja necessário o uso de informações extras a respeito da cena. Neste trabalho, também são exploradas informação de cor e de confiança produzidas pela câmera RGB-D.

Uma vez que a magnitude do gradiente de uma imagem RGB usualmente sofre influência

da iluminação ou de sombras presentes na cena, utiliza-se o mapa de cromaticidade no espaço de cores CIE L*a*b, dado por

$$C(u,v) = \sqrt{a(u,v)^2 + b(u,v)^2}, \tag{3.3}$$

onde a e b são as coordenadas cromáticas. A sombra, idealmente, está incorporada no canal L e, por isso, não interfere na cromaticidade.

O mapa de confiança dos dados de profundidade também mostrou-se uma boa fonte de informação a respeito do contorno das mãos. Em câmeras RGB-D do tipo ToF (*Time-of-Flight*), a informação de profundidade é obtida a partir da relação entre o sinal emitido e o recebido, sendo tipicamente luz infravermelha (IR). Quanto menor for a porção de luz refletida, menor é a quantidade de luz detectada e menor a amplitude do sinal (SWADZBA et al., 2007). Por este motivo, a amplitude do sinal recebido também é utilizada como medida de confiança.

O nível de confiança (luz refletida) depende tanto da reflectância da superfície quanto da orientação da mesma em relação à fonte emissora da luz (em geral, a própria câmera). No cenário utilizado neste trabalho, geralmente tem-se a câmera em posição descendente monitorando as mãos e a superfície de interação (com a possibilidade de haver também outros objetos não-planares). As mãos tendem a apresentar um alto valor de confiança: no cenário descrito, sua normal é aproximadamente alinhada com a da luz emissora, e a pele humana apresenta uma reflectância considerável na região próxima do infravermelho (KANZAWA; KIMURA; NAITO, 2011).

Porém, os ângulos normais da superfície ao longo da borda da mão são maiores do que da luz emissora, e regiões locais com um ângulo da normal amplo (de $70°$ a $90°$) capturados por sensores de profundidade tem sua amplitude de retorno reduzida significativamente (NGUYEN; IZADI; LOVELL, 2012). Como consequência, o mapa de confiança apresenta transições ao longo das bordas da mão que se destacam na imagem, as quais são capturadas pela magnitude do gradiente. Desta forma, além de aumentar o valor da energia ao longo das bordas da mão, o uso do gradiente da confiança também é capaz de remover a influência de gradientes de profundidade espúrios, os quais foram provavelmente corrompidos por ruído.

Por fim, o mapa de energia E é dado por

$$E(u,v) = \left(||\nabla d|| + \beta \frac{\epsilon^2}{\epsilon^2 + d^2} ||\nabla C(u,v)|| \right) ||\nabla c(u,v)||, \tag{3.4}$$

onde $\nabla c(u,v)$ é o gradiente normalizado do mapa de confiança, d é a função da distância ao plano definida na Equação 3.2, β controla a influência da cromaticidade e ϵ controla quão pró-

xima do plano a informação de cor começa a ser utilizada na composição do mapa de energia. Como explicado anteriormente, a informação de profundidade é boa o suficiente para segmentar a mão quando esta encontra-se distante do plano (ou seja, quando d é grande), porém esta tarefa torna-se mais complicada à medida em que a mão se aproxima do plano. Por esta razão, é dada uma importância progressiva à informação de cromaticidade quando d é pequeno. Valores considerados adequados para β e ϵ são 0.15 e 0.007, respectivamente, de acordo com experimentos realizados.

Uma vez que as fronteiras dos *blobs* resultantes são frequentemente ruidosas (devido principalmente à baixa qualidade do mapeamento uv entre profundidade e cor feito pelo *middleware* fornecido pela câmera), é feita a filtragem desses *blobs* através do filtro da mediana. O resultado de aplicar a segmentação *watershed* quando a mão encontra-se completamente encostada no plano (com exceção do dedo) pode ser vista na Figura 3.5, juntamente com os componentes individuais da energia E.

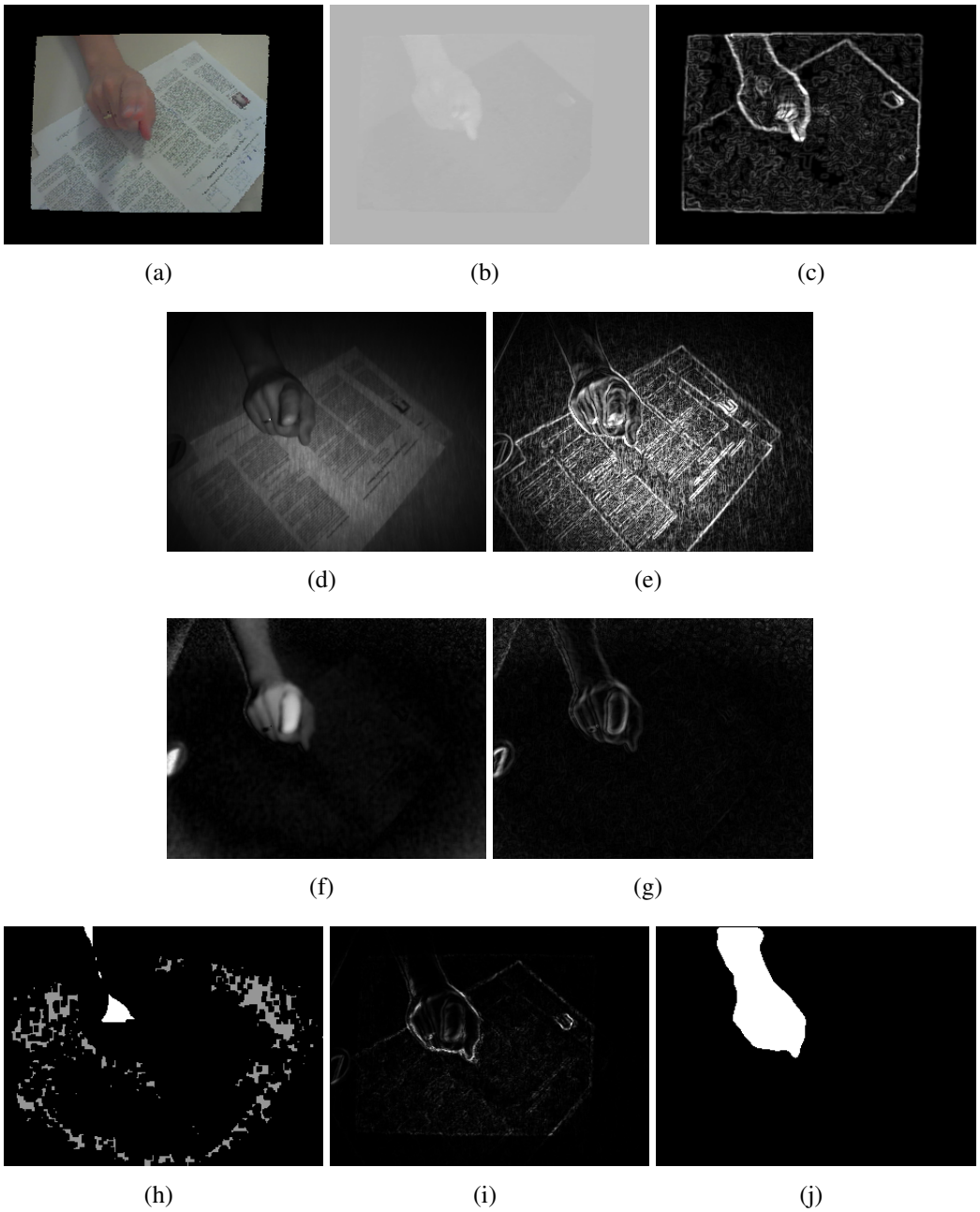
3.3 Detecção das Mãos Dentre os Objetos Sobre o Plano

Esta seção abrange o processo proposto para a identificação dos objetos acima do plano com características que o definam como mão ou um objeto qualquer. Visando baixa complexidade computacional, optou-se por não se fazer uma análise de forma da mão (que pode ser bastante variada), mas ao invés uma rápida avaliação baseada em cor de pele. Mais especificamente, foi feita a utilização de um algoritmo de detecção de cor de pele (CHAI; NGAN, 1999) juntamente com uma técnica que explora a confiança dos dados de profundidade gerados na região da mão, a qual apresenta comportamento diferente da confiança produzida por uma série de objetos pesquisados. Ao final deste módulo, apenas os *blobs* que correspondem a mãos (de acordo com esse processo) são repassados às etapas posteriores.

3.3.1 Detecção Utilizando Informação de Cor de Pele

O procedimento para a segmentação de objetos acima do plano descrito até então leva em consideração algumas propriedades esperadas em mãos (como a redução da confiança da profundidade nas bordas), porém não é capaz de distinguir mãos de outros objetos. Por esta razão, o próximo passo é responsável por explorar a informação de cor de pele com o objetivo de identificar as mãos dentre todos os objetos segmentados.

Figura 3.5 – Etapas intermediárias e resultado final da segmentação baseada na transformada *watershed*. (a) Imagem RGB. (b) Cromaticidade AB. (c) Gradiente da cromaticidade AB. (d) Confiança. (e) Gradiente da confiança. (f) Distância ao plano. (g) Gradientes da distância ao plano. (h) Marcadores gerados. (i) Função de energia. (j) Resultado da transformada *watershed*.



(a) (b) (c)

(d) (e)

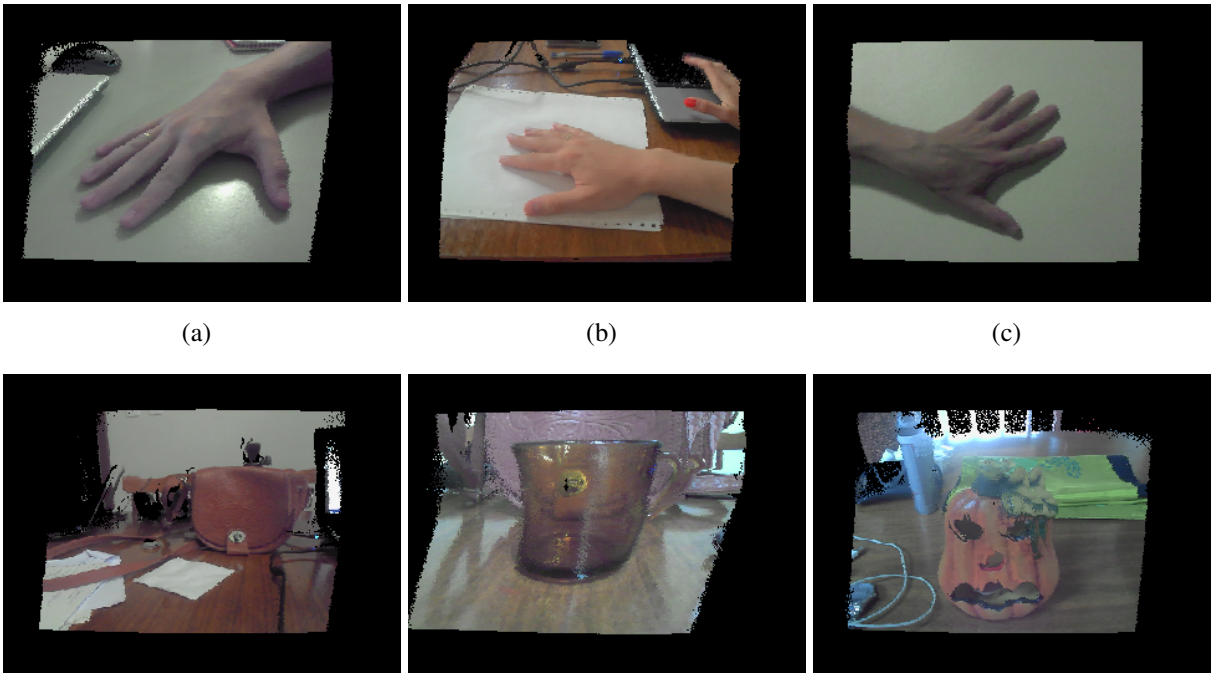
(f) (g)

(h) (i) (j)

Fonte: Compilado pelo autor.

A utilização de técnicas de segmentação de cor com o propósito de se localizar *pixels* com cor de pele é uma tarefa difícil e propensa a erros, e várias técnicas foram propostas no passado. Estas diferenciam-se pelo uso de distintos espaços de cor, esquemas de classificação e compromisso entre complexidade e tempo de execução. No escopo deste trabalho não há a necessidade de um esquema de detecção de pele com alta taxa de acerto; o principal objetivo consiste em

Figura 3.6 – Algumas das imagens de *dataset* criado. Figuras (a), (b) e (c) são corretamente classificadas como mãos. A bolsa (d) e a xícara (d) são classificados como objetos com cor de pele, apenas, mas a abóbora de adorno (e) é incorretamente classificada como mão.



(a) (b) (c)

(d) (e) (f)

Fonte: Compilado pelo autor.

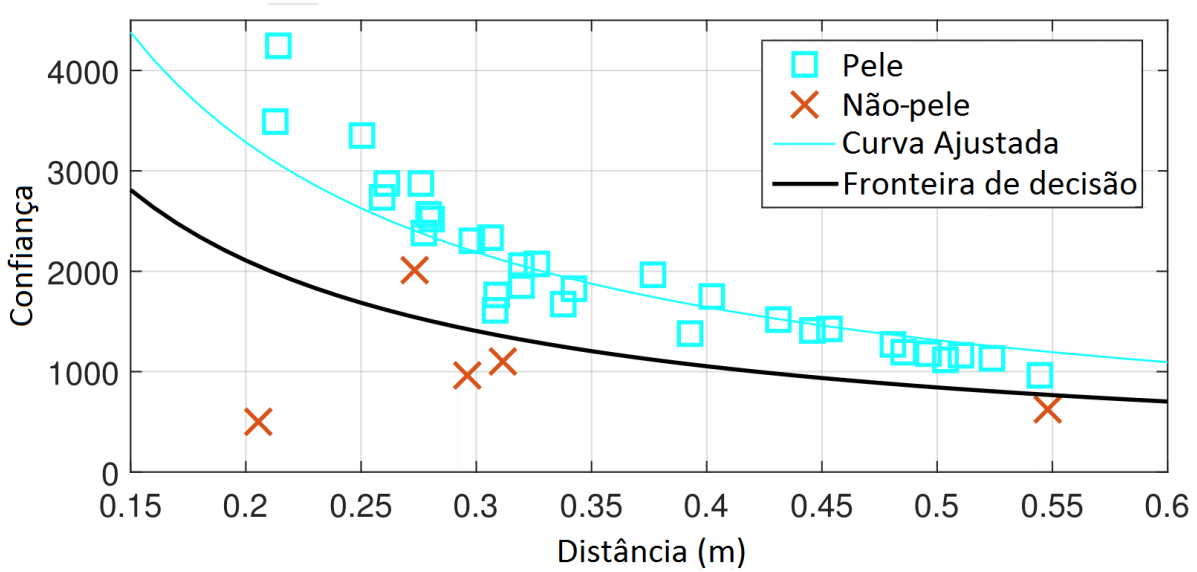
examinar cada um dos objetos sobre a mesa de uma forma muito rápida e decidir quais deles são mãos.

Uma classe popular de detectores de pele é baseada em regiões predefinidas de um dado espaço de cores. Neste trabalho, adotou-se uma abordagem de segmentação de cor (CHAI; NGAN, 1999) que é rápida e produz bons resultados. O espaço de cor utilizado é o YCrCb pois, de acordo com os autores, há a possibilidade de se trabalhar com crominância (representada pelos canais Cr e Cb) e luminância (representada pelo canal Y) separadamente. Ainda de acordo com o autor, a cor de pele de diferentes raças humanas se diferencia principalmente pelo tom escuro ou claro da pele, as quais são características codificadas apenas por Y e não por Cr e Cb. Desta forma, os limiares de cor de pele são dados por

$$0 < Y < 256 \wedge 133 < Cr < 173 \wedge 77 < Cb < 127. \tag{3.5}$$

Esta técnica, porém, comumente produz falsos positivos, como pode ser visto na Figura 3.8. Isso ocorre pelo simples fato de que qualquer objeto cuja cor estiver dentro do intervalo proposto será considerado como tendo cor de pele. Desta forma, buscou-se a utilização de informações extras que possam auxiliar na identificação de tais objetos, como será explicado a seguir.

Figura 3.7 – Confiança *versus* distância para mãos (quadrados azuis), modelo ajustado (curva azul), fronteira de decisão (curva preta) e objetos com cor de pele (cruzes vermelhas).



Fonte: Compilado pelo autor.

3.3.2 Detecção Utilizando Informação de Confiança

A pele é conhecida por apresentar uma reflectância consideravelmente alta no espectro infravermelho, como explicado (e explorado) na Seção 3.2. Uma vez que a câmera captura a amplitude do retorno, é natural assumir que essa resposta também depende da distância do objeto até a câmera. Para medir o relacionamento entre confiança e distância para *pixels* com cor de pele, foi criado um pequeno *dataset* com variação na iluminação e distância entre as mãos e a câmera (utilizou-se uma câmera *SoftKinetic DS325 RGB-D* em todos os experimentos), juntamente com uma máscara (criada de forma manual) informando os *pixels* que pertencem à mão presente na cena.

Os dados coletados estão sumarizados na Figura 3.7, sugerindo uma relação na forma $c = \lambda/d$, de tal maneira que se espera que os produtos cd são aproximadamente constantes. O valor $\lambda = 421.5635$ foi estimado como o valor médio para os produtos cd usando todas as amostras da base de dados, e a curva ajustada $c = \lambda/d$, a qual pode ser vista na Figura 3.7, indica uma boa aproximação. Também foi computado o desvio padrão σ de cd, a fim de que amostras relacionadas à cor de pele para as quais $|cd - \lambda| > k\sigma$ não sejam aceitas, onde o número de desvios padrão k controla o intervalo de aceitação/rejeição. Assim, rejeita-se *pixels*

que passam o teste da cor de pele dado pela Equação 3.5 se

$$c < (\lambda - k\sigma)/d, \tag{3.6}$$

salientando que decidiu-se utilizar apenas o limiar inferior uma vez que a maioria dos objetos sem cor de pele testados possuem menor confiança do que a mão. Na realidade, esta fronteira de decisão para $k = 2.5$ também está ilustrada na Figura 3.7, juntamente com cinco amostras de objetos que não possuem pele mas que passam no teste de cor. A fronteira de decisão escolhida não exclui nenhuma das amostras reais de pele, mas é capaz de corretamente rejeitar quatro das cinco amostras de objetos sem pele. Finalmente, um dado *blob* é validado como mão se uma fração T_h (a qual foi atribuída experimentalmente o valor de 0.2) de *pixels* do *blob* passar no teste de cor dado pela Equação 3.5 e pelo teste de confiança e profundidade dado pela Equação 3.6. A Figura 3.6 mostra alguns exemplares de pele e de outros objetos com cor de pele utilizados em nossos experimentos. Nesses exemplos, os três primeiros objetos foram corretamente rejeitados, enquanto o quarto foi corretamente aceito. Já o quinto objeto foi incorretamente aceito pelo algoritmo proposto.

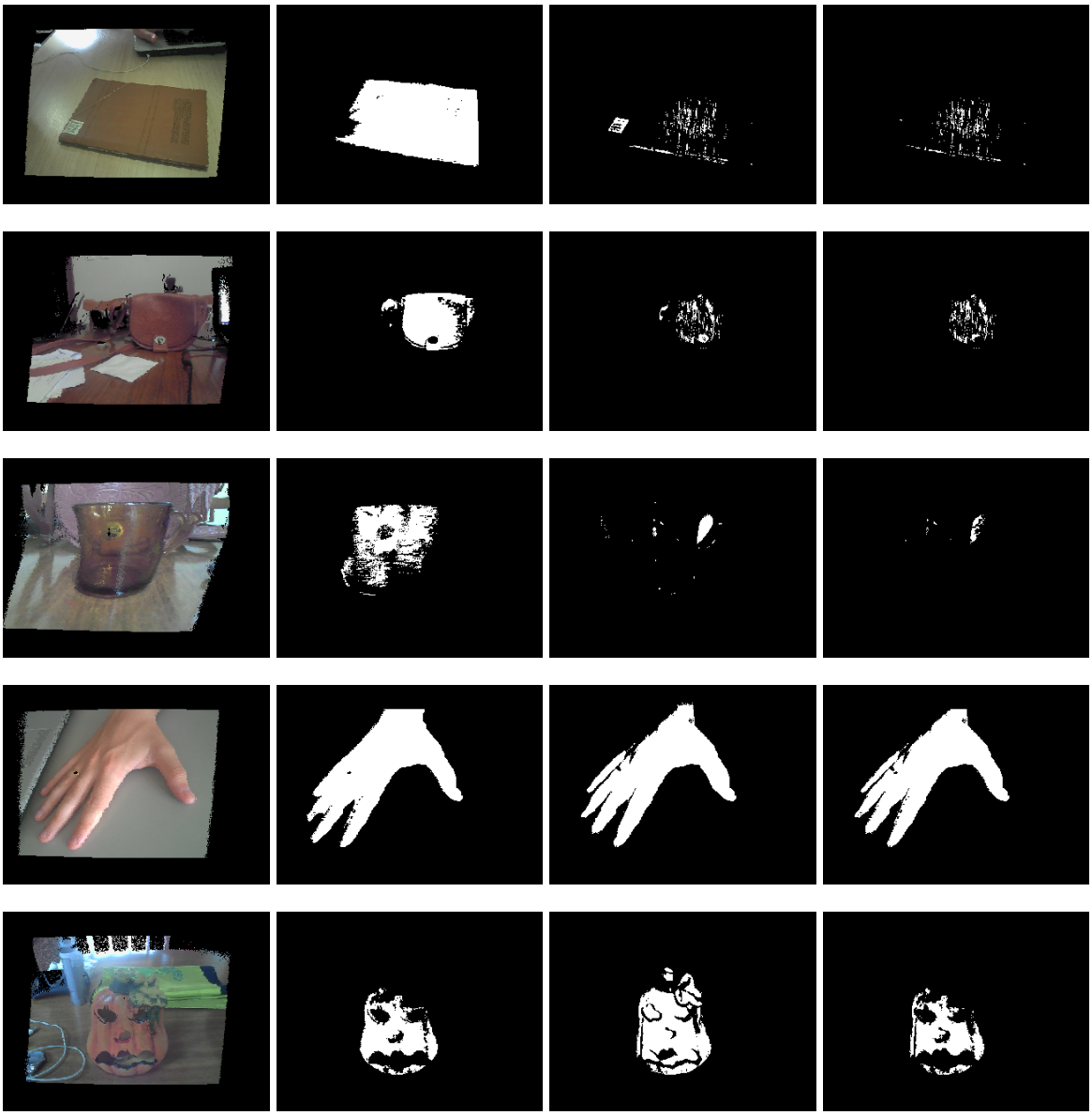
3.4 Detecção da Ponta dos Dedos

Assumindo que a identificação das mãos na cena ocorreu de forma correta, é possível agora focar os esforços no formato da mão. Mais precisamente, deseja-se saber a localização das pontas dos dedos a fim de verificar se um ou mais deles está tocando a superfície de interação. Nesta etapa, assume-se que os dedos presentes na cena não estejam encostados uns nos outros, de forma a ser possível detectar cada um deles individualmente. Considerando-se que a projeção de um dedo é grosseiramente uma região retangular com uma das pontas arredondadas, espera-se que seja produzido um esqueleto morfológico (ou eixo medial) (SERRA, 1982) relativamente linear tal que:

- A Transformada da Distância (TD) ao longo do esqueleto deva ser aproximadamente pequena, uma vez que os dedos são estruturas finas quando comparadas com a mão;
- A TD deve ser aproximadamente constante ao longo do esqueleto, devido à geometria esperada de um dedo projetado.

Neste trabalho, utiliza-se um método rápido para o cálculo de um esqueleto multiescala (TELEA; WIJK, 2002), o qual é pouco sensível a ruídos presentes na forma 2D do objeto e que

Figura 3.8 – Resultados para o teste de cor dado pela Equação 3.5 e 3.6. Imagens contendo objetos com cor de pele (primeira coluna) segmentadas pela cor de pele (segunda coluna) e pela confiança (terceira coluna). O resultado final da segmentação encontra-se na quarta coluna.



Fonte: Compilado pelo autor.

também calcula a TD desta forma 2D. Dado um esqueleto S, é feita a análise de cada ramo b partindo-se da ponta em direção ao interior do esqueleto. É então computado o valor

$$C_k^b = d_0^b + \sum_{i=1}^{k} |d_i^b - d_{i-1}^b|, \tag{3.7}$$

onde d_i^b denota a TD do $i^{ésimo}$ *pixel* pertencente ao ramo b ($i = 0$ denota a ponta do dedo). É importante notar que C_k^b é o valor da TD na ponta do dedo acrescido da variação acumulada (em valores absolutos) ao longo dos primeiros k pontos do ramo. Em um esqueleto relacionado

a um dedo, espera-se que ambos os termos sejam pequenos, como explicado anteriormente. Por isto, o ramo b é detectado como um dedo se $C_K^b < T_f$, onde K é o número de pontos avaliados no esqueleto (ao qual foi atribuído o valor de 15 de acordo com experimentos realizados), e $T_f = 15$, de acordo com a espessura observada em um dedo de proporções padrão a uma distância de 30cm da câmera. Neste trabalho, considera-se que a ponta do dedo encontra-se no ponto extremo do esqueleto, o que normalmente corresponde ao meio da unha deste mesmo dedo.

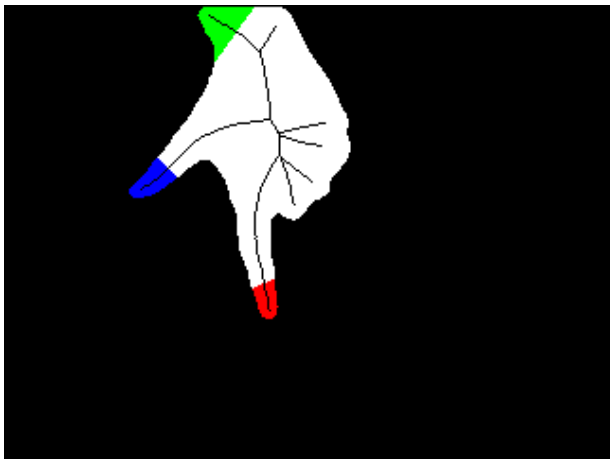
A Figura 3.9(a) ilustra o procedimento de detecção de dedos. Mais precisamente, é mostrada a máscara binária obtida através do processo de segmentação e detecção de cor de pele proposto neste trabalho, juntamente com os esqueletos computados. Dos 8 ramos gerados pelo processo de esqueletonização, dois deles (marcados em azul e vermelho) foram corretamente detectados como dedos. A Figura 3.9(b) mostra o gráfico de d_i^b para três candidatos a dedo, marcados com cores distintas na Figura 3.9(a). Como pode ser observado, os valores de d_i^b são aproximadamente constantes para os dedos, mas eles aumentem rapidamente para o ramo relacionado à mão (mostrado em verde). No contexto deste trabalho, p_j consiste em atribuir uma ponta de dedo de um quadro $t-1$ a uma ponta de dedo de um quadro t. Já s_i é dado pelo custo (a distância euclidiana entre as coordenadas de imagens entre dois dedos) de se associar um dedo de $t-1$ a um dedo de t.

Como a detecção de dedos é feita de forma independente a cada quadro, não há uma associação direta entre cada dedo em quadros adjacentes. Porém, como esta informação é necessária para o rastreamento dos dedos ao longo do tempo, faz-se uso do algoritmo Húngaro (KUHN, 1955). Essa técnica, a qual resolve problemas de atribuição em tempo polinomial, recebe como entrada uma matriz $n \times m$ onde o elemento da $i^{ésima}$ linha e $j^{ésima}$ coluna corresponde ao custo de resolver o problema p_j a partir da proposta de resolução s_i.

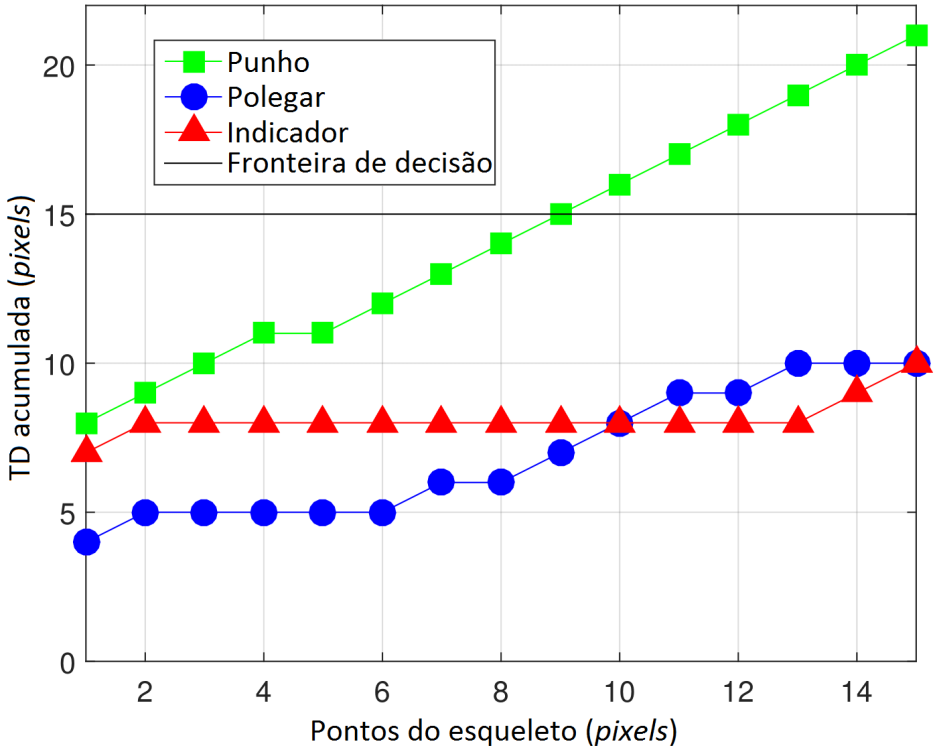
3.5 Identificação da Interação

Esta seção foca majoritariamente na interação propriamente dita que ocorre quando um dedo encosta no plano. Essa detecção é responsável não apenas pelas características físicas do toque, como a distância do dedo à superfície e a localização, em coordenadas de imagem, do toque, como também da coerência temporal de sucessivos contatos com o plano realizados por um mesmo dedo. Desta forma, torna-se possível a criação de formas mais complexas de interação, como desenho, duplo clique, entre outros.

Figura 3.9 – Identificação das pontas de dedos. (a) Esqueleto gerado para uma dada mão. (b) O valor da transformada da distância ao longo dos ramos indicados pelas regiões coloridas em (a).



(a)



(b)

Fonte: Compilado pelo autor.

3.5.1 Detecção do Toque

Dada a localização $\boldsymbol{x}$, em 3D de um dado dedo sob análise no tempo t, é computada sua distância $\delta_t = d(\boldsymbol{x}_t)$ ao plano de interação utilizando-se a Equação 3.2. Caso $\delta_t < T_p$, considera-se que o dedo está tocando a superfície. A definição do limiar T_p depende da altura do dedo e de erros acumulados no processo como um todo (estimação do plano, detecção da ponta do dedo,

ruído do sensor RGB-D, entre outros), e escolheu-se $T_p = 0.02$m por apresentar resultados satisfatórios nos cenários testados. É importante mencionar que, idealmente, δ_t jamais será zero, uma vez que a distância do dedo ao plano é computada a partir da parte superior da ponta do dedo, enquanto apenas a parte inferior do mesmo toca o plano.

3.5.2 Suavização da Trajetória de Toque

Uma vez que a localização da ponta de um dedo tende a conter erros, o Filtro de Kalman é utilizado para atenuar as trajetórias obtidas. Seja $\boldsymbol{u}_t = (u_t, v_t)^T$ a coordenada de imagem observada de um dedo ao longo do tempo. O vetor de estados é dado por $\boldsymbol{y}_t = (U_t, V_t, \dot{U}_t, \dot{V}_t)^T$, onde $(U_t, V_t)^T$ são as coordenadas filtradas e $(\dot{U}_t, \dot{V}_t)^T$ a velocidade. O modelo de movimento é dado por

$$\boldsymbol{y}_{t+1} = F\boldsymbol{y}_t + \boldsymbol{w}, \tag{3.8}$$

onde

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

é a matriz de transição considerando-se uma velocidade constante com $\Delta t = 1$, o que indica que a medida da localização dos dedos é feita a cada quadro, e $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \eta I_4)$ é o ruído do processo dado por uma distribuição normal multivariada e isotrópica de média 0 e matriz de covariância ηI_4, sendo η um valor pequeno (ao qual foi atribuído o valor de 10^{-4} de acordo com experimentos) de forma a permitir velocidades variáveis, e I_4 é a matriz identidade 4×4. Logo, assume-se que a perturbação η é igual e independente em cada componente do vetor de estados $\boldsymbol{y}_t$.

A equação de medida (que relaciona o vetor de estados $\boldsymbol{y}_t$ com a observação $\boldsymbol{u}_t$) é dada por

$$\boldsymbol{u}_t = H\boldsymbol{y}_t + \boldsymbol{v}_t, \tag{3.10}$$

onde

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.11}$$

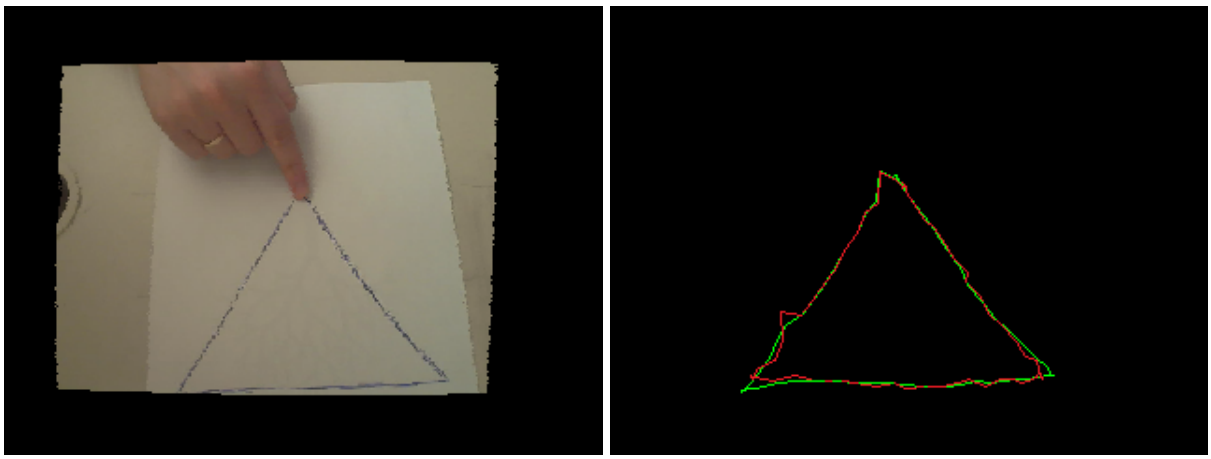
e $\boldsymbol{v}_t$ é o ruído de observação dependente do tempo, ou seja, as componentes de posição do vetor de observação são compostas pela posição observada juntamente com um ruído aditivo.

O comportamento do Filtro de Kalman é diretamente afetado pelo relacionamento entre o ruído de processo e de observação. Quando o ruído de processo é muito menor, o modelo de movimento é priorizado, levando a trajetórias mais suaves. Por outro lado, os pontos observados são priorizados quando o ruído de observação é muito menor. Idealmente, é desejável dar maior peso para o modelo de observação quando a medição for "confiável", e mais importância para o modelo de movimento, caso contrário. Neste trabalho, considera-se que a observação da localização da ponta de um dedo $\boldsymbol{u}_t$ é confiável quando a confiança correspondente c_t produzida pela câmera é grande, e quando a distância δ_t da ponta do dedo ao plano é menor (uma vez que é mais provável que esteja tocando o plano). Baseado nestas hipóteses, optou-se por $\boldsymbol{v}_t \sim \mathcal{N}(\mathbf{0}, \rho_t I_2)$, onde

$$\rho_t = \beta_0(1 - \exp(-\beta_1 \delta_t / c_t))I_2, \tag{3.12}$$

β_0 e β_1 são parâmetros (com valores de 0.001 e 100000 baseado em experimentos, respectivamente), e I_2 é a matriz identidade 2×2. Valores maiores para β_0 e β_1 priorizam o modelo de movimento, e o valor de ρ_t tipicamente oscila entre 0.001 e 0.003. É importante notar que $\rho_t \approx 10\eta$ quando a confiabilidade for a maior possível, o que levaria a uma maior fidelidade em relação à localização do dedo. À medida que a confiabilidade decresce, a suavização da curva ganha progressivamente mais peso. Na Figura 3.10, é possível comparar a curva suavizada pelo Filtro de Kalman e a mesma curva sem suavização.

Figura 3.10 – Efeito do uso do filtro de Kalman em uma curva. (a) Último quadro da sequência de desenho. (b) Desenho sem o uso do Filtro de Kalman em vermelho e com o uso do Filtro em verde.



(a) (b)

Fonte: Compilado pelo autor.

3.6 Discussão

Este capítulo apresentou a técnica proposta, de forma a detalhar tanto os desafios relacionados ao uso de informação de cor e profundidade da cena em um sistema de interação utilizando uma câmera RGB-D, quanto as soluções propostas para cada um dos problemas. A primeira parte focou especialmente na detecção da superfície de interação, gerando como saída a equação do plano que o representa. Em seguida, objetos sobre esta superfície foram devidamente segmentados, sendo então selecionados apenas aqueles que possuíam cor dentro do intervalo definido para cor de pele confiança dentro do intervalo. Posteriormente, foi mostrada a técnica para detecção da localização da ponta dos dedos, bem como daqueles que estão interagindo com o plano.

4 RESULTADOS EXPERIMENTAIS

Este capítulo visa demonstrar os resultados obtidos com a técnica proposta a partir do desenvolvimento de um protótipo que explora algumas possibilidades de interação com o plano (e acima dele). Os resultados variam tanto em relação ao ambiente no qual o sistema é testado quanto à interação pretendida pelo usuário.

Com base nos toques detectados, há uma série de possibilidades de interação com o plano que podem ser exploradas. Neste trabalho, foi implementado um protótipo que permite ao usuário o desenho de formas utilizando o dedo de acordo com os seguintes eventos:

Duplo clique: dois toques na superfície realizados pelo mesmo dedo e dentro de um pequeno intervalo de tempo (um segundo), com o propósito de limpar a imagem.

Desenho: toque no plano de forma contínua com o dedo. O usuário pode desenhar com mais de um dedo simultaneamente, e cada dedo irá gerar uma linha distinta. No protótipo desenvolvido, o modo desenho é ativado quando um toque é detectado por, no mínimo, um segundo.

***Spraying*:** se o usuário clicar "s", cada dedo presente na cena irá gerar uma pintura em forma de "*spray*" na posição em que se encontra a ponta do dedo. O raio do *spray* é proporcional à distância do dedo ao plano. O uso de teclado, neste caso, foi motivado apenas pela simplificação da prova de conceito, sendo que a ativação por meio de gestos seria a forma ideal de lidar com este procedimento.

Também são mostrados os resultados qualitativos e quantitativos envolvendo o tempo de execução e a acurácia da técnica. Mais precisamente, é feita uma avaliação tanto do erro de localização do toque quanto da porcentagem de quadros em que o sistema detectou corretamente a interação pretendida pelo usuário.

Utilizou-se uma câmera *DS325 RGB-D* (Softkinetic, 2016a) como dispositivo de entrada com o SDK versão $1.5.0.1$. Ela mede o tempo que a luz de infravermelho (emitida pela própria câmera) leva para retornar ao dispositivo, técnica conhecida como *time-of-flight* (ToF). A cadência da câmera é de 30 quadros por segundo, e seu alcance nominal vai de 0.15m a 1m. De acordo com o fabricante, o erro na resolução da profundidade é de, no máximo, 1.4cm para objetos distantes 1m da câmera.

O sistema foi desenvolvido em C++ e utilizou a biblioteca OpenCV versão $2.4.11$. Todos os testes foram executados em um computador com as seguintes especificações:

- Processador: Intel Core i7-2700k CPU 3.50 GHz

- Memória (RAM): 16.0 GB
- Sistema Operacional: Windows 10 64 bits

Utilizando apenas um núcleo do processador e imagens RGB-D 320×240 como entrada, o tempo médio de execução para cada quadro foi de 109ms, o que equivale a 10 quadros por segundo. Uma análise mais detalhada do tempo de execução de cada módulo do sistema é apresentada na Tabela 4.1.

Tabela 4.1 – Tempo Médio de Execução dos Módulos da Técnica.

Módulo do Sistema	**Tempo Médio**
Segmentação do plano e de objetos sobre ele	57ms
Identificação de objetos com cor de pele	2ms
Localização da ponta dos dedos	40ms
Detecção do toque	10ms
TOTAL	**109ms**

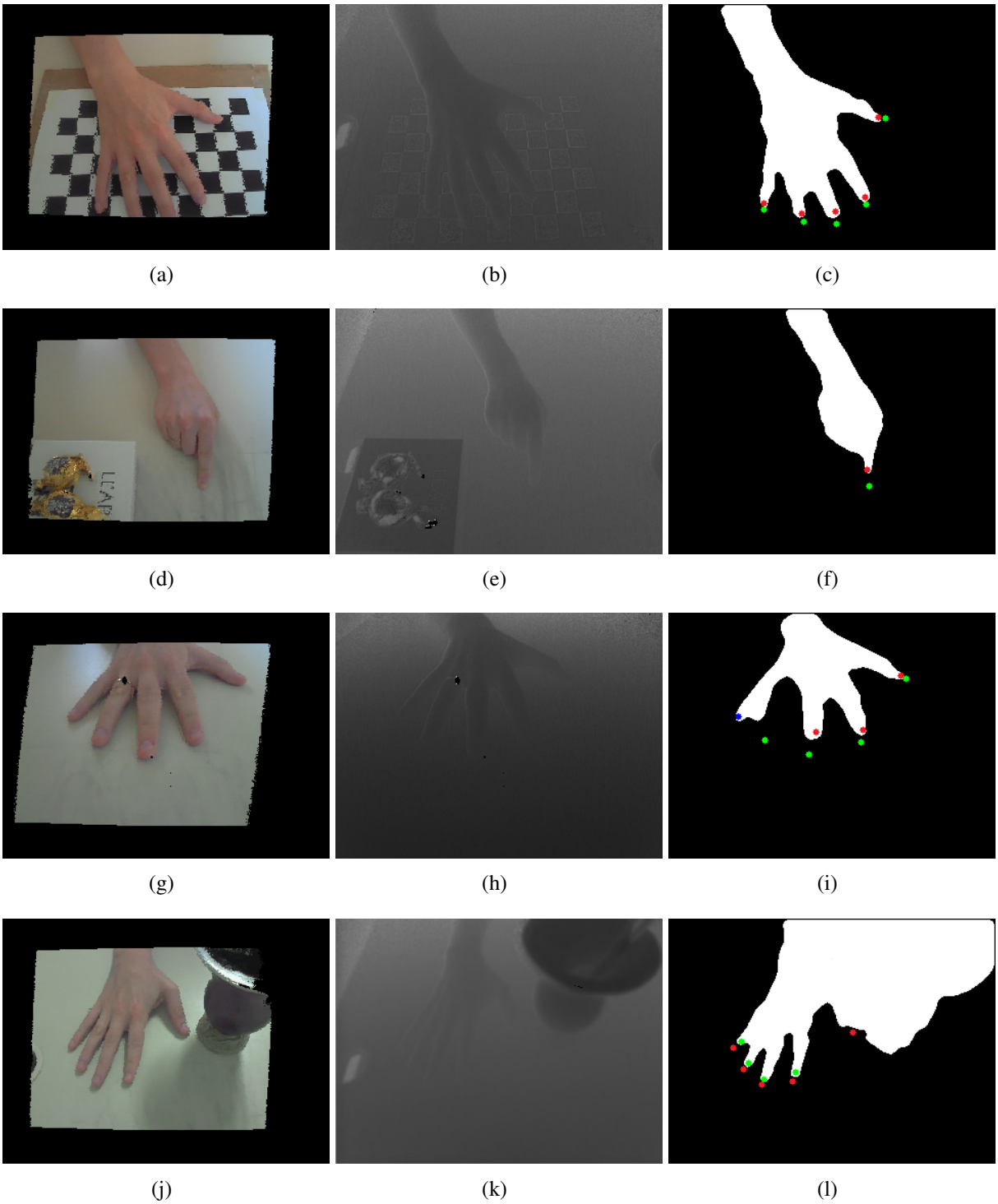
Fonte: Compilado pelo autor.

4.1 Avaliação da Segmentação da Mão e Identificação dos Dedos

A Figura 4.1 mostra diferentes cenários de utilização da técnica juntamente com o respectivo resultado para a segmentação das mãos e identificação da ponta dos dedos. Pontos verdes representam a localização real da ponta do dedo (neste caso, considera-se a ponta do dedo como sendo o meio da unha do dedo), pontos vermelhos a localização prevista pela técnica (a qual corresponde à ponta do esqueleto morfológico gerado para o dedo) e o ponto azul indica que a posição dada pela técnica encontra-se exatamente sobre a localização real. A Figura 4.1(a) apresenta o resultado da segmentação da mão acima de um tabuleiro de xadrez. Neste cenário, é interessante notar que, apesar de serem planos, os quadrados escuros apresentam um valor alto na respectiva imagem de profundidade (Figura 4.1(b)). Isso ocorre pois superfícies escuras absorvem boa parte da luz de infravermelho (ALHWARIN; FERREIN; SCHOLL, 2014), o que prejudica a qualidade dos dados de profundidade da câmera. Apesar disso, a técnica proposta foi capaz de segmentar a mão corretamente.

Já na Figura 4.1(d), é possível observar um objeto não-planar presente na cena que é corretamente eliminado na etapa de verificação da cor de pele. Na Figura 4.1(g), o anel usado pelo usuário provocou uma reflexão especular no infravermelho, o que pode ter prejudicado a segmentação do dedo, fazendo com que o mesmo não fosse detectado nas etapas posteriores.

Figura 4.1 – Resultado da técnica para a segmentação das mãos. (a-c) Mão segmentada sobre tabuleiro de xadrez. (d-f) Mão segmentada e objetos rejeitados. (g-i) Mão deitada na mesa incorretamente segmentada. (j-l) Mão e objeto incorretamente segmentados.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fonte: Compilado pelo autor.

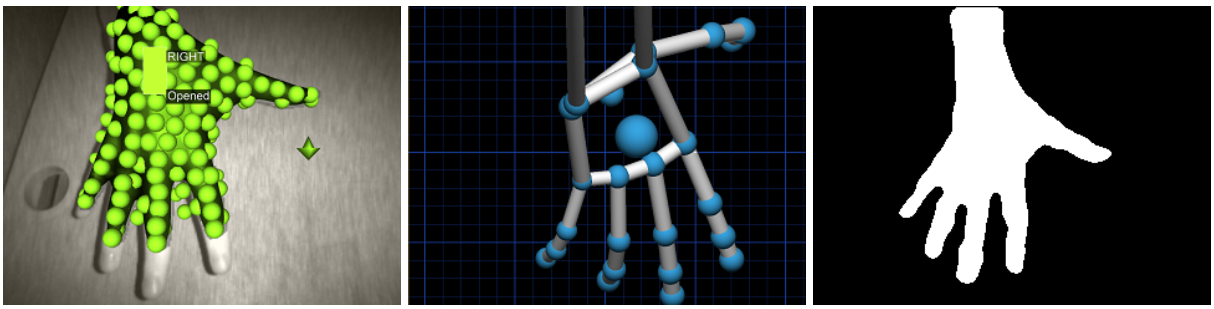
Por fim, a Figura 4.1(j) mostra um caso onde o resultado da segmentação acaba incorporando mão e objeto em um mesmo *blob*. Neste caso, a incorporação da região do plano existente entre os dois objetos foi provocada pela falta de marcadores de fundo nesta região. Por esta razão,

um dos dedos interagindo com o plano não foi detectado.

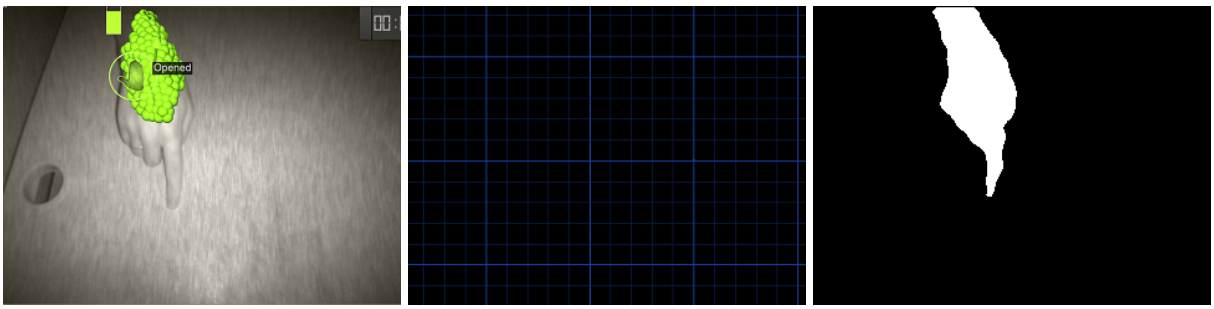
Também foi feita a comparação do método proposto com dois produtos comerciais focados em detecção de pose da mão e identificação de dedos: o *middleware iiSu* versão 3.6.0.2 para a detecção da mão, o qual também opera através da câmera *Softkinetic DS325 RGB-D*, e o *software Leap Motion Diagnostic Visualizer* 2.3.1 (Leap Motion, Inc., 2016b). Em todos os testes envolvendo o sensor *Leap Motion*, este foi posicionado no mesmo local que a câmera RGB-D e com mesma orientação (de acordo com inspeção visual) de forma a haver uma comparação justa.

A Figura 4.2 mostra dois cenários distintos: a mão distante do plano (primeira linha) e próxima a ele (segunda linha). Em ambos os cenários, a técnica proposta foi capaz de segmentar corretamente a mão. Já a Figura 4.2(a) mostra que o *software iiSu* é capaz de reconhecer a mão no cenário proposto nesse trabalho quando ela está distante do plano, mas falha quando a mesma se encontra próxima a ele, como na Figura 4.2(d). Situação semelhante ocorre com o *Leap Motion*, o qual detecta a mão longe da superfície, como pode ser visto na Figura 4.2(b), mas é incapaz de detectá-la quando ela encontra-se encostada na mesa (Figura 4.2(e)). Além disso, é importante mencionar que não foram encontradas técnicas na literatura para segmentação de mãos que utilizam câmera RGB-D no ambiente proposto.

Figura 4.2 – Comparação com produtos comerciais para a mão distante do plano (primeira coluna) e próxima a ele (segunda coluna). (a,d) Resultados para o *middleware iiSu*. (b,e) Resultados para o *software Leap Motion Visualizer*. (c,f) Resultados para a técnica proposta.



(a) (b) (c)



(d) (e) (f)

Fonte: Compilado pelo autor.

4.2 Acurácia da Técnica na Detecção de Toques e Duplos Cliques

Para estimar a acurácia da técnica em relação a toques e duplos cliques, foram gravados 14 vídeos de dois usuários utilizando o protótipo, nos quais eles efetuaram toques (incluindo eventos de desenho) e duplos cliques, em um total de 1584 quadros. Posteriormente, foi feita a marcação manual de cada quadro de acordo com o estado do dedo (no ar ou tocando o plano), resultando em 1358 quadros onde o dedo toca o plano e 226 onde não o toca. No processo de marcação, um evento de toque foi considerado correto caso fosse detectado pelo sistema em um intervalo não maior do que 300ms em relação ao toque real.

Já para eventos de duplo clique, foi feita a marcação manual do momento em que cada duplo clique (31 no total) realizado pelos usuários foi efetuado. De forma análoga à detecção no modo desenho, o evento de duplo clique foi considerado correto caso fosse detectado pelo sistema em um intervalo não maior do que 300ms.

4.2.1 Detecção de Toques

A acurácia da técnica na detecção de eventos de toque é sumarizada na Tabela 4.2. Como pode ser observado, o sistema foi capaz de detectar 1293 toques executados, ou 95%, com 35 detecções falsas (ou 2% do total de número de quadros). É importante mencionar que esses resultados avaliam o *pipeline* completo do sistema proposto, incluindo erros ocasionados pela incorreta detecção de mãos e/ou pontas dos dedos.

Tabela 4.2 – Matriz de Confusão da Técnica em Relação a Eventos de Toque.

		Real	
		P	N
Detectado	P	1293	35
	N	14	146

Fonte: Compilado pelo autor.

Mais precisamente, esses erros que levam a uma eventual detecção incorreta do dedo afetaram 96 quadros, ou 6% do total, e alguns deles podem ser observados na Figura 4.3. Tanto na Figura 4.3(a) quanto na Figura 4.3(b), é possível observar que, à medida que o dedo se aproxima da câmera, a qualidade do mapeamento *uv* decai, tornando o dedo maior do que ele é na realidade. Já na Figura 4.3(c), o usuário inclina demais o dedo indicador, de forma que o mesmo fique muito próximo da mão. Acredita-se que esses foram os motivos para a rejeição dos dedos nesses quadros.

Figura 4.3 – Quadros onde não foram detectados dedos. (a-b) Dedo próximo à tela. (c) Dedo com inclinação acentuada em relação à câmera.

(a) (b) (c)

Fonte: Compilado pelo autor.

4.2.2 Detecção de Duplos Cliques

O próximo resultado mostra a acurácia do sistema proposto para a detecção de eventos de duplo clique. Como pode ser observado na tabela 4.3, o sistema foi capaz de detectar 26 deles (84%), com apenas dois falsos positivos, o que corresponde a menos de 1% do total de quadros. A Figura 4.4 ilustra a distância estimada do dedo para o plano ao longo do tempo para um dos vídeos do *dataset*, além de indicar o estado detectado pelo sistema. Nessa sequência de 114 quadros ($\approx$ 13s), o usuário conclui o ato de duplo clique em torno de 1s, 3s, 5s, 8s, 10s e 13s. O sistema é capaz de detectar todos eles, com exceção do duplo clique que ocorre em torno dos 5s.

Tabela 4.3 – Matriz de Confusão da Técnica em Relação a Eventos de Duplo Clique.

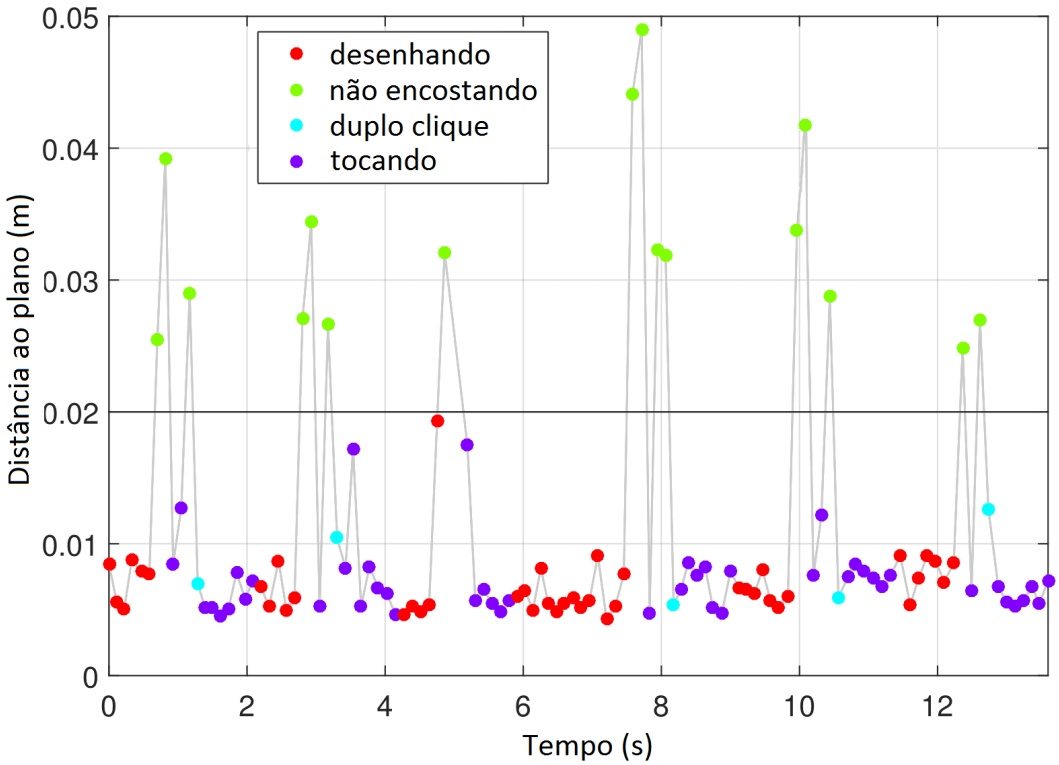
		Real	
		P	N
Detectado	P	26	2
	N	5	1460

Fonte: Compilado pelo autor.

4.2.3 Modo Desenho

Em relação ao modo desenho, foi feita uma avaliação qualitativa em desenhos realizados por cinco usuários (sem experiência prévia com o sistema), todos de pele clara, como pode ser visto na Figura 4.5, em um ambiente controlado. Mais especificamente, foi apresentada a eles uma folha de papel contendo uma forma geométrica (círculo, quadrado e estrela). Em seguida,

Figura 4.4 – Distância da ponta de um dedo ao plano ao longo do tempo e os eventos detectados.



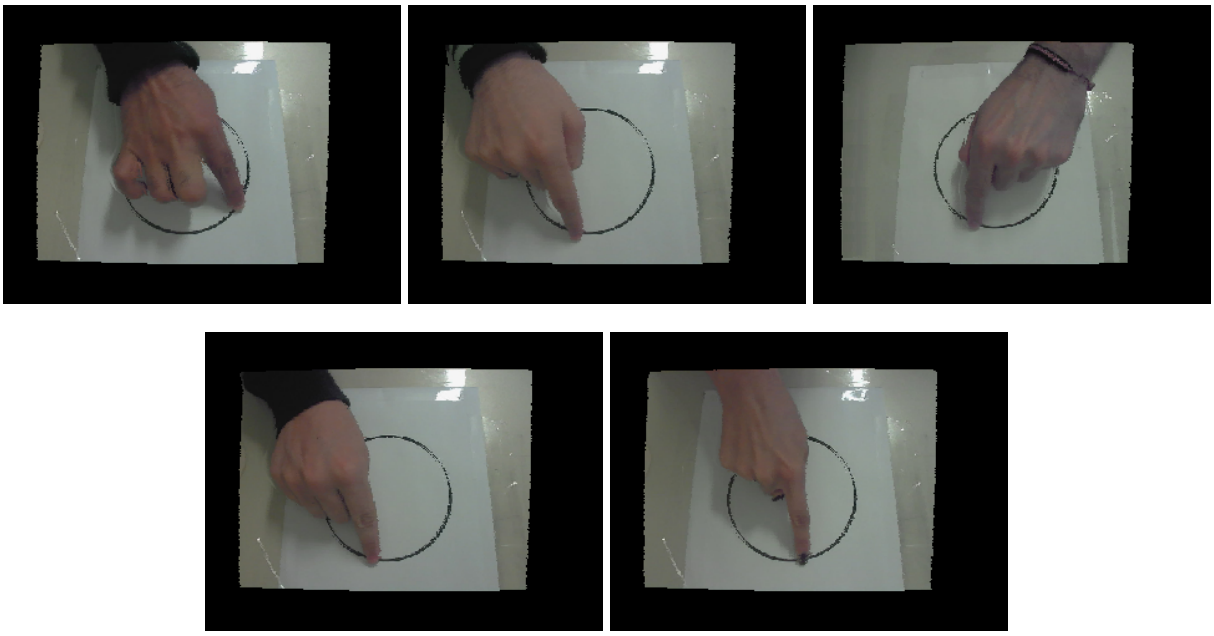
Fonte: Compilado pelo autor.

foi pedido que eles desenhassem acima do contorno do desenho com seu dedo indicador.

Durante a interação, não foi permitido aos usuários o acompanhamento do resultado gerado pela técnica. Desta forma, eles ficaram impossibilitados de corrigir a velocidade ou trajetória dos dedos com o propósito de gerar um desenho mais fidedigno. Os resultados produzidos pelos cinco usuários sobrepostos à forma desejada são mostrados na Figura 4.6. É possível perceber que, em geral, o desenho realizado pelos usuários segue corretamente a trajetória do traçado presente no papel.

Também foi avaliada a acurácia da técnica proposta em relação à localização dos toques durante o modo desenho. Para isso, foi feita a marcação manual da localização verdadeira da ponta do dedo no plano da imagem durante o modo desenho (1002 quadros). Em seguida, foi computado o erro (distância euclidiana) do ponto real de toque para a localização detectada pelo sistema. A média obtida por essa avaliação foi de 8.4 ± 4.4 *pixels* (a mediana foi de 7.6 *pixels*). Dado que foram processadas imagens de tamanho 320×240, o erro médio foi menor do que 4% da menor dimensão da imagem. Além disso, é importante mencionar que a largura média de um dedo tocando o plano nos experimentos realizados foi de 10.9 ± 1.6 *pixels*, de forma que o erro médio de localização da ponta do dedo possui a mesma magnitude da largura da ponta

Figura 4.5 – Exemplo ilustrando a cor de pele dos participantes dos experimentos.



Fonte: Compilado pelo autor.

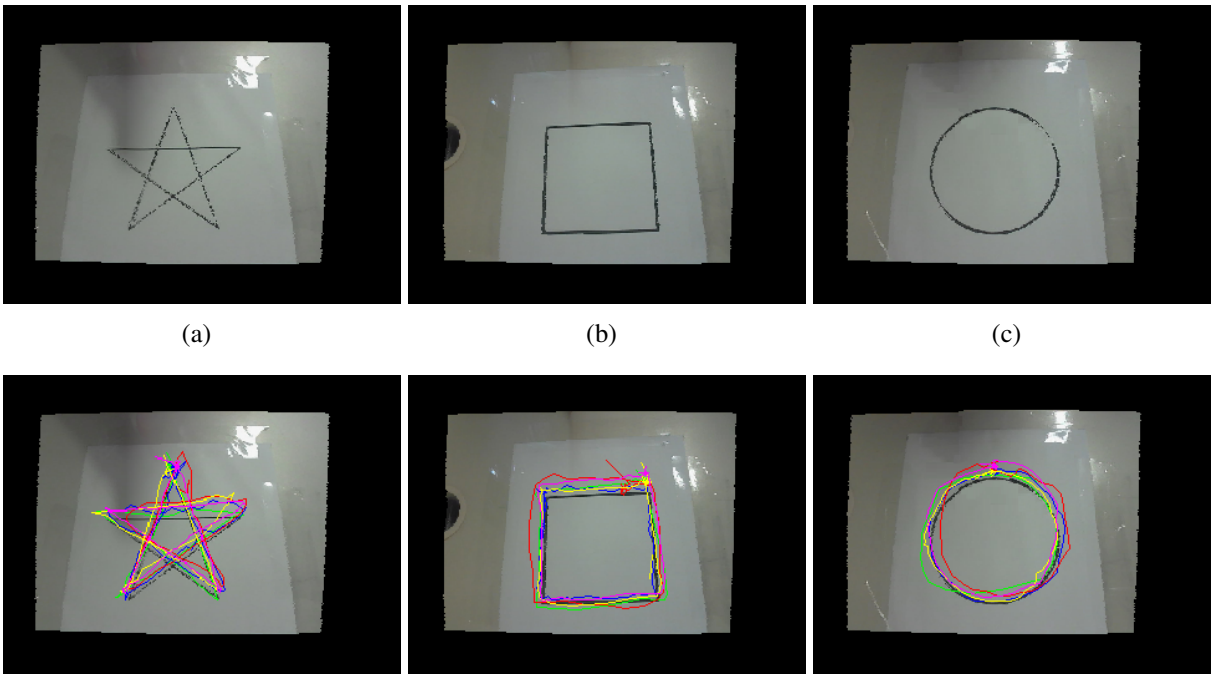
do dedo projetado.

Por fim, a Figura 4.7 apresenta o resultado da interação com o plano utilizando mais de um dedo. Neste desenho, o usuário inicia com os dois dedos posicionados sobre a mesa e os movimenta livremente pela superfície, mantendo a distância fixa entre os dedos.

4.2.4 Modo *Spray*

Na Figura 4.8, é possível visualizar alguns desenhos obtidos a partir do uso do modo *spray*, no qual a altura da ponta do dedo ao plano controla o raio de alcance do *spray*. Nesses resultados, a cor codifica a distância da ponta do dedo ao plano utilizando o mapa de cores "outono", onde o vermelho denota distâncias menores. Como pode-se perceber, marcações mais compactas aparecem com tons avermelhados e mais espalhadas com tons amarelados. Cabe ressaltar que poderia-se oferecer que o usuário escolhesse a cor de cada "spray" em uma interface de desenho, mas o objetivo nessa monografia é apenas ilustrar a relação entre o raio do "spray" e a distância do dedo ao plano.

Figura 4.6 – Desenhos realizados por cinco usuários (em vermelho, verde, azul, amarelo e magenta) sobrepostos ao formato desejado. (a) Estrela. (b) Quadrado. (c) Círculo. (d) Resultado para o desenho da estrela. (e) Resultado para o desenho do quadrado. (f) Resultado para o desenho do círculo.

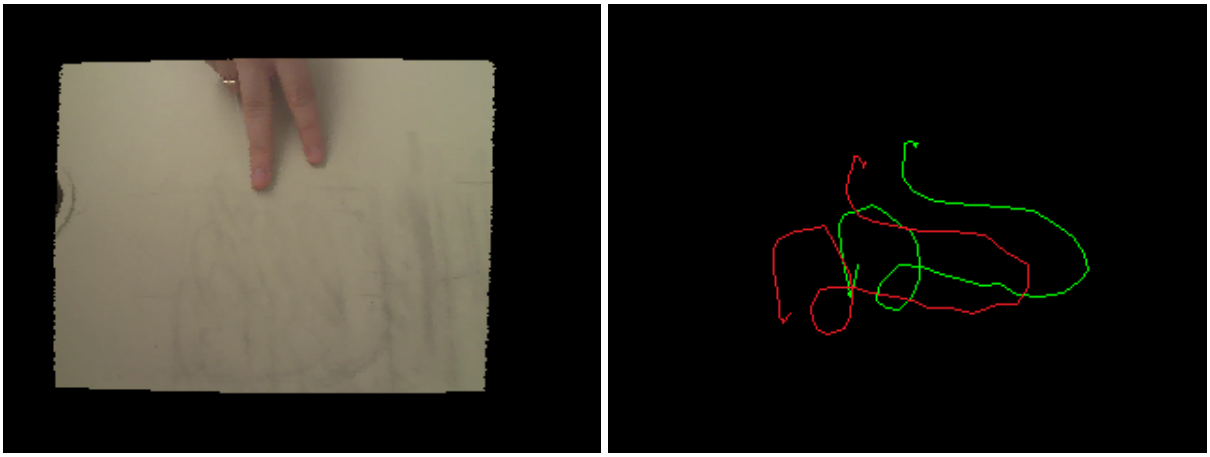


(a) (b) (c)

(d) (e) (f)

Fonte: Compilado pelo autor.

Figura 4.7 – Interação no modo desenho utilizando mais de um dedo. (a) Último quadro da sequência no modo desenho. (b) Resultado utilizando o dedo indicador (linha verde) e dedo médio (linha vermelha).



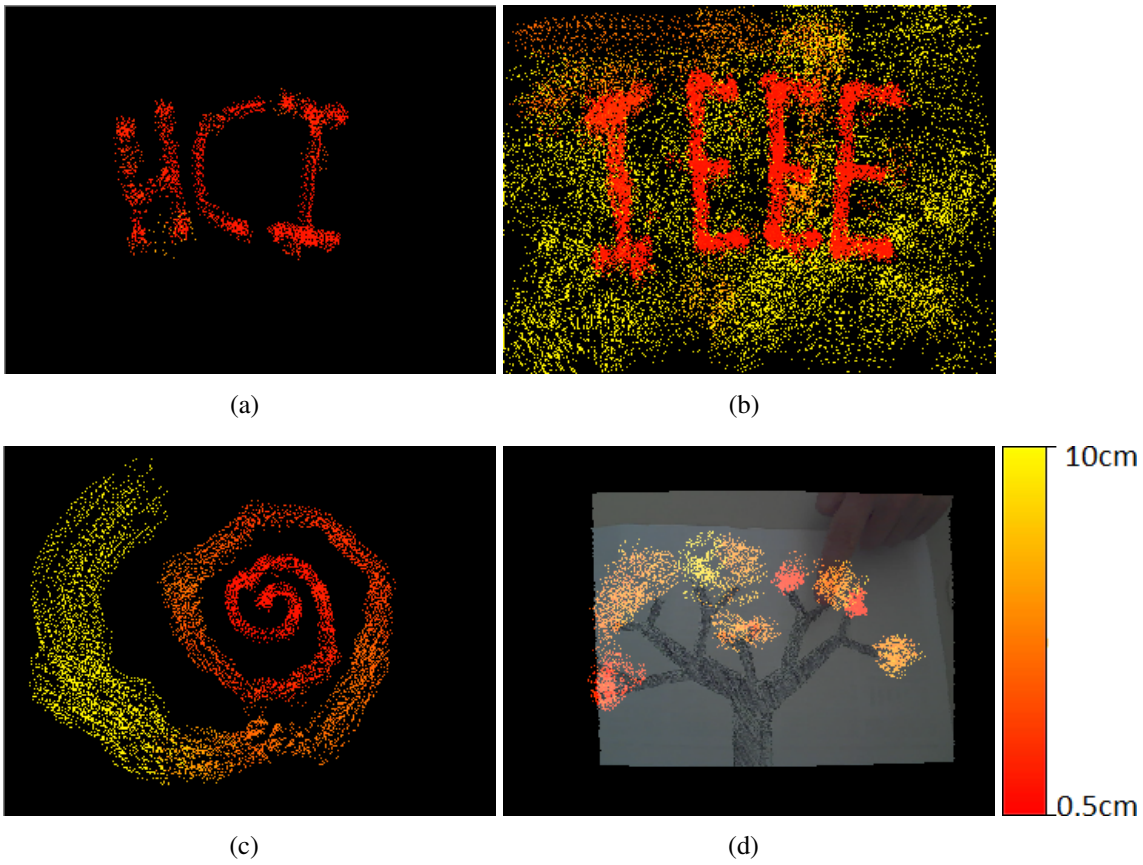
(a) (b)

Fonte: Compilado pelo autor.

4.3 Discussão

Esse capítulo apresentou resultados do uso da técnica a partir do protótipo desenvolvido. A primeira avaliação envolveu o tempo de execução da técnica. Em seguida, foi feita uma

Figura 4.8 – Exemplos de uso do *spray*. (a) Usuário desenha a sigla HCI. (b) Usuário desenha a sigla IEEE. (c) Usuário faz desenho livre. (d) Usuário pinta a copa de uma árvore impressa em um papel posicionado no plano.



(a) (b)

(c) (d)

Fonte: Compilado pelo autor.

análise e comparação da etapa de segmentação da mão e identificação dos dedos com produtos comerciais partindo de uma mesma posição de instalação da câmera. Posteriormente, a acurácia da técnica foi medida, tanto em relação ao erro na localização do toque quanto à taxa de acerto em relação ao evento efetuado pelo usuário. Por fim, avaliou-se aspectos particulares dos modos de desenho e "spray" implementados no protótipo.

Os resultados apresentados nesse capítulo demonstram que a técnica é capaz de gerar a informação necessária para potenciais aplicações de interação com superfícies planares, apesar de não ser executada em tempo real. A acurácia da técnica apresentou bons resultados mesmo sendo utilizada por usuários não treinados, e a incapacidade dos produtos comerciais de operarem neste cenário demonstra uma oportunidade a ser explorada.

5 CONSIDERAÇÕES FINAIS

Essa dissertação apresentou uma técnica para interação com superfícies planares utilizando uma câmera RGB-D em posição descendente. A partir de uma imagem no espaço de cores RGB, as coordenadas 3D de cada ponto e a respectiva confiança de cada um deles de acordo com a câmera, o algoritmo proposto identifica o plano de interação, juntamente com a mão e os dedos do usuário, assim como sua distância ao plano de interação. O trabalho envolveu uma revisão de pesquisas relacionadas e detalhou a abordagem proposta, demonstrando através de um protótipo aspectos qualitativos e quantitativos do algoritmo proposto. Este capítulo está dividido em três seções. A primeira menciona as contribuições geradas pelo método. A segunda aborda as limitações encontradas, bem como discute possíveis soluções. A última sessão propõe uma série de aspectos da técnica que podem ser expandidos em trabalhos futuros.

5.1 Contribuições

Sistemas de Interação Humano-Computador procuram estreitar o relacionamento entre dispositivos eletrônicos e usuários, visando principalmente a agilidade de comunicação e o conforto na forma com que os comandos são transmitidos para os equipamentos. Pesquisas que abordam essa problemática, em geral, não se detiveram no desenvolvimento de técnicas que dessem ao usuário a possibilidade de utilizar objetos do dia a dia (como uma mesa) para transmitir informações ao seu dispositivo ao mesmo tempo que não provocassem fadiga (como a síndrome do braço de gorila).

Procurando suprir essa falta, o trabalho descreveu uma técnica de interação em superfícies planares utilizando somente uma câmera RGB-D. As principais contribuições científicas deste trabalho são a introdução de um novo método de segmentação do plano e objetos acima do plano, a detecção das pontas de dedos baseado em esqueletonização multiescala e um teste simples porém eficiente para detecção de mão baseado em informação de confiança, profundidade e cor. Uma versão simplificada da segmentação do plano de interação e objetos acima dele foi publicada anteriormente (WEBER; JUNG; GELB, 2015).

O método proposto pode detectar eventos de toque e de duplo clique com uma taxa de acerto de 95% e 84%, respectivamente, e a interface de desenho proposta mostra uma potencial aplicação para a técnica, embora testes mais aprofundados sobre a usabilidade do sistema sejam necessários. Em particular, a técnica pode ser utilizada por ferramentas propostas recentemente (LIMPAECHER et al., 2013) para reconhecimento e aperfeiçoamento de desenhos

livres criados pelos usuários. Desta forma, é possível transmitir comandos de uma forma muito semelhante à interação baseada em toque em telas ou *touchpads*, porém utilizando a superfície que lhe for mais confortável. É também dada a possibilidade de se posicionar um desenho qualquer sobre o plano e contornar suas bordas com o dedo, bem como colori-lo virtualmente (utilizando, por exemplo, o modo *spray* apresentado no protótipo).

Em geral, os resultados apresentados mostram que os desenhos realizados por usuários inexperientes se aproximaram do traçado original. A maior parte dos eventos de duplo clique, mesmo ocorrendo em janelas de tempo particularmente pequenas, pôde ser detectada. Além disso, a informação da distância do dedo ao plano mostrou-se suficientemente precisa para a criação do modo *spray*, tornando possível a coloração virtual de desenhos existentes na cena, por exemplo.

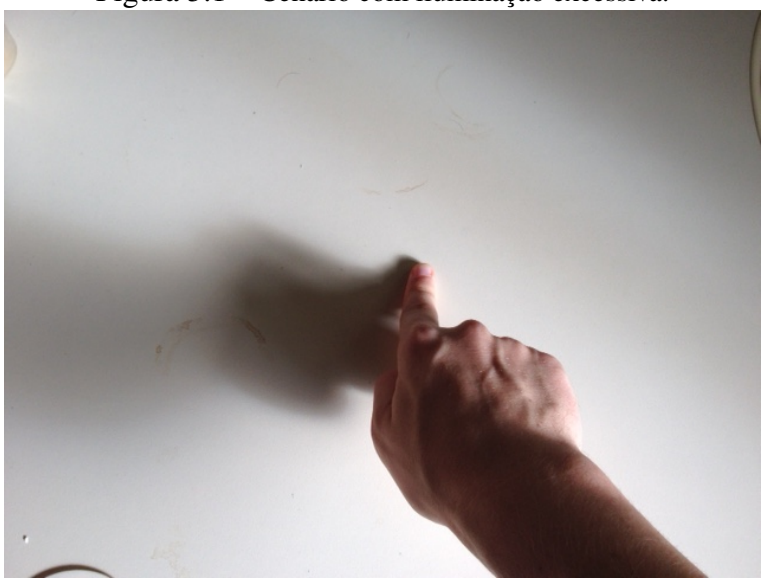
5.2 Limitações

Durante o desenvolvimento da técnica, foram detectadas certas limitações. A principal delas está relacionada ao tempo de execução, o qual impacta diretamente na qualidade da localização das pontas de dedos que participam da interação. Neste caso, uma atualização mais frequente a respeito da localização de um dedo pode tornar o resultado do Filtro de Kalman ainda menos suscetível a dados espúrios. Além disso, acredita-se que haverá uma melhora também na acurácia de detecção dos eventos de duplo clique pelo fato de haver mais informação sobre a distância do dedo ao plano.

Outra limitação encontra-se na intensidade de iluminação do ambiente. Em cenários onde ela é muito excessiva como na Figura 5.1, ocorre uma queda brusca na qualidade dos dados de profundidade gerados pela câmera RGB-D. Tal fato é oriundo da interferência causada no sistema de captura da luz de infravermelho da câmera. Porém, já há trabalhos mostrando ser possível obter a profundidade da cena mesmo em condições extremas de iluminação (O'TOOLE et al., 2015), o que pode eliminar esta restrição.

Além disso, a escolha pelo uso de informação de cor de pele limita o uso da técnica a usuários que não estejam vestindo luvas e não possuam tatuagens na mão, bem como impossibilita que o mesmo segure objetos diversos enquanto interage com o sistema. Finalmente, como uma análise de forma não foi realizada para validação de um objeto segmentado como a mão (apenas cor), objetos com cor de pele podem ser interpretados de forma errônea como uma mão.

Figura 5.1 – Cenário com iluminação excessiva.



Fonte: Compilado pelo autor.

5.3 Trabalhos Futuros

Como proposta para trabalhos futuros, propõe-se inicialmente a implementação em GPU tendo em vista a melhora da performance da técnica. Essa paralelização pode ser feita em todos os trechos de código onde uma das imagens de entrada tem de ser processada *pixel* a *pixel* e sem dependência entre eles, como por exemplo no teste de aceitação do plano do RANSAC, definição dos marcadores e da energia da transformada *watershed*, o gradiente para o processo de obtenção dos esqueletos dos *blobs*, entre outros. Adicionalmente, sugere-se o uso de modelos de mão mais sofisticados, de modo que seja possível não apenas ignorar a busca por cor de pele mas também proporcionar uma variedade ainda maior de interações, como por exemplo uma sequência de gestos acima do plano. Uma variabilidade maior de cores de pele dentre os usuários do protótipo também pode consolidar ainda mais as conclusões obtidas a respeito da acurácia da técnica.

Outro possível melhoramento se refere a uma análise da usabilidade do protótipo. Este estudo pode gerar informações valiosas a respeito das características desejadas pelo usuário, como uma maior taxa de quadros por segundo em detrimento de menor precisão na localização do toque, possibilidade de interação com superfícies não planares ou a necessidade de inclusão de interação com objetos, por exemplo.

REFERÊNCIAS

ALHWARIN, F.; FERREIN, A.; SCHOLL, I. Ir stereo kinect: improving depth images by combining structured light with ir stereo. In: PACIFIC RIM INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 13., 2014, Gold Coast, QLD, Aus. **Proceedings...** New York, US: Springer, 2014. p. 409–421.

BENKO, H.; JOTA, R.; WILSON, A. Miragetable: freehand interaction on a projected augmented reality tabletop. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 30., 2012, Austin, Texas. **Proceedings...** New York, NY: ACM, 2012. p. 199–208.

BERGH, M. Van den; GOOL, L. V. Combining rgb and tof cameras for real-time 3d hand gesture interaction. In: WORKSHOP ON APPLICATIONS OF COMPUTER VISION, 11., 2011, Kona, HI. **Proceedings...** Piscataway, NJ: IEEE, 2011. p. 66–72.

BORING, S.; JURMU, M.; BUTZ, A. Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays. In: ANUAL CONFERENCE OF THE AUSTRALIAN COMPUTER-HUMAN INTERACTION SPECIAL INTEREST GROUP, 21., 2009, Melbourne, Aus. **Proceedings...** New York, US: ACM, 2009. p. 161–168.

CHAI, D.; NGAN, K. N. Face segmentation using skin-color map in videophone applications. **Circuits and Systems for Video Technology, IEEE Transactions on**, Los Alamitos, v. 9, n. 4, p. 551–564, 1999.

CHENG, H.; YANG, L.; LIU, Z. A survey on 3d hand gesture recognition. **Circuits and Systems for Video Technology, IEEE Transactions on**, Los Alamitos, PP, n. 99, p. 1–1, 2015.

DAI, J.; CHUNG, C.-K. Touchscreen everywhere: On transferring a normal planar surface to a touch-sensitive display. **Cybernetics, IEEE Transactions on**, Los Alamitos, USA, v. 44, n. 8, p. 1383–1396, 2014.

DAI, J.; CHUNG, R. Making any planar surface into a touch-sensitive display by a mere projector and camera. In: COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS , 10., 2012, Providence, RI. **Proceedings...** Washington, DC: IEEE, 2012. p. 35–42.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, New York, USA, v. 24, n. 6, p. 381–395, 1981.

HARRISON, C.; BENKO, H.; WILSON, A. D. Omnitouch: wearable multitouch interaction everywhere. In: ANNUAL ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, 24., 2011, Santa Barbara, CA. **Proceedings...** New York, USA: ACM, 2011. p. 441–450.

KANZAWA, Y.; KIMURA, Y.; NAITO, T. Human skin detection by visible and near-infrared imaging. In: CONFERENCE ON MACHINE VISION APPLICATIONS, 12., 2011, Nara, Japan. **Proceedings...** [S.l.]: Citeseer, 2011.

KHOSHELHAM, K.; ELBERINK, S. O. Accuracy and resolution of kinect depth data for indoor mapping applications. **Sensors**, v. 12, n. 2, p. 1437–1454, 2012.

KIM, D. et al. Retrodepth: 3D silhouette sensing for high-precision input on and above physical surfaces. In: ANNUAL ACM CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 32., 2014, Toronto, Canada. **Proceedings...** New York, USA: ACM, 2014. p. 1377–1386.

KJELDSEN, R. et al. Interacting with steerable projected displays. In: INTERNATIONAL CONFERENCE ON AUTOMATIC FACE AND GESTURE RECOGNITION, 5., 2002, Washington, DC. **Proceedings...** Los Alamitos, CA: IEEE, 2002. p. 402–407.

KLOMPMAKER, F.; NEBE, K.; FAST, A. dsensingni: a framework for advanced tangible interaction using a depth camera. In: INTERNATIONAL CONFERENCE ON TANGIBLE, EMBEDDED AND EMBODIED INTERACTION, 6., 2012, Kingston, ON, Canada. **Proceedings...** New York, USA: ACM, 2012. p. 217–224.

KRUEGER, M. W. **Artificial reality II**. [S.l.]: Addison-Wesley Professional, 1991.

KUHN, H. W. The hungarian method for the assignment problem. **Naval research logistics quarterly**, v. 2, n. 1-2, p. 83–97, 1955.

Leap Motion, Inc. **Leap Motion**. 333 Bryant Street Ste. LL150 San Franciso, CA, USA, 2016. Disponível em: <https://developer.leapmotion.com/documentation/cpp/supplements/Leap_Visualizer.html>. Accesso em: 15/03/2016.

Leap Motion, Inc. **Using the Diagnostic Visualizer**. 333 Bryant Street Ste. LL150 San Franciso, CA, USA, 2016. Disponível em: <https://developer.leapmotion.com/documentation/cpp/supplements/Leap_Visualizer.html>. Accesso em: 15/03/2016.

LETESSIER, J.; BÉRARD, F. Visual tracking of bare fingers for interactive surfaces. In: ANNUAL ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, 17., 2004, Santa Fe, NM. **Proceedings...** New York, USA: ACM, 2004. p. 119–122.

LIANG, H.; YUAN, J.; THALMANN, D. Parsing the hand in depth images. **Multimedia, IEEE Transactions on**, IEEE, v. 16, n. 5, p. 1241–1253, 2014.

LIMPAECHER, A. et al. Real-time drawing assistance through crowdsourcing. **ACM Transactions on Graphics (TOG)**, ACM, v. 32, n. 4, p. 54, 2013.

MALIK, S.; LASZLO, J. Visual touchpad: a two-handed gestural input device. In: INTERNATIONAL CONFERENCE ON MULTIMODAL INTERFACES, 6., 2000, State College, PA. **Proceedings...** New York, USA: ACM, 2004. p. 289–296.

MEYER, F. Color image segmentation. In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING AND ITS APPLICATIONS, 4., 1992, Maastricht, Netherlands. **Proceedings...** London, England: IET, 1992. p. 303–306.

NGUYEN, C. V.; IZADI, S.; LOVELL, D. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In: INTERNATIONAL CONFERENCE ON 3D IMAGING, MODELING, PROCESSING, VISUALIZATION AND TRANSMISSION, 2., 2012, Zurich, Switzerland. **Proceedings...** Piscataway, NJ: IEEE, 2012. p. 524–530.

O'TOOLE, M. et al. Homogeneous codes for energy-efficient illumination and imaging. **ACM Transactions on Graphics (TOG)**, New York, NY, v. 34, n. 4, p. 35, 2015.

PISHARADY, P. K.; SAERBECK, M. Recent methods and databases in vision-based hand gesture recognition: A review. **Computer Vision and Image Understanding**, Amsterdam, Netherlands, v. 141, p. 152–165, 2015.

RAUTARAY, S. S.; AGRAWAL, A. Vision based hand gesture recognition for human computer interaction: a survey. **Artificial Intelligence Review**, New York, USA, v. 43, n. 1, p. 1–54, 2015.

SERRA, J. **Image analysis and mathematical morphology, v. 1**. [S.l.]: Academic press, 1982.

Softkinetic. **DepthSense Cameras**. Boulevard de la Plaine 11, 1050 Brussels, Belgium, 2016. Disponível em: <http://www.softkinetic.com/Products/DepthSenseCameras>. Accesso em: 08/03/2016.

Softkinetic. **iiSu Middleware**. Boulevard de la Plaine 11, 1050 Brussels, Belgium, 2016. Disponível em: <http://www.softkinetic.com/Products/iisuMiddleware>. Accesso em: 15/03/2016.

SUAREZ, J.; MURPHY, R. R. Hand gesture recognition with depth images: A review. In: SYMPOSIUM ON ROBOT AND HUMAN INTERACTIVE COMMUNICATION, 21., 2012, Paris, France. **Proceedings...** Los Alamitos, USA: IEEE, 2012. p. 411–417.

SUPANCIC III, J. S. et al. Depth-based hand pose estimation: data, methods, and challenges. In: INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2015, Santiago, Chile. **Proceedings...** Los Alamitos, USA: IEEE, 2015. p. 1868 – 1876.

SWADZBA, A. et al. A comprehensive system for 3d modeling from range images acquired from a 3d tof sensor. In: INTERNATIONAL CONFERENCE ON COMPUTER VISION SYSTEMS, 5., 2007, Bielefeld, Germany. **Proceedings...** [S.l.], 2007.

TELEA, A.; WIJK, J. J. V. An augmented fast marching method for computing skeletons and centerlines. In: SYMPOSIUM ON DATA VISUALISATION, 4, 2002, Barcelona, Spain. **Proceedings...** Aire-la-Ville, Switzerland: Eurographics Association, 2002. p. 251–ff.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: COMPUTER VISION AND PATTERN RECOGNITION, 8., 2001, Kauai, HI. **Proceedings...** Piscataway, USA: IEEE, 2001. p. I–511.

WANG, C.; LIU, Z.; CHAN, S.-C. Superpixel-based hand gesture recognition with kinect depth camera. **Multimedia, IEEE Transactions on**, IEEE, v. 17, n. 1, p. 29–39, 2015.

WEBER, H.; JUNG, C. R.; GELB, D. Hand and object segmentation from rgb-d images for interaction with planar surfaces. In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 22., 2015, Québec City, Canada. **Proceedings...** Piscataway, USA: IEEE, 2015. p. 2984–2988.

WELLNER, P. The digitaldesk calculator: tangible manipulation on a desk top display. In: ANNUAL ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, 4., 1991, Hilton Head, South Carolina. **Proceedings...** New York, USA: ACM, 1991. p. 27–33.

WILSON, A. D. Touchlight: an imaging touch screen and display for gesture-based interaction. In: INTERNATIONAL CONFERENCE ON MULTIMODAL INTERFACES, 6., 2004, State College, PA. **Proceedings...** New York, USA: ACM, 2004. p. 69–76.

WILSON, A. D. Using a depth camera as a touch sensor. In: INTERNATIONAL CONFERENCE ON INTERACTIVE TABLETOPS AND SURFACES, 5., 2010, Saarbrücken, Germany. **Proceedings...** ACM: New York, USA, 2010. p. 69–72.

WREN, C.; IVANOV, Y. Volumetric operations with surface margins. **Computer Vision and Pattern Recognition: Technical Sketches**, 2001.

ZHANG, Z. Microsoft kinect sensor and its effect. **MultiMedia**, Piscataway, USA, v. 19, n. 2, p. 4–10, 2012.