Pós-Graduação em Ciência da Computação

LUCAS SILVA FIGUEIREDO

HAFT:
UMA FERRAMENTA PARA INTERAÇÃO NATURAL

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE
2012

Lucas Silva Figueiredo

HAFT: Uma Ferramenta para Interação Natural

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR(A): **Veronica Teichrieb**

RECIFE
2012

Catalogação na fonte
Bibliotecário Jefferson Luiz Alves Nazareno CRB 4-1758

F475h Figueiredo, Lucas Silva
HAFT: Uma Ferramenta para Interação Natural / Lucas Silva Figueiredo. – 2012.
117 f.: fig., tab.

Orientadora: Veronica Teichrieb
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn. Ciência da Computação, Recife, 2012.
Inclui referências e apêndice.

1. Ciência da computação. 2. Computação gráfica. 3. Interação Natural I. Teichrieb, Veronica (orientadora). II. Título.

004 CDD (22. ed.) UFPE-MEI 2017- 276

Lucas Silva Figueiredo

HaFT: Uma Ferramenta para Interação Natural

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação

Aprovado em: 14/03/2012.

BANCA EXAMINADORA

_____

Prof. Dr. Geber Lisboa Ramalho
Centro de Informática / UFPE

_____

Prof. Dr. Carlos Hitoshi Morimoto
Instituto de Matemática e Estatística / USP

_____

Prof. Dr. Enrique Andrés López Droguett
Departamento de Engenharia de Produção / UFPE

_____

Profa. Dra. Veronica Teichrieb
Centro de Informática / UFPE

Agradecimentos

Todos os familiares que apoiaram e atrapalharam de certa forma o progresso deste trabalho. A vida sem altos e baixos não faz muito sentido e no fim das contas os defeitos e qualidades de cada pessoa são a mesma coisa. Agradeço ao meu pai pelo apoio e eventual aperreio com questões da família, sou grato por fazer parte disso. A caio, vera, ana e clara por motivos similares.

A todos os companheiros de trabalho pelo afeto e companheirismo, que é tão presente mesmo em um ambiente de esforço coletivo, com metas estabelecidas e prazos a cumprir, tem um jeito específico de dar a volta por cima destas situações estressantes. Especialmente a vt, por todo o cuidado e orientação em todo o processo.

A minha companheira de vida, sempre presente quase como uma parte de mim, que quando falta fico consideravelmente perdido, mesmo que ela não saiba disso.

A meus queridos amigos de quem me distanciei tanto nesses anos, as restrições de tempo e a distância física acabam sendo ferramentas torturantes para boas amizades.

Resumo

Nesta dissertação é apresentada uma ferramenta para o rastreamento de mãos e faces, apelidada de HAFT como abreviação para o nome na língua inglesa Hand and Face Tracker. O rastreamento se dá através do gerenciamento de nuvens de pontos da imagem, tendo como destaque um resultado robusto com uma taxa de atualização interativa. Para detectar e seguir mãos e faces, HAFT faz uso de modelos de segmentação de cor de pele, classificadores de Haar, e processos como agrupamento (*labeling*), extração de arestas, operadores morfológicos, análise do fluxo ótico, entre outros algoritmos de visão computacional. Como estudo de caso, a ferramenta de rastreamento proposta é aplicada em um cenário de interação baseada em gestos, dentro do contexto de jogos musicais, através do jogo intitulado Guitars on Air. Além de propor um desafio real no setor de entretenimento, o estudo de caso é apresentado como uma aplicação para avaliação de ferramentas de rastreamento, promovendo uma análise tanto individual, quanto comparativa. Desta forma, HAFT é analisado em paralelo a dois outros métodos de interação por gestos: o primeiro através do uso de luvas de cor laranja, e o segundo através do sensor de profundidade Kinect.

Palavras-chave: Rastreamento de Mãos. Rastreamento de Faces. Nuvem de Features. Interação por Gestos. Interação Natural. Jogos Musicais. Visão Computacional.

Abstract

This dissertation presents a tool designed for hand and face tracking called HAFT. The tracking is performed through the management of clouds of points in the image, achieving a robust result with an interactive *frame* rate. In order to detect and follow hands and faces, HAFT makes use of Bayesian models of skin color segmentation, Haar classifiers, labeling processes, edge extraction, morphological operators, optical-flow analysis, among other computer vision algorithms. As case study, the proposed tracking tool is applied in a gesture interaction scenario, within the context of musical games, by using a developed game called Guitars on Air. In addition to proposing a real challenge in the entertainment industry, the case study is presented as an application for evaluation of tracking tools, providing individual and comparative analysis of such tools. Thereby, HAFT is analyzed along with two other tracking methods for gesture interaction: the first performs the tracking through the detection of orange colored gloves, and the second is the corporal interaction device called Kinect.

Keywords: Hand Tracking. Face Tracking. Flocks of Features. Gestural Interaction. Natural Interaction. Musical Games. Computer Vision.

Lista de Figuras

Sumário

1 INTRODUÇÃO

O presente trabalho disserta sobre o escopo de Interação Natural (VALLI 2005), buscando compreender e avançar no sentido de uma relação mais intuitiva entre seres humanos e máquinas. Mais especificamente, a área abarcada é a de Interação Corporal. Ou seja, no presente trabalho são estudadas interações a partir da leitura do corpo do usuário, com o objetivo de investigar o presente cenário tecnológico e em seguida explorar uma das formas de realizar este tipo de interação através do desenvolvimento, aprimoramento e validação de uma técnica planar de rastreamento de partes do corpo.

1.1 Definição do problema

O rastreamento do corpo humano através de sistemas computacionais é um desafio presente e amplamente discutido no contexto de interação. Este rastreamento pode ocorrer de diversas formas, como através de sensores inerciais anexos ao corpo, ou de acessórios vestíveis contendo dicas visuais como marcadores ou cores marcantes.

No entanto através do uso de dispositivos ou acessórios ligados ao corpo do usuário, este têm sua liberdade restringida de forma leve (como por exemplo com o uso de luvas coloridas) ou intensa (em casos em que o usuário é levado a vestir uma roupa repleta de marcadores). Esta restrição pode inviabilizar alguns cenários de aplicação, como casos em que o usuário não teria tempo para vestir o acessório em questão ou que simplesmente o acessório requerido não estivesse acessível, inviabilizando por exemplo o uso do rastreamento corporal para aplicações Web visto que boa parte dos usuários que a primeira vista teriam acesso ao aplicativo não possuiriam o acessório em questão. Além disso, apesar de existirem iniciativas na literatura em busca de métodos de rastreamento corporal, existe um espaço em aberto para ferramentas de rastreamento de baixo custo (que não fazem uso de sensores de profundidade, por exemplo). Isto visto que os métodos existentes tendem a apresentar lacunas em requisitos essenciais para uma ferramenta de rastreamento que visando um uso abrangente deve superar dificuldades como, por exemplo, um alto tempo de resposta, imprecisão dos resultados e borramento de câmera (motion blur).

1.2 Objetivos

Motivado por esta problemática o presente trabalho visa focar em técnicas de rastreamento que não fazem uso de dispositivos ou acessórios anexos ao corpo do usuário. Mais especificamente, nesta dissertação serão trabalhados os seguintes objetivos:

1. Estudo, desenvolvimento e teste de uma ferramenta de rastreamento chamada HAFT;

2. Estudo, desenvolvimento e teste de um jogo musical chamado Guitars on Air tendo em vista o seu uso como estudo de caso e ferramenta de avaliação para a ferramenta HAFT. Como opção estratégica, neste trabalho o rastreamento corporal se enquadrará especificamente no rastreamento de mãos e faces dos usuários.

Sobre o rastreamento de mãos e faces, é possível classificar os métodos de acordo com a sua dimensionalidade, separando em métodos 2D (ou planares) e métodos 3D. As técnicas planares de rastreamento corporal, de forma geral, retornam menos informações de rastreamento em comparação com técnicas 3D, as quais, por exemplo, recuperam informações de profundidade. Porém em contrapartida as técnicas planares trazem vantagens em relação à carga de processamento requerida, tendo um tempo de execução, de forma geral, muito mais baixo, sendo assim mais propícias a aplicações de interação que requerem uma resposta mais rápida do sistema em relação aos gestos corporais do usuário. Desta forma, neste trabalho foi explorado o rastreamento planar como alternativa para aplicações que se propõem a fornecer uma interação fluida através de gestos corporais.

Mais especificamente, a técnica de rastreamento implementada e aprimorada neste trabalho se propõe a rastrear simultaneamente mãos e faces através do gerenciamento de nuvens de pontos. Este tipo de rastreamento faz uso de uma câmera monocular comum como único dispositivo de captura, visando alcançar uma vasta aplicabilidade do trabalho desenvolvido dado que estes dispositivos estão amplamente presentes no contexto atual, acoplados a laptops e celulares, por exemplo, ou acessíveis em separado por um baixo custo.

Mais além, ainda como motivação para uma investigação mais aprofundada sobre formas de rastreamento baseadas em nuvens de pontos é possível ressaltar características como baixo tempo de processamento e robustez a uma série de dificuldades presentes no rastreamento corporal como sensibilidade a movimentos rápidos e a diferentes posicionamentos relativos da mão por exemplo. Estas características tendem a afetar menos os métodos baseados em nuvens de pontos devido ao tratamento em aspecto micro que estes métodos empenham ao seguir determinado alvo durante a interação. Desta forma, a ferramenta HAFT se apoia neste tipo de rastreamento vislumbrando o uso deste tipo de técnica como uma saída para a obtenção de resultados de qualidade que possam ser repassados para uma aplicação que vise à interação do usuário através da sua face ou das mãos. Assim, a ferramenta proposta visa explorar este tipo de rastreamento buscando extrapolar os resultados obtidos previamente em outros métodos da literatura que fazem uso do conceito de nuvem de pontos. Desta forma, o objetivo base deste

trabalho é a ferramenta em si, considerando em paralelo todos os aspectos atrelados a esta, e visando a ferramenta como uma alternativa de baixo custo e alto desempenho para o rastreamento de mãos e faces voltado para aplicações que promovem uma interação natural.

Como mencionado, a implementação descrita nesta dissertação constitui uma ferramenta chamada HAFT (como abreviação do título na língua inglesa "*Hand And Face Tracker*"). Os principais requerimentos da ferramenta em questão são os de fornecer o rastreamento simultâneo de vários alvos (mãos e faces) em tempo real, visando alcançar robustez a movimentos rápidos, suporte a posições e configurações distintas das mãos e faces, bem como a oclusões parciais e sobreposições dos alvos rastreados (por exemplo, mãos por sobre a face). A quantidade de alvos rastreada é arbitrária, de forma que é suportado o rastreamento das mãos e faces de vários usuários simultaneamente, desde que exista espaço visual do ponto de vista da câmera utilizada. De forma geral, o intuito é que a ferramenta proposta possa ser aplicada em um cenário real de interação, dando suporte a aplicações que fazem uso de interações por gestos. Abaixo estão listados os principais critérios de referência:

- Baixo Custo. Através do uso de uma câmera monocular como único dispositivo de captura a ferramenta proposta pode ser utilizada com baixo custo financeiro associado.

- Acesso. Além do baixo custo associado, as câmeras monoculares hoje estão presentes em diversos dispositivos como laptops e smartphones de forma que a técnica utilizada se torna acessível em diversos cenários já em seu primeiro momento ou após a realização de um porte para plataformas móveis por exemplo.

- Execução em Tempo Real. A técnica de rastreamento utilizada em HAFT tem como objetivo fornecer respostas em relação ao posicionamento de diversos alvos (mãos e faces) em tempo real (24 quadros por segundo).

- Movimentos Rápidos. De forma geral, as técnicas presentes na literatura tendem a apresentar problemas durante o rastreamento de movimentos rápidos do usuário, o que prejudica a interação limitando o usuário gerando uma preocupação extra deste para com o sistema. A ferramenta proposta tem como objetivo tratar este problema através do gerenciamento de nuvens de pontos.

- Oclusão Parcial. Através do uso de nuvens de pontos a técnica de rastreamento utilizada é capaz de tratar casos de oclusão parcial, permitindo que o rastreamento continue ininterruptamente após estas ocorrências.

- Detecção. A ferramenta proposta tem como meta fornecer respostas sobre os alvos rastreados na cena de forma dinâmica, permitindo que usuários entrem e saiam de cena quando desejado e para isso está acoplada a técnicas de detecção de mãos e faces integradas ao rastreamento.

Os critérios apresentados fazem parte de uma meta maior que trata de fornecer uma ferramenta usável em cenários reais para interação natural. Desta forma, a construção da ferramenta HAFT é guiada através dos requisitos citados, buscando maximizar este conjunto de exigências como um todo, sem que determinada característica seja levada em conta isoladamente em detrimento das demais. Assim, por exemplo, o conceito de execução em tempo real por vezes é levado com menos rigor em casos que por exemplo é necessário executar um algoritmo para detecção de faces, o qual pode levar mais do que 100 milissegundos (atingindo no máximo 10 quadros por segundo), desta forma, mesmo infringindo o conceito de tempo real durante a detecção, em uma aspecto mais geral a ferramenta se torna mais completa abrangendo mais cenários de uso.

Além de uma série de testes isolados, como estudo de caso, o rastreamento realizado por HAFT é aplicada em um cenário de interação dentro do contexto de jogos musicais, através do jogo intitulado Guitars on Air. Tendo em vista que a técnica de rastreamento utilizada é aplicável para mãos e faces de forma similar, e que durante testes preliminares o rastreamento de faces se mostrou, de forma geral, consideravelmente mais estável, o estudo de caso proposto se propõe a avaliar a ferramenta especificamente em relação ao rastreamento de mãos, com o objetivo de entender as principais dificuldades da técnica proposta em um cenário de uso que inclui fatores complicadores como movimentos rápidos, mudanças de gestos, oclusões parciais e totais, etc. Desta forma, entendendo as dificuldades da forma de rastreamento adotada em um cenário de uso mais complexo, é possível tratar estas dificuldades com foco nas principais necessidades da ferramenta, visando um futuro aprimoramento que afeta mutuamente o rastreamento de mãos e faces.

Além de propor um desafio relacionado a um cenário real no setor de entretenimento, o estudo de caso é apresentado como uma aplicação com foco para a avaliação de ferramentas de rastreamento de mãos, promovendo uma análise tanto individual, quanto comparativa das ferramentas utilizadas. A proposta de uso do Guitars on Air como ferramenta de avaliação é associada principalmente às mecânicas de jogo presentes nos jogos musicais. Através do conceito de air guitar (prática em que uma pessoa simula a performance de um guitarrista fingindo vestir uma guitarra imaginária) associado às mecânicas de jogos musicais o jogo Guitars on Air oferece uma proposta de interação que requer da ferramenta de rastreamento de mãos utilizada um baixo tempo de resposta, bem como um nível de precisão configurável e robustez à movimentos rápidos. Mais além, através de métricas associadas às mecânicas do jogo proposto é possível mesurar as capacidades da ferramenta de rastreamento de mãos usada. Com o intuito de realizar uma análise comparativa, HAFT é testado em paralelo a dois outros métodos de interação por gestos: o primeiro através o uso de luvas de cor laranja, e o segundo através do dispositivo de interação corporal Kinect. O principal objetivo nesta comparação é entender as principais dificuldades da ferramenta HAFT para sua aplicação livre em cenários reais de interação; visto que as ferramentas utilizadas como base comparativa representam formas de rastreamento validadas (de forma bem sucedida) em cenários semelhantes ao proposto o principal intuito é capturar as

lacunas presentes na ferramenta proposta. Os principais critérios de referência associados ao estudo de caso estão listados abaixo:

- Aplicação Real. Simular uma aplicação real de interação por gestos em que o usuário possui certa liberdade manipular o sistema, estando livre para explorar o ambiente e o sistema da melhor forma que encontrar tendo como única meta executar uma tarefa especificada.
- Interface Intuitiva. Em adição o sistema (Guitars on Air) deve ser claro ao comunicar para o usuário o que se passa no decorrer da tarefa.
- Ferramenta de Avaliação. O estudo de caso proposto tem como meta fornecer métricas associadas à características do rastreamento como precisão, tempo de resposta e robustez à movimentos rápidos, fornecendo dados sobre a experiência do usuário para análise futura associando à performance do usuário à qualidade do rastreamento de um ponto de vista individual (analisando uma única ferramenta de rastreamento) e comparativo (ao analisar o comportamento de duas ou mais ferramentas de rastreamento em paralelo).

1.3 Organização do Documento

A organização do presente texto segue a seguinte ordem: no Capítulo 2, é explanado o conceito de Interação Natural, que permeia todo o trabalho realizado; o Capítulo 3 se trata de uma revisão da literatura dentre os principais métodos de rastreamento do corpo humano; o Capítulo 4 fornece todo o detalhamento da técnica de rastreamento desenvolvida, explicitando o passo-a-passo executado para que se obtenha o resultado final; no Capítulo 5 são explicitadas as motivações e implementações realizadas para a concretização da ferramenta de avaliação Guitars on Air, bem como o detalhamento da realização dos testes com usuários; por fim, o Capítulo 6 apresenta as conclusões chave do trabalho bem como as sugestões de trabalhos futuros.

2

PARADIGMA DE INTERAÇÃO NATURAL

Desde que a tecnologia faz parte da vida humana, há uma preocupação inerente ao seu uso. O desenvolvimento de novas ferramentas está atraledo, mesmo que inconscientemente, ao conceito de usabilidade.

Mesmo os artefatos mais antigos, como lanças e machadinhas, eram moldados, por exemplo, para evitar ferimentos durante o uso. Cada ferramenta, ao se mostrar promissora no auxílio de determinada tarefa, de uma forma análoga à seleção natural que agiu sobre as espécies de seres vivos, passava por aprimoramentos, deixando para trás experimentações malsucedidas (CHARYTONOWICZ 2000.

2.1 Complexidade x Tempo

No entanto, com o progresso experimentado pela humanidade, é percebido como regra geral que a complexidade de uso das tecnologias desenvolvidas aumenta de forma significativa. De forma geral, as ferramentas presentes na pré-história eram intuitivas, com um baixo tempo de treinamento associado. Lanças, machadinhas, ou cinzéis (Figura 2.1), possuíam aparência simples e uma forma de uso assimilável e reproduzível apenas pela observação, a qual constituía em segurar e mover a ferramenta em determinada direção.

Figura 2.1: Cinzéis armazenados no Museu Histórico Nacional. Estas ferramentas possuíam extremidade afiada, como uma lâmina curta, para partir outras estruturas (PROUS *et al.* 2002).

Figura 2.2: Cabine do avião norte-americano “Lockheed U-2”, conhecido como “Dragon Lady” (CHARYTONOWICZ 2000).

Ao longo dos séculos, as tecnologias passaram a assumir tarefas mais elaboradas, e como consequência, em alguns casos as formas de interação entre humanos e máquinas passaram a requerer mais treinamento, até o ponto em que se tornaram necessários anos da vida do usuário a fim de usufruir determinado equipamento. Abaixo, a Figura 2.2 ilustra a cabine (cockpit) de um avião norte-americano fabricado durante a década de 1960, apresentando notável complexidade de operação. Neste caso, por exemplo, além da formação escolar requerida, um piloto de combate da Força Aérea dos Estados Unidos da América (USAF) passa por treinamento em um curso de graduação com duração de 54 semanas (BASEOPS 2012).

O surgimento de computadores também está sujeito à regra de uma alta complexidade de uso. Os primeiros modelos sequer possuíam interfaces gráficas, e mesmo com o surgimento dos Computadores Pessoais (PCs), acompanhados de monitores, o alto grau de complexidade permaneceu evidente. Entende-se este fato como a consequência de que, mesmo havendo indícios de esforços na busca pelo conceito de usabilidade (MYERS 1998), não existiam estudos extensivos, pois a área de Interação Humano-Computador (IHC) ainda estava por surgir.

2.2 Prelúdio sobre IHC

Em 1960, a USAF (Força Aérea Norte-Americana) desenvolveu um sistema para alertar pilotos sobre condições perigosas. O sistema usava mensagens de voz gravadas para sugerir comandos como girar, subir, evitar colisões eminentes, dentre outros. Durante estudos, foi percebido que ao usar uma voz feminina para emitir os comandos, a taxa de erro dos pilotos diminuía consideravelmente, além do tempo de resposta se tornar menor.

Este resultado foi atribuído principalmente ao fato da maioria dos pilotos na época serem

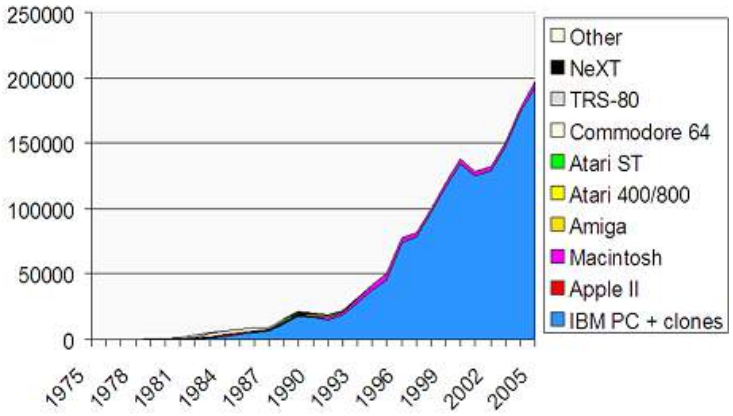


Figura 2.3: Número de vendas anuais de diversos modelos de PCs (em milhares de unidades) entre 1975 e 2005 (REIMER 2009).

homens, o que fazia com que dedicassem um maior nível de atenção à voz feminina. Esta pesquisa deu origem ao termo "fatores humanos", ressaltando a importância do entendimento do usuário ao interagir com um determinado sistema. Todavia, apesar desta constatação, este conceito permaneceu adormecido por mais de duas décadas (HARRISON 2007).

Em 1980, após empresas como a Intel, a Apple e a IBM (entre outras) haverem lançado os primeiros PCs, houve uma mudança significativa em relação ao uso de computadores de uma forma geral. A Figura 2.3 ilustra bem a popularização dos PCs ao longo dos anos. Estes passaram a estar presentes nos mais diversos setores, sendo usados, por exemplo, como simuladores na indústria, ferramentas de escritório e fontes de entretenimento dentro de domicílios (KANELLOS 2002; BBC 2002). Foi neste contexto que surgiu a área de Interação Humano-Computador (IHC). A demanda de uso daqueles equipamentos aflorou a necessidade de novas pesquisas com foco nos usuários e em como estes percebiam e reagiam às máquinas. Iniciaram-se estudos voltados a interfaces gráficas mais intuitivas, dispositivos de entrada que permitiam uma interação mais rápida e precisa, entre outros tópicos. Somente então o conceito de "fatores humanos" recebeu o devido respaldo, dando nome a uma das principais conferências de IHC existentes: Conference on Human Factors in Computing Systems, conhecida como CHI (HARRISON *et al.* 2007).

2.3 Cenário Tecnológico

Atualmente há um contexto semelhante ao apresentado na década de 1980, porém em proporções consideravelmente maiores. Um indício desta constatação é o massivo acesso a dispositivos portáteis observado nos últimos anos (incluindo *smartphones, tablets, laptops,* etc.), que está mapeado no gráfico da Figura 2.4.

Ademais, o público alvo destas novas tecnologias é muito mais abrangente do que os antigos usuários de PCs; restrições de idade, de condições financeiras, e até deficiências físicas e cognitivas são fatores que influenciam cada vez menos no momento de aquisição destes dispositivos (TRENDS 2011; SVOBODA *et al.* 2009). Novas tecnologias surgem dia após

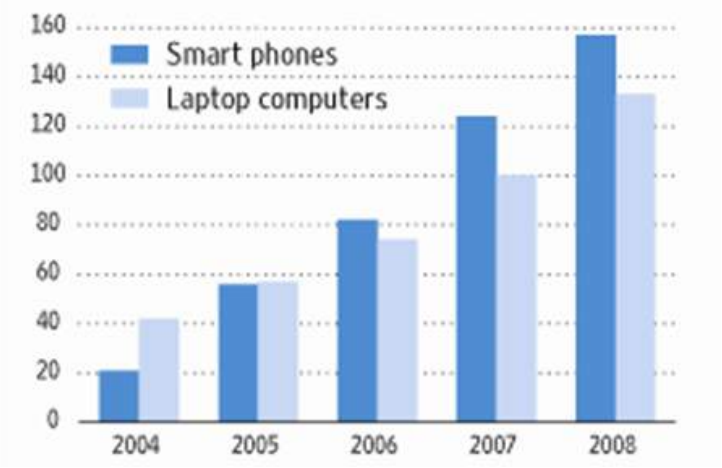


Figura 2.4: Vendas anuais de laptops e smartphones. É importante ressaltar que os números na coluna do gráfico representam milhões de unidades vendidas (WINGFIELD 2009).

dia, ocupando novos espaços, transformando em realidade o conceito de Computação Ubíqua, que aponta a tecnologia computacional como uma presença em todo o ambiente experimentado pelo ser humano, de forma natural, fazendo parte de cada aspecto da vida humana (CHEN *et al.* 2006). Os avanços na miniaturização de dispositivos, aliados ao surgimento de ferramentas para a comunicação sem fio, dispositivos *plug-and-play*, processadores portáteis e novas tecnologias sensitivas abriram as portas para pesquisas sobre novas formas de interação. De fato, ao cercar usuários com tecnologias sensitivas, as quais se encontram presentes no ambiente de forma geral, é possível explorar a interação de forma mais invisível, de modo que o usuário possa ser lido através dos sensores e assim suas intenções possam ser interpretadas entendendo indicativos como movimentos corporais, comunicação vocal e expressões faciais.

Neste sentido surge a ideia de se investir no conceito de uma interação mais invisível, que abstrai para o usuário a forma como os comandos são interpretados. Todavia, é ineficaz enquadrar essa nova necessidade de pesquisa como uma demanda de investimentos em IHC. Ao longo das últimas três décadas, IHC evoluiu, se expandiu e se diversificou abrangendo hoje tópicos que não eram cogitados àquele primeiro momento. Estudos de IHC que antes se resumiam à Ciência da Computação hoje abrangem áreas de conhecimento como Psicologia, Design e Comunicação. Da mesma forma, a gama de aplicações estudadas que antes se restringia a ferramentas para atividades de escritório (por exemplo: editores de planilhas e de textos), se expandiu para diversos outros campos como jogos digitais, *e-commerce*, etc. Outros aspectos como os tipos de interfaces e os estudos sobre usuários se multiplicaram de forma que a área de IHC, hoje, é mantida por uma comunidade vasta e multi-facetada, unida por um fraco vínculo em torno da ideia de usabilidade (CARROLL 2009). Assim, dentro do cenário apresentado, a área de IHC tende a se expandir ainda mais, no entanto, dentro desta, é possível definir um escopo mais preciso.

2.4 Interação Natural

O termo Interação Natural nasce em resposta a este momento tecnológico. Dentro de um contexto de Computação Ubíqua, a interação deve ocorrer de forma invisível, pois dada a presença de sistemas em todo o ambiente, é inviável exigir dos usuários um aprendizado específico para cada interface. Desta forma, é sugerida uma inversão de papéis, na qual ao invés do usuário interpretar o sistema em questão, este último passa a interpretar o usuário, percebendo suas intenções e usando-as para alcançar uma interação bem sucedida. Este, aliado a outros conceitos como liberdade, uso de metáforas e feedback em tempo real, dão origem ao campo de pesquisa chamado Interação Natural (VALLI 2005).

Formalmente, tem-se que uma interação natural é definida em termos de experiência humana: as pessoas se comunicam naturalmente através de gestos, expressões faciais, movimentos corporais e da fala, da mesma forma que percebem o mundo olhando em volta, ouvindo e manipulando materiais físicos.

Desta forma, o pressuposto fundamental é que os usuários devem interagir com a tecnologia da mesma forma como eles interagem com o mundo real na vida cotidiana, como a evolução genética e a educação os instruiu (VALLI 2005). Em suma, Interação Natural abre um canal de comunicação direto entre o usuário e o mundo digital, pondo a parte mediadores, como por exemplo *joysticks*, *mouses*, teclados e sensores vestíveis como luvas hápticas (*Haptic Gloves*) e HMDs (*Head Mounted Displays*).

2.4.1 Herança Genética e Cultural

São os milhares de anos de evolução humana e os primeiros anos de vida de cada pessoa que definem quais atividades e interações são naturais e quais não são. Apesar de subjetiva, esta definição justifica por que é natural manipular esferas coloridas, e por que não o é usar um teclado de computador, por exemplo (VALLI 2005).

Em uma primeira perspectiva, atividades naturais são aquelas para as quais os seres humanos estão adaptados, implicitamente escritas nas estruturas da mente e do corpo. Portanto, as interações simples, realizadas por mulheres e homens pré-históricos que perduraram pelos tempos, podem ser consideradas como atribuições padrão para o organismo humano. A partir desta perspectiva, atividades como a manipulação de objetos leves, o ato de levar alimentos à boca com as mãos, fazer rabiscos e mesmo caminhar podem ser consideradas naturais.

Em adição, atividades que têm uma origem cultural, são consideradas como ancestrais (ex.: gestos dêiticos e alguns símbolos simples). Os primeiros anos de vida de cada ser humano são responsáveis pela mais intensa fase de desenvolvimento do cérebro. O número de sinapses efetivadas até os três primeiros anos de idade gira em torno de mil trilhões (Figura 2.5). Apesar da complexidade das conexões aumentar com o passar dos anos, um ser humano adulto tem cerca de metade destas sinapses, pois a partir dos onze anos de idade, o cérebro da criança passa

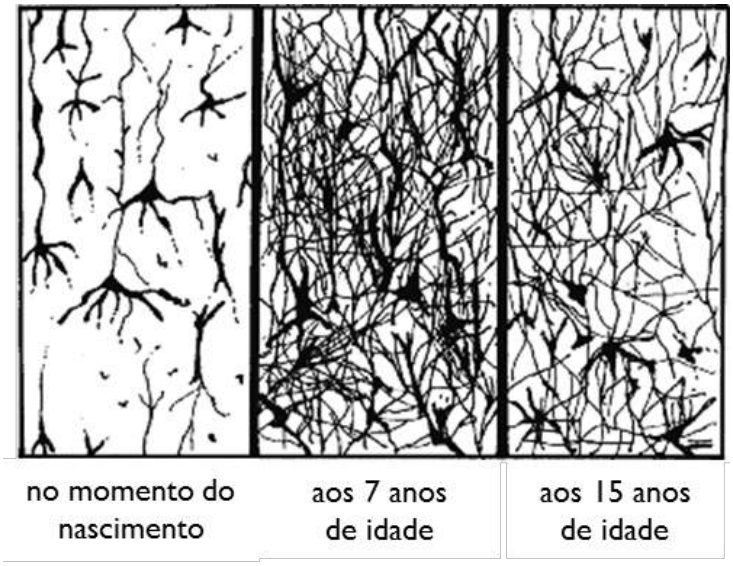


Figura 2.5: Representação da densidade de sinapses em um cérebro humano em três diferentes fases de desenvolvimento (BROTHERSON 2005).

por um processo de "poda" para se livrar de conexões extras, não-usadas. Por outro lado, após uma sinapse ser formada, a mesma pode ser reforçada caso o estímulo que a gerou seja repetido, e este reforço pode ocorrer até o ponto de torna-la permanente. Desta forma, atividades como a comunicação verbal, gestos, expressões faciais, são assimiladas e executadas sem a necessidade de um esforço consciente significativo do cérebro (BROTHERSON 2005).

Ambos os aspectos abrem margem para distintas definições de Interação Natural devido à multiplicidade de heranças genéticas e culturais. Povos que se mantiveram consideravelmente isolados, por exemplo, acabam por carregar uma carga genética que os distingue, mesmo que minimamente, de outros. O mesmo ocorre em relação à herança cultural. De fato, ao se tratar da forma como se passam os primeiros anos de vida de um ser humano, existem inúmeras possibilidades de diferentes influências e estímulos. Desta forma, considerando os estudos em Interação Natural, ao se focar em um grupo específico, é relevante que sejam compreendidas as influências exercidas nas crianças daquele grupo; no entanto em uma visão prática mais geral, em um cenário globalizado, é factível determinar interações genéricas.

2.4.2 Carga Cognitiva

A memória do ser humano é dividida em três áreas:

1. Memória sensorial;
2. Memória de trabalho ou de curto prazo;
3. Memória de longo prazo.

O propósito da memória de trabalho é o de raciocinar e aprender. Informações são copiadas da memória sensorial ou da memória de longo prazo para a memória de trabalho com

o objetivo de serem processadas em seguida. No entanto, a memória de trabalho é limitada; em adultos a capacidade de armazenamento gira em torno de sete elementos, variando entre cinco e nove a depender da pessoa (VALLI 2005). Denomina-se carga cognitiva a quantidade de elementos presentes simultaneamente na memória de trabalho (MILLER 1994).

Entende-se por elemento cada componente transferido para a memória de trabalho. Estes podem ser referências para imagens, sons, letras, palavras, equações, etc. O seu conceito é volátil, dependente da experiência de cada pessoa. Elementos solitários são agrupados baseando-se no seu significado ou em características definidas em um momento passado, dando origem aos esquemas. Assim, esquemas são arranjos alocados na memória de longo prazo, formados a partir de associações feitas na memória de trabalho (SWELLER 1988).

Uma vez que há um estímulo, o cérebro procura por esquemas, e inicia aquele que é menos custoso em termos de esforço. Caso um sistema induza esquemas simples, a interação se torna mais direta e menos cansativa; quão maior for o nível de abstração, maior será o esforço cognitivo. Logo, a interação será natural caso estimule esquemas consolidados devido ao uso, comuns à humanidade de forma geral, liberando a carga cognitiva para que seja ocupada pelos desafios da aplicação em si (VALLI 2005).

2.4.3 Liberdade

Mais além, a interação não deve ser mecânica e tampouco as pessoas, ao usarem o sistema, devem se sentir parte de um mecanismo. O conjunto de regras que atua sobre a interação deve se aproximar das regras que atuam no mundo real, replicando limitações e liberdades conhecidas e implicitamente aceitas. Desta forma, por exemplo, em um ambiente virtual onde o usuário controla a locomoção de um avatar, ter uma placa informando "pare" é menos natural do que uma parede virtual grande o suficiente para que não seja atravessada (VALLI 2005).

Da mesma forma, em um cenário ideal, o uso de dispositivos de entrada com fios, botões ou anexados ao usuário é desencorajado. Este não deve se sentir preso ou dependente de equipamentos e ser descreditado de suas intenções (transmitidas através da fala, de gestos, ou de qualquer outro meio) ao perceber que seus comandos são reconhecidos somente por meio de um canal específico e limitado. Em um cenário ideal de interação, inclusive o uso de câmeras, ou de outros sensores à distância, deve ser ocultado levando o usuário a estar concentrado na tarefa que deve executar e não na forma que deve interagir; isto evitará preocupações como para qual direção tem que estar voltado, qual é o limite de movimento que pode fazer para não sair do alcance da câmera, etc. Todas estas concessões devem transmitir uma sensação positiva de liberdade para o usuário, além de diminuir a carga cognitiva, pois a interação será mais direta a partir do momento em que o usuário não precise assimilar cada mecânica da interface (VALLI 2005).

2.4.4 Movimento

Seres humanos associam movimento à vida; um objeto ou uma cena em movimento é capaz de atrair a atenção, inclusive por um longo período de tempo, em comparação com uma cena estática ou repetitiva. De forma geral, movimento é sinal de que algo está se modificando, que uma novidade está por vir; por isso se torna um elemento importante se tratando de interação.

Em sistemas computacionais, o conceito de movimento pode ser usado de duas formas opostas, como descrito em (VALLI 2005):

1. Por um lado, o sistema é capaz de atrair a atenção dos usuários ao mover objetos visíveis;

2. Por outro lado, pode ser implantando um reconhecimento de movimentos voluntários do usuário, os quais foram executados com o propósito de atrair a atenção do sistema.

O presente trabalho visa contribuir para permitir interações mais naturais, tomando como base cada uma destas vertentes, a fim de explorar mais a fundo este novo conceito de interação e entender o seu potencial para o contexto tecnológico atual e futuro. Primeiramente, tendo como foco o desenvolvimento de uma técnica de rastreamento, para o uso de movimentos como entrada na interação entre seres humanos e máquinas, ou seja, o escopo trabalhado dentro de Interação Natural neste trabalho está relacionado à interação através de gestos corporais. Em seguida com o desenvolvimento de uma ferramenta para avaliação de técnicas de rastreamento de mãos, conectando os conceitos de movimentos visualizados e produzidos pelo usuário. Além disso, tendo sempre em vista os conceitos apresentados de Carga Cognitiva, Liberdade e Movimento com o objetivo final de fornecer a desenvolvedores um método sólido para interações naturais.

3

RASTREAMENTO DO CORPO HUMANO

O conceito de Interação Natural propõe interfaces que entendem as intenções do usuário de forma que este transmite suas intenções intuitivamente, interagindo com sistemas computacionais da mesma forma como o faz no dia-a-dia com pessoas e objetos. Este conceito permite abordar todas as formas de comunicação que um ser-humano é capaz de executar, desde a fala, toque, gestos, expressões faciais, olhares entre outras. O mesmo se aplica aos sentidos: visão, audição, tato, etc.

Idealmente, estes tópicos deveriam ser tratados de forma integrada, combinando os dados para gerar soluções completas. No entanto, os atuais avanços em tecnologias sensitivas e técnicas de processamento de sinais ainda não permitem uma abordagem com este nível de abrangência. Uma das relevantes linhas de pesquisa estudadas na área de Interação Natural é a interpretação de movimentos corporais. A linguagem corporal se enquadra como uma das mais importantes formas de comunicação. O famoso estudo de Albert Mehrabian chamado "Silent Messages" (MEHRABIAN 1971) aponta a linguagem não verbal como uma forma de comunicação significativa, dado o objetivo de transmitir uma mensagem. Mais especificamente, o estudo trata da comunicação entre dois indivíduos, e analisa as três formas de expressão do emissor: verbal, vocal e visual. Como conclusão o experimento revela que, em relação ao grau de aceitação de determinada mensagem, cada aspecto influencia com intensidade diferente, cabendo à linguagem corporal (componente visual) 55% de relevância (Figura 3.1).

Ao contrário do raciocínio que pode ser induzido, o experimento não sugere que o aspecto visual seja usado isoladamente, em detrimento dos demais; para que uma mensagem seja comunicada com sucesso, o ideal é que os três aspectos sejam congruentes. Porém, caso cada componente se comporte de maneira disforme, a maior chance é de que o receptor leve em conta o aspecto visual para determinar, por exemplo, se a mensagem é verdadeira ou falsa.

A relevância da comunicação corporal é percebida desde as primeiras investigações da área de IHC; setores de pesquisa se dedicam a estudos na área de rastreamento do corpo humano que datam desde o início da década de 1980 (AGGARWAL *et al.* 1997). Desde então, os métodos que se propõem a tal tarefa passaram a abranger diferentes formas de perceber o corpo do usuário.

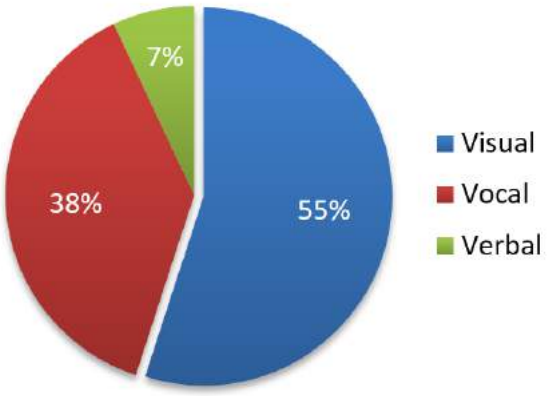


Figura 3.1: Grau de aceitação de receptores na comunicação, separando os aspectos visuais (linguagem corporal), vocais (tonalidades e timbres de voz) e verbais (palavras da mensagem).

Figura 3.2: Classificação de técnicas de rastreamento do corpo humano a partir de critérios de uso.

Uma das possibilidades explorada é o uso de dispositivos e acessórios anexos ao corpo para auxiliar o rastreamento. O rastreamento por sua vez se trata do processo de entender o posicionamento do corpo humano para em seguida interpretar os gestos realizados. Neste sentido, há dispositivos que possuem sensores inerciais associados como acelerômetros e giroscópios (TITTERTON 2004) e assim conseguem medir o deslocamento e rotação de partes do corpo. Por outro lado também existem acessórios com cores específicas e/ou emissores de luz que fornecem as informações necessárias a uma unidade de captura (WANG *et al.* 2009; LEE 2008).

Há também métodos que dispensam o auxílio de acessórios, deixando o corpo do usuário mais livre. Dentro deste escopo existem técnicas monoculares (SHAN *et al.* 2009) e mais recentemente algumas que realizam o rastreamento através de sensores de profundidade (MICROSOFT 2010). Desta forma, em termos de uso, as técnicas de rastreamento do corpo humano podem ser classificadas como apresentado na Figura 3.2.

Figura 3.3: Luvas táteis (haptic gloves). Topo: CyberGlove II (CYBERGLOVE 2010); Esquerda: P5 Glove (CYBERWORLD 2012); Direita: Rutgers Master II-ND (POPESCU *et al.* 1999).

3.1 Acessórios Anexos

Uma das formas de capturar os movimentos corporais é através do uso de acessórios ou dispositivos anexos ao corpo. Hardware com capacidades sensitivas como, luvas táteis (Figura 3.3), e sensores inerciais são capazes de prover informações sobre a posição e a orientação de partes do corpo. Estes equipamentos podem estar ligados por fios a uma unidade de processamento ou por tecnologias sem fio. De forma geral, fornecem informações precisas e em tempo real, pois os componentes sensitivos possuem circuitos específicos para executar tarefas consideravelmente simples se analisadas em separado.

Outra forma de capturar os movimentos do corpo é através do rastreamento de acessórios facilmente detectáveis por câmeras. Neste caso, ao contrário de existir um sensor preso ao corpo do usuário, sinais visuais são fornecidos e capturados, facilitando o rastreamento do corpo. Estes sinais variam desde luvas com cores destacadas (KANERVA *et al.* 2006; FIGUEIREDO *et al.* 2009), luvas coloridas (WANG *et al.* 2009), marcadores fiduciais para luvas (VEIGL *et al.* 2002) ou vestes (FIALA 2007), a roupas cobertas de luzes infravermelhas comumente usadas em técnicas de motion capture (MOESLUND *et al.* 2001; RASKAR *et al.* 2007; HERDA *et al.* 2000). Estes sinais visuais funcionam como dicas para o processo do rastreamento; ao se introduzir acessórios de aspecto diferenciado no ambiente, se torna mais fácil detectá-lo e distinguir as partes interativas da cena (Figura 3.4).

Por fim, existem abordagens mistas, que procuram unir vantagens de ambos os lados. O Nintendo Wii (NINTENDO 2006), console de videogame disponível desde 2006, conta com uma barra emissora de luz infravermelha e joysticks sem-fio (Figura 3.5) equipados com acelerômetros

Figura 3.4: Topo: marcadores fiduciais (VEIGL *et al.*2002); Esquerda: luva multicor (WANG *et al.* 2009); Direita: luvas amarelas (FIGUEIREDO *et al.* 2009) como exemplos de acessórios como pistas visuais para o rastreamento do corpo.

e uma câmera de alta taxa de atualização (100Hz) que capta apenas sinais infravermelhos (LEE 2008). Desta forma, apesar de a câmera estar no Wii Remotes, ou Wiimotes, consequentemente ao corpo do usuário e a barra emissora estar em uma posição fixa no ambiente, distanciada do usuário, a emissão-recepção de luz infravermelha funciona também como uma pista visual, de forma semelhante às citadas acima, sugerindo apenas uma mudança de referencial.

Em (SONY 2010), outro exemplo de caso híbrido é apresentado como Playstation Move (PS Move), associado ao console Playstation 3 (PS3). O PS Move apresenta um funcionamento semelhante ao Wiimote, fazendo uso de sensores inerciais, câmera e sinais luminosos. Porém, além do referencial usado neste caso ser o mais comum (uma câmera estacionária e um sinal

Figura 3.5: Esquerda: PS Move (joystick) e PS Eye (câmera), acessórios para o Playstation 3 (SONY 2010). Direita: Wii Remote, equipado com uma câmera infravermelha na extremidade superior, e Wii Sensor Bar, com 5 LEDs infravermelhos em cada extremidade lateral (NINTENDO 2006).

visual anexo ao dispositivo ou parte do corpo de interesse), o PS Move funciona no espectro de luz visível. Um globo emissor de luz na extremidade do controle funciona a partir de LEDs RGB, desta forma, a cor escolhida para ser rastreada é aquela que tem maior destaque no ambiente, dentro do espaço visualizado pela câmera (SONY 2010).

De forma geral, ambos os dispositivos funcionam bem na prática, com baixo tempo de resposta e alta precisão. A partir desta combinação, os consoles citados são capazes de anular o acúmulo de erro gerado pelos sensores inerciais, em uma espécie de calibração constante a partir do captura visual dos sinais emitidos. Por outro lado, o uso de sensores inerciais permite o contínuo rastreamento mesmo após oclusões totais do dispositivo.

3.2 Corpo Livre

Além de técnicas de rastreamento através de dispositivos ou acessórios anexos, existem outras formas de rastrear os movimentos corporais do usuário, deixando o seu corpo livre. Estas técnicas tendem a trazem o benefício da liberdade atrelada à interação do usuário, conceito importante em Interação Natural. Nos casos em que se faz necessário o uso de um acessório anexo, o usuário precisa dedicar parte de sua atenção em como está segurando o aparelho, se está apontando na direção correta, além de se preocupar em não lançá-lo e ter que lidar com o seu peso constantemente. Desta forma, mesmo ergonomicamente bem moldados, estes dispositivos tendem a limitar o usuário durante a interação, tanto em relação ao aspecto físico quanto ao intelectual. O mesmo se aplica as luvas táteis, luvas coloridas e roupas com marcadores.

Por outro lado, existem alternativas que não recorrem a acessórios ou dispositivos extras ligados ao usuário. Técnicas de rastreamento neste sentido podem usar câmeras comuns (monoculares) ou outros sensores mais elaborados, capazes de recuperar as informações de profundidade do ambiente.

3.2.1 Sensores de Profundidade

Sensores de profundidade são capazes de determinar a posição no eixo z para cada *pixel* da cena capturada. Desta forma, é possível destacar e agrupar objetos, os quais em uma imagem de câmera comum compartilhariam o mesmo plano. Assim, a terceira coordenada é bem aproveitada em cenários de interação, em que o usuário se posiciona em frente ao sensor se destacando, em termos de profundidade, em relação ao restante do ambiente.

Dispositivos como o Kinect (MICROSOFT 2010), Xtion PRO e Xtion PRO LIVE (ASUS 2011) ilustrados na Figura 3.6, através de um padrão projetado em luz infravermelha e uma câmera capaz de capturá-lo, se tornam sensores de profundidade, associando a cada *pixel*, um valor no eixo z. Com o auxílio da terceira coordenada, através de um processamento executado em hardware, é possível destacar o corpo humano das demais partes do ambiente, definindo um esqueleto básico com vinte pontos para algumas articulações e extremidades do corpo como

Figura 3.6: Sensores de profundidade. Esquerda: Kinect (MICROSOFT 2010); Direita: Xtion PRO (ASUS 2011).

ombros, punhos e face (SHOTTON *et al.* 2011).

Além do esqueleto simplificado do corpo humano, existem trabalhos que tem como alvo outras partes do corpo no intuito de obter um rastreamento mais refinado. Tendo o rastreamento de mãos como foco, por exemplo, ao contrário do único ponto que se obtém como representação, é possível obter a configuração desta com 26 graus de liberdade (OIKONOMIDIS *et al.* 2011).

No entanto, apesar da inovação trazida por sensores de profundidade, estes revelam alguns problemas para a interação corporal e os seus cenários de uso. As tecnologias oferecidas neste sentido demonstram limitações em relação à iluminação devido ao uso de luz infravermelha. O reconhecimento do padrão emitido é prejudicado caso haja incidência de luz solar através de janelas ou vãos abertos no ambiente, e espaços a céu aberto tornam o processo ainda mais difícil dado que a luz solar emite também espectro infravermelho em intensidade superior a destes sensores. Mais além, há indícios de atrasos na transmissão de comandos para jogos que usam o Kinect, em alguns casos girando em torno de 150 milissegundos (MINKLEY 2010), que segundo (VALLI 2005) é considerado o limite de atraso para uma interação natural, ou seja, nestes casos o atraso pode tornar a interação incômoda.

3.2.2 Câmeras Monoculares

Em contrapartida, existem técnicas monoculares de rastreamento dos movimentos corporais, fazendo uso apenas de uma câmera comum hoje presente em praticamente qualquer dispositivo como PCs, laptops, tablets e smartphones. Por isso, técnicas deste tipo trazem vantagens em relação à mobilidade e ao custo, pois não envolvem um segundo equipamento para capturar a cena. Além disso, ao contrário dos sensores de profundidade apresentados, suportam uso em ambientes a céu aberto ou com alta incidência de luz solar, pois não fazem uso da emissão/reflexão de luz infravermelha.

Por outro lado, técnicas de rastreamento monocular tendem a apresentar dificuldades extras, pois não contam com o auxílio de informações adicionais de sensores de movimento ou profundidade. Estas técnicas são de forma geral, baseadas inteiramente em algoritmos de visão computacional e assim estão sujeitas a problemas como sensibilidade a variações de iluminação, borramento por movimento (motion blur), atrasos (devido ao alto tempo de processamento que alguns algoritmos exigem), oclusão, etc. No entanto, a problemática apresentada não trata de limitações conceituais em relação à tecnologia utilizada, como por exemplo nos casos

previamente citados sobre sensores de profundidade e movimento em relação às restrições de uso que implicam. Ao contrário, as dificuldades acima se comportam como desafios constantemente atacados por técnicas monoculares, com o objetivo final de obter uma solução eficiente e robusta para cenários de interação de forma geral. É nesta linha que o trabalho proposto trata estes problemas, visando contornar as dificuldades citadas através da associação de técnicas de Visão Computacional voltadas para propósitos de interação corporal.

Além destes desafios intrínsecos à interação através da captura monocular, o fato de ter o corpo humano como alvo de rastreamento adiciona complexidade ao problema. O corpo humano é extremamente articulado, possuindo uma série de membros e articulações. Além disso, a aparência de cada usuário é singular, desde a cor da pele ao modo como se veste, exigindo da técnica de rastreamento maleabilidade. Ou seja, mesmo o modelo do corpo humano sendo bem conhecido anatomicamente, obedecendo sempre a um mesmo padrão de proporções e articulações, não é possível rastreá-lo com técnicas de rastreamento de modelos usuais como algumas utilizadas, por exemplo, para aplicações de Realidade Aumentada baseada em Modelos (LIMA *et al.* 2009).

Tendo em vista os problemas levantados, o desafio de rastrear o corpo humano é atacado de várias formas, muitas vezes com foco em uma parte específica do corpo, como mãos (MAHMOUDI *et al.* 1993), face (VADAKKEPAT *et al.* 2008), olhos (COUTINHO *et al.* 2010), etc. Ao contrário do que pode ser erroneamente deduzido, o rastreamento do corpo por inteiro, como regra geral, não é um super-conjunto dos demais. Técnicas de rastreamento do corpo geralmente apresentam restrições em relação a algumas posições (THAYANANTHAN *et al.* 2008), como não ser robusto a posições laterais do corpo e não conseguir rastrear uma pessoa sentada, por exemplo. A distância mínima do usuário em relação à câmera também pode inviabilizar algumas aplicações, pois se faz necessário um enquadramento do corpo completo do usuário na imagem capturada. Mais além, de forma geral, técnicas de rastreamento do corpo como um todo tendem a retornar poucos detalhes sobre partes específicas como mãos e faces por exemplo.

Desta forma, o trabalho aqui proposto tem foco no rastreamento de mãos e faces como forma de interação, motivado pelo uso deste tipo de rastreamento no contexto de entretenimento. Os movimentos da mão são, de forma geral, expressivos e tem aplicabilidade extensa (JOHN *et al.* 2009; RODRIGUEZ *et al.* 2010; HACKENBERG *et al.* 2011) e o rastreamento de faces pode contribuir para tarefas adicionais como identificação do usuário, e modos de visualização direcionados, simulando displays 3D a partir da movimentação do ponto de vista do usuário (LEE 2008).

3.2.2.1 Rastreamento 3D

Rastrear mãos e faces com propósito de interação não é uma tarefa trivial; a literatura envolvida é plena de diferentes métodos para atingir este objetivo. Existem rastreadores 3D que tentam recuperar a localização das mãos (BRAY *et al.* 2007) e das faces (MURPHY-CHUTORIAN *et al.* 2009), com seis graus de liberdade (DoF) ou mais em rastreamentos de

Figura 3.7: Treinamento (topo) e reconhecimento (base) de gestos com a mão através de classificadores em cascata. Antes de executar o processo de reconhecimento a estimativa da posição e tamanho da mão é realizada em 2D (STENGER 2006).

mãos que levam em consideração os diferentes posicionamentos dos dedos.

Em geral, o rastreamento de faces possui algumas facilidades em relação ao de mãos, devido principalmente, ao fato de que as configurações da face não mudam significativamente. Mesmo com variações como movimentos da boca e dos olhos, o aspecto geral da face permanece o mesmo. Inclusive, existem rastreadores de face 3D que alcançam os requerimentos principais para aplicações interativas, tais como velocidade de execução e robustez do rastreamento, (WANG *et al.* 2007; SEEINGMACHINES 2012).

Por outro lado, os métodos de rastreamento 3D de mãos tendem a apresentar limitações, como a demanda de um tempo maior de processamento e sensibilidade a movimentos rápidos (STENGER 2003), o que é um ponto negativo para aplicações interativas. Em alguns casos o procedimento de rastreamento de mãos é executado rápido o suficiente para ser utilizado numa aplicação em tempo real, no entanto estão sujeitos à outras restrições como a baixa amplitude de movimentos e configurações dos dedos (STENGER 2006), fator que limita a aplicabilidade da técnica.

3.2.2.2 Rastreamento 2D

Mesmo tendo em vista como objetivo final o rastreamento 3D, técnicas 2D (ou planares) são capazes de prover informações suficientes para uma série de interações em relação à movimentos espaciais. Além disso, técnicas planares de rastreamento e detecção de faces e mãos dão suporte ao reconhecimento de expressões faciais (SHAN *et al.* 2009) e gestos (STENGER 2006; IKE *et al.* 2007), como ilustrado na Figura 3.7. Em adição, o rastreamento planar pode dar suporte à uma técnica 3D, indicando uma região de busca ótima como entrada, diminuindo o seu tempo de execução e aprimorando a confiabilidade do resultado.

Além disso, iniciativas de rastreamento 2D são propensas a atingir uma alta taxa de atualização. Em (BRADSKI 1998), é proposto um método para rastreamento de faces chamado CAMSHIFT, com resultados ilustrados na Figura 3.8. A biblioteca OpenCV (FRAN 2004)

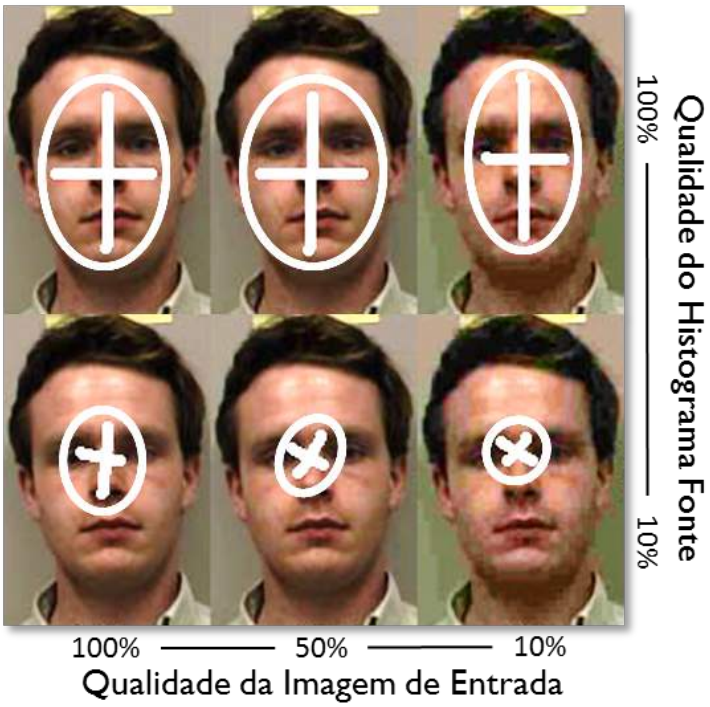


Figura 3.8: Resultados do uso da técnica CAMSHIFT em diferentes níveis de qualidade (em relação à compressão JPG) das imagens e histogramas de entrada (BOYLE 2001).

fornece um método derivado do CAMSHIFT, que generaliza o objeto rastreado, sendo iniciado por uma seleção da imagem capturada realizada em tempo de execução. Como inicialização, um histograma é criado a partir de uma região retangular da imagem da câmera, em seguida esta região é rastreada e expandida considerando uma segmentação baseada no histograma. Assim o alvo rastreado pode se tornar qualquer objeto ou parte do corpo, incluindo as mãos do usuário. Esta abordagem se beneficia do fato de que o objeto a ser rastreado é fornecido no real cenário da aplicação. Porém também apresenta problemas, como a precisão da área rastreada, que não é bem definida e ocasionalmente é indesejavelmente expandida, abrangendo regiões da imagem que não deveriam ser rastreadas. Na Figura 3.8 é possível observar o uso da técnica CAMSHIFT voltada para o rastreamento de faces sob diferentes circunstâncias de qualidade da imagem utilizada para a formação do histograma (imagem de treinamento) quanto também da qualidade da imagem utilizada durante o rastreamento (imagem de entrada); a variação da qualidade da imagem se refere à taxa de perda associada a uma compressão realizada no formato JPEG.

Há trabalhos que além das informações de cor do objeto rastreado, fazem uso de características visuais da imagem de forma localizada (features) para realizar o rastreamento. De forma geral, features são pontos na imagem que representam alto contraste em relação à vizinhança (em geral pontos que fazem parte de arestas na imagem real), e são mais facilmente rastreáveis em uma sequência de *frames*. Extraindo e seguindo um largo conjunto de features internos e próximos à região rastreada, é possível seguir determinado alvo durante a interação. Em relação ao CAMSHIFT, o uso de features traz alguns benefícios no sentido de um maior controle do rastreamento, pois cada feature possui um nível de informação inerente, fornecendo dados a

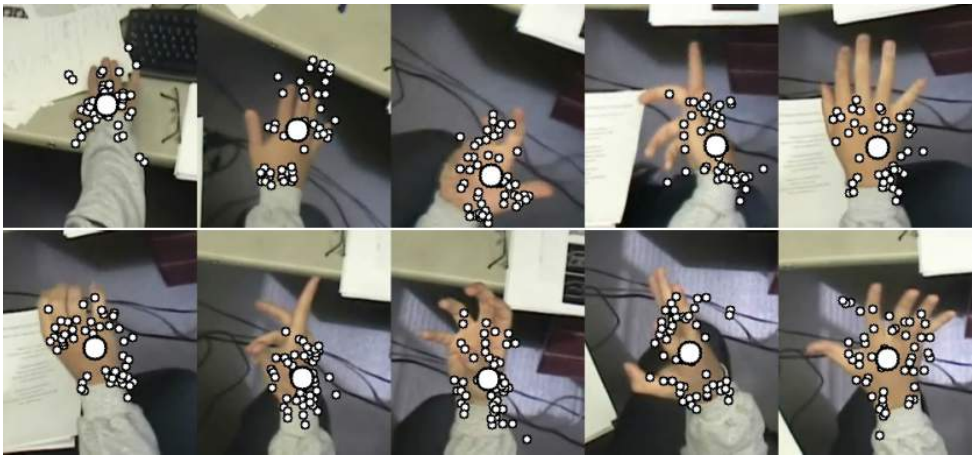


Figura 3.9: Resultados do rastreamento de mãos através do gerenciamento de uma nuvem de features (KÖLSCH *et al.* 2005).

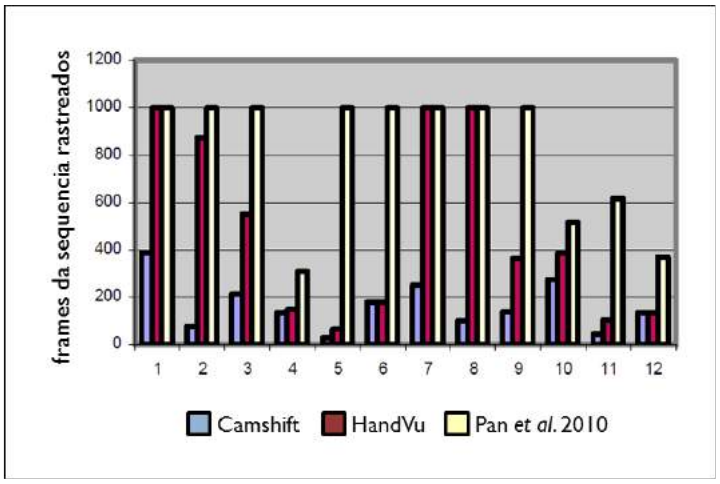
respeito de sua velocidade (indicando o fluxo do movimento) e do seu tempo de vida (o que é útil para abordagens temporais). Desta forma, se usadas adequadamente, estas informações são úteis para obter um rastreador mais robusto. Em (KÖLSCH *et al.* 2005), um método de rastreamento através de nuvem de features (ou flocks of features), chamado HandVu (Figura 3.9), é descrito e comparado com o CAMSHIFT (BRADSKI 1998), obtendo resultados superiores em relação à robustez a falhas de rastreamento.

Em seguida, outros trabalhos podem ser citados por explorar abordagens similares à nuvem de features (HOEY *et al.* 2006; ONG *et al.* 2009). Em (PAN *et al.* 2010) é proposto um método de rastreamento de mãos baseado em (KÖLSCH *et al.* 2005), que propõe o controle da nuvem de features, realizado a partir de remoções e inserções contínuas das mesmas, com o propósito de manter um número fixo de features rastreadas. Além disto, este controle faz uso da informação de velocidade das features para selecionar um ponto mediano que represente o conjunto. Desta forma é possível evitar que a mão deixe de ser rastreada por conta da interferência de outros alvos com cor de pele que se movam mais lentamente, como regiões do background ou a própria face (Figura 3.10).

Mais além, este método de gerenciamento da nuvem de features proposto em (PAN *et al.* 2010) é comparado sob diversas condições: de iluminação, incluindo ambientes fechados e a céu aberto; de movimentação, com movimentos lentos, comuns e rápidos da mão; com mudanças de gestos (posicionamento relativo dos dedos e da palma); e de complexidade do pano de fundo da cena tanto em relação à dinamicidade quanto à quantidade de elementos com cor de pele (Figura 3.11). Os métodos usados na comparação são o CAMSHIFT (BRADSKI 1998), o HandVu (KÖLSCH *et al.* 2005) e a técnica proposta por (PAN *et al.* 2010), sendo o critério de avaliação a quantidade de *frames* que cada técnica é capaz de rastrear antes de ocorrer uma falha (perda do alvo). Em todas as 12 sequências de vídeo a técnica de (PAN *et al.* 2010) apresentou resultados superiores às demais, reforçando o conceito de gerenciamento da nuvem de features.

O presente trabalho propõe uma ferramenta para o rastreamento de mãos e faces chamada

Figura 3.10: Quatro *frames* de uma sequencia de rastreamento através do controle de uma nuvem de features com base na velocidade (PAN *et al.* 2010). Através do uso deste tipo de informação é possível priorizar a mão em relação à face.



ID	Mudaça de Gestos	Mudança de Pano de Fundo	Ambiente Interno / Externo	Velocidade da Mão	Complexidade do Pano de Fundo
1	Não	Não	Interno	Lenta	Pouca Pele[1]
2	Sim	Sim	Interno	Lenta	Pouca Pele[1]
3	Sim	Sim	Interno	Lenta	Normal[2]
4	Sim	Sim	Interno	Rápida	Normal[2]
5	Sim	Sim	Interno	Normal	Complicado[4]
6	Não	Não	Externo	Normal	Pouca Pele[1]
7	Sim	Sim	Externo	Normal	Normal[2]
8	Sim	Sim	Externo	Normal	Normal[3]
9	Sim	Sim	Externo	Normal	Normal[3]
10	Sim	Sim	Externo	Normal	Complicado[5]
11	Sim	Sim	Externo	Rápida	Complicado[5]
12	Sim	Sim	Externo	Rápida	Complicado[5]

Legenda da Complexidade do Pano de Fundo:
1. Uma pessoa;
2. Uma pessoa, roupas, porta de madeira;
3. Duas pessoas, roupas, porta de madeira;
4. Duas pessoas, móveis de madeira;
5. Muitas pessoas, rua.

Figura 3.11: Comparação entre três técnicas de rastreamento de mãos através de 12 sequências de vídeo cada uma com 1000 *frames* (PAN *et al.* 2010). Variações de valocidade da mão, mudanças de gestos e pano de fundo e cenários externo (outdoor) e internos (indoor) foram levadas em consideração. A métrica usada para avaliar as técnicas foi a quantidade de *frames* rastreados com sucesso até que ocorresse uma falha de rastreamento. Em todas as sequências testadas, o método de (PAN *et al.* 2010) obtever resultados superiores aos demais.

HAFT (como abreviação para Hand and Face Tracker). HAFT por sua vez é um rastreador baseado no trabalho descrito em (PAN *et al.* 2010), capaz de rastrear simultaneamente as mãos e a face do usuário a uma taxa de atualização interativa. O foco no controle da nuvem de features de maneira robusta, conduz os features para dentro da região rastreada (mãos ou face) através da realocação das mesmas e a partir de uma aplicação de pesos diferenciada para a face e para as mãos. Mais além, a segmentação da cor da pele no HAFT é voltada para cenários de interação, evitando ruídos internos (falso-negativos) e externos (falso-positivos), fato que reforça as regiões rastreáveis e exclui previamente regiões da imagem que poderiam gerar futuras features erroneamente extraídas (*outliers*).

Tendo como foco um rastreamento através do gerenciamento de nuvens de pontos, a ferramenta proposta visa explorar esta forma de rastreamento com o intuito de expandir a técnica apresentada em (PAN *et al.* 2010) e aplica-la em um cenário real de interação como descrito no Capítulo 5. Ou seja, dados os indícios de que através do uso de um conjunto de heurísticas para controlar a nuvem de pontos é possível obter um rastreamento em tempo real e robusto a variações de representação (expressões faciais e movimentos dos dedos das mãos) e movimentos rápidos do alvo rastreado, a ferramenta HAFT é proposta como uma solução que integra estes conceitos a uma visão mais usual, sendo apresentada como um protótipo capaz de rastrear simultaneamente diversos alvos presentes na cena. Os detalhes e as principais contribuições serão descritas no próximo capítulo.

4

HAFT - *HAND AND FACE TRACKER*

A técnica proposta neste trabalho tem como objetivo rastrear continuamente mãos e faces servindo como ferramenta para uma interação natural com o sistema. Desta forma, conceitos como liberdade, robustez à movimentos rápidos e baixo tempo de resposta orientam as decisões ao longo da proposição da técnica. O seu fluxo de execução principal é composto por seis etapas: segmentação, agrupamento, detecção, rastreamento, avaliação e refinamento (Figura 4.1). A cada iteração, a partir de uma nova imagem de entrada, alvos são detectados na cena e passam a ser rastreados juntamente com aqueles pré-existentes. Cada alvo pode representar uma mão, uma face, ou outra parte da cena que faz parte de uma região da imagem com alta incidência de *pixels* de interesse, neste caso, *pixels* avaliados como cor de pele. Assim, precedendo a detecção, é executada a etapa de segmentação de cor de pele, seguida pelo agrupamento e definição de contornos de regiões de interesse, o que acelera bastante o processo de detecção de faces por exemplo. Após o rastreamento dos alvos, é executada a etapa de avaliação para detectar e tentar recuperar-se de falhas, seguida da etapa de refinamento que prepara os dados para serem usados pela aplicação de destino.

4.1 Ajustes e Testes

O desenvolvimento da ferramenta HAFT se deu de forma iterativa, em uma sequencia de ciclos compostos por três fases: implementação, ajustes e teste. Desta forma, boa parte do processo de definição dos parâmetros utilizados foi obtida de forma em consequência de sucessivos testes realizados durante o desenvolvimento. Os testes realizados neste processo, de forma geral são testes preliminares, tendo como principal motivação definir o funcionamento completo do rastreamento de forma eficiente, mas deixando em aberto outras possíveis configurações que por ventura podem vir a ser mais sólidas em uma análise global.

Para a realização destes testes preliminares, foi utilizado um ambiente de laboratório, com boa iluminação proveniente de uma série de lâmpadas fluorescentes. As variações de iluminação exploradas se deram através do controle destas lâmpadas, acedendo e apagando-as. A câmera utilizada para a captura das imagens de entrada foi uma webcam comum de 1.3 mega*pixels*,

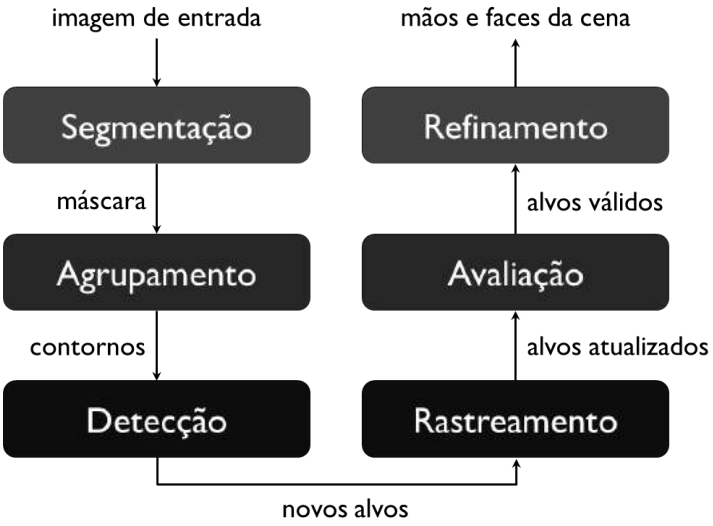


Figura 4.1: Fluxo de execução simplificado da técnica HAFT.

marca Clone e com capacidade de capturar 30 quadros por segundo nas resoluções de 320 x 240 e 640 x 480. A resolução utilizada durante os testes em questão foi de 640 x 480.

Dada a metodologia adotada para o desenvolvimento da ferramenta, cabe de antemão citar como um trabalho futuro, a aplicação de testes exaustivos tendo como objetivo a maximização de resultados positivos de cada uma das etapas do rastreamento.

4.2 Segmentação

A segmentação de cor de pele, em geral, serve de auxílio para técnicas de rastreamento do corpo humano, principalmente para o rastreamento de faces (MURPHY-CHUTORIAN *et al.* 2009) e mãos (STENGER 2006), pois o uso de roupas não afeta o resultado nestes casos. O processo de segmentação consiste basicamente da análise de uma imagem a fim de distinguir quais *pixels* desta são de interesse e quais não o são, tendo como resultado uma imagem binária com valores acima de zero para *pixels* de interesse, e valores zero para aqueles que não são de interesse.

A escolha do modelo de cor tem forte influência no resultado final de uma segmentação. Assim existem técnicas que adotam os mais diversos modelos como HSV, YCbCr, LUV, RGB, etc. Além disso, a segmentação pode considerar cada *pixel* de forma independente (VEZHNEVETS *et al.* 2003), ou usar um critério de decisão que leva em conta a influência da vizinhança (RUIZ *et al.* 2004).

4.2.1 Modelos de Cor

A escolha do modelo de cor tende a ter forte impacto dado o objetivo de segmentar pele humana em uma imagem. O modelo RGB é composto por coordenadas relativas às cores

primárias do sistema aditivo: R (Red) representando o vermelho, G (Green) representando o verde e B (Blue) representando o azul. Este modelo é comumente o modelo nativo de dispositivos de captura como webcams. No entanto o modelo RGB mescla informações de crominância e luminância entre as três coordenadas (R, G e B), dificultando um discernimento entre cor e não cor de pele que seja independente das condições de iluminação do ambiente. Para contornar este problema alguns métodos recorrem ao uso das coordenadas RGB normalizadas (GOMEZ *et al.* 2002), minimizando a diferença de intensidade luminosa.

Existem também técnicas que fazem uso de outros modelos de cor. O modelo HSV por exemplo, divide suas coordenadas entre matiz (H, ou hue), saturação (S) e valor (V, que representa a intensidade luminosa do *pixel*). Assim, é possível discernir aspectos inerentes à pele pondo à parte as condições de iluminação da cena. Apesar de fornecer uma boa representação, o modelo HSV apresenta alguns problemas como resultados da conversão de uma imagem previamente em RGB. A conversão para o modelo HSV tende a avaliar a crominância de regiões muito próximas de branco puro ou preto puro como vermelho, que por suas vez é a mesma crominância da pele. Além disso, o modelo HSV não trata o brilho de acordo com a percepção humana, ou seja, variando unicamente a componente H (relacionada a matiz) é possível ter variações de luminância; por exemplo, variando H de uma de azul para amarelo transmite uma sensação de luminosidade maior (mesmo mantendo as componentes S e V fixadas). O modelo YCbCr por sua vez, trata a crominância com duas coordenadas e possui um modelo de luminância mais acurado.

Cada um dos modelos apresentados traz características positivas para a detecção de cor de pele, incluso o RGB por dispensar o tempo necessário para conversão (VEZHNEVETS *et al.* 2003). No entanto, é possível tirar proveito da combinação de características em modelos diferentes. Um método proposto por (PETRESCU *et al.* 2011) apresenta um critério de detecção aplicável no modelo YCbCr que calcula a saturação (componente S do modelo HSV) para servir como peso atenuante em regiões de pele que devido à iluminação ou à características da pele do indivíduo são retratadas na imagem com baixa saturação (muito próximo à tons de cinza).

No presente trabalho, com intuito de aprimorar a técnica de segmentação de cor de pele, quatro critérios de classificação foram testados. É importante ressaltar que os critérios apresentados a seguir consistem de valores fixos usados como limiar para a decisão de se determinado *pixel* é ou não classificado como cor de pele e estes valores são fornecidos pelos respectivos autores como uma escolha genérica para que seja abstraído o uso de câmeras específicas ou condições de iluminação ambiente específicas para a segmentação. Desta forma, os valores citados pelos autores podem ser manipulados para otimizar a segmentação em um caso específico como será discutido mais a frente na subseção 4.2.2. Os quatro critérios de classificação testados estão descritos a seguir:

1. Critério de classificação usando o modelo de cor sobre o modelo RGB (impondo limiares sobre as três componentes R, G e B, cada uma variando de 0 a 255). Este critério abrange dois cenários de iluminação (KOVAC *et al.* 2003);

(a) Pele sob iluminação uniforme da luz do dia:

$R > 95 \& G > 40 \& B > 20$,

$R - G > 15 \& R > G \& R > B$

Este primeiro critério visa captar *pixels* que representam a pele sob iluminação uniforme, ou seja em seu aspecto comum e para isso impõe condições de que a componente vermelha seja minimamente maior que as demais e além disso que a azul seja a menor dentre todas.

(b) Pele sob iluminação de luz artificial ou iluminação lateral da luz do dia;

$R > 220 \& G > 210 \& B > 170$,

$|R - G| \leq 15$

Este segundo critério visa o caso de *pixels* de tom de pele que estão sob forte iluminação e assim possuem altos valores nas três coordenadas, além disso, existe uma restrição adicional para que a diferença entre as coordenadas vermelha e verde não seja grande, evitando captar *pixels* com alta saturação.

2. Critério usando o modelo de cor HSV, em um formato no qual as coordenadas V e S variam entre 0 e 1 e a coordenada H entre 0 e 360 (TSEKERIDOU *et al.* 1998);

   $V \geq 0.4$,

   $0.2 \leq S \leq 0.6$,

   $0 \leq H \leq 25 | 335 \leq H \leq 360$

   Este modelo exclui aqueles *pixels* demasiadamente escuros, que dificilmente são distinguíveis em relação à crominância. Além disso, são selecionados somente *pixels* com um valor intermediário de saturação, visto que os tons de pele no modelo HSV não chegam a ter saturação máxima (o que ocorre no caso de cores puras como laranja, amarelo e vermelho) e tampouco saturação mínima (que representa o caso dos tons de cinza). A restrição da coordenada H permite que esta varie de 0 a 25 (da matiz vermelha à laranja) ou entre 335 e 360 (matiz avermelhada com influencia pequena de tons de roxo). Visto que a componente H apresenta um comportamento cíclico é possível entender estas duas condições como uma única com o objetivo de selecionar matizes de tom avermelhado.

3. Critério com base no modelo RGB normalizado para reduzir a influência de variações de iluminação (GOMEZ *et al.* 2002);

   (a) Normalização das coordenadas:

   $r = R/(R+G+B) g = G/(R+G+B) b = B/(R+G+B)$

A normalização das coordenadas permite tratar de maneira similar dois *pixels* que apesar de representarem a pele sob iluminações diferentes, apresentam relações entre as componentes que são similares.

(b) Critério de classificação:

$r/g > 1.185\&$

$(r \times b)/(r+g+b)^2 > 0.107\&$

$(r \times g)/(r+g+b)^2 > 0.112$

4. Critério de classificação que faz uso combinado do modelo de cor YCbCr com uma estimativa da saturação do *pixel* (PETRESCU *et al.* 2011);

$s = \sqrt{((Cr-128)^2 + (Cb-128)^2)}$

$K = x(s+y)$

$Cr > 148.8162 - 0.1626Cb + 0.4726K$

$Cr > 1.2639Cb - 33.7803 + 0.7133K$

onde x é um valor entre 0.53 e 0.6 e y um valor entre 5 e 6.5. Os valores usados para x e y foram respectivamente 0.53 e 5.

Cada um destes critérios usa regras definidas que relacionam os componentes dos respectivos modelos de cor para classificar determinado *pixel*. A Figura 4.2 ilustra resultados dos quatro critérios testados.

4.2.1.1 Treinamento

No entanto, além de critérios de classificação baseados em regras pré-definidas, existem técnicas que fazem uso de conjuntos de dados de treinamento para definir se um *pixel* é ou não cor de pele (VEZHNEVETS *et al.* 2003). O treinamento consiste basicamente em construir histogramas para serem utilizados posteriormente em um modelo Bayesiano. Para alimentar os histogramas um conjunto de fotografias deve ser selecionado e segmentado manualmente. Usando um software de edição de imagens, regiões de cor de pele são selecionadas e uma imagem máscara é criada, com a mesma resolução da imagem original, sendo cada *pixel* representado em branco (se considerado como um *pixel* com cor de pele) ou em preto (se não).

Primeiramente, a base de dados que foi utilizada para a alimentação dos histogramas neste trabalho está disponível em (JONES *et al.* 1999). Esta base de dados é composta por imagens coletadas aleatoriamente na Web. Assim, apesar de portar uma quantidade grande de dados (cerca de 13.500 imagens) parte das imagens não representava um cenário real contendo símbolos ou desenhos, além de edições manuais comuns nesse meio.

De forma geral, é possível explorar uma troca entre abrangência e precisão ao se tratar da base usada como treinamento. Ou seja, caso o treinamento se dê partir de imagens que compartilhem um cenário comum com a aplicação é possível ter uma maior precisão nos

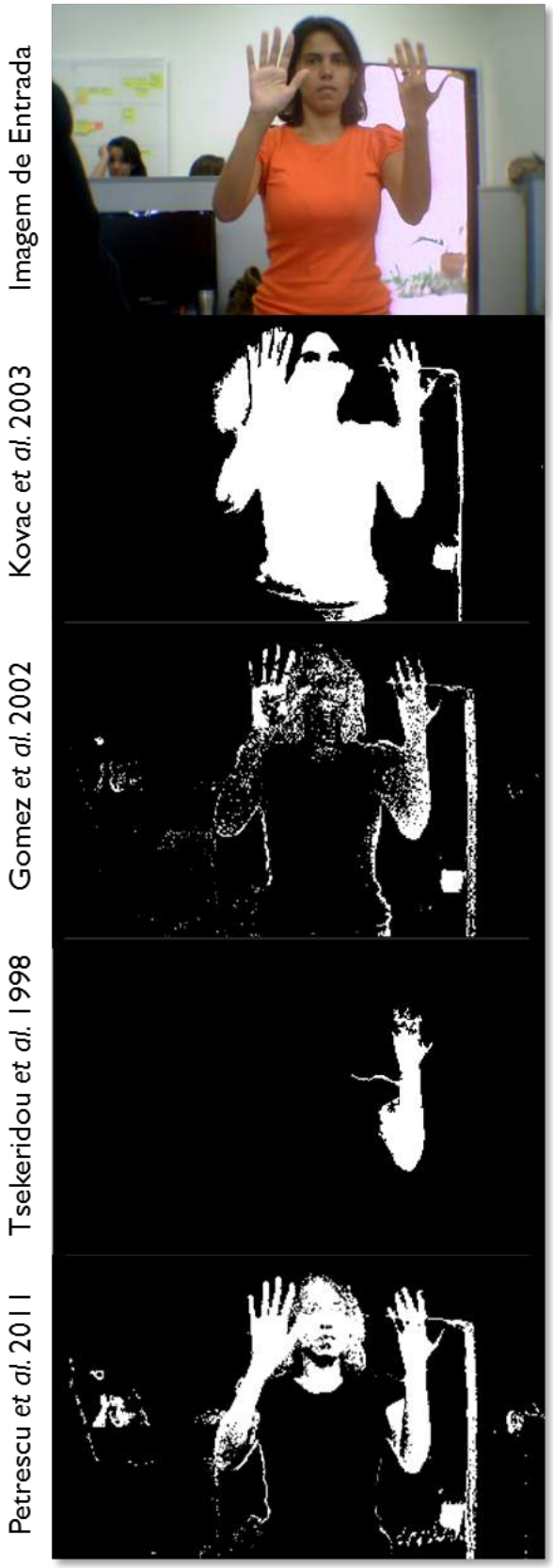


Figura 4.2: Resultados da segmentação de cor de pele a partir de quatro critérios de classificação analisados.

resultados. Assim, caso fosse considerado o objetivo de detectar regiões de cor de pele na Web, a base de dados disponibilizada em (JONES *et al.* 1999) seria aplicável, no entanto esta não é adequada para o presente trabalho, cujo objetive é ser uma ferramenta para interação que faz uso de imagens capturadas por webcams. Então, um novo conjunto de dados para treinamento foi construído; parte dele pode ser visualizado na Figura 4.3.

Neste novo conjunto foram retratados potenciais usuários, com diferentes tonalidades de pele, expondo as faces e mãos (frente e costas), para assimilar as variações de tom de pele. No total, 12 imagens foram utilizadas de resolução 656×558, e para cada uma foi gerada uma máscara de mesma resolução indicando os *pixels* tom de pele. O total de *pixels* cor de pele nesta base de dados foi de 555.647 e o total de *pixels* considerados como não cor de pele foi de 3.836.929.

A partir das máscaras fornecidas, dois histogramas são criados: o primeiro armazena as ocorrências dos *pixels* com cor de pele (positivo), e o segundo as ocorrências de *pixels* que não representam pele (negativo). Dada a base de dados histogramas foram criados para os três modelos de cor já citados (RGB, HSV e YCbCr), com o objetivo de analisar comparativamente a influência do modelo de cor no resultado final de segmentação. Assim, para o modelo HSV, foram gerados histogramas em 1D usando o componente H (hue) que determina a matiz além de restrições fixas sobre as coordenadas S e V, para o YCbCr, histogramas 2D a partir das coordenadas Cb e Cr com restrições fixas adicionas para a coordenada Y e para o modelo RGB, histogramas 3D.

Após alguns testes, optou-se por selecionar empiricamente o número de barras (bins) de cada histograma. De fato, não é sustentável utilizar a proporção 1:1 de número de bins em relação aos possíveis valores que cada componente pode assumir. Os possíveis valores que a componente H (de HSV) pode assumir variam de 0 a 359 (360 valores no total), no entanto se forem criados 360 bins alguns componentes acidentalmente terão poucas ocorrências e consequentemente pouca representatividade gerando buracos ou falhas no histograma, o que não deveria ocorrer. Este fato se agrava quando o número de dimensões do histograma é incrementado, para as coordenadas Cb e Cr o número total de valores é 65.536 e para o histograma RGB 3D chega a 16.777.216, de forma que mesmo em bases de dados extensas com muitas imagens como no caso da base fornecida em (JONES *et al.* 1999), é impraticável obter uma amostragem suficiente para que o histograma se mostre coeso. Assim, o número de bins por coordenada usado para os histogramas foi selecionado após uma série de testes com o objetivo de formar um histograma coeso dada a base de treinamento utilizada e ao mesmo tempo, perder o mínimo de informação que diferencia cada uma das regiões dos espaços de cor utilizados. Este número foi selecionado após a experimentação de uma segmentação com parâmetros fixos variando somente a quantidade de bins do histograma usado como base. Após sucessivas experimentações foi selecionado aquele resultado que demonstrou o menor ruído interno nas regiões de cor de pele sem que a segmentação passasse a tomar outras regiões que não são cor de pele como válida. Para o modelo HSV o número de bins selecionado para o histograma foi de 90, para CbCr 64

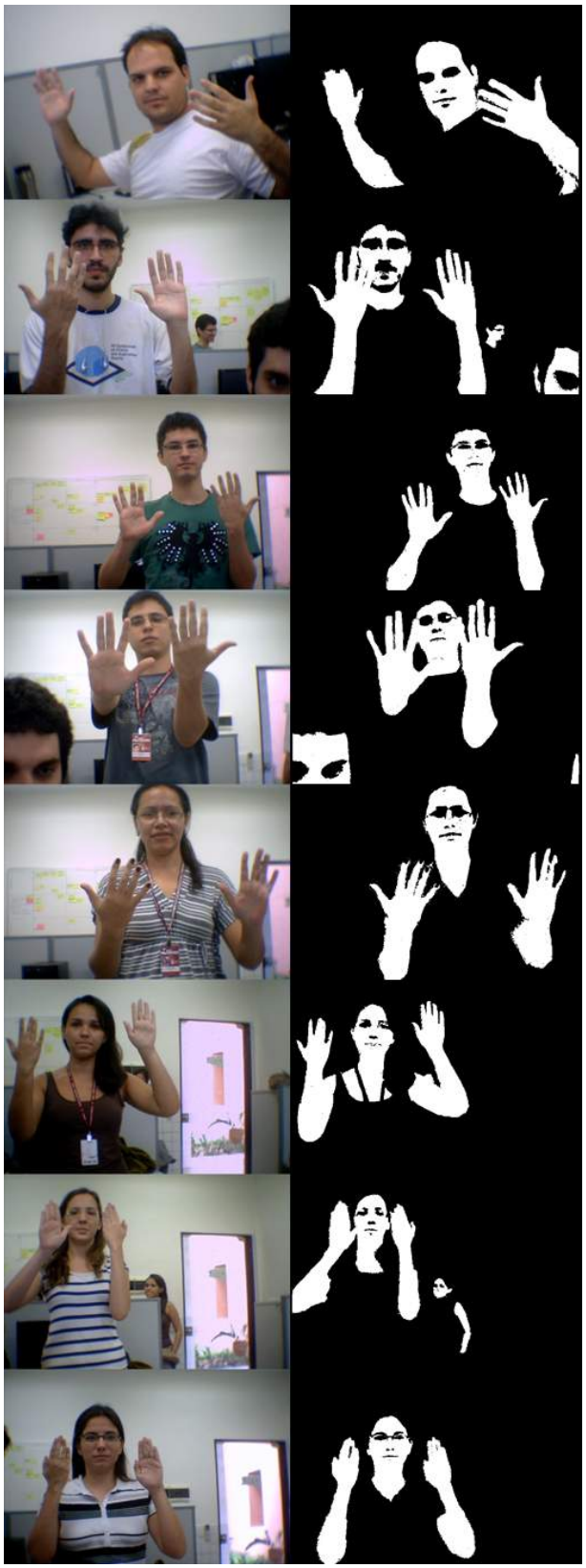


Figura 4.3: Exemplos de pares de imagens usados para alimentar os histogramas. Esquerda: imagem capturada por uma Webcam comum. Direita: máscaras binárias geradas manualmente com o auxílio de um software de edição de imagens.

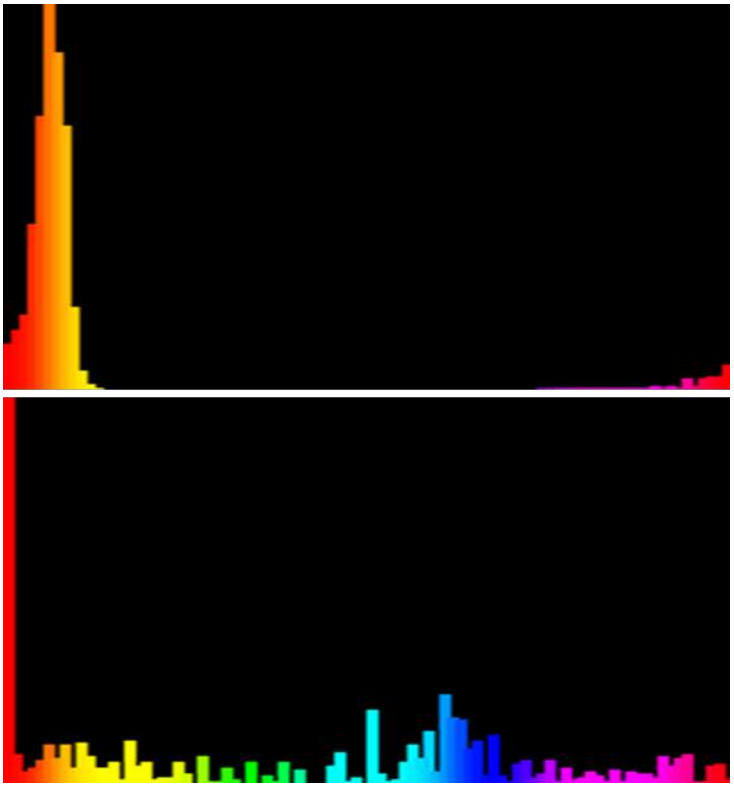


Figura 4.4: Histogramas com 90 colunas (bins), montado a partir da componente H do modelo HSV, ilustrando a ocorrência de *pixels* com cor de pele (topo) e *pixels* que não representam a pele (base). A cor das colunas representa a crominância dos *pixels* usados para alimentá-la. Para propósitos de visualização, os histogramas estão normalizados; de fato, o total de ocorrências do histograma negativo é superior em uma ordem de grandeza em relação ao positivo.

(totalizando 4096 bins), e para o RGB 16 (também 4096 bins no total). A Figura 4.4 ilustra os histogramas gerados a partir do modelo HSV.

Uma vez que os histogramas são gerados, os dados de treinamento são adicionados em tabelas de auxílio (Look-Up Tables ou LUTs). Estas tabelas são montadas para facilitar a checagem se determinado valor é válido ou não, economizando tempo de processamento. O cálculo para definir a probabilidade de um *pixel* ser ou não cor de pele é determinado por um classificador Bayesiano demonstrado no Grupo de Equações 4.1. As tabelas de checagem são usadas para armazenar as probabilidades pré-computadas de cada *pixel* ser cor de pele, evitando assim que o cálculo seja realizado durante o rastreamento propriamente dito, acelerando o processo de segmentação.

Grupo de Equações 4.1. Modelo Bayesiano para determinar a probabilidade da coordenada H de um *pixel* ser cor de pele.

$$
\begin{aligned}
P(pele) &= T_s/(T_s+T_n) \\
P(\neg pele) &= T_n/(T_s+T_n) \\
P(H|pele) &= (s[H])/T_s \\
P(H|\neg pele) &= (n[H])/T_n
\end{aligned}
$$

$$p = (P(H|pele)P(pele))/(P(H|pele)P(pele) + P(H|\neg pele)P(\neg pele))$$

onde *pele* representa a hipótese em questão, T_s e T_n representam o total de *pixels* nos histogramas de cor de pele e não cor de pele respectivamente, e $s[H]$ e $n[H]$ são os números de ocorrências no bin do histograma referentes à componente H (em cada um dos histogramas). H, por sua vez é a evidência em questão. O valor de p representa a probabilidade da hipótese pele estar representada na evidência H, ou seja, p representa a chance de observar pele dado o valor concreto de H. Este valor varia de acordo com a base de dados, pois o número de ocorrências de determinado *pixel* no histograma positivo ou negativo faz flutuar a chance do mesmo ser classificado de um lado ou de outro. Assim, o limiar aplicado para determinar a probabilidade mínima em cada caso foi determinado empiricamente, após sucessivos testes variando unicamente a probabilidade até que o melhor caso visual fosse encontrado. Estes testes foram realizados usando como base quatro imagens da base de dados usada como entrada para a construção dos histogramas, realizando uma comparação visual em paralelo a fim de encontrar o melhor valor para ser usado como limiar de probabilidade mínima em cada caso. Estas quatro imagens foram selecionadas por representarem cenas problemáticas, com problemas representativos tendo em vista o objetivo de segmentar cor de pele. As quatro imagens representam minimamente os seguintes problemas:

1. Pessoas com diferentes tons de pele, tons de pele muito claro e também escuro.
2. Tecidos com cores muito próximas a tons de pele.
3. Tecido laranja, ou seja, de mesmo matiz que cores tom de pele.
4. Regiões de tom de pele sob diferentes iluminações, tendo alguns pontos com alta incidência de luz (muito próximas da cor branca devido à resposta especular da pele) e outros com quase nenhuma (muito próximas da cor preta).
5. Regiões de madeira, apresentando cores muito próximas de um tom de pele escuro.
6. Parede avermelhada sob diferentes intensidades luminosas apresentando cores de tom rosa claro, confundível com tom de pele.
7. Outros objetos de diversas cores na cena, revelando falhas específicas dos métodos em relação a cores intensas como rosa-choque, amarelo e verde-limão.
8. Parede branca, facilmente atribuída a tons de pele em métodos que facilitam o critério de classificação (o mesmo que critério de aceitação ou validação) para casos de alta reflexão especular em que a pele assume tons muito próximos ao branco.

Todas estas características, assim como o resultado da execução de diversos métodos para segmentação podem ser observadas nas imagens disponibilizadas no Apêndice. É importante ressaltar que o classificador Bayesiano funciona de forma análoga para os modelos YCbCr

e RGB, diferenciando somente no momento do acesso ao número de ocorrências devido à dimensionalidade do respectivo histograma.

Apesar das componentes S e V do modelo HSV e Y do modelo YCbCr não serem utilizadas para montar os histogramas, durante a segmentação estas são consideradas com o propósito de descartar ruído e regiões indesejadas. De fato, a componente H do modelo HSV é insuficiente para determinar se um *pixel* é cor de pele. O uso da saturação como limiar é necessário pois esta componente assume o valor 0 quando a cor avaliada é muito próxima a tons de cinza, fato que explica a alta incidência de *pixels* considerados vermelhos no histograma negativo da Figura 4.4. Isto ocorre porque quando a saturação da cor em questão é zero ou próxima de zero, sua crominância (hue) assume um valor padrão já que não é possível definí-la com precisão. Outras regiões indesejadas podem sugir na cena com alta saturação como artefatos vermelhos ou laranjas, e usando apenas a coordenada H estas regiões seriam identificadas como cor de pele. Assim, um limiar superior e um inferior são usados sobre a componente S de cada *pixel* da imagem de entrada. O mesmo é feito com as componentes V (HSV) e Y (YCbCr) para evitar regiões muito escuras ou muito claras que podem ser representadas.

4.2.2 Propagação de Influência

Além de tratar *pixels* de forma isolada, existem métodos que propõem considerar a vizinhança de cada ponto para que este receba influências dos arredores (RUIZ *et al.* 2004). O método de segmentação proposto neste trabalho faz uso deste conceito de propagação de influência combinado com o uso conjunto de critérios de classificação em diferentes modelos de cor, os quais foram apresentados na subseção anterior, assim como sugerido em (SAWANGSRI *et al.* 2005).

O principal conceito que guia a segmentação proposta é expandir grupos de *pixels* que tem alta probabilidade de ser cor de pele, para outras regiões com menor probabilidade. Assim, sete critérios de classificação foram usados: os quatro primeiros citados na subseção 4.2.1 com a adição de um critério para cada conjunto de histogramas de treinamento nos três espaços de cor RGB, HSV e YCbCr. Cada um destes sete critérios de classificação implementados foi ajustado em dois modos: um modo que minimiza a quantidade de falso-positivos, revelando somente regiões com alta chance de ser cor de pele, estes *pixels* foram nomeados com o alias de iniciadores (ou starters); e um segundo modo que minimiza a quantidade de falsos-negativos, no intuito de cobrir ao máximo as regiões de pele, mesmo ao custo de incluir ruído no critério de aceitação, recebendo o alias de espalhadores (ou spreaders). Mais além, dado o fato de que os critérios de classificação testados possuem natureza independente, podendo ser combinados para um resultado mais completo, o agrupamento dos mesmos foi realizado. A partir de testes e observação, alguns critérios de classificação foram selecionados para definir o alias de cada *pixel*, sob os seguintes parâmetros: **Iniciadores:**

1. Histogramas HSV:

   (a) $probabilidade > 43\%$

   (b) $35 < S < 140$

   (c) $110 < V < 190$

2. Histogramas YCbCr:

   (a) $probabilidade > 50\%$

   (b) $60 < Y < 190$

Espalhadores:

1. Histogramas HSV:

   (a) $probabilidade > 1\%$

   (b) $30 < S < 170$

   (c) $40 < V < 230$

2. Histogramas YCbCr:

   (a) $probabilidade > 30\%$

   (b) $40 < Y < 210$

3. Histogramas RGB

   (a) $probabilidade > 5\%$

4. PETRESCU *et al.* 2011

   (a) $K = 0.4 \times (saturao + 5)$

A seleção dos valores apresentados acima foi obtida de forma empírica, como explicado na subseção 4.1. Neste caso, cada uma das componentes que não foram usadas como base pelo cálculo de probabilidade teve seu lmiar inicialmente igual aos 4 critérios citados e implementados (descritos na subseção 4.2.1) e em seguida foram ajustados através de testes em paralelo com as mesmas quatro imagens.

Para a execução da segmentação usando a propagação de influência proposta, cada *pixel* da imagem segmentada passa a guardar um valor correspondente ao seu potencial de propagação. Inicialmente este valor é inicializado com 0, sendo atualizado no decorrer do processo. Primeiramente, a execução percorre a imagem da esquerda para a direita, e de cima para baixo, até que seja encontrardo um *pixel* iniciador. A partir deste momento uma busca em largura é iniciada para abordar e expandir o potencial do *pixel* atual para os vizinhos (usando conectividade-4). A expansão de potencial perdura até que seja encontrado um *pixel* que não seja iniciador. Caso o *pixel* encontrado seja um espalhador, deste momento em diante o potencial de

propagação passa a decair sempre que espalhado para os vizinhos, a menos que atinja um vizinho iniciador e então o potencial volta a ser acumulado. Em ambos os casos, ainda durante a busca em largura, se o *pixel* atual é espalhador mas não herdou potencial de propagação ($potencial = 0$) ou se o *pixel* atual não é espalhador ou iniciador (não considerado cor de pele), então a busca em largura cessa, e a iteração comum volta a proceder em busca de um novo *pixel* iniciador. Todos os *pixels* visitados durante a busca em largura são marcados, de forma que não estão propensos a iniciar uma nova busca em largura redundante.

O acúmulo potencial de *pixels* iniciadores se dá através da simples soma do potencial do *pixel* gerador (*pixel* pai na busca em largura) somado a um valor base de 3. Este valor base é definido empiricamente, tendo em vista o objetivo de que os *pixels* iniciadores acumulem uma quantia de potencial maior que 1. Esta decisão é tomada visto o critério de que a condição de parada para a expansão é que o potencial propagado seja maior que 1.

Mas ao mesmo tempo existe o interesse de que este acúmulo não adicione um valor tão distante da unidade pois há o intuito de que os *pixels* tenham o potencial de propagação acumulado associado ao raio de alcance de sua propagação futura e assim não é recomendado que um *pixel* iniciador se propague por uma região muito maior. Ou seja, analisando o decaimento linear de um *pixel* que acumulou em seu potencial de propagação o valor de 200, este seria capaz de expandir o seu potencial e assim ativar uma região de *pixels* espalhadores em um raio de 200 *pixels*. Usando outros valores como base, por exemplo usando o valor 10 como o valor acumulado por *pixels* iniciadores, bastaria que 20 *pixels* iniciadores estivessem em conjunto para a ativação de uma região de 200 *pixels* espalhadores. Com o valor base de 3, o número de iniciadores necessários para tal é de 67 *pixels*, sugerindo uma expansão mais comedida. Além disso uma função atenuante é aplicada ao decrescer o potencial de propagação entre os *pixels* espalhadores. Ou seja, o decaimento de potencial não ocorre pela simples subtração de 1 do potencial do *pixel* pai para o *pixel* filho durante a busca em largura. Uma nova função é aplicada ao potencial de propagação com o intuito de controlar o decaimento de forma específica para que se obtenha a expansão desejada de cada conjunto de *pixels*. A função que determina o decaimento é definida de acordo com a Equação 4.2.

Equação 4.2. Modelo de propagação de potencial durante a busca em largura realizada na segmentação de cor de pele.

$$p_{filho} = p_{pai} \times (min(ln(p_{pai})/5, 0.99))$$

em que p_{filho} representa o potencia de propagação do *pixel* vizinho e p_{pai} o potencial de seu gerador. Esta função induz que regiões com muitos *pixels* iniciadores, após acumularem bastante potencial se expandam por uma quantidade considerável de espalhadores, enquanto que regiões com poucos iniciadores praticamente não se expandam, desencorajando a propagação de ruído.

É importante ressaltar que a Equação 4.2 define a função de decaimento do potencial de propagação tendo em vista imagens de entrada com a resolução de 640 x 480 *pixels*. A função

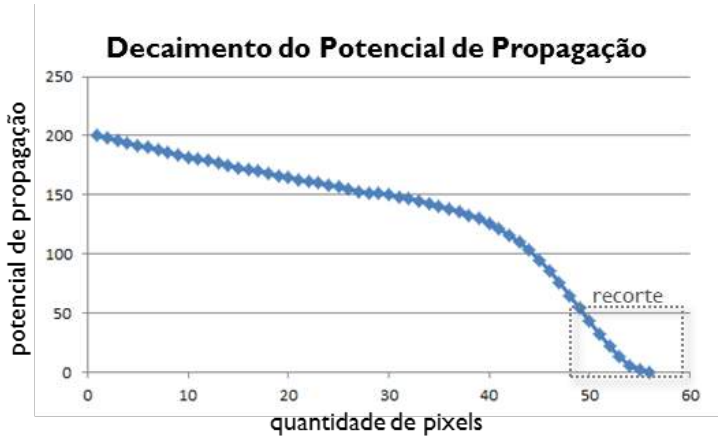


Figura 4.5: Amostragem do decaimento do potencial de propagação de *pixels* espalhadores. Esta amostragem demonstra que um *pixel* espalhador que tem valor de propagação próximo a 200, possui potencial para expandir-se por até aproximadamente mais 55 *pixels*, enquanto que se observada a amostragem dentro da pespectiva do recorte, é possível observar que um potencial de valor 50 não é capaz de expandir por mais do que 7 *pixels*.

de decaimento foi construída tendo em vista esta resolução dado que ao longo do processo de definição da segmentação as imagens usadas como entrada em cenas interativas obtinham esta mesma resolução. Desta forma para o uso em outras resoluções a função deve ser aplicada de forma adaptada, obedecendo o comportamento desejado de expansão comedida. A Figura 4.5 ilustra o decaimento do valor de propagação representado pela função em questão, variando de um potencial de 200 a potencial de 0. Desta forma, a segmentação de regiões de cor de pele é realizada expandindo pontos que possuam alta probabilidade de ser cor de pele, no entanto de forma comedida, assumindo um potencial associado a cada grupo e expandindo de acordo com ele (Figura 4.6).

Como funcionalidade opcional, como sugerido em (PHUNG *et al.* 2003), é possível limitar a expansão de grupos impedindo que *pixels* de aresta, através de um algoritmo de extração como o Canny (CANNY 1986), sejam agentes propagadores. Um conjunto de resultados comparativos pode ser visto na Figura 4.7, que ilustra alguns pontos chave para a motivação da implementação do método proposto. Artefatos das cenas que possuem cores muito próximas a tons de pele foram evitados como as camisas nas imagens de entrada 1 e 2. Este resultado foi alcançado sem que alvos de interesse como os braços da mulher na imagem 2 fossem erroneamente ignorados.

4.3 Agrupamento

Após a segmentação de cor de pele, uma máscara de resolução igual à imagem de entrada é fornecida. A depender do método de segmentação utilizado, possivelmente a máscara não será binária, possuindo valores intermediários entre 0 e 255. No entanto todos os *pixels* com valor maior do que 0 são considerados válidos.

Figura 4.6: Ilustração da execução da propagação de potencial em um conjunto de *pixels*. Para um *pixel* ser considerado cor de pele é necessário que seja um iniciador, ou então que seja um espalhador com potencial maior do que 1.

Assim, a partir da máscara fornecida como saída da segmentação, regiões com *pixels* interligados são agrupadas através de um algoritmo de detecção de contornos. Somente contornos externos são detectados, desta forma, incluindo regiões internas que a priori seriam consideradas inválidas dado o resultado da segmentação. Esta opção tem impacto relevante pois em geral os contornos internos representam regiões da face como olhos e boca, ou regiões sombreadas da mão. Além disso, contornos internos podem representar ruído dentro de uma região de cor de pele (Figura 4.8). De forma geral, em todos estes casos, é desejável que estas regiões permaneçam válidas para as etapas de detecção e rastreamento.

Além de detectar os contornos existentes na cena, nesta etapa também é aplicada uma remoção de ruído, através da eliminação de contornos com uma área menor que 2000 *pixels*2 para imagens de uma resolução de 640 x 480. Este valor representa 0.0065% da área total da imagem, e em caso do uso do mesmo processo em imagens de oturas resoluções, esta porcentagem deve ser usada como guia. Uma região da imagem que representa uma porcentagem tão baixa de forma geral pode ser excluído como elemento sem interesse para o rastreamento, visto que representa ou uma falha de segmentação ou um objeto muito pequeno ao fundo da imagem e que simplesmente não é considerado de interesse. Assim, o intuito é eliminar contornos irrelevantes para acelerar o processamento e diminuir a chance de falha das etapas futuras. Dado que o objetivo da ferramenta é detectar e rastrear alvos continuamente independente da quantidade e de em que momento de tempo a interação está ocorrendo (por exemplo, o usuário saiu de cena por um instante e pode ou não demorar a voltar), todas as regiões de cor de pele são rastreadas e sempre que aparece uma região nova, os algoritmos de detecção são acionados para inferir se o

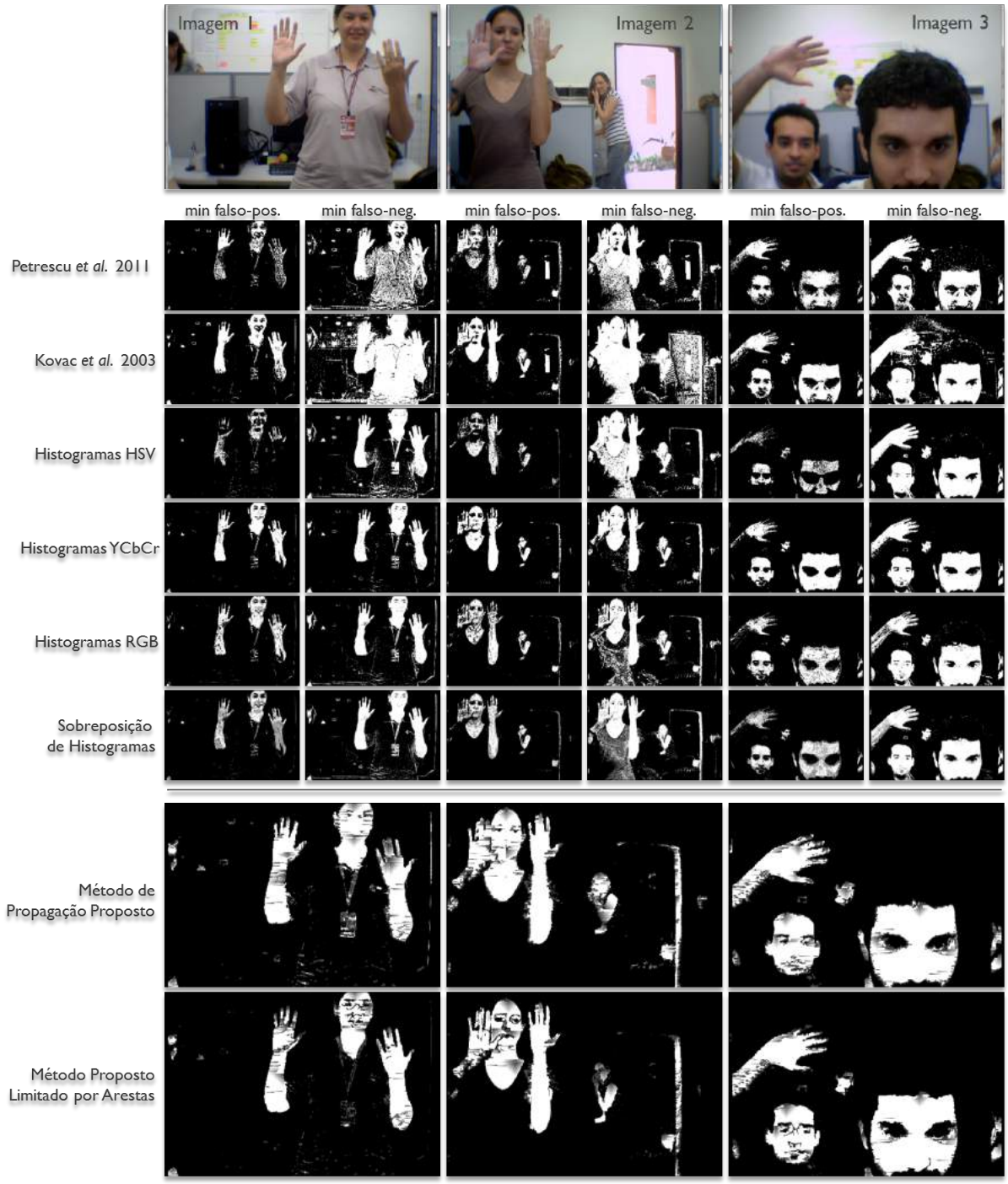


Figura 4.7: Resultados da segmentação de cor de pele em três imagens de entrada. Cada linha representa um método de segmentação distinto. Para os métodos usados como base na propagação, os resultados dos modos de minimização de falsos-positivos e falsos-negativos estão dispostos em duas colunas adjacentes.

Figura 4.8: Resultado da segmentação usando o histograma RGB (topo) e resultado do agrupamento através de contornos externos (base). Ruído interno às regiões de cor de pele é comum neste tipo de método de segmentação e o tipo de agrupamento usado tende a resolver parte dele.

novo alvo é válido. Assim, pelos motivos já levantados, é de considerável importância eliminar pequenos grupos gerados por ruído ou pequenos artefatos no fundo da cena.

Por fim, é durante esta etapa também que são definidas as novas regiões dos alvos rastreados com sucesso no *frame* anterior. Para cada alvo rastreado corretamente no *frame* anterior, se faz necessário encontrar uma região válida no *frame* seguinte, caso contrário é inferido que o alvo saiu de cena ou que está completamente ocludido e consequentemente não será mais rastreado. Como resultado, a etapa de agrupamento fornece um vetor de contornos após estes contornos estarem definidos uma busca é feita para associar um contorno para cada alvo proveniente do *frame* anterior. Cada contorno resultante da etapa de remoção de ruído é marcado na imagem máscara com um ID. Assim, dado um ponto qualquer na imagem é possível saber se este está ou não contido em um contorno e caso esteja, sabe-se qual é este contorno. A busca pelo contorno de cada alvo se dá através do ponto guia. Cada alvo rastreado possui um conjunto de informações associadas e dentre elas existe um conjunto de features e um ponto guia. Caso a posição do ponto guia não esteja respaldada por contorno algum no *frame* atual (devido à mudança das posições das regiões de pele na cena), a busca segue para a feature mais próxima do ponto guia, e assim por diante (Figura 4.9). Caso nenhuma das features esteja representada por um contorno, o alvo é perdido.

4.4 Detecção

A partir dos contornos extraídos na etapa anterior, os algoritmos de detecção passam a agir para inferir se os alvos são mãos ou faces. Para cada contorno do *frame* atual, primeiramente

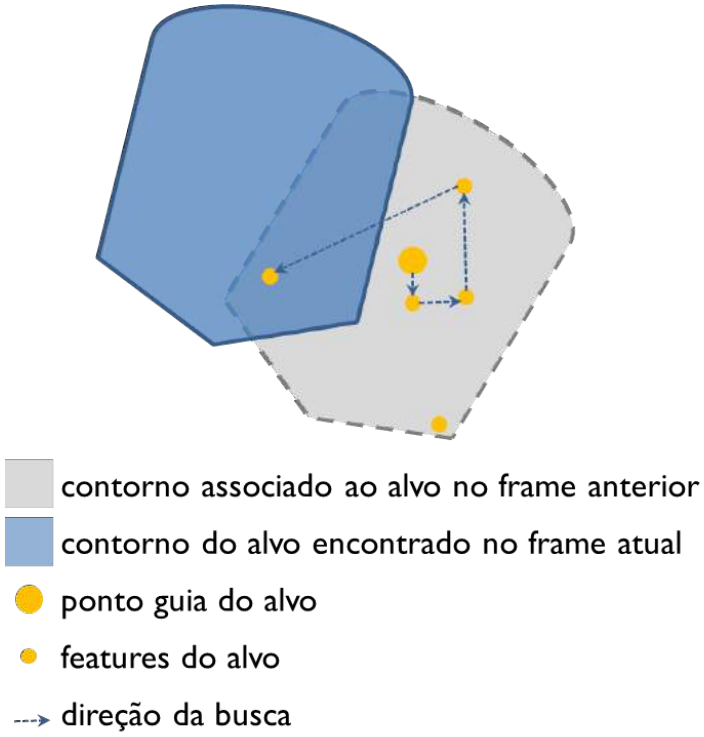


Figura 4.9: Representação da busca por um novo contoro para um alvo rastreado com sucesso no *frame* anterior. A busca cessa assim que o novo contorno é encontrado.

é verificado se este já está associado a um alvo e consequentemente está sendo rastreado. Caso o contorno não esteja sendo rastreado ele imediatamente se torna um alvo de tipo indefinido. Para que seja rastreado futuramente, um conjunto de features é extraído na região do contorno através do algoritmo Good Features to Track ou GFTT (SHI *et al.* 1994). A quantidade de features extraída é dependente do tamanho do contorno (perímetro), dada pela Equação 4.3.

Equação 4.3.

$$nfeatures = perimetro/16$$

A inferência do tipo de um alvo não é imediata. Para evitar detecções incoerentes (falsos-positivos), antes que o tipo de um alvo seja determinado, é necessário que este seja detectado com sucesso durante uma quantidade consecutiva de *frames*. Além disso, o custo de execução da detecção de faces é alto, e assim, não é aconselhável que a detecção de faces seja executada imediatamente para cada novo alvo presente na cena. Assim, um mecanismo de controle é acoplado a cada alvo, funcionando como contadores de detecções bem sucedidas e de *frames* de espera.

Mais especificamente, antes que um alvo seja considerado como face, este precisa esperar cerca de 10 *frames* sendo rastreado como um alvo de tipo indefinido. Após o tempo de espera,

é necessário que o alvo seja detectado como face em 3 *frames* consecutivos. Caso haja uma falha, o alvo retorna a esperar, no entanto desta vez por mais tempo (20 *frames*). Este mecanismo ajuda a acelerar o rastreamento e a garantir que uma região aleatória de cor de pele não seja interpretada como face.

Para a detecção de mãos, as restrições são consideravelmente menores. O algoritmo de detecção de mãos implementado é executado muito rapidamente, pois usa apenas o contorno propriamente dito, iterando uma única vez em todo ele. Além disso, a detecção de mãos trata aspectos bem específicos, sendo dificilmente sujeita a falso-positivos. Assim, sempre que surge um novo contorno, a detecção de mãos é executada para desvendar se este se trata de uma mão. O mesmo mecanismo de segurança é utilizado em relação a detecções consecutivas, no entanto neste caso o número exigido é 2 e não há tempo de espera pois a execução da detecção de mãos implementada é consideravelmente menos sujeita à casos de falso-positivo além de atingir um tempo de execução irrisório podendo ser executada para cada um dos grupos não identificados da imagem a cada quadro.

Desta forma, todas as regiões de cor de pele presentes na cena são rastreadas e eventualmente detectadas caso sejam uma mão ou face. Esta política de comportamento da ferramenta dá liberdade para o usuário transitar pelo ambiente e também para novos usuários entrarem em cena e colaborarem com o primeiro. Esta característica da ferramenta está alinhada com o conceito de liberdade presente em interações naturais. O intuito é de que seja um objetivo da ferramenta entender o usuário quando este quer interagir, e não o contrário.

4.4.1 Detecção de Faces

A detecção de faces é realizada através de um algoritmo baseado em classificadores Adaboost em cascata de features de Haar (VIOLA *et al.* 2004; SHAN *et al.* 2009). A implementação utilizada está disponível na biblioteca OpenCV (BRADSKI *et al.* 2008). Este tipo de algoritmo apresenta altas taxas de sucesso para a detecção de faces, e de forma geral, para tal tarefa, possui execução em tempo satisfatório (VIOLA *et al.* 2004). No entanto, para uma aplicação interativa, o ideal é que a execução seja flúida de forma que o usuário não necessite dedicar atenção a entender o que se passa durante a execução do sistema.

Desta forma, como entrada para a detecção de faces, ao contrário da imagem completa da cena, capturada pela webcam, são apenas enviadas as regiões específicas para acelerar o processo e assegurar que as faces detectadas sejam reais, respaldadas por regiões de cor de pele. O tamanho de janela utilizado para a execução do algorimto de detecção de faces foi de $30x30$ *pixels*.

4.4.2 Detecção de Mãos

A partir dos contornos retornados, para cada alvo que permanece indefinido, a detecção de mãos é executada. De fato, existem métodos de detecção avançados que demonstram sucessos

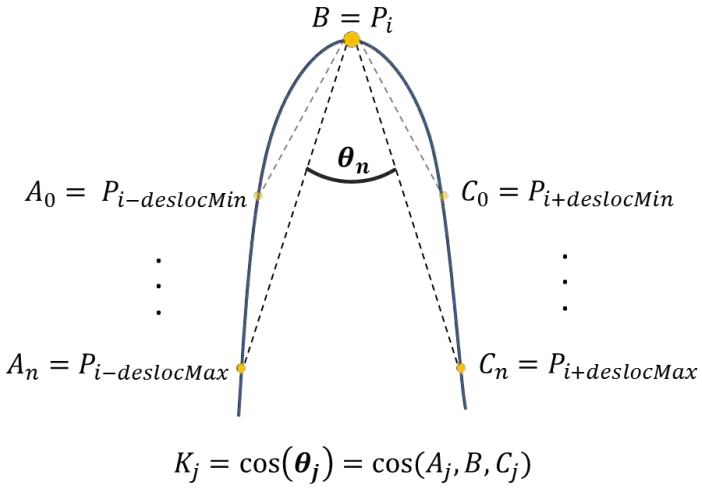


Figura 4.10: Cálculo do cosseno de cada ponto do contorno. Para cada ponto P_i, n cossenos K_j são calculados, sendo aquele de maior valor associado à P_i.

para as mais distintas poses das mãos e em baixas resoluções. No entanto estes métodos tendem a demandar um alto tempo de execução chegando a necessitar de minutos para que a execução esteja finalizada (MITTAL *et al.* 2011).

Assim, o método de detecção de mãos proposto neste trabalho, dado o propósito de atingir uma taxa de atualização interativa, é baseado nos contornos da região segmentada de cor de pele. Durante a execução, cada *pixel* de borda (perímetro do contorno) é visitado somente uma vez, tornando a execução do método rápida o suficiente para ter baixo impacto durante o fluxo de rastreamento. Assim, para que a mão seja detectada, é necessário que o contorno externo desta forneça informações suficientes. Então, a técnica proposta sugere um método de detecção que demanda uma posição da mão do usuário com pelo menos três dos dedos de uma mão do usuário esticados revelando oscilações no contorno da mão que possam ser detectadas. Como sugerido em (PAN *et al.* 2010), para cada *pixel* do contorno um valor de cosseno de ângulo é associado. Este cálculo é feito a partir de vizinhos compreendidos entre um deslocamento mínimo e um deslocamento máximo a partir de cada um dos pontos do contorno (Figura 4.10).

Em (PAN *et al.* 2010) é sugerido que este deslocamento varie entre 10 (mínimo) e 30 (máximo) pontos de distância e n neste caso é 20, ou seja, todos os casos de deslocamento são testados. No entanto, o uso de valores fixos para esta tarefa torna a detecção falível para diferentes escalas. Para contornar este problema, foi realizada uma análise da relação entre o tamanho do contorno como um todo (perímetro), e os respectivos tamanhos de dedos mínimos e médios em máscaras nas quais a mão se encontrava presente. Após uma série de amostragens, foram obtidas as funções de aproximação apresentadas na Figura 4.11. Desta forma, foi atribuído o valor estimado do dedo mínimo ao deslocamento mínimo e o valor do dedo médio ao deslocamento máximo, de acordo com as funções descritas na Equação 4.4.

Equação 4.4. Funções usadas para o cálculo do tamanho estimado dos dedos médio e mínimo em função do tamanho do perímetro.

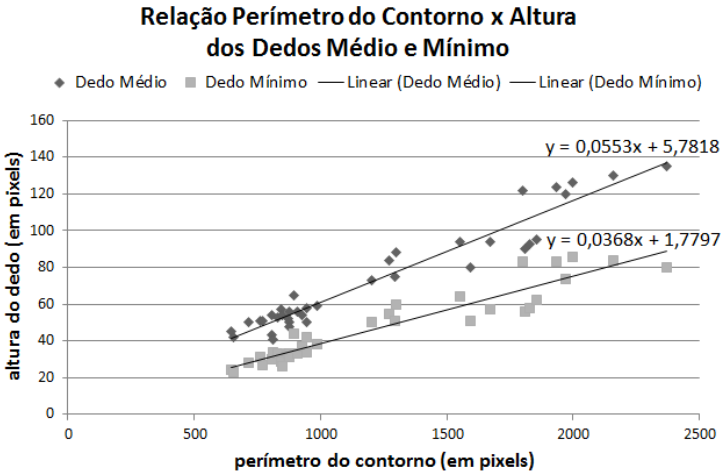


Figura 4.11: Amostragem das alturas dos dedos médio e mínimo em *pixels*, relacionadas com o tamanho do contorno (perímetro), assim obtendo funções de aproximação para a estimativa dos tamanhos dos dedos em relação à cada contorno selecionado.

$$dedoMed = (0.0553 perimetro) + 5,7818 \; dedoMin = (0.0368 perimetro) + 1,7797$$

Além disso, o número de testes para encontrar o maior valor do cosseno $(valorn)$ foi reduzido a 3, fazendo apenas checagens para o deslocamento máximo, deslocamento mínimo e para o deslocamento intermediário $((maximo\breve{}minimo)2)$. A quantidade reduzida de testes traz ganhos em tempo de execução sem prejudicar a qualidade do resultado. A Figura 4.12 ilustra resultados a partir de ambos os métodos. É possível perceber que para que a detecção seja robusta à mudanças na escala, manter um valor fixo para os deslocamentos é inviável. Além disso, nestes casos as falhas nos contornos geradas por ruídos acabam por ter forte influência no resultado gerando artefatos com "falsos" cossenos e dificultando as etapas futuras para a detecção. O limiar usado para definir pontos de interesse é de um cosseno maior do que 0, assim, somente considerando pontos que têm ângulo menor ou igual a 90. Em adição, para cada *pixel* válido, o produto vetorial entre este e os pontos vizinhos utilizados é calculado para determinar a direção do *pixel*. Esta direção indica se o *pixel* faz parte de um pico ou um vale no contorno, ou seja, indica se o ângulo formado é interno ou externo.

Durante o cálculo de cossenos ao longo do contorno, aqueles pontos adjacentes que foram considerados válidos e que possuem a mesma direção são agrupados. Em seguida grupos com tamanho menor que $dedoMin3$ são removidos pois representam ruídos menores do que o pico de um dedo ou um vale adjacente a dois dedos. Grupos com tamanho maior que $dedoMed1.5$ também são removidos pois de forma geral representam picos ou vales muito grandes para representar um dedo, e são distúrbios comuns de serem observados como vales por exemplo em distorções pontiagudas geradas pela segmentação, selecionando uma linha do cenário ao fundo junto ao contorno de uma região de cor de pele que está sendo analisada. Os valores

Figura 4.12: Cálculo de cossenos de cada um dos pontos do contorno. Na coluna da esquerda, resultados através do método de detecção em (PAN *et al.* 2010); na coluna da direita os resultados do método proposto. Pontos esverdeados representam cossenos internos ("abraçando" regiões de pele) enquanto que pontos avermelhados retratam cossenos externos. O limiar para o menor valor de cosseno é de 0, ou seja, somente ângulos iguais ou inferiores a 90 são considerados.

apresentados para descartar grupos de *pixels* de contorno muito pequenos ($dedoMin3$) ou muito grandes ($dedoMed1.5$) foram obtidos através de uma sequência de testes tentando evitar que grupos de interesse fosse descartados e ao mesmo tempo que aqueles que são grupos provenientes de ruído fosse eliminados. Estes testes se realizaram através de uma sucessiva sequência de ajustes e observação, de forma que os valores obtidos provavelmente não são ótimos para caso em questão e merecem um trabalho futuro aplicando testes mais extensivos e investigativos a fim de aperfeiçoar esta etapa do processo de detecção.

Em seguida é sugerido por (PAN *et al.* 2010) que sejam encontrados cinco picos (grupos verdes) e quatro vales (grupos vermelhos) que estejam intercalados em sequência ao longo do contorno. No entanto, esta opção descarta alvos que por ventura, mesmo o usuário fazendo corretamente a pose, não estejam representando claramente os cinco dedos, devido a falhas na segmentação.

Além disso, requerer os cinco dedos para a detecção eleva a probabilidade de que exista ruído entre estes. Assim o método proposto altera esta restrição para que a pose mínima necessária seja a de três dedos esticados. Apesar de abrir precedente para mais casos de falsos-positivos, esta abordagem permite a inserção de novas restrições geométricas que eliminam os casos duvidosos. O padrão procurado é identificado através das relações sobre as distâncias entre pontos centrais de picos e vales (Figura 4.13).

As restrições aplicadas estão descritas abaixo:

1. As distâncias entre vales (linhas vermelhas) devem ser menores que as distâncias entre picos (linhas verdes);

2. As distâncias entre picos devem ser menores que as distâncias entre picos e vales (linhas amarelas);

3. As distâncias entre picos não devem diferir entre si por um fator maior que 1.3; Pois neste caso significaria que a distância entre a ponta de um dedo e a ponta do dedo adjacente é 30% maior do que a distância entre outros dois dedos, enquanto que o comportamento esperado é que todas as distâncias entre as pontas de dedos que estão adjacentes seja consideravelmente parecida. O valor de 30% foi estipulado por testes preliminares e pode ser ajustado dependendo do grau de restrição desejado a ser aplicado no padrão geométrico;

4. As distâncias entre picos e vales não devem ser superiores ao tamanho de $dedoMin$.

Assim, através de algumas restrições geométricas, é possível evitar falsos-positivos além de que exigindo menos dedos expostos para a detecção, a taxa de sucesso aumenta (Figura 4.14).

Na Figura 4.14, os motivos de falha na coluna esquerda ocorrem principalmente devido a resultados da segmentação que omitem um dos dedos (maioria dos falsos-negativos), e susceptibilidade à curvas geradas por ruído (no topo à esquerda) o que é tratado através das restrições geométricas sugeridas obtendo resultado coerente (no topo à direita).

Figura 4.13: Padrão usado para a detecção de mão. Para que a mão seja detectada basta que três dedos estejam expostos e algumas restrições geométricas sejam cumpridas.

Figura 4.14: Comparação dos resultados entre o método de detecção usado em (PAN *et al.* 2010) (coluna esquerda) e proposto no presente trabalho coluna direita.

4.5 Rastreamento

A técnica de rastreamento proposta utiliza o conceito de nuvem de features. O principal intuito é garantir que as features rastreadas permaneçam dentro da área da imagem que representa o alvo de interesse. Desta forma, o grupo de features fornecido pela etapa de detecção passa a ser rastreado continuamente, sendo renovado sempre que exista uma deficiência em relação à quantidade de features. Este gerenciamento ocorre através do uso de uma série de heurísticas detalhadas nas subseções seguintes.

4.5.1 Fluxo Ótico

O *frame* subsequente após uma inicialização bem sucedida é usado como entrada para um algorítmo de fluxo ótico (optical-flow), baseado no método piramidal Lucas-Kanade (BOUGUET 2001). A implementação utilizada neste trabalho se econtra disponível na biblioteca de código aberto OpenCV (BRADSKI *et al.* 2008). O algoritmo de fluxo ótico é responsável por, a partir de um conjunto de features de uma imagem anterior, usando imagem atual, encontrar as correspondências destes mesmos pontos na imagem seguinte (*frame* atual). Assim, este algoritmo é capaz de estimar o movimento de regiões da imagem com detalhes sobre a direção e intensidade do mesmo.

Desta foma, o algoritmo de fluxo ótico é executado para cada alvo presente na cena, caso este tenha sido rastreado com sucesso no *frame* anterior, retornando as novas posições das features presentes no *frame* passado e assim possibilitando estimar o movimento do alvo e encontrá-lo no *frame* atual. No entanto, existe a possibilidade de que estas correspondências estimadas não sejam coerentes, devido a razões como oclusões, movimentos bruscos (motion blur) e ruído inerente à captura. Assim, o fluxo ótico está sujeito a encontrar novas posições para as features antigas que causem dispersão da nuvem ao contrário de mantê-la consistente, facilitando uma futura falha de rastreamento.

4.5.2 Remoção

No intuito de superar este problema, uma abordagem de remoção de features é aplicada ao resultado do fluxo ótico. Para cada nova feature encontrada, é verificado se esta pertence ao contorno esperado através de uma checagem de ID, como demonstrado na Figura 4.9. Caso a feature em questão não se enquadre no contorno esperado, esta é removida. Desta forma, se garante que todas as features rastreadas pertençam ao contorno desejado e não se espalhem por outras regiões da imagem dispersando a nuvem.

Outro critério de remoção é aplicado de acordo com a distância entre features de um mesmo conjunto. No decorrer da execução do fluxo ótico, é comum que features originadas de regiões diferentes acabem sendo levadas para um ponto comum. De forma geral, além da influência que os movimentos na cena podem exercer para reunir as features em uma mesma

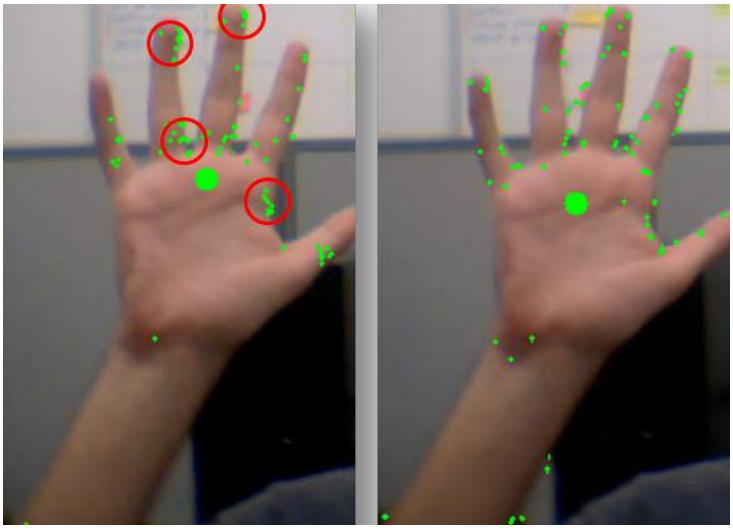


Figura 4.15: Na esquerda o resultado do fluxo ótico sem que seja executada a remoção por proximidade; círculos vermelhos indicam agrupamentos indesejados de features. Na direita está ilustrado o resultado com auxílio da remoção; nesto caso, mesmo que existam features próximas, estas serão removidas em seguida e não mais consideradas ao longo do rastreamento.

região, o algoritmo de fluxo ótico tende a fluir para regiões de alto gradiente, o que eventualmente lança features de locais distintos para uma subregião de alto gradiente na nova imagem. Além disso, após as features se agruparem em uma região, estas não tendem a se separar novamente, pois a partir deste momento passam a compartilhar o mesmo patch ou subregião de origem. Desta forma, as features restantes são checadas no intuito de encontrar pares de pontos muito próximos uns dos outros. Para cada par de features, a distância euclidiana é calculada e caso esta distância seja menor que o limiar (10*pixels* de distância) uma das features é removida. Na Figura 4.15 podem ser visualizados ambos os casos com e sem remoção por proximidade.

4.5.3 Ponto Guia

Após a remoção de features inválidas o rastreamento prossegue com o cálculo do ponto guia. O ponto guia serve como indicativo de preferência, sinalizando que porção da região rastreada é mais importante. Como exemplo, podem ser vistos na Figura 4.15 os pontos guias das mãos em ambas imagens, representados como círculos verdes de maior raio. É o ponto guia por exemplo, que indica por onde deve ser iniciada a busca pela nova região (contorno) de cada alvo em um novo *frame* (Figura 4.9). Em (PAN *et al.* 2010) é sugerido que o ponto guia seja selecionado como sendo o ponto mediano do conjunto, aquele que minimiza a soma das distâncias entre si e os demais pontos, como descrito na Equação 4.5.

Equação 4.5. Cálculo do ponto guia como sendo o ponto mediano de um conjunto de features.

$$P_g = \underset{i}{argmin} \sum_j^i d_i^j$$

onde P_g representa o ponto guia selecionado e d_j^i representa a distância entre a feature i e a feature j. Assim a feature mais central tende a ser selecionada como ponto guia, considerando simultaneamente a densidade de pontos ao longo da região, de forma que o ponto guia tende a se localizar em partes da região com alta densidade de features. Em adição, em (PAN *et al.* 2010) é proposto o uso de pesos baseados na velocidade das features, de forma que uma feature que possua velocidade mais alta que as demais, tenha maior probabilidade de ser selecionada como ponto guia. A aplicação dos pesos é adicionada segundo a Equação 4.6.

Equação 4.6. Adição de um fator baseado na velocidade das features como peso no cálculo do ponto mediano.

$$P_g = \underset{i}{argmin} \sum_j^i d_i^j \times \left(\frac{\widehat{v_j}}{\widehat{v_j} + \widehat{v_i}} \right)^2$$

em que (v_i) e (v_j) representam a velocidade (distância percorrida em *pixels* entre o *frame* passado e o atual) das features i e j, respectivamente.

A aplicação de pesos baseados na velocidade é indicada para determinar o ponto guia de mãos, dado que estas, como alvos de rastreamento, tendem a se mover com muito maior frequência e intensidade do que outros alvos, como faces ou regiões com tom de pele ao fundo da cena. Assim, a escolha de pontos com maior velocidade tende a dar preferência à mão quando por exemplo, dois contornos de dois alvos se confundem por conta de sobreposições de um sobre o outro durante a utilização do sistema.

No entanto, a formulação deste peso proposto traz problemas em relação à *outliers*, que por sua vez são features mal posicionadas após a execução do fluxo ótico, com potencial de obter velocidades muito superiores às demais features. Nestes casos, o ponto guia passa a ser o *outlier* mesmo este não sendo representativo, por simplesmente deter uma alta velocidade, reduzindo significativamente o somatório de distâncias e acabando por se tornar o ponto resultante da minimização. Além disto, mesmo na ausência de *outliers*, este mecanismo de pesos induz que o ponto guia do alvo a cada *frame* seja aquele com maior velocidade, o que gera um comportamento instável. Desta forma, neste trabalho, é proposta uma alternativa para esta questão, selecionando o ponto guia como a média ponderada das coordenadas x e y das features multiplicadas pela relevância (no caso das mãos, a velocidade) de cada uma (Equação 4.7).

Equação 4.7. Cálculo do ponto guia como média ponderada da posição de todas as features do conjunto, usando o fator de relevância de cada uma como peso.

$$P_g.x = \frac{\sum f_i.x \times f_i.relevancia}{\sum f_i.relevancia}$$
$$P_g.y = \frac{\sum f_i.y \times f_i.relevancia}{\sum f_i.relevancia}$$

onde f_i representa a i-ésima feature tendo relevancia como um atributo inerente.

Como resultado, é possível observar um comportamento mais estável do ponto guia, tendendo a se posicionar em partes da região de interesse onde existam mais features com alta relevância. Além de aplicar um peso para o cálculo do ponto guia em alvos detectados como mãos, é proposta neste trabalho também a aplicação de pesos para este mesmo cálculo quando se tratando da face. No entanto, para a face, o fator de relevância utilizado não é a velocidade e sim a idade de cada feature. Dado que a face é um alvo muito mais estável (estático) que a mão, e que também possui regiões de alto contraste de cores, bem estabelecidas como olhos e boca, é usual que as features sejam rastreadas continuamente durante vários *frames*, sem necessidade de serem removidas e assim sendo possível acumular um contador para representar sua idade. Este mecanismo faz com que mesmo se uma face e uma mão estejam compartilhando o mesmo contorno temporariamente, o ponto guia de cada alvo permaneça nas regiões específicas de cada um dos seus respectivos alvos, permitindo que após a sobreposição seja terminada, o rastreamento continue normalmente (Figura 4.16). Por fim, alvos indefinidos não possuem critério de relevância, associando à todas as features o valor de relevância 1 como padrão.

Na Figura 4.16 está representada em uma sequência de três *frames* o rastreamento de três alvos. A representação de cada alvo conta com um retângulo (sinalizando a região de interesse), uma série de pequenos pontos que ilustram as features rastreadas e um círculo indicando a posição do ponto guia. Cada alvo é ilustrado com uma cor específica que denota o seu tipo. Assim, a cor usada para faces é azul, para mãos é verde, e para alvos indefinidos é usada a cor amarela. Primeiramente, é possível observar que mesmo em uma pequena sequência de três *frames* o ponto guia da mão (círculo verde) calculado através do método proposto em (PAN *et al.* 2010) sofre alterações bruscas demonstrando instabilidade que resultam em frequentes tremidas (*jitter*). Enquanto que no método proposto, como resultado da média ponderada das features, o ponto guia demonstra um comportamento mais estável, permanecendo em uma região específica da palma da mão.

A sequência ilustrada é resultado da captura de um momento da cena em que os alvos da mão e da face passam a se confundir, pois compartilham a mesmo contorno da região de cor de pele. Desta forma, features do alvo da mão passam a ser extraídas na região da face e features do alvo da face passam a ser extraídas na região da mão fazendo com que as regiões de interesse (retângulos) dos dois se confundam. No entanto, através do método proposto, no *frame* 786 a após a separação dos contornos dos alvos, o ponto guia permanece sobre a mão, fazendo com que a região de interesse se estabilize e o rastreamento continue normalmente. Enquanto que através do cálculo proposto por (PAN *et al.* 2010) é observado que o ponto guia da mão passa a ocupar uma posição sobre a face, levando a busca por um novo contorno a selecionar o contorno da face para o alvo da mão. Desta forma, o contorno da mão deixa de ser rastreado como um alvo antigo e passa a ser considerado um novo alvo de caráter indefinido até que seja identificado novamente como uma mão.

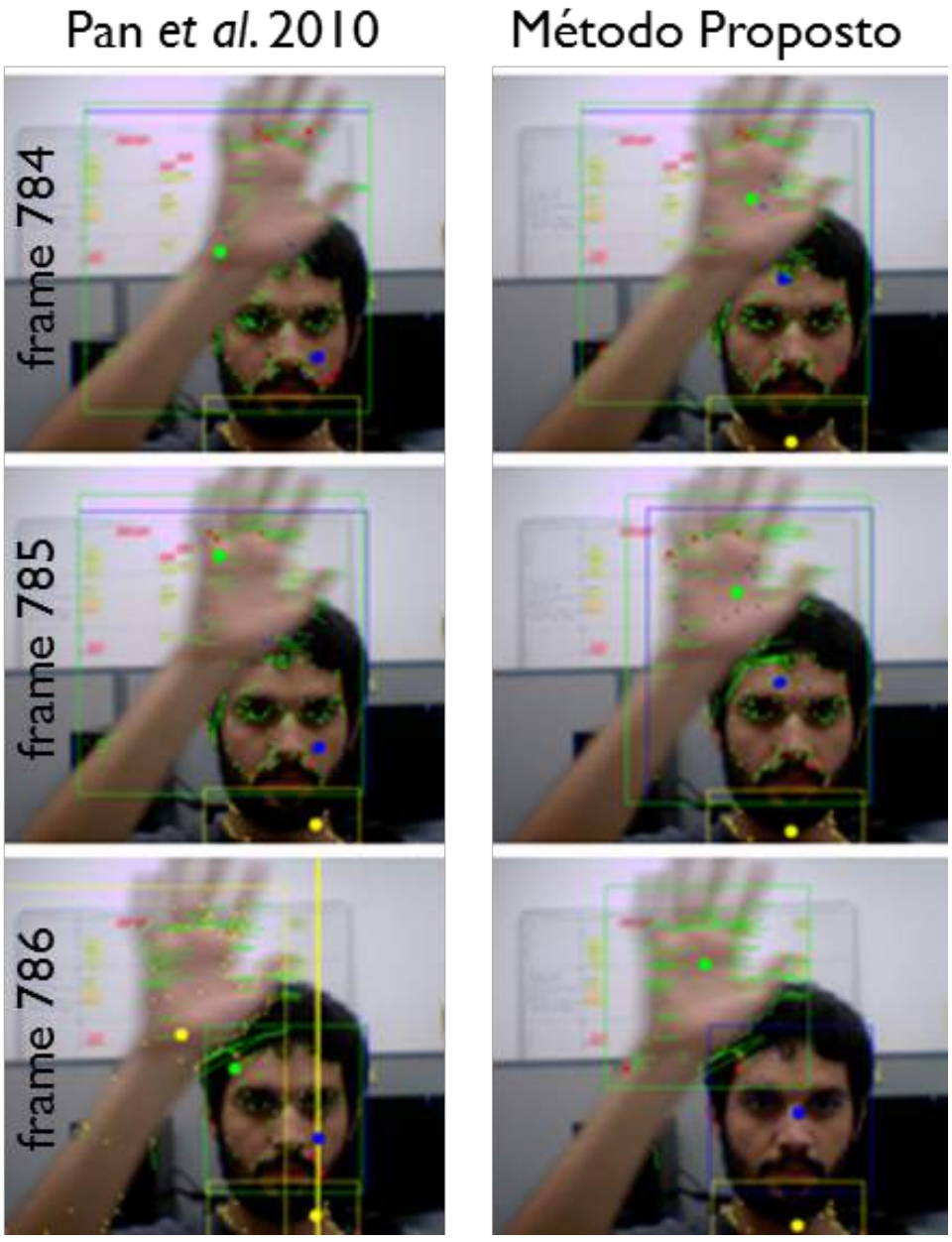


Figura 4.16: Análise em paralelo entre o método de cálculo do ponto guia proposto e o método apresentado em (PAN *et al.* 2010). Amarelo: alvos indefindos. Azul: faces. Verde: mãos.

4.5.4 Realocação

Após o cálculo do ponto guia, um procedimento de realocação das features mais distantes é realizado com o intuito de manter a nuvem coesa. Esta etapa busca pelos 5% de features mais distantes do ponto guia e as transfere para novas posições intermediárias, fazendo-as ocupar o ponto médio entre a posição do ponto guia e as suas próprias antigas posições. Esta etapa contribui para que em casos como os mostrados na Figura 4.16, as features sejam auxiliadas a se concentrar em torno do ponto guia. Na Figura 4.16, as features a serem realocadas são renderizadas como pontos vermelhos.

4.5.5 Complemento

Em seguida uma nova extração de features é realizada para suprir em quantidade aquelas que foram removidas previamente. A etapa de complemento só pode ser realizada após o cálculo do ponto guia, pois caso contrário estaria influenciando fortemente no seu resultado a partir de novas features com baixa relevância. Esta etapa de extração é similar àquela realizada no momento da detecção de um novo alvo. Através do algoritmo GFTT (SHI *et al.* 1994), novas features são extraídas no mesmo contorno do alvo rastreado. A quantidade de features extraída é determinada pelo tamanho do contorno (perímetro), no entanto, para evitar que uma quantidade massiva de features seja extraída em ambos alvos no momento em que dois contornos se fundem (Figura 4.16), uma função atenuante é aplicada, considerando o tamanho do contorno do alvo nos últimos 200 *frames*. 200 *frames* é um valor estipulado para se obter uma amostragem satisfatória do comportamento do alvo, e assim estimar a quantidade de features considerando parte do histórico do alvo e não somente o quadro atual. Este número se comportou de forma satisfatória durante os testes realizados, no entanto pode ser ajustado por exemplo em caso que a instabilidade é recorrente e o tamanho do contorno não é considerado representativo mesmo analisando o alvo em 200 quadros passados. É esperado que o número de features de um alvo seja equivalente ao $perimetro16$. Esta divisão visa obter um número suficiente de features para enquadrar uma destas a cada 16 *pixels* do contorno, obtendo assim uma distribuição razoável de features ao longo do contorno. A obtenção deste número também se deu de forma empírica e este pode ser ajustado visando o rastreamento de mais ou menos features. No entanto vale ressaltar que com poucas features o rastreamento se torna mais falível visto que se torna mais fácil a perda de todas as features, enquanto que ao adicionar mais pontos é possível que o rastreamento se torne lento. Desta forma o valor estipulado em relação ao contorno é um valor que se enquadra como um meio termo dentre estes aspectos e que apresenta um bom comportamento neste cenário aplicado. Em relação à atenuação no cálculo do número máximo de features para cenários com altas variações, considera-se o histórico do alvo como mencionado anteriormente, assim, caso a variação de perímetro esteja muito acima da média observada nos últimos 200 *frames*, o número de features é calculado a partir de uma pequena variação sobre o perímetro antigo.

Figura 4.17: Exemplo de reinicialização automática. Através da predição de uma nova região de interesse, features são extraídas após uma falha para que o alvo continue sendo rastreado.

4.6 Avaliação

Após o rastreamento dos alvos, é possível que os resultados atingidos não sejam satisfatórios. De fato, é esperado também que a etapa de rastreamento, apesar de todas as heurísticas implementadas, por ventura apresente falhas como associar dois alvos distintos a uma mesma região ou simplesmente perder o alvo completamente. No intuito de contornar problemas deste tipo, uma fase de avaliação é executada.

Primeiramente são percorridos os alvos em busca daqueles que perderam todas as features. Apesar do número de features por alvo ser suficiente para cobrir toda região, existem casos em que todas as features são removidas por terem suas novas posições avaliadas fora do contorno do alvo. Assim, uma tentativa de reinicialização automática é realizada nestes casos. As últimas duas regiões de interesse (Region Of Interests ou ROIs), as quais representam retângulos que encobrem as features dos alvos (bounding boxes), são utilizadas para prever a posição da próxima ROI (Figura 4.17). Então, features são extraídas internamente a estas ROIs, com a condição de que façam parte de regiões de cor de pele. Caso um número mínimo de 10 *features* seja extraído, o alvo é considerado como reinicializado e o rastreamento prossegue; caso contrário, o alvo é perdido.

Em seguida, durante a avaliação, são procurados pares de alvos que compartilhem o mesmo contorno. Em caso positivo, é verificado se pelo menos um dos alvos se encontra com 80% ou mais de sua ROI encoberta pelo outro alvo, pois nestes casos, dois alvos se encontram rastreando a mesma região simultaneamente, demandando um tempo de processamento extra sem adicionar nenhum ganho ao rastreamento. Assim, nestes casos, os alvos são combinados, e aquele que possuía o menor conjunto de features é eliminado.

4.7 Refinamento

Por fim, após todo o processo de detecção, rastreamento e avaliação estarem completados no *frame* atual, os dados são preparados para que uma aplicação use-os como entrada na interação. Cada alvo representa um conjunto de features com um ponto guia e uma ROI associados, além de ser atribuído a um contorno de uma região de cor de pele na imagem de entrada. A etapa de refinamento faz uso destes dados para fornecer informações extras sobre o alvo como, por exemplo, a orientação (através do cálculo de momentos) e uma elipse que compreenda o alvo de forma aproximada. Além disso a etapa de refinamento pode fornecer uma aproximação de uma coordenada 3D do alvo estimada através de uma função sobre o valor da área do seu contorno. Por fim, a etapa de refinamento também pode prover uma suavização da coordenada do ponto guia, para que a aplicação não precise lidar com as tremidas (*jitter*) inerentes à estimativa deste ponto.

O resultado final de um *frame* pode ser visto na Figura 4.18, em que três alvos são rastreados simultaneamente: duas mãos e uma face. No caso da Figura 4.18, a face está sendo rastreado como um alvo não indentificado visto que para a detecção da face, o algoritmo utilizado necessita de uma região livre mínima em volta da cabeça a qual não existe na figura apresentada, pois a região da testa do usuário está cortada pelo limite superior da imagem. Ou seja, as informações relativas ao alvo que representa a face estão renderizadas em amarelo pois a face ainda não foi devidamente detectada e no momento é considerada como um alvo indefinido. Os pontos vermelhos representam features a serem realocadas, os círculos de maior raio representam os pontos guia de cada alvo e os retângulos suas respectivas ROIs.

4.8 Atualização

O último processamento realizado em um *frame* é a atualização dos alvos. Cada alvo rastreado armazena o seguinte conjunto de informações:

- Conjunto de features;
- Número máximo de features permitido;
- Contorno: perímetro da região de cor de pele em que o alvo se encontra;
- ROI: bounding box das features do alvo;
- Ponto Guia;
- Elipse: indica a posição e orientação do alvo;
- Tipo: face, mão ou indefinido;

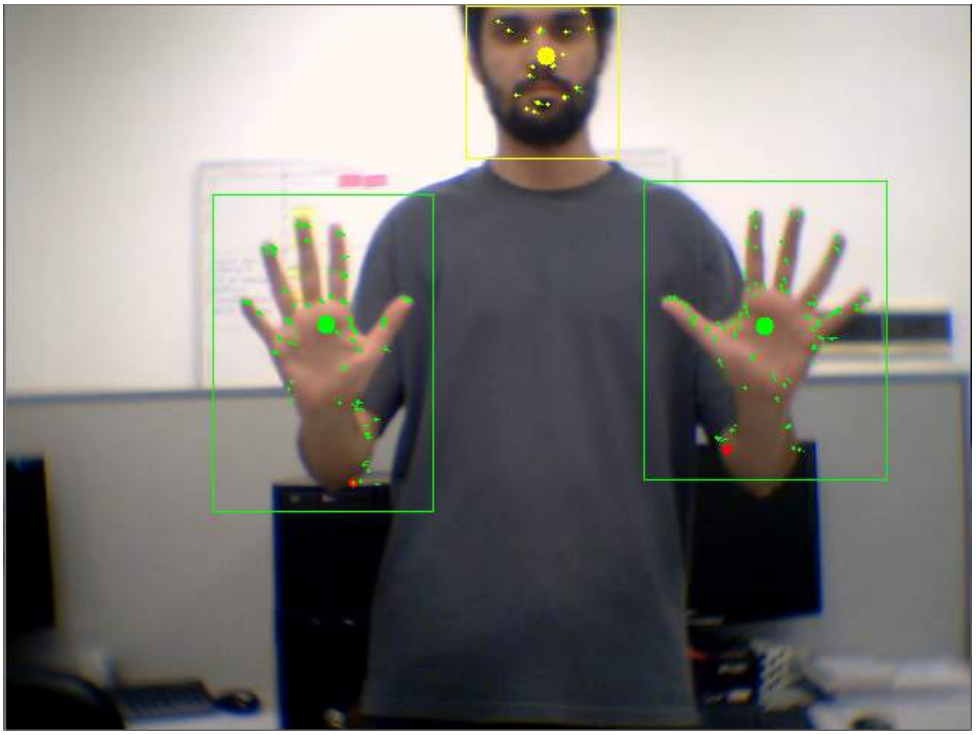


Figura 4.18: Resultado final do rastreamento após a execução de todas as etapas em um *frame*.

- Contadores de detecção: indicam a quantidade de vezes necessária para que o alvo seja definido.

Ao final do processo de rastreamento como um todo, todos os alvos são atualizados para que os dados do *frame* atual sejam armazenados então como informações do *frame* anterior, visto que o processo foi finalizado e um novo *frame* será capturado em seguida. Cada alvo armazena 200 versões anteriores de si, com o intuito de manter um histórico para que inferências futuras sejam facilitadas em relação, por exemplo, ao tamanho do perímetro esperado.

5

ESTUDO DE CASO: *GUITARS ON AIR*

Como estudo de caso para o HAFT, um jogo chamado Guitars on Air, baseado em interações por gestos, é proposto tanto como uma aplicação de entretenimento quanto como uma ferramenta para avaliação do rastreamento de mãos. Apoiado no conceito de air guitar aliado a interfaces de jogos musicais, o estudo de caso proposto provê uma análise entre três diferentes métodos de rastreamento de mãos. Além disso, o conceito explorado por Guitars on Air tem como objetivo inserir estas técnicas de rastreamento em um cenário real, no qual o usuário está livre para interagir com a aplicação da melhor forma que considerar. Assim, ao contrário de tarefas restritas para avaliar precisão e robustez do rastreamento, os usuários devem jogar Guitars on Air para que dados coletados durante o desempenho sejam usados na avaliação.

5.1 Rastreamento de Faces

Durante a fase de desenvolvimento da ferramenta HAFT vários testes preliminares foram realizados para averiguar o comportamento da ferramenta em alguns cenários. Durante estes testes o rastreamento de faces bem como rastreamento de mãos foi testado e analisado.

De forma geral, durante os testes realizados ao longo do desenvolvimento, o rastreamento de faces se comportou de forma superior em relação ao rastreamento de mãos. Para rastrear faces, o funcionamento do algoritmo presente em HAFT se mostrou mais robusto devido ao fato de que as features presentes na nuvem se mantêm com mais facilidade sobre a face visto que a face oferece regiões de alto contraste mais fáceis de serem seguidas ao longo dos quadros capturados. A Figura 5.1 ilustra uma série de resultados do rastreamento de faces, retratando diversas características do mesmo.

Assim, de forma geral o rastreamento se comporta de forma consideravelmente mais estável quando se trata do alvo face. Por outro lado, ao rastrear a mão, uma série de desafios adicionais surge devido à pouca estabilidade que a mão demonstra em relação ao seu comportamento durante a interação. A mão muda de configuração (posicionamento dos dedos) com muita frequência durante a cena. Além disso, a velocidade média da mão durante a interação do usuário tende a ser muito maior do que a da face. Por fim, a mão não dispõe de “artefatos” ou

Figura 5.1: Resultados do rastreamento de faces através da ferramenta HAFT, sob diversas condições.

partes com alto contraste ao contrário da face que engloba as regiões dos olhos e da boca por exemplo. Estas regiões da face são fáceis de serem rastreadas por features e como no caso da mão isto não ocorre, a dificuldade para a permanência (rastreamento contínuo) das features na mão é maior, ocasionando uma perda de features muito maior.

Todos estes pontos tornam a mão um alvo consideravelmente mais difícil de ser rastreado. Vale ressaltar que o rastreamento utilizado tanto para mãos como para faces é o mesmo, diferindo simplesmente no uso de um peso distinto para o cálculo do ponto guia. Assim, quaisquer aprimoramentos apontados para o rastreamento de um dos alvos (da mão por exemplo) podem também ser aplicados para encontrar melhorias em ambos os casos. Desta forma, o estudo de caso proposto neste trabalho visa estudar o rastreamento de mãos com o intuito de entender as dificuldades da ferramenta em um cenário adverso, pondo à prova o rastreamento proposto e assim buscando entender quais os principais pontos em aberto para projetar futuras melhorias para o HAFT como um todo. Desta forma o estudo de caso aqui apresentado como o jogo Guitars on Air visa avaliar ferramentas de rastreamento de mãos unicamente. No entanto é importante frisar que a partir da avaliação de HAFT como um rastreador de mãos é possível entender e incorporar melhorias associadas à ferramenta como um todo.

5.2 Avaliação do Rastreamento

O uso de técnicas para o rastreamento de mãos é de grande aplicabilidade para interfaces naturais, no entanto é necessário analisar quais técnicas são propícias para determinada aplicação visto que de forma geral cada método presente no estado da arte da área apresenta restrições de desempenho relacionadas a um ou outro aspecto. Desta forma, a avaliação de métodos de rastreamento de mãos deve ser capaz de analisar aspectos técnicos relativos ao uso final da técnica, os quais são determinantes para o sucesso do rastreamento nos mais distintos campos de aplicação visto que os aspectos analisados são genéricos e definem a qualidade do rastreamento independentemente do seu uso.

Além disso, a avaliação pode se dar de forma subjetiva, através de estudos de caso voltados para a experiência do usuário, por exemplo com a aplicação de questionários qualitativos. A análise subjetiva serve de suporte para o entendimento do comportamento de determinada técnica de rastreamento em um cenário real, tendo como base de análise a própria impressão do usuário em relação ao seu controle sobre a aplicação via o rastreamento utilizado. Desta forma, a análise subjetiva carrega uma grande importância associada ao uso em si da técnica, revelando uma perspectiva de aplicabilidade da técnica mais realista.

5.3 Aspectos Técnicos

Os principais aspectos técnicos relacionados a métodos de rastreamento de mãos são o tempo de resposta, a precisão e a robustez aos diversos casos de falha, como nas mudanças de

iluminação, movimentos rápidos, mudança da configuração dos dedos e oclusões parciais.

- Tempo de Resposta: o tempo de resposta do rastreamento é de extrema importância para definir a aplicabilidade do mesmo. Um rastreamento que ocorre em tempo real, a uma taxa mínima de 30 quadros por segundo, permite uma interação suave e agradável para o usuário. No entanto, visto que o rastreamento em si não define a aplicação, mais tempo de processamento será demandado pelas camadas que farão uso dele. Assim, quanto menor o tempo de resposta do rastreamento, mais tempo poderá ser utilizado pela aplicação de modo a ainda respeitar o limite de uma interação em tempo real.

- Precisão: a precisão é o aspecto relacionado a problemas como drift e jitter. O drift ocorre quando o rastreamento, por usar informações de quadros anteriores para estimar a nova posição da mão, tende a retornar como resultado para o quadro atual um posicionamento intermediário entre as posições reais da mão no quadro anterior e atual. Desta forma, o drift resulta em um atraso na interação mesmo não representando uma carga de processamento maior pois o rastreamento se comporta como se estivesse seguindo a mão e não como se identificasse sua posição exata no quadro atual. Já o jitter está relacionado a imprecisão presente no rastreamento que é apresentado como sucessivas tremidas da posição encontrada para representar a mão. Este ocorre principalmente em métodos que não fazem uso da informação prévia para encontrar a posição atual do alvo rastreado, apresentando um resultado que se mostra coerente se analisado isoladamente em relação unicamente ao quadro atual, no entanto durante a sequência de quadros revela um aspecto instável com tremidas que podem dificultar a interação.

- Robustez: a robustez de uma técnica de rastreamento trata basicamente das condições de falha às quais ela está sujeita. As condições que em geral implicam em falhas são mudanças de iluminação, mudanças de configuração dos dedos para técnicas que fazem uso do modelo 3D da mão, movimentos rápidos da mão e oclusões parciais em casos que a mão não se mostra completamente para o dispositivo de captura. Em geral, é comum que algoritmos de rastreamento tratem estes problemas com uma recuperação de falhas através de técnicas de detecção, pois muitas vezes estes problemas são pontuais durante a interação. No entanto, o ideal é que o rastreamento persista mesmo sob estas condições adversas para não depender unicamente de um método de detecção e tampouco gerar uma quebra na interação.

5.4 Jogos Musicais e Interação por Gestos

O conceito de jogos musicais se refere a videogames nos quais a jogabilidade está associada, em grande parcela, à execução de uma trilha sonora. Na última década os jogos

Figura 5.2: Metáfora de interação proposta em Frets on Fire (BARBANCHO *et al.* 2009), na qual o usuário, ao segurar o teclado, simula que está vestindo uma guitarra elétrica.

musicais passaram a ocupar um espaço significativo no mercado de jogos. Adagio (GAMES (2008)), FreQuency (MUSIC (2001)) e DJ Hero (ACTIVISION (2010)) são alguns exemplos que podem ser citados, variando de simples implementações em Action Script voltadas para a Web, até títulos mais sofisticados, destinados à última geração de consoles.

Como reflexo, neste nicho, afloram avanços para uma interação mais natural. Em Frets on Fire (BARBANCHO *et al.* 2009), um jogo de código aberto para PCs, é sugerida uma forma alternativa para o uso do teclado, na qual o usuário é levado a segurá-lo com ambas as mãos, simulando que possui uma guitarra elétrica (Figura 5.2), utilizando assim o conceito de metáfora sugerido como artifício para interações naturais (VALLI 2005).

Jogos como Guitar Hero (WIXON 2007) e Rock Band (HARMONIX 2007) levaram este conceito um passo adiante, adotando versões de joysticks no formato de versões menores de instrumentos musicais, como ilustrado na Figura 5.3.

De forma geral, a interação em jogos digitais é uma área em constante evolução, passando por dispositivos de entrada simples (com poucos botões), a controles mais sofisticados, incluindo características como sensores de movimento, e rastreamento do corpo humano através da captura de padrões infravermelhos projetados (NINTENDO 2006; SONY 2010; MICROSOFT 2010).

Mais além, abordagens baseadas em movimento podem ser aplicadas a jogos musicais como Guitar Hero e Rock Band, através do conceito de tocar uma guitarra imaginária, que

Figura 5.3: Dispositivos de entrada para jogos musicais como Guitar Hero (WIXON 2007) e Rock Band (HARMONIX 2007).

consiste em simular os gestos de um guitarrista sem que exista um instrumento fisicamente presente. Desta forma, pode-se aplicar a prática de air guitar (CRANE 2006) como meio de interação.

Em (MÄKI-PATOLA *et al.* 2006) uma guitarra virtual é proposta, sendo controlada apenas pelos movimentos das mãos do usuário e provendo feedback sonoro sempre que o usuário demonstra intenção de tocar a guitarra movendo as mãos. Em (PAKARINEN *et al.* 2008) é proposta uma guitarra virtual similar, com saída de som realística e com rastreamento baseado em emissão e reflexão de luz infravermelha. Mais adiante, em (FIGUEIREDO *et al.* 2009), uma biblioteca de código aberto é provida, oferecendo como suporte à aplicações uma interface de gestos para guitarras virtuais (Figura 5.4).

Estes exemplos se mostraram bem sucedidos como ferramentas de interação, revelando, na maioria dos casos, satisfação e diversão por parte do usuário. No entanto, estes trabalhos citados fazem uso dos simuladores de guitarra virtual muito mais no sentido de uma interface de expressão musical, deixando inexplorada a faceta destas interfaces como dispositivos de entrada para jogos musicais. Além disso, nestes casos, o rastreamento das mãos é realizado através de dicas visuais (luvas de cor de destaque ou peças de metal para a reflexão de luzes infravermelhas). O estudo de caso proposto neste trabalho apresenta um jogo chamado Guitars on Air, que unifica a prática de air guitar com um foco de entretenimento interativo, levando o conceito de jogos musicais a um novo nível de interação.

5.5 Mecânicas do Jogo

As mecânicas de jogo presentes em jogos musicais tem correlação direta com os desafios apresentados anteriormente como aspectos técnicos críticos para métodos de rastreamento de mãos. Este fato encoraja o uso de jogos deste tipo para a avaliação destas técnicas. Os jogos

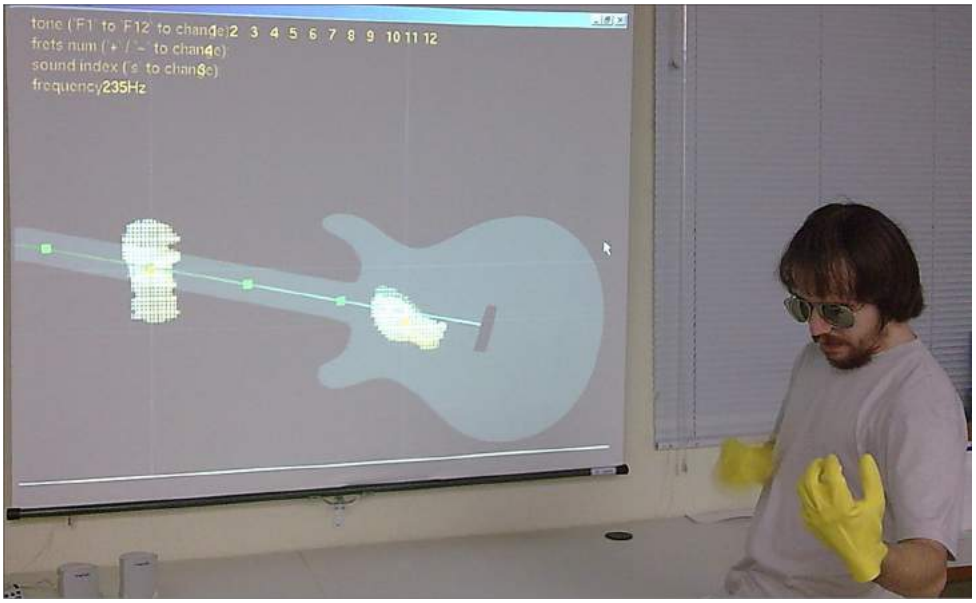


Figura 5.4: Controle de uma guitarra virtual através do uso de luvas amarelas (FIGUEIREDO *et al.* 2009).

musicais que de forma geral abstraem o uso de uma guitarra, com um joystick em formato de guitarra de plástico por exemplo, requerem que o usuário execute um comando tendo a mão esquerda em uma determinada posição e a direita executando o gesto de tocar dentro de um curto espaço de tempo. Este mesmo conceito é aplicável à interação com uma guitarra virtual através de gestos espaciais, com base no conceito de air guitar.

Guitars on Air é um jogo de música baseado em interações por gestos. Assim como em Guitar Hero e Rock Band, o objetivo neste jogo é executar os movimentos indicados por indicadores (no caso do Guitars on Air, setas coloridas) que surgem continuamente, enquanto a música tema de sessão é tocada. A execução dos gestos deve ocorrer de forma sincronizada com os elementos sonoros da música. Desta forma, o usuário é capaz de associar diretamente o movimento realizado com a música tocada, simulando a influência de seus gestos no resultado sonoro da experiência.

Guitars on Air foi construído sob a utilização de uma biblioteca de código aberto desenvolvida pelo autor e provida em (FIGUEIREDO *et al.* 2009), no qual uma guitarra virtual é simulada rastreando apenas as mãos do jogador. A seguir, como forma de simplificar o detalhamento da proposta, será tratado como um padrão na descrição do jogo, que a mão esquerda do jogador é a que remete ao braço da guitarra virtual, e a mão direita é aquela por sobre o corpo da guitarra.

5.5.1 Gestos Espaciais

Além da simulação de uma guitarra virtual, em (FIGUEIREDO *et al.* 2009) é descrito um método de interação no qual é implementado o reconhecimento de dois tipos de movimentos das mãos. O primeiro movimento é relacionado com o balanço vertical da mão direita. Este

movimento representa a intenção do usuário de tocar a guitarra virtual, passando a mão direita por sobre as cordas imaginárias, em um movimento análogo àquele feito em uma guitarra real quando se deseja tocar todas as 6 cordas em um único gesto. O segundo movimento está relacionado à translação horizontal da mão esquerda do usuário (que está sobre o braço da guitarra), e é análogo ao movimento real do guitarrista quando este deseja mudar de tons graves para agudos (e vice-versa) enquanto tocando notas ou acordes.

A simplicidade dos gestos considerados é fundamental para atingir uma interação bem sucedida, que ocorra com naturalidade. O usuário não deve se preocupar em aprender e lembrar movimentos complexos durante o uso do aplicativo, pois a carga cognitiva deve ser a menor possível (VALLI 2005). Então, os gestos usados devem ser associados imediatamente a uma tarefa geralmente conhecida, e que também seja fácil o suficiente de ser realizada sem experiência prévia, revelando uma curva de aprendizagem que é iniciada em um ponto alto de familiarização relacionada à tarefa.

Além disso, outros conceitos de interação natural foram levados em consideração no intuito de formalizar este conceito de interação gestual. Questões como a liberdade do usuário em relação à maneira de executar o gesto e como o ambiente pode ser explorado encoraja o uso dos gestos descritos para interações mais espontâneas.

5.5.1.1 Movimentos da Mão Direita: Setas

Com o intuito de incrementar a interação do jogo, tornando-o mais dinâmico e também mais coerente com a metáfora proposta, as duas direções do movimento vertical da mão direita foram consideradas: de cima para baixo e vice-versa. Como uma indicação de qual direção deve ser usada durante o movimento, cada comando é expresso por uma seta apontando para cima ou para baixo, correspondendo ao movimento da mão direita.

Cada seta surge no topo de uma plataforma e passa a descer por esta até atingir o limite inferior da mesma (Figura 5.5). O momento correto para realizar o comando demandado por cada seta é aquele em que a seta atinge a região inferior da plataforma, e é um momento sincronizado com o elemento sonoro relativo à seta. Desta forma, a execução do movimento ocorre no mesmo momento em que este som se torna evidente e assim, o jogador sente, de certo modo, que está tocando a música. Para induzir esta sensação com maior intensidade, um retorno (feedback) sonoro é implementado, de forma que quando o jogador comete erros (não coordena o movimento com a seta no tempo correto) o volume da música diminui, da mesma forma que quando o movimento de uma seta é executado corretamente o volume é elevado.

O momento correto de realizar o movimento é indicado por uma linha base (Figura 5.5) da plataforma em perspectiva (que é o suporte plano pelo qual as setas descendem), na parte inferior da tela. A partir deste mecanismo de setas descendentes, o usuário sabe de antemão o próximo comando que deve ser executado e pode preparar a direção do balanço da mão da direita, sendo capaz de realizá-lo dentro de um intervalo de tempo de 200 ms. Ou seja, durante a interação, se tratando da seta que deve ser executada é possível realizar o gesto requerido dentro

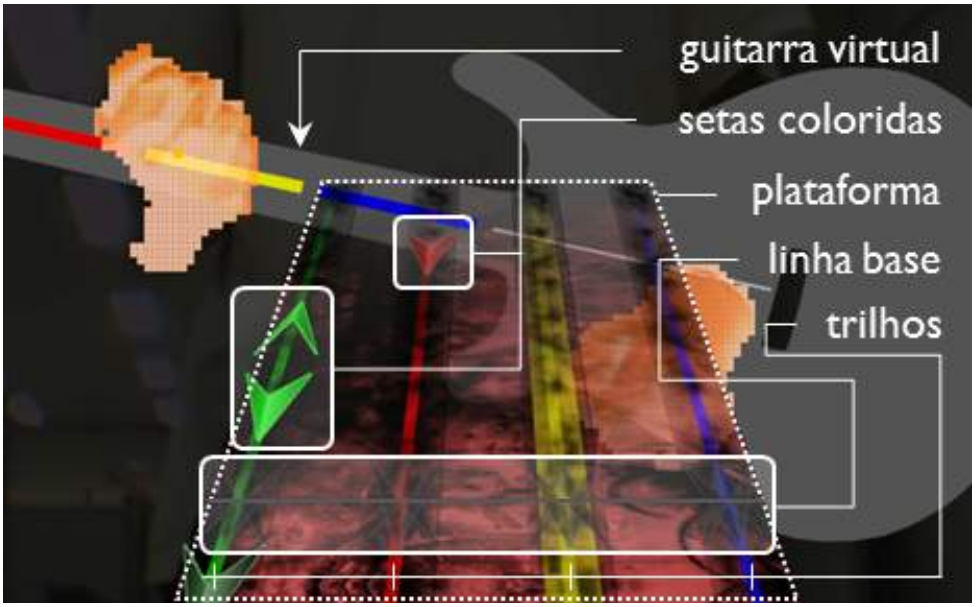


Figura 5.5: Primeira interface contruída para o Guitars on Air.

de um intervalo de tempo de 200 ms. Ao se aproximar da linha base, a se torna executável e passa 100 ms até estar exatamente sobre a linha, após isso a seta ainda se encontra executável após os próximos 100 ms, assim o usuário tem uma margem de execução que o permite antecipar ou atrasar um pouco a execução do gesto em relação ao momento perfeito que seria quando a seta está exatamente sobre a linha base.

O intervalo total de 200 ms foi definido para dar uma boa margem de tempo para a execução dos usuários. Este tempo pode ser ajustado afim de aumentar a restrição para que os gestos sejam mais precisos no aspecto temporal, ou ser menos restrito para dar uma maior margem de tempo para usuários com dificuldade de executar o gesto como uma reação precisa. Além de dar uma boa margem de tempo para a execução do gesto da mão direita, este intervalo permite captar o atraso associado à interação como um todo, revelando uma característica importante do rastreamento em análises comparativas. Isto é, através de uma comparação entre ferramentas, mantendo os mesmos parâmetros para a interação e aplicando os mesmos testes para cada usuário. Desta forma é possível entender o atraso de cada ferramenta de forma comparativa. Este aspecto é relevante de ser analisado visto que o atraso real de um rastreador não é medido somente pelo tempo de resposta, mas também está associado ao drift do rastreamento. Este e outros conceitos podem ser percebidos em uma das primeiras interfaces gráficas construídas para o Guitars on Air, ilustrada na Figura 5.5.

Ainda sobre a mecânica de interação dos gestos realizados pela mão direita, é importante ressaltar que o usuário não é penalizado por mover as mãos caso não exista a demanda eminente de execução de um comando, ou seja, nenhuma das setas se encontra na região da linha base. Assim, o usuário está livre para se mover de qualquer forma desejada, devendo apenas focar a atenção nos comandos por vir para executá-los corretamente. Esta estratégia simplifica a interação do usuário, propondo uma tarefa clara para ser executada como o jogador preferir.

5.5.1.2 Movimentos da Mão Esquerda: Trilhos

O movimento da mão esquerda é usado como uma metáfora de performances de guitarristas, por exemplo, em momentos de execução de um solo ou diversos acordes em diferentes regiões do braço da guitarra. Assim, cada comando por vir (representado por uma seta, como descrito anteriormente), deve ser executado em uma região específica do braço da guitarra virtual. A mão esquerda deve ser então colocada na posição horizontal indicada no momento exato que a mão direita realiza o balanço (movimento vertical na direção requisitada pela seta executada). Trilhos com cores diferentes são usados como uma indicação das diferentes posições possíveis, e assim cada seta vem apoiada no seu trilho correspondente, assumindo suas cores (Figura 5.5).

Cada trilho, e suas setas correspondentes são desenhados em cores diferentes para facilitar a orientação do usuário, de forma que seja de fácil identificação a posição do braço da guitarra em que cada seta deve ser executada. Apesar de algumas exceções como no jogo Adagio (GAMES 2008), esta mecânica é bastante difundida na maioria dos jogos musicais (HARMONIX 2007; ACTIVISION 2010). Além disso, o trilho selecionado, ou seja, aquele em que a mão esquerda repousa, é iluminado e alargado como uma indicação para o usuário. A seleção do trilho é a relativa posição horizontal da mão esquerda do usuário. Assim, cada vez que o usuário mover a mão esquerda em uma distância suficiente para mudar de trilho, este novo trilho fica em foco e o anterior retorna ao seu formato normal, com o aspecto de uma linha fina.

O número de trilhos pode ser ajustado para outros valores (no caso da Figura 5.5 este número é 4), entretanto, durante o desenvolvimento foi empiricamente verificado que este número não pode crescer muito. De fato, requerer do jogador a mudança de posição entre cinco trilhos já é uma tarefa complexa e pode influenciar de forma negativa a experiência. De forma geral, quanto mais trilhos são apresentados, mais refinada deve ser a habilidade do usuário, e já que o objetivo é construir um jogo com uma usabilidade de alto alcance, não é interessante introduzir usuários ao Guitars on Air com um nível de dificuldade muito alto. Desta forma, todos os testes com usuários foram realizados usando uma guitarra virtual de três trilhos, um verde, um vermelho e um amarelo.

Estas mecânicas de interação estão correlacionadas com os aspectos técnicos citados previamente, pois exigem do rastreamento os três principais requisitos para que a interação ocorra com sucesso: baixo tempo de resposta, precisão e robustez. Mais além, as mecânicas associadas a este tipo de jogo permitem medir os problemas de determinada técnica de interação. Um alto tempo de resposta bem como o problema de drift influenciam diretamente de forma negativa no gesto da mão direita ao tentar tocar a guitarra, visto que o espaço de tempo para esta execução é pequeno. Problemas como jitter atrapalham muito o usuário ao tentar selecionar um trilho com a mão esquerda. Estas mecânicas de interação exigem que o rastreamento seja robusto à movimentos rápidos, pois caso contrário muitas falhas irão ocorrer já que o usuário está constantemente movendo as mãos com velocidade. Ou seja, dentre os aspectos técnicos citados como desafios para o rastreamento de mãos, é possível avaliar a maioria destes com a

aplicação de um jogo musical. Todos estes aspectos citados podem ser medidos ao armazenar os casos de erro do usuário durante a interação com o jogo.

5.5.2 Medição da Performance do Usuário

No intuito de medir o desempenho do jogador, assim como fornecer ao mesmo um estímulo para atingir o seu melhor desempenho em uma sessão de jogo, um sistema de pontuação foi desenvolvido. Seguindo os modelos apresentados em Guitar Hero (WIXON 2007) e Rock Band (HARMONIX 2007), três principais medições foram armazenadas ao longo da sessão de jogo.

A primeira se trata do número consecutivo de acertos (streak), armazenando e incrementando um contador cada vez que o jogador faz um movimento correto e zerando o mesmo cada vez que um erro ocorre. No final da sessão de jogo (fim da música jogada), o valor mais alto (max streak) obtido neste contador é registrado. Além da habilidade do jogador e domínio da interação, este tipo de medição é um indicativo de quão focado e imerso na experiência o jogador está, sendo capaz de realizar continuamente movimentos corretos por um longo período de tempo. Além disso, tratando Guitars on Air na perspectiva de uma ferramenta para avaliação de rastreadores de mão, este contador acaba revelando informações relacionadas à estabilidade do rastreamento, visto que no caso de um rastreador que perde os alvos com certa frequência ao longo da interação limita o usuário a atingir um max streak baixo.

Outra importante forma de medir o desempenho é a porcentagem de setas executadas corretamente em relação à quantidade de setas perdidas. O motivo de uma seta não ser executada corretamente pode variar. Assim, para cada seta há o armazenamento de se ocorreu um acerto ou um erro, e no caso de erro qual foi a razão dentre as seguintes opções:

1. A mão esquerda não estava ocupando o trilho correto no momento em que a direita cruzou a corda da guitarra virtual (erro batizado de wrong rail);

2. A mão direita cruzou a corda na direção inversa à requerida pelo comando visualizado na seta (erro batizado de wrong direction);

3. A mão direita não cruzou sobre as cordas da guitarra virtual no espaço de tempo solicitado (erro batizado de sleep).

Desta forma, é possível saber não apenas se o jogador foi bem no seu desempenho, assim como por quais razões e com que intensidade e frequência ocorreram suas falhas. Mais além, quando a mão direita cruza a linha base (demonstrando a intenção de tocar) e existe uma seta a ser acertada (dentro dos limites da linha base), independente de o movimento configurar um acerto ou erro, o lapso de tempo entre o movimento e a posição média de onde a seta está é armazenado. Este último dado compreende o atraso entre o gesto do usuário e o momento ideal para que a seta seja executada.

Por fim, um medidor de pontuação (placar) é armazenado. A pontuação consiste de um valor derivado de alguns conceitos combinados, representando o desempenho em geral do jogador, e tem como principal objetivo estimular o progresso do mesmo. O placar é inicializado com o valor 0 e incrementado em 1 sempre que o jogador executa um movimento corretamente. Caso os últimos cinco movimentos foram corretos, os próximos pontos serão multiplicados por 2. Se os últimos dez forem corretos, o multiplicador passa de 2 para 3, e caso os últimos quinze movimentos forem corretos, o multiplicador será 4. Sempre que o jogador perde uma nota (ocorre uma falha na execução de uma seta), o multiplicador volta a ser 1. Esta mecânica de jogo encoraja o usuário a continuar acertando corretamente as setas que seguem surgindo, assim como serve também de auxílio para mantê-lo focado.

5.6 Sistema de Retorno (Feedback)

Dado o conceito do jogo apresentado, para que o usuário assimile as mecânicas implementadas, um sistema de feedback visual e sonoro se faz necessário. Desta forma, oferecendo uma apresentação visual que guie claramente o usuário às informações importantes, evitando que este perca o foco, assim como provendo uma resposta sonora que funcione como um agente em prol da imersão. Com o propósito de fornecer respostas às ações do usuário em tempo real, uma série de funcionalidades de feedback visual e sonoro foram implementadas. O objetivo principal destas funcionalidades é fornecer ao usuário sinalizações intuitivas relacionadas aos seus movimentos, bem como relacionadas às demandas geradas pelas mecânicas do jogo. Assim, durante o jogo, existem basicamente quatro casos diferentes que demandam feedbacks específicos:

1. Mudança de Trilhos: ocorre quando o usuário move sua mão esquerda com a intenção de mudar de faixa (ou trilho), saindo do trilho selecionado. Este movimento é ilustrado pela iluminação do trilho selecionado e pelo aumento da sua largura, assim como uma indicação visual da posição horizontal da mão esquerda ao longo do braço da guitarra virtual.

2. Movimentos Livres: são movimentos da mão direita sem intenções diretas dentro do jogo, ou seja, comumente representam apenas ajustes da posição da mão direita para se preparar para a seta que está por vir. É importante mostrar ao usuário que seus movimentos continuam sendo rastreados mesmo que estes não tenham um impacto direto no jogo. Assim, cada vez que sua mão direita se move, seu status vertical é ilustrado.

3. Setas Descendentes: as setas surgem no topo da plataforma e ao longo do tempo descem, aproximando-se da região da linha base, até que chegue o momento de ser tocada. Assim, a seta ganha foco com o tempo, assumindo caráter quase invisível a princípio (quando está longe de ser tocada), em seguida ganhando destaque gradativamente até alcançar a zona jogável. Além disso, uma seta guia é mostrada na

linha base. Esta última corresponde à próxima nota a ser tocada, e se torna mais clara quando o jogador antecipa o próximo movimento corretamente, com a mão direita localizada na posição vertical correta para acertar a nota na direção correta.

4. Tentativa de Acerto; se trata do momento em que o usuário movimenta sua mão direita na tentativa de cruzar a corda da guitarra virtual com o objetivo de acertar a seta no momento correto, ou seja dentro do espaço de tempo de 200 ms em que a seta está mais próxima à linha base. Este é um dos momentos decisivos durante a interação, pois ocorre em um espaço de tempo limitado. Desta forma, se o usuário antecipar ou atrasar demasiadamente o momento de atingir a seta (através do gesto com a mão direita), esta se modifica para uma versão mais escura da sua cor e, ao mesmo tempo, o volume da música tocada diminui, dando ao usuário a sensação de que ele realmente errou uma nota da música. No entanto, se o usuário jogar corretamente, a seta assume uma cor mais intensa e o volume da música é normalizado.

A partir dos conceitos apresentados, foi desenvolvida uma interface gráfica em dois estágios, através de um método iterativo de implementação de protótipos e testes, para que a interface se tornasse intuitiva, com o objetivo de alcançar um sistema de feedback intuitivo para a experiência do usuário.

5.6.1 Guitarra Flutuante e Base de Trilhos

A primeira interface desenvolvida foi resultado da junção direta da guitarra virtual flutuante (simulando que o usuário está vestindo-a) apresentada em (FIGUEIREDO *et al.* 2009) e da base de trilhos estática, como mostrada no jogo Rock Band (HARMONIX 2007). A primeira questão observada neste modelo de interface foi que a guitarra virtual não estava sendo associada aos trilhos e setas do jogo. Então, o braço da guitarra foi colorido, fazendo uma correspondência entre os trastes (divisões) da guitarra e os respectivos trilhos (Figura 5.5). No entanto, durante as sessões de jogo o usuário deve estar focado na tarefa de acertar as setas descendentes, e é sugerido que todas as informações visuais fornecidas como feedback estejam concentradas em torno destes componentes. Desta forma, não é recomendável induzir o usuário a dividir a atenção entre a base de trilhos estática e a guitarra virtual dinâmica (flutuante).

O efeito observado desta configuração foi que o usuário, em pouco tempo de jogo, desistia de olhar para a guitarra e passava a focar apenas na linha base de trilhos. No entanto, ao tomar esta opção, o jogador passa a perder informações importantes a respeito do posicionamento relativo das mãos em relação à guitarra virtual, e assim está sujeito a perder alguns movimentos por não mover a mão direita em uma distância suficiente. Estes testes revelaram a necessidade de unir a guitarra e a base de trilhos em uma visualização combinada.

Figura 5.6: Interface final construída para o Guitars on Air.

5.6.2 Elementos Unificados

Para contornar os problemas observados, uma interface integrada foi desenvolvida. A representação da guitarra se tornou estática junto à plataforma dos trilhos. Cada mão passou a ser representada por um cursor com a aparência específica para as mãos direita (mão segurando uma palheta) e esquerda (mão segurando o braço da guitarra), como ilustrado na Figura 5.6. Esta interface se mostrou mais objetiva para o usuário, incrementando a experiência como um todo.

5.7 Avaliação de Rastreadores de Mão

Além do jogo em si como uma aplicação de entretenimento, neste trabalho o Guitars on Air é utilizado como uma ferramenta para avaliação de rastreadores de mão. Devido às mecânicas inerentes aos jogos musicais, as quais requerem que os jogadores realizem comandos específicos (através de movimentos das mãos). Estes comandos têm a precisão controlável através da quantidade de trilhos usados (nos testes apresentados este número foi 4) e em um espaço de tempo ajustável de acordo com margem vertical da linha base (que no caso apresentado possui altura suficiente para que cada seta possua um período 200 ms para ser tocada). Desta forma, é possível analisar diferentes métodos de entrada de forma comparativa. Assim, todas as estatísticas armazenadas durante a sessão de jogo são levadas em consideração para avaliar as técnicas de rastreamento em questão.

Ao todo, para a realização da avaliação, três métodos de rastreamento de mãos foram

incorporados ao protótipo de Guitars on Air:

1. Luvas de cor Laranja: o primeiro método consiste em um algoritmo de detecção de cor, aplicado para rastrear um par de luvas laranjas como sugerido em (MÄKI-PATOLA *et al.* 2006; FIGUEIREDO *et al.* 2009). O intuito de usar esta técnica em uma avaliação comparativa se dá devido ao fato de que as mesmas já foram testadas em um cenário de aplicação que se assemelha muito ao do Guitars on Air.

2. Kinect: o segundo método analisado foi o rastreamento provido pelo sensor de profundidade Kinect. O uso do Kinect nesta análise comparativa é de relevância no sentido de prover como base comparativa uma ferramenta que já é consolidada e usada em larga escala para aplicações similares de interação voltadas para entretenimento.

3. HAFT: o terceiro método de rastreamento utilizado foi provido pela ferramenta aqui proposta intitulada HAFT, com o propósito de validá-la em um cenário de uso real, além de compará-la com os métodos citados acima.

Com o propósito de avaliar o jogo proposto, bem como a interação provida pelos rastreadores citados um estudo de caso foi preparado. Primeiramente, através de um software de composição para o formato utilizado em Guitars on Air, uma lista de comandos correlacionados à música Paint it Black dos Rolling Stones foi gerada.

Esta música apresenta algumas características que se mostram importantes em uma primeira experiência com um jogo musical: um padrão de execução é repetido de forma que permite ao ouvinte predizer os próximos passos da canção após ouvi-la por poucos segundos; além disso, esta música apresenta uma dinâmica de forte marcação rítmica e não muito rápida, dando a jogadores iniciantes tempo suficiente para mudar de posições entre um comando e outro. O ambiente de performance foi preparado para os usuários, com o objetivo de prover uma porção de espaço livre e uma distância para a visualização do a interface gráfica com clareza e conforto como mostrado na Figura 5.7. O sensor Kinect e a câmera usada foram posicionados em uma altura aproximada a 1,5m e 1,4m, respectivamente e a tela de visualização a uma altura de 1,1m, logo abaixo dos dois sensores. O posicionamento do usuário demonstrou variações entre 1,5m e 2,5m de distância em relação à tela. Durante as sessões, os usuários não eram limitados formalmente, e assim se moviam livremente no ambiente oferecido. De fato, o único requisito sugerido aos usuários foi que estes tentassem cumprir a tarefa proposta com o melhor desempenho possível, tentando maximizar a pontuação alcançada.

Com o intuito de realizar testes para averiguar tanto o funcionamento do Guitars on Air como jogo quanto como ferramenta de avaliação, além de prover uma análise das ferramentas de rastreamento citadas, um grupo de usuários foi selecionado para participar das atividades. Os testes foram realizados com 12 usuários, 6 mulheres e 6 homens em uma faixa de idade entre 18 e 26 anos. O número de 12 usuários é suficiente para a realização de testes iniciais com o intuito de apontar pontos críticos nas diferentes partes dos sistemas utilizados, no entanto

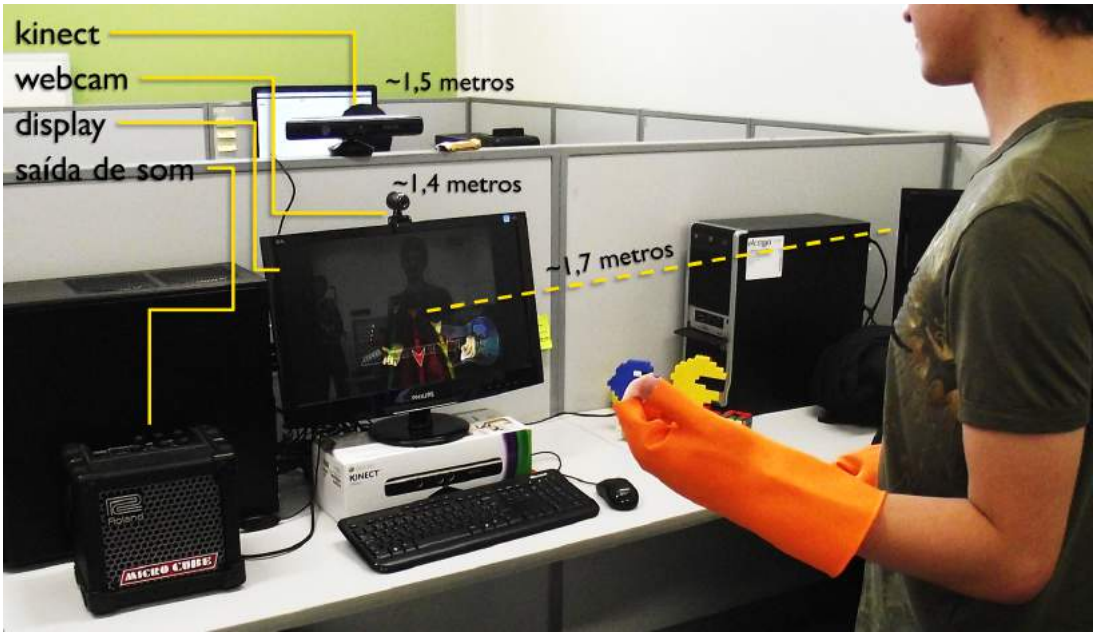


Figura 5.7: Cenário de uso montado para os testes realizados com o Guitars on Air. Neste caso, as mãos do usuário estão sendo rastreadas através de um algoritmo para a detecção de luvas laranja como sugerido em (MÄKI-PATOLA *et al.* 2006; FIGUEIREDO *et al.* 2009).

esta quantidade não permite uma análise com alto grau de precisão devido à baixa amostragem. Assim, um dos importantes trabalhos futuros é a realização de uma sequência de testes mais extensa, com um número de usuários maior; atividade que não pôde ser realizada em tempo hábil no escopo deste trabalho.

Cada um dos usuários jogou quatro vezes usando a mesma canção como base (Paint it Black), e em cada uma das vezes variava-se o método de rastreamento de mãos utilizado, sendo a primeira destas vezes não contabilizada nos testes, servindo somente para a familiarização do usuário com o aplicativo. A ordem de aplicação de cada técnica de rastreamento foi variada para cada usuário, com a intenção de evitar uma tendência nos resultados por conta do aprendizado acumulado nas primeiras experiências. Ou seja, cada usuário experimentou as três formas de rastreamento em uma ordem permutada em relação às experiências de outros usuários.

Primeiramente, cada usuário foi apresentado ao sistema com uma breve explicação e em seguida eles eram postos a jogar por 2 minutos, para dar a oportunidade destes se familiarizarem com o ambiente de interação e a aplicação em si, fazendo com que se sentissem mais confortáveis durante a avaliação. Em seguida, foi requisitado a cada usuário que este procurasse atingir a maior pontuação possível no jogo até que o último comando fosse executado, totalizando uma duração de 2 minutos e 21 segundos. Os dados de desempenho então eram registrados e o usuário passava a jogar utilizando o próximo método de rastreamento. Assim, cada técnica de rastreamento de mãos foi testada 12 vezes, 4 delas sendo a primeira experiência do usuário com o jogo, 4 sendo o ponto intermediário na sessão do usuário e 4 sendo a última técnica testada.

De acordo com (PINELLE *et al.* 2008), existem doze tópicos que são comumente proble-

máticos em relação à usabilidade em jogos digitais, dos quais cinco, mais focados na interação em si, podem ser aplicados diretamente ao cenário deste trabalho. Estas cinco problemáticas estão listadas abaixo:

1. Respostas às ações do usuário imprevisíveis; detecção falha das intenções; respostas inconsistentes em relação às ações de entrada.

2. Difícil controle das ações no jogo; controles muito sensíveis; controles não-naturais.

3. A resposta para as ações do usuário não ocorrem em um período de tempo satisfatório; tempo de resposta lento interfere na interação; comandos anteriores e atuais se mostram conflitantes durante o processo.

4. Sequência de comandos exageradamente complexa; curva de aprendizagem é muito íngreme; sequências são complexas, longas e estranhas, tornando o jogo difícil de ser jogado.

5. Representações visuais são difíceis de serem interpretadas; má visualização da informação; muita confusão na tela; muitos elementos na tela ao mesmo tempo; difícil de visualizar e distinguir conteúdo interativo e não interativo.

Assim, depois das sessões de jogo, cada usuário respondeu um questionário com foco na experiência, baseado nos cinco tópicos listados acima. Em adição, algumas perguntas relacionadas especificamente às formas de rastreamento foram inseridas, por exemplo, se o uso de luvas laranja em algum momento incomodou. Para quantificar as respostas, um esquema de intensidade similar ao proposto em (KALAWSKY 1999) foi utilizado. Para cada pergunta, o usuário foi levado a responder cada um das dezessete perguntas do questionário em uma escala de 5 (se concordava 100%) a 1 (discordava completamente), também conhecida como Likert Scale (LIKERT 1932).

5.8 Resultados e Discussão

As respostas dos usuários em relação à interface do Guitars on Air pode ser observada na Figura 5.8.

Por sua vez, a Figura 5.9 ilustra as respostas do usuários para perguntas relacionadas especificamente à cada uma das ferramentas de rastreamento.

A Figura 5.8 e a Figura 5.9 ilustram as respostas dos usuários às perguntas dos questionários variando na escala de 1 a 5 como mencionado anteriormente. Os gráficos apresentados se comportam como histogramas. Assim, as colunas da Figura 5.8 representam quantos usuários responderam o quesito com a referente pontuação (expressa abaixo da coluna no eixo horizontal). A Figura 5.9 ilustra gráficos similares tendo as linhas (ou barras horizontais) como representação de quantos usuários respondeu o quesito com a pontuação indicada.

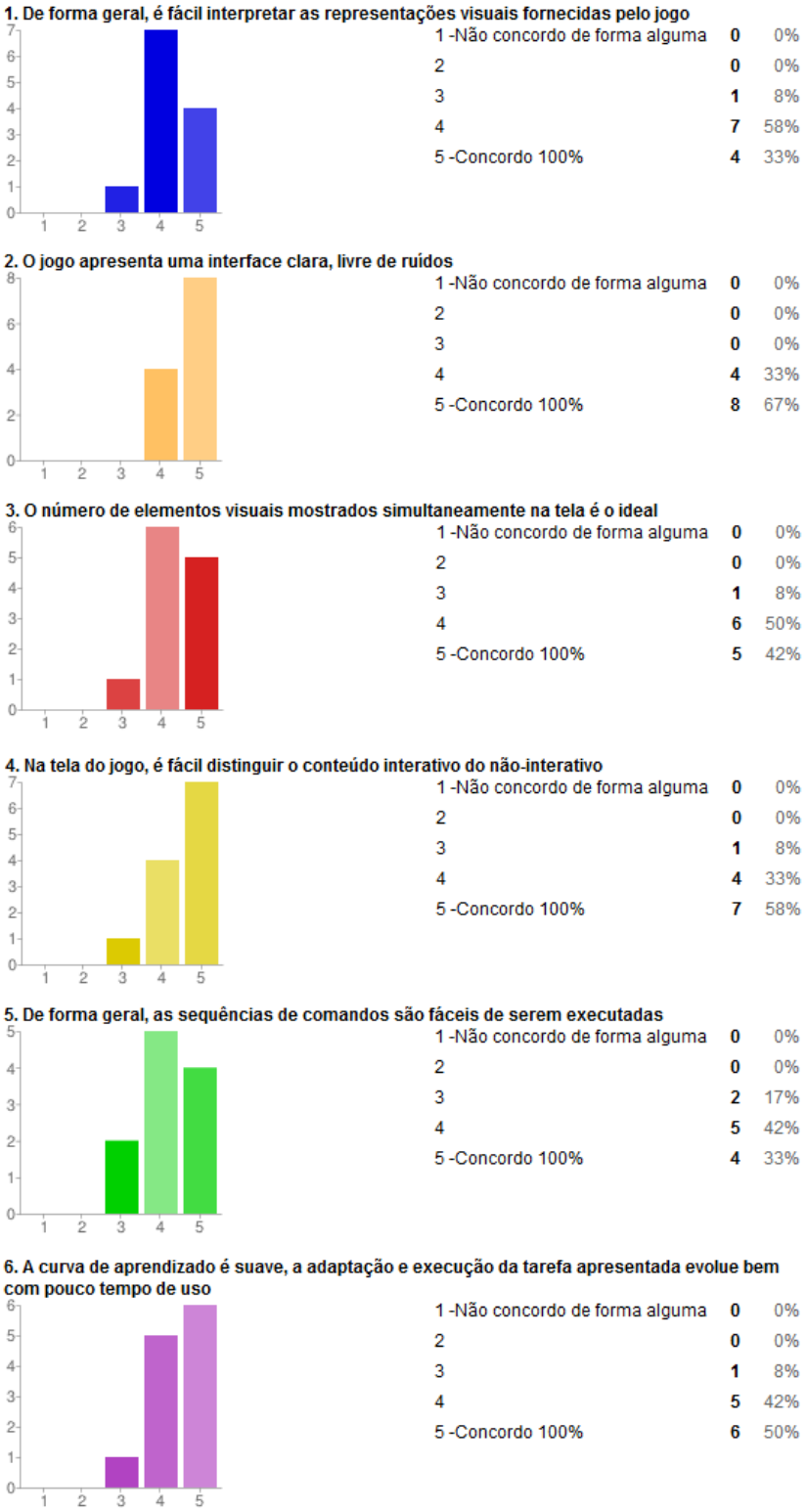


Figura 5.8: Respostas dos usuários sobre o jogo Guitars on Air.

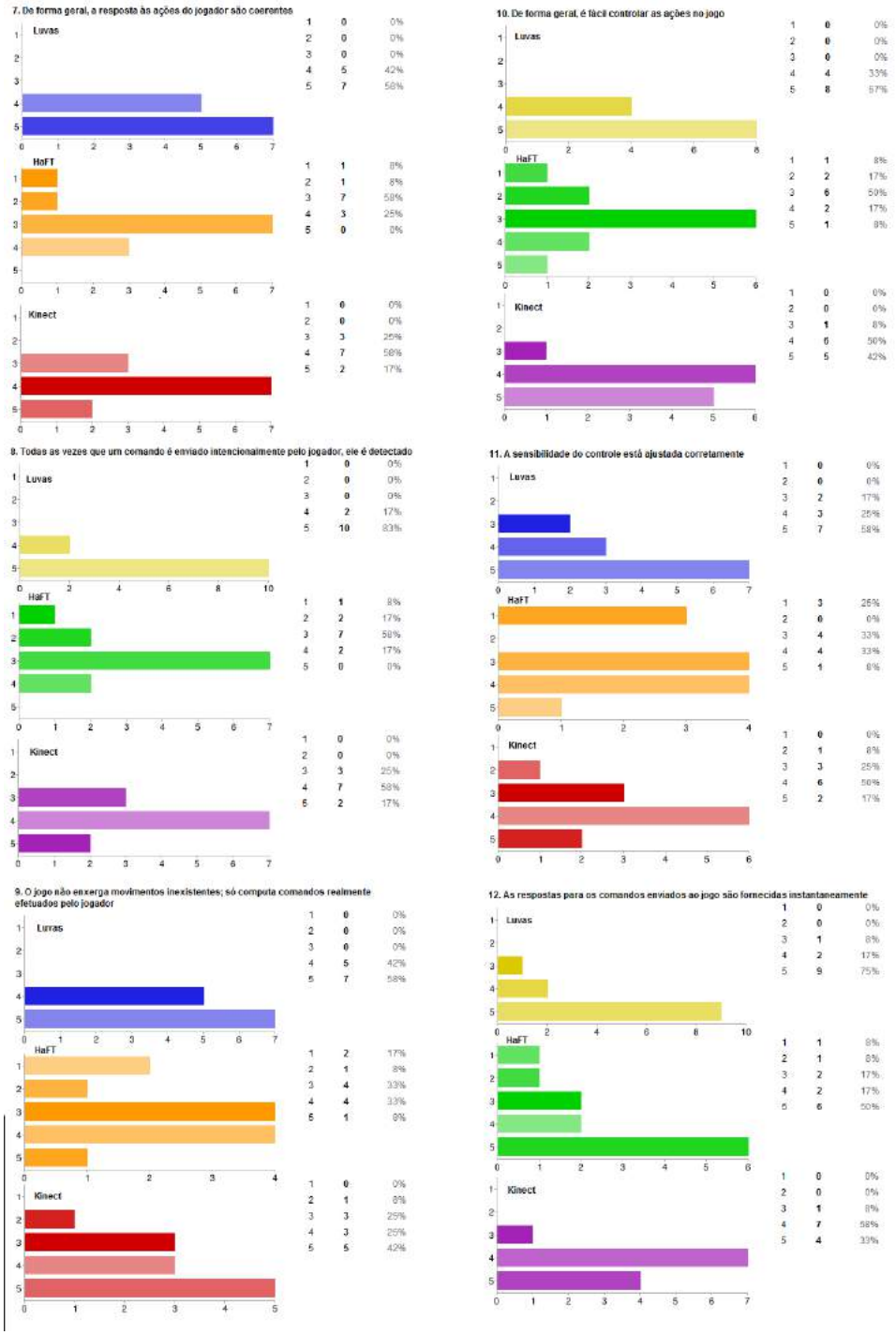


Figura 5.9: Respostas dos usuários às perguntas relacionadas a cada uma das ferramentas de rastreamento.

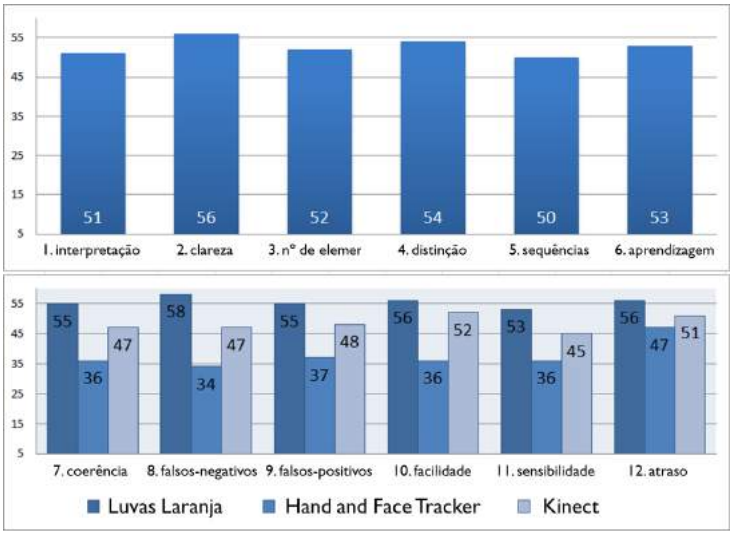


Figura 5.10: Soma das respostas do questionário realizado.

A Figura 5.10 ilustra os resultados do questionário mostrando a pontuação dos mesmos quesitos presentes nas Figuras 5.8 e 5.9, no entanto de forma mais concisa, tendo colunas como a representação da soma das respostas de todos os usuários. Dado que 12 usuários participaram dos testes e o máximo de pontos que cada quesito pode ter é 5, a pontuação total máxima para cada quesito é 60 e a mínima é 12.

Primeiramente, analisando a Figura 5.10, é possível observar que em relação à implementação do jogo em si (parte superior da figura, contendo respostas genéricas em relação ao jogo e independentes do método de rastreamento), considerando a interface e os comandos requisitados durante as sessões, a pontuação geral foi altamente satisfatória. Todos os quatro aspectos relacionados à interface e os dois relacionados à mecânica do jogo receberam altas pontuações, sendo bem avaliados pelos usuários. Estes seis aspectos estão listados abaixo:

1. Facilidade de interpretação.
2. Clareza da interface.
3. Número adequado de elementos na tela.
4. Fácil distinção sobre elementos interativos e não-interativos.
5. Fácil execução das sequências de comandos
6. Curva de aprendizagem suave

De fato, analisando estes mesmos aspectos nos gráficos mais detalhados da Figura 5.8 é possível observar que todos os usuários responderam 3 ou mais para estes quesitos, com uma maior concentração nos valores de 4 e 5. Ou seja, de forma geral, os usuários se mostraram satisfeitos com a interface do jogo, assim como com o fluxo de comandos e sua curva de aprendizado.

As perguntas seguintes (parte inferior da Figura 5.10) foram direcionadas à avaliação específica de cada método de rastreamento. De forma geral, os usuários revelaram que o método de rastreamento com luvas foi que demonstrou o mais alto grau de satisfação em todos os quesitos. As questões respondidas foram sobre os seguintes tópicos:

7. A coerência entre os gestos realizados e reconhecidos.
8. Ocorrência de falsos-negativos (foi perguntado aos usuários se alguns de seus movimentos não foram reconhecidos).
9. Ocorrência de falsos-positivos (foi perguntado se o jogo levou em conta algum movimento que não chegou a ser realizado pelo usuário).
10. A facilidade de uso de forma geral.
11. O ajuste de sensibilidade.
12. O atraso no tempo de resposta (foi perguntado se o usuário sentiu que suas ações demoravam certo tempo para serem levadas em conta pelo jogo).

De forma geral, do ponto de vista do usuário a ferramenta HAFT se comportou aquém das demais; em todos os quesitos é possível observar a soma das respostas dos usuários é menor para o HAFT (Figura 5.10). Analisando mais a fundo, observando a Figura 5.9, é possível ressaltar o quesito sobre sensibilidade de controle (questão 11) como um dos pontos em que boa parte dos usuários apresentou dificuldade extrema. Esta sensibilidade está diretamente associada ao problema de jitter (tremidas no resultado do rastreamento ao longo do tempo). De fato, o rastreamento provido por HAFT não apresenta nenhum tratamento para o problema de jitter, fornecendo como resultado do rastreamento a posição 2D do ponto guia o qual está sujeito à mudanças sempre que novas features são adicionadas ou removidas da nuvem, além de sofrer influência da velocidade das features presentes. Assim, o comportamento do ponto guia se apresentou de forma caótica devido à própria natureza do algoritmo de rastreamento baseado em nuvem de pontos. Este aspecto influenciou consideravelmente no desempenho dos usuários, assim, em uma leitura mais abrangente é possível relacionar este problema como um dos pilares de dificuldade dos usuários em relação ao uso do HAFT. Desta forma, possivelmente por conta deste impacto, este aspecto por consequência afetou a avaliação dos usuários em vários outras questões como os quesitos 7 e 11 que estão relacionados respectivamente à coerência do rastreamento e ao controle de forma geral.

Outro quesito que captou parte significativa da impressão negativa dos usuários em relação ao HAFT foi o quesito 8. A pergunta sobre a detecção dos comandos realizados deixa claro que para os usuários o HAFT funciona de médio a mal neste ponto, concentrando as respostas em torno dos valores 3, 2 e 1, enquanto que para o Kinect estas estão concentradas em torno do valor 4 e para as Luvas Laranja e maior concentração das respostas é sobre o valor

5. Este quesito revela uma das principais dificuldades da ferramenta proposta que se trata da ausência de um método eficiente de detecção e recuperação automática. Foi observado durante os testes que devido a diversas adversidades como oclusões, movimentos rápidos, limites do campo de visão da câmera, entre outros, o rastreamento proposto por HAFT passava por falhas e era perdido. Enquanto que os outros dois métodos de rastreamento (Kinect e Luvas Laranja) contavam com uma forma de detecção automática, o HAFT requeria que o usuário realizasse um gesto específico para detectar novamente as mãos e assim seguir o rastreamento. Ou seja, mesmo na ocorrência de falhas com o Kinect e as Luvas Laranja a reinicialização se dava de forma automática enquanto que com o HAFT era necessário primeiramente que o usuário percebesse não estar mais controlando devidamente a guitarra virtual, para daí então realizar o gesto requerido para a detecção. Esta dificuldade se repetiu durante vários testes com vários usuários e se reflete em parte no quesito 8, pois esta é uma das causas para o usuário realizar os gestos sem que estes sejam detectados devido à uma falha de rastreamento prévia que não foi recuperada pois seria necessária a execução de uma nova detecção, requerendo do usuário um gesto específico da mão. Mais uma vez, este tópico pode ser compreendido como influenciador nas demais respostas com as dos quesitos 7 e 9, influenciando na avaliação dos usuários como um todo.

Assim, estes dois tópicos críticos sobre o jitter presente no HAFT e a ausência de uma detecção automática como ocorre com o Kinect e com as Luvas Laranja são avaliados, a partir das respostas obtidas nos questionários e da observação da desempenho dos usuários, como os principais aspectos falhos da ferramenta, tendo em vista um uso livre da mesma em uma aplicação como o Guitars on Air.

Em relação outras duas ferramentas testadas, é possível afirmar que o rastreamento através das Luvas Laranja apresentou o melhor retorno em relação à avaliação dos usuários. De forma geral, como pode ser visto na Figura 44, a pontuação somada do rastreamento usando luvas atingiu valores muito próximos de 60 (valor máximo), indicando que em relação à facilidade de uso e de controle da aplicação do Guitars on Air, as luvas se apresentaram como uma solução adequada e muito bem aceita pelos usuários. De fato, em todos os quesitos o rastreamento com luvas se mostra superior aos demais, concentrando opiniões sempre em torno dos valores 4 e 5, com destaque para os quesitos 7, 8, 9 e 10 em que fica claro que o uso das luvas realiza um rastreamento com precisão e pouquíssimas ocorrências de falsos-positivo e/ou falsos-negativo.

Em seguida, os resultados dos questionários apontam o Kinect como uma ferramenta de rastreamento intermediária, considerada satisfatória na avaliação geral, obtendo resultados em torno do valor 4 na maior parte dos quesitos. Ainda em relação ao Kinect, é importante ressaltar o quesito 12 que trata do atraso da resposta de rastreamento. Neste quesito é possível observar que nas demais ferramentas, os usuários na sua maioria optaram por dar o valor máximo (5), indicando que o atraso nestas seria muito pequeno, enquanto que em relação ao Kinect a maioria dos usuários optou pelo valor 4, indicando um indício de atraso que não é crítico mas que está presente. Este quesito levanta a hipótese de que o Kinect apresenta um atraso um pouco maior

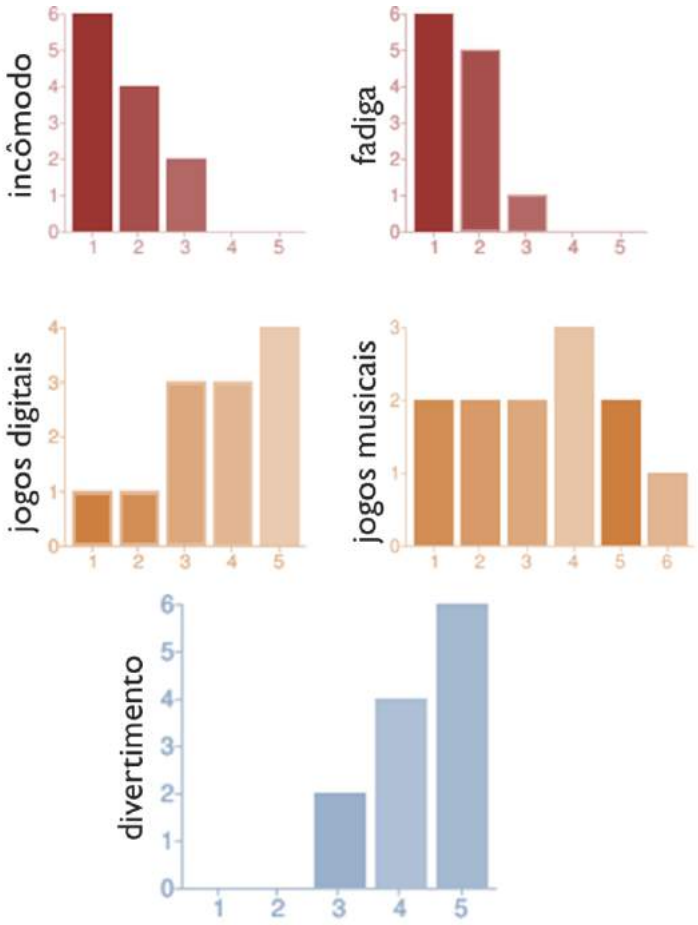


Figura 5.11: Perguntas adicionais realizadas. Os gráficos mostram o número de ocorrencias (colunas) de cada uma das respostas quantitativas (linhas). Topo: questões relacionadas à ergonomia. Centro: experiência pré-existente dos usuários com jogos digitais e mais especificamente jogos musicais. Base: grau de divertimento geral durante a experiência.

que as demais ferramentas.

Em adição aos quesitos já citados, um conjunto de perguntas complementares foi respondido pelos usuários (Figura 5.11). Nestas perguntas a mesma escala foi utilizada e a Figura 45 ilustra nos eixos verticais de cada gráfico a quantidade de respostas que cada valor na escala recebeu, com exceção da pergunta sobre jogos musicais que ofereceu uma opção a mais (valor 6) para abarcar usuários com familiaridades excepcionais com estes tipos de jogos.

Analisando a Figura 5.11, é possível observar que ambos, o incômodo do uso das luvas e a fadiga durante a interação e se mostraram como problemas menores, de baixa relevância, os quais não comprometeram o divertimento da experiência. No entanto, estas questões se analisadas em sessões de jogo mais longas (dezenas de minutos ou horas de jogo), podem causar o distanciamento do usuário. Por fim, são mostrados os graus de experiência preexistentes dos usuários em relação a jogos digitais e musicais especificamente, que como resultado revelam que o grau de experiência com jogos musicais varia de forma uniforme entre o conjunto de usuários que participou dos testes, mostrando que o fato da alta compreensão da interface e das mecânicas utilizadas em Guitars on Air não está associado somente a experiências prévias com jogos do

método de rastreamento	% de acertos	média do nº máximo de acertos consecutivos	atraso médio (milissegundos)
Luvas Laranjas	80,2	38,1	7,23
Kinect	68,8	29,5	14,27
HaFT	50,4	14,5	7,26

Figura 5.12: Tabela comparativa entre os métodos de rastreamento avaliados, a partir de dados obtidos através do Guitars on Air, com quesitos relacionados à robustez e velocidade. Obs.: os dados usados para gerar esta tabela se encontram no Apêndice.

mesmo tipo.

A tabela ilustrada na Figura 5.12 ilustra os valores médios dos dados coletados ao longo das sessões de jogo. A partir desta tabela é possível perceber que o desempenho dos usuários foi significativamente melhor quando usadas as luvas para o rastreamento. E por sua vez, o Kinect demonstrou desempenho superior ao método de rastreamento proposto neste trabalho. Esta primeira análise demonstra coerência em relação às avaliações do questionário e esta provida pelos dados fornecidos pelo Guitars on Air, revelando que o mesmo, como ferramenta de avaliação para métodos de rastreamento de mãos, fornece resultados coerentes em relação à experiência do usuário.

Sobre a velocidade de reconhecimento dos gestos, o Kinect apresentou uma defasagem de 7 milissegundos (ms) em relação aos outros dois métodos. A medição do atraso é realizada através do cálculo da média dos atrasos em cada um dos casos em que o usuário realiza um comando indicado por uma seta. O atraso de cada um destes momentos consiste na diferença de tempo entre o instante em que a seta estava exatamente sobre a linha base e o instante em que o gesto do usuário foi computado. Assim, este atraso leva em conta todo o processo de captura e rastreamento, além de estar associado ao usuário podendo este influenciar em um atraso maior ou menor dependo da sincronia (ou seja, capacidade de acertar as notas no momento exato) em que o usuário se encontra em relação ao jogo. Assim, é possível que a amostragem de usuários por ser pequena, seja insuficiente para definir com precisão a diferença comparativa de atrasos entre as ferramentas, no entanto vale ressaltar que os indícios apresentados na tabela da Figura 46 levantam a hipótese que o Kinect revela um atraso mínimo em relação às demais ferramentas. É importante ressaltar também que esta hipótese também surgiu ao analisar os questionários respondidos pelos usuários, especialmente o quesito 12. Assim, este ponto permanece como um ponto em aberto a ser averiguado futuramente com maior precisão (visto que na realidade, o atraso de 14,27 ms pode ser maior ou menor), deixando como base a hipótese de que o Kinect apresenta um atraso de interação perceptível ao usuário e maior do que outras ferramentas baseadas puramente em algoritmos de visão como o caso do HAFT e das Luvas Laranja. É importante também ressaltar que uma diferença de atraso de 7 ms é praticamente imperceptível ao ser humano e tem pouco ou quase nenhum impacto na interação.

Em relação à taxa de acerto apresentada pelo HAFT, uma das razões atribuídas ao resultado obtido, é o fato de que a detecção de mãos utilizada não é automática como ocorre nos casos das luvas e do Kinect (nestes casos as mãos são identificadas independente do gesto realizado). Assim, sempre que ocorre uma falha de rastreamento durante o uso de HAFT, primeiramente é necessário que esta seja percebida pelo usuário que em seguida passa a tentar recuperar o rastreamento fazendo gestos específicos para a detecção, o que acaba levando a perda de notas, além da distração causada que atrapalha o desempenho de forma geral. Além disso o problema de jitter foi observado nos testes como um dos principais tópicos críticos presentes no HAFT o qual não se apresentava de forma tão incisiva nas demais ferramentas. Este problema se revelou como um grande desafio para que os usuários controlasse a aplicação devidamente, causando além de sucessivos erros de interação, um desestímulo por parte dos usuários visto que a interface não correspondia às intenções e aos gestos executados.

Mais além, é importante ressaltar que a realização dos testes descritos com os 12 usuários é de extrema importância para a identificação de pontos críticos da ferramenta HAFT no entanto é insuficiente para entender com precisão o comportamento da mesma no cenário de entretenimento proposto. A Figura 5.13 ilustra em separado o desempenho de cada um dos usuários e é possível observar um caso em que o desempenho com as três ferramentas foi muito semelhante (pontilhado em laranja), além de alguns casos em que o usuário atingiu pontuações superiores ao usar a ferramenta HAFT em relação àquela sessão em que o mesmo usuário utilizou o Kinect (pontilhados em amarelo). Assim, é demonstrado que a análise através de valores médios em uma amostragem pequena é insuficiente para afirmar com precisão o quão melhor cada uma das ferramentas é entre si. Além disso, é possível levantar a hipótese de que em um cenário mais controlado, menos sujeito a falhas de rastreamento e fazendo uso de uma versão com o problema de jitter amenizado no HAFT, esta seria capaz de obter resultados superiores ao Kinect em relação ao desempenho em jogo do usuário.

O estudo de caso apresentados se mostrou como um grande desafio para a ferramenta de rastreamento proposta. De fato o HAFT funcionou como uma ferramenta que suporta uma série de cenários adversos (como o suporte a movimentações rápidas), no entanto ainda não se encontra em um estágio que possa ser aplicada para interações livres em aplicações de entretenimento. Por outro lado, o estudo de caso aplicado ao HAFT permitiu encontrar os principais pontos em aberto ao também comparar o HAFT com ferramentas já adaptadas e propícias para o cenário do Guitars on Air. Desta forma a aplicação do Guitars on Air serviu como ponto de comparação sólido para a compreensão dos principais aspectos falhos de HAFT e assim serviu ao seu propósito de estudo de caso e análise para a ferramenta proposta.

Ainda analisando o desempenho dos métodos de rastreamento utilizados, vale ressaltar o tempo de processamento obtido de cada um destes. O tempo de processamento de cada uma das técnicas testadas é consideravelmente baixo, de forma que cada uma das três cumpriu o requisito de ser executada em tempo real, com taxa de atualização igual ou maior que 30 quadros por segundo (também *frames* per second ou fps). Para que o Kinect forneça novas posições

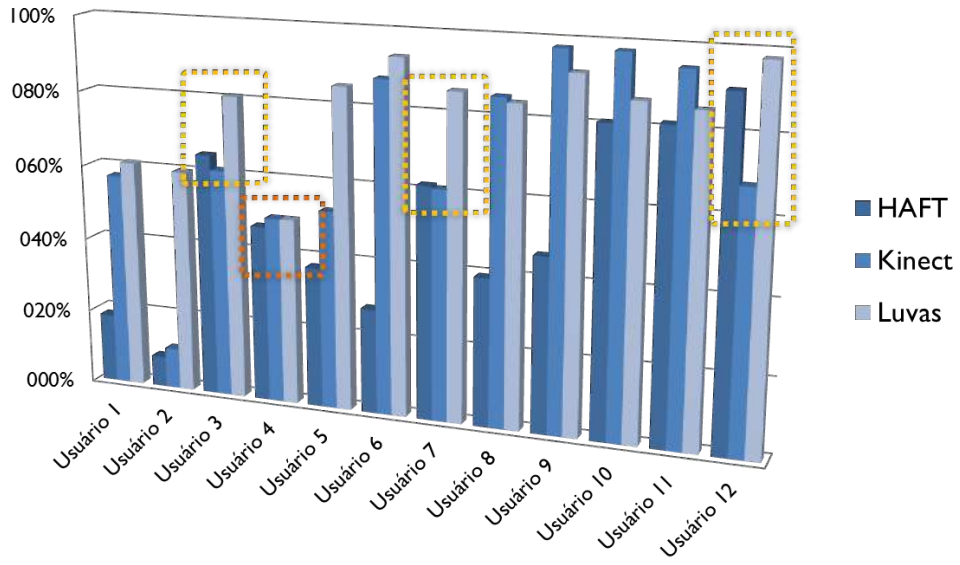


Figura 5.13: Gráfico ilustrando o desempenho em separado de cada um dos usuários ao usar cada uma das técnicas de rastreamento de mãos testada.

do esqueleto rastreado, um tempo médio de 5 ms é necessário, HAFT por sua vez, leva 7 ms. Ambos os tempos de execução são consideravelmente baixos, no entanto o Kinect apresenta atrasos no final do processo de interação. Este fato sugere que apesar de o Kinect prover um método de rápida execução, o seu resultado final de rastreamento a cada *frame* acumula um atraso relacionado à posição do *frame* anterior, produzindo um efeito de deslizamento (drift) de forma que para que a posição se estabilize o usuário precisa ficar parado por um pequeno instante de um ou dois *frames* de duração. O rastreamento por detecção de luvas de cor laranja por sua vez, é executado em um tempo médio de 3 ms. Desta forma, o tempo de processamento total do jogo varia entre 14 e 18 ms, a depender do método de rastreamento usado como entrada.

Todos os testes aplicados ao Guitars on Air foram realizados com imagens capturadas por uma Webcam comum, na resolução de 320 x 240 *pixels* a uma taxa de 30 fps. O processador utilizado foi o Intel(R) Core(TM) i5-2300 CPU @ 2.80 GHz, com acesso a uma memória de 4.00 GB RAM, e todo o processamento foi executado sobre o sistema operacional Windows 7 Enterprise x64.

6
CONCLUSÃO

Nesta dissertação é apresentada uma ferramenta para o rastreamento de mãos e faces chamada HAFT. Primeiramente, é demonstrada uma etapa de segmentação de cor de pele, voltada para propósitos de interação, distinguindo grupos de interesse de ruídos e partes irrelevantes do ambiente. Ainda sobre a segmentação, uma série de testes e comparações foi realizada, demonstrando que o método proposto é coeso, agrupando e expandindo setores cor de pele em cada frame capturado.

Em seguida, a partir do conceito de gerenciamento de uma nuvem de features foi demonstrado o rastreamento simultâneo de vários alvos (mãos e faces), de forma robusta a oclusões parciais entre os mesmos. Para tal, a técnica proposta faz uso de pesos como fatores de relevância para cada uma das features rastreadas, sugerindo o uso de um ponto guia como a média ponderada da nuvem rastreada. Entre outras contribuições, está a etapa de segmentação de cor de pele, propondo um método de propagação de influência de pixels iniciadores para espalhadores, fazendo uso de critérios de classificação mesclados dentre vários modelos de cor, e assim demonstrando resultados que evitam regiões de ruído e agrupam de forma mais eficiente regiões de cor de pele. Mais além, um método de detecção de mãos é proposto, como evolução daquele apresentado em (PAN *et al.* 2010), obtendo melhores resultados tanto em relação as taxas de falsos-positivos quanto falsos-negativos.

Os resultados foram analisados de forma comparativa através de um estudo de caso que representa um cenário real de aplicação no setor de entretenimento. A ferramenta utilizada para os testes foi desenvolvida no presente trabalho; associando o conceito de jogos musicais à prática de air guitar, o jogo Guitars on Air foi apresentado como um protótipo coerente para o uso em avaliações de métodos de rastreamento de mãos. Um total de 12 usuários participaram de testes, usando o Guitars on Air como ferramenta para a avaliação de três distintos métodos de rastreamento: rastreamento de cor para a detecção de luvas laranja, rastreamento através do sensor de profundidade Kinect e por fim o rastreamento provido pela ferramenta aqui apresentada, HAFT. Após as sessões de teste, cada usuário respondeu um questionário quantitativo visando avaliar tanto o jogo em si, quanto os métodos de rastreamento utilizados. Como resultado, Guitars on Air se mostrou um jogo com interface concisa e de fácil interpretação. Além disso,

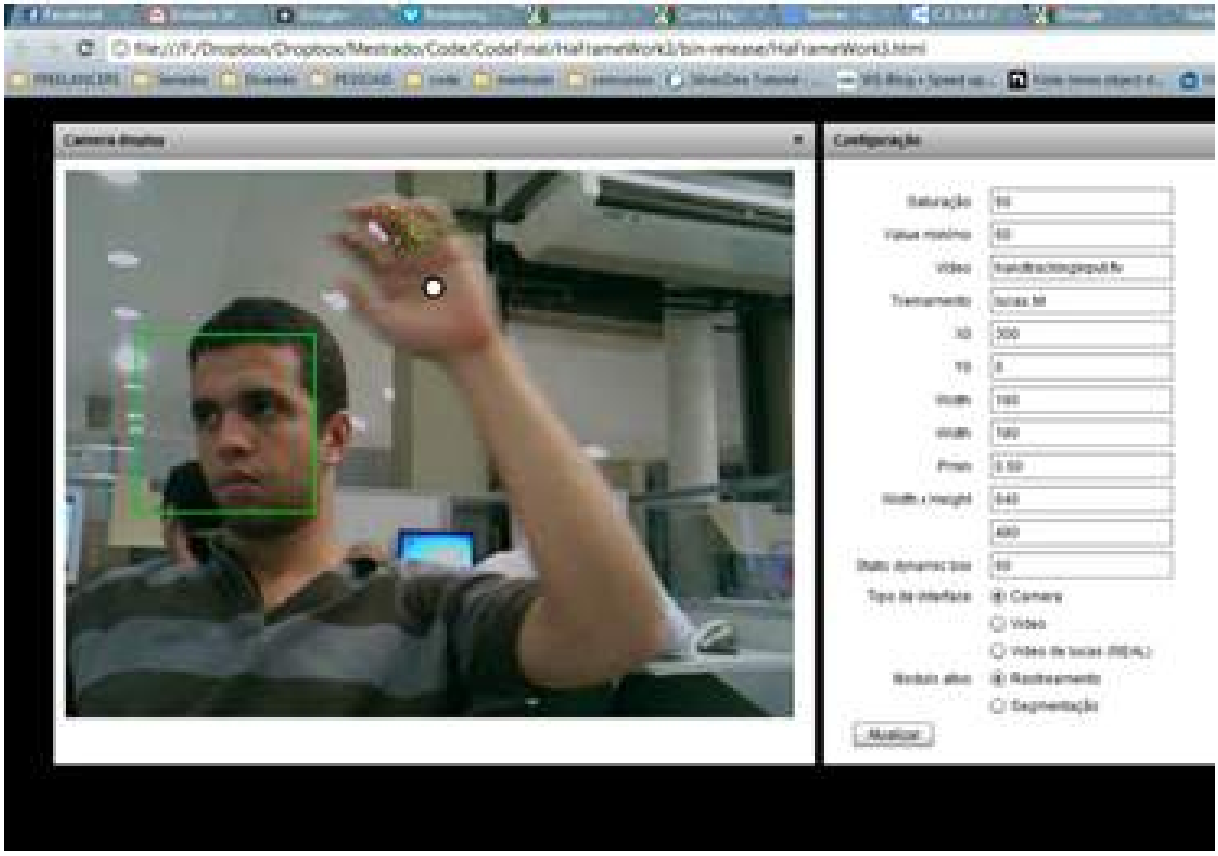


Figura 6.1: Porte do HAFT para a linguagem de programação Action Script 3, possibilitando seu uso através de navegadores Web (SOUZA 2012).

os dados gerados pelas sessões de jogo sobre os resultados de rastreamento de cada técnica testada foram correlacionados com as respectivas respostas dos usuários, demonstrando que como ferramenta de avaliação, Guitars on Air provê uma análise condizente com a experiência do usuário.

Dos três métodos de rastreamento de mãos testados, aquele baseado na detecção de luvas obteve o melhor resultado, seguido pelo Kinect e em seguida pela técnica proposta, cada um apresentando uma taxa média de acertos de $80,2\%$, $68,8\%$ e $50,4\%$ respectivamente. No entanto em relação aos atrasos entre gestos realizados pelo usuário até o momento em que estes são efetivados no jogo, o Kinect apresentou resultados inferiores ($14ms$ de atraso) em relação aos outros dois métodos de rastreamento ($7ms$).

Mais além, como reflexo do baixo tempo de processamento e recurso de memória demandado por HAFT, este se mostrou como um método de alta aplicabilidade, sendo portado para Action Script 3 como resultado de uma colaboração (SOUZA 2012), com o objetivo de ser usufruído na plataforma Web. A Figura 6.1 ilustra os resultados obtidos da técnica em uma versão otimizada para a plataforma Web.

Como trabalhos futuros, é necessário investir no sentido de detectar as mãos do usuário de forma “invisível”, sem que este tenha que explicitamente demandar pelo comando de detecção. Assim, além do uso da técnica de detecção proposta, é necessária a adição de outras mecânicas.

Desta forma, é sugerido que seja adicionado à etapa de avaliação, presente na fase de rastreamento, uma rotina dedicada à detecção de mãos baseada também no comportamento do alvo e não somente na sua forma. Ou seja, a partir do padrão de movimentação do alvo, aliado a outros fatores como o comportamento das features ao longo do tempo e ao formato do alvo, aplicar critérios de decisão para classifica-lo como mão.

Ainda sobre a etapa de avaliação, é possível estender esta para que suporte oclusões totais em curtos períodos de tempo, através de uma análise do comportamento do alvo. Assim, caso o usuário mova a mão, por exemplo, para fora da região de alcance (fora da imagem capturada pela câmera), caso em um curto espaço de tempo o alvo volte à posição esperada, é dada continuidade ao rastreamento, sem a necessidade de uma nova inferência através dos métodos de detecção. Por fim, HAFT pode ser integrado com uma técnica de rastreamento 3D de mãos (THAYANANTHAN *et al.* 2006) ou faces (WANG *et al.* 2007) com o objetivo de acelerar o processo de rastreamento bem como reduzir a chance de falhas destes algoritmos.

Em relação à ferramenta de avaliação Guitars on Air, alguns aprimoramentos de mecânica são considerados como futuros passos. A introdução de efeitos associados a alguns movimentos de guitarristas como bends, slides e vibratos (Figura 6.2), incrementando a jogabilidade e assim imersão durante o jogo. Estes artifícios estão associados diretamente à prática de tocar guitarra, funcionado como uma metáfora gestual. O efeito de slide consiste em o guitarrista tocar uma nota e mantê-la pressionada movendo o dedo que pressiona a corda sobre o braço da guitarra em um movimento horizontal (no caso, movendo a mão esquerda). A metáfora pensada neste caso para o Guitars on Air é apresentada na parte inferior da Figura 6.2, indicando que após tocar uma nota na guitarra virtual, o usuário deve mover sua mão esquerda de um trilho para o outro ao longo do tempo. O efeito de vibrato funciona de forma parecida, também como metáfora gestual. O vibrato é análogo ao slide, no entanto o efeito é realizado sem que o guitarrista tire o dedo da casa em que está pressionado, realizando movimentos, mas mantendo a posição pressionada no braço da guitarra. Assim, o efeito de vibrato projetado para ser incorporado ao Guitars on Air é análogo ao slide no entanto o usuário deve mover sua mão horizontalmente realizando oscilações mantendo a mão esquerda no mesmo trilho, sem passar para um trilho vizinho. Estas metáforas adicionais adicionam artifícios de imersão ao jogo, fazendo com que o usuário fique mais focado em sua atividade e a realize com maior dedicação e atenção. Além destas metáforas adicionais, efeitos como esses adicionam informações para avaliação em um espaço de possibilidades mais abertos do que o usado até o momento, que por hora possibilita somente duas opções, acerto ou erro.

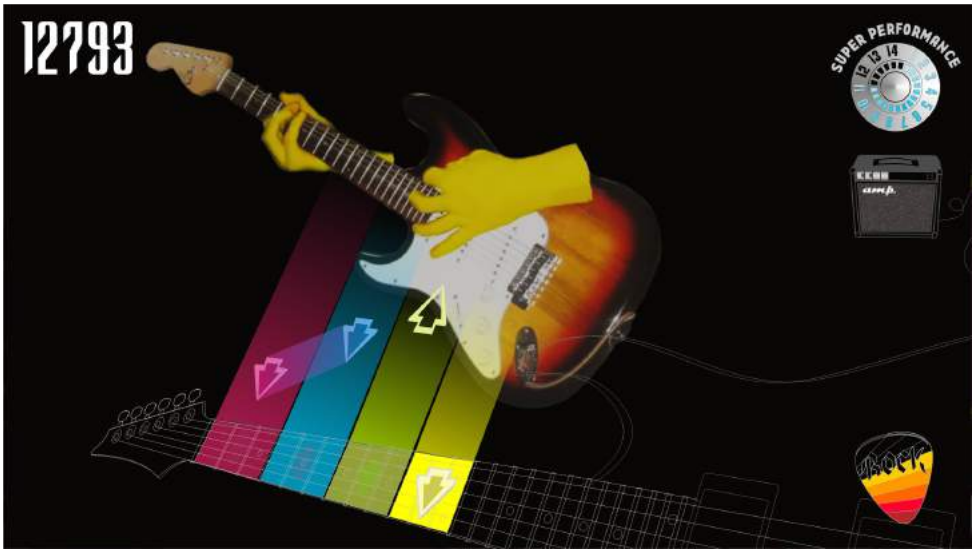


Figura 6.2: Representação conceitual dos efeitos de vibrato (topo) e slide (base) para o jogo Guitars on Air.

REFERÊNCIAS

ACTIVISION. **DJHero**. 2010.

AGGARWAL, J. K.; CAI, Q. Human motion analysis: a review. In: NONRIGID AND ARTICULATED MOTION WORKSHOP, 1997. PROCEEDINGS., IEEE. 1997. pages 90–102.

ASUS. **Xtion PRO, Xtion PRO LIVE**. 2011.

BARBANCHO, A. M. et al. Automatic edition of songs for guitar hero/frets on fire. In: MULTIMEDIA AND EXPO, 2009. ICME 2009. IEEE INTERNATIONAL CONFERENCE ON. 2009. pages 1186–1189.

BASEOPS. **JSUPT - USAF Military Pilot Training Information**. 2012.

BOUGUET, J.-Y. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. **Intel Corporation**, volume 5, number 1-10, pages 4, 2001.

BOYLE, M. **The effects of capture conditions on the CAMSHIFT face tracker**. Alberta, Canada: Department of Computer Science, University of Calgary, 2001. (2001-691-14).

BRADSKI, G.; KAEHLER, A. **Learning OpenCV**: computer vision with the opencv library. "O'Reilly Media, Inc.", 2008.

BRADSKI, G. R. Real time face and object tracking as a component of a perceptual user interface. In: APPLICATIONS OF COMPUTER VISION, 1998. WACV'98. PROCEEDINGS., FOURTH IEEE WORKSHOP ON. 1998. pages 214–219.

BRAY, M.; KOLLER-MEIER, E.; VAN GOOL, L. Smart particle filtering for high-dimensional tracking. **Computer Vision and Image Understanding**, volume 106, number 1, pages 116–129, 2007.

BROTHERSON, S. E. **Understanding brain development in young children**. NDSU Extension Service Fargo, ND, 2005.

CANNY, J. A computational approach to edge detection. **IEEE Transactions on pattern analysis and machine intelligence**, number 6, pages 679–698, 1986.

CARROLL, J. M. Human–computer interaction. **Encyclopedia of Cognitive Science**, 2009.

CHARYTONOWICZ, J. Tomorrow's Ergonomics. In: PROCEEDINGS OF THE HUMAN FACTORS AND ERGONOMICS SOCIETY ANNUAL MEETING. 2000. volume 44, number 33, pages 6–194.

CHEN, C. et al. Visualizing the Evolution of HCI. **People and Computers XIX—The Bigger Picture**, pages 233–250, 2006.

COUTINHO, F. L.; MORIMOTO, C. H. A depth compensation method for cross-ratio based eye tracking. In: PROCEEDINGS OF THE 2010 SYMPOSIUM ON EYE-TRACKING RESEARCH & APPLICATIONS. 2010. pages 137–140.

FIALA, M. Magic mirror system with hand-held and wearable augmentations. In: VIRTUAL REALITY CONFERENCE, 2007. VR'07. IEEE. 2007. pages 251–254.

FIGUEIREDO, L. S. et al. An open-source framework for air guitar games. In: GAMES AND DIGITAL ENTERTAINMENT (SBGAMES), 2009 VIII BRAZILIAN SYMPOSIUM ON. 2009. pages 74–82.

FRANCOIS, A. R. **CAMSHIFT tracker design experiments with Intel OpenCV and SAI**. UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES INST FOR ROBOTICS AND INTELLIGENT SYSTEMS, 2004.

GAMES, M. **Adagio**. 2008.

GOMEZ, G.; MORALES, E. Automatic feature construction and a simple rule induction algorithm for skin detection. In: ICML WORKSHOP ON MACHINE LEARNING IN COMPUTER VISION. 2002. volume 31.

HACKENBERG, G.; MCCALL, R.; BROLL, W. Lightweight palm and finger tracking for real-time 3D gesture control. In: VIRTUAL REALITY CONFERENCE (VR), 2011 IEEE. 2011. pages 19–26.

HARRISON, S.; TATAR, D. The three paradigms of HCI. **SIGCHI, Alt Chi. Session**, pages 1–21, 2007.

HERDA, L. et al. Skeleton-based motion capture for robust reconstruction of human motion. In: COMPUTER ANIMATION 2000. PROCEEDINGS. 2000. pages 77–83.

HOEY, J. et al. Tracking using Flocks of Features, with Application to Assisted Handwashing. In: BMVC. 2006. pages 367–376.

IKE, T.; KISHIKAWA, N.; STENGER, B. A Real-Time Hand Gesture Interface Implemented on a Multi-Core Processor. In: MVA. 2007. pages 9–12.

INC., C. **P5 Glove - Virtual Reality Data Glove**. 2012.

JOHN, C.; SCHWANECKE, U.; REGENBRECHT, H. Real-time volumetric reconstruction and tracking of hands and face as a user interface for virtual environments. In: VIRTUAL REALITY CONFERENCE, 2009. VR 2009. IEEE. 2009. pages 241–242.

JONES, M.; REHG, J. Statistical color models with application to skin detection. In: COMPUTER VISION AND PATTERN RECOGNITION, 1999. IEEE COMPUTER SOCIETY CONFERENCE ON. 1999. volume 1, pages –280 Vol. 1.

KALAWSKY, R. S. VRUSE—a computerised diagnostic tool: for usability evaluation of virtual/synthetic environment systems. **Applied ergonomics**, volume 30, number 1, pages 11–25, 1999.

KANELLOS, M. **Personal Computers**: more than 1 billion served. 2002.

KARJALAINEN, M. et al. Virtual air guitar. **Journal of the Audio Engineering Society**, volume 54, number 10, pages 964–980, 2006.

KÖLSCH, M.; TURK, M. Flocks of features for tracking articulated objects. **Real-Time Vision for Human-Computer Interaction**, pages 67–83, 2005.

KOVAC, J.; PEER, P.; SOLINA, F. 2D versus 3D colour space face detection. In: VIDEO/IMAGE PROCESSING AND MULTIMEDIA COMMUNICATIONS, 2003. 4TH EURASIP CONFERENCE FOCUSED ON. 2003. volume 2, pages 449–454.

LEE, J. C. Hacking the nintendo wii remote. **IEEE pervasive computing**, volume 7, number 3, 2008.

LIKERT, R. A technique for the measurement of attitudes. **Archives of psychology**, 1932.

LIMA, J. et al. Online monocular markerless 3d tracking for augmented reality. **Abordagens Práticas de Realidade Virtual e Aumentada**, pages 1–30, 2009.

MAHMOUDI, F.; PARVIZ, M. Visual hand tracking algorithms. In: GEOMETRIC MODELING AND IMAGING–NEW TRENDS, 2006. 1993. pages 228–232.

MEHRABIAN, A. et al. **Silent messages**. Wadsworth Belmont, CA, 1971. volume 8.

MICROSOFT, C. **Kinect for Xbox 360**. URL: http://www.xbox.com/en-US/xbox-360/accessories/kinect.

MILLER, G. A. The magical number seven, plus or minus two: some limits on our capacity for processing information. **Psychological review**, volume 101, number 2, pages 343, 1994.

MINKLEY, J. **Rare**: kinect lag "not an issue". 2010.

MITTAL, A.; ZISSERMAN, A.; TORR, P. H. Hand detection using multiple proposals. In: BMVC. 2011. pages 1–11.

MOESLUND, T. B.; GRANUM, E. A survey of computer vision-based human motion capture. **Computer vision and image understanding**, volume 81, number 3, pages 231–268, 2001.

MURPHY-CHUTORIAN, E.; TRIVEDI, M. M. Head pose estimation in computer vision: a survey. **IEEE transactions on pattern analysis and machine intelligence**, volume 31, number 4, pages 607–626, 2009.

MUSIC, H. **Frequency**. 2001.

MUSIC, H. **Rock Band**. 2007.

MYERS, B. **A brief history of human-computer interaction technology**. Interactions, 1998.

NINTENDO. **Nintendo Wii**. 2006.

OIKONOMIDIS, I.; KYRIAZIS, N.; ARGYROS, A. Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. In: COMPUTER VISION (ICCV), 2011 IEEE INTERNATIONAL CONFERENCE ON. 2011. pages 2088–2095.

ONG, E.-J. et al. Robust facial feature tracking using selected multi-resolution linear predictors. In: IEEE 12TH INTERNATIONAL CONFERENCE ON, 2009. 2009. pages 1483–1490.

PAKARINEN, J.; PUPUTTI, T.; VÄLIMÄKI, V. Virtual slide guitar. **Computer Music Journal**, volume 32, number 3, pages 42–54, 2008.

PAN, Z. et al. A real-time multi-cue hand tracking algorithm based on computer vision. In: VIRTUAL REALITY CONFERENCE (VR), 2010 IEEE. 2010. pages 219–222.

PETRESCU, S. et al. **Color segmentation**. US Patent 8,055,067.

PHUNG, S. L.; BOUZERDOUM, A.; CHAI, D. Skin segmentation using color and edge information. In: SIGNAL PROCESSING AND ITS APPLICATIONS, 2003. PROCEEDINGS. SEVENTH INTERNATIONAL SYMPOSIUM ON. 2003. volume 1, pages 525–528.

PINELLE, D.; WONG, N.; STACH, T. Heuristic evaluation for games: usability principles for video game design. In: PROCEEDINGS OF THE SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS. 2008. pages 1453–1462.

POPESCU, V.; BURDEA, G.; BOUZIT, M. Virtual reality simulation modeling for a haptic glove. In: COMPUTER ANIMATION, 1999. PROCEEDINGS. 1999. pages 195–200.

PROUS, A. et al. **OS MACHADOS PRÉ-HISTÓRICOS NO BRASIL DESCRIÇÃO DE COLEÇÕES BRASILEIRAS E TRABALHOS EXPERIMENTAIS**: fabricação de lâminas, cabos, encabamento e utilização1. 2002.

RASKAR, R. et al. Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators. In: ACM SIGGRAPH 2007 PAPERS, New York, NY, USA. ACM, 2007. (SIGGRAPH '07).

REIMER, J. **Total share**: 30 years of personal computer market share figures. 2009.

RODRIGUEZ, S.; PICON, A.; VILLODAS, A. Robust vision-based hand tracking using single camera for ubiquitous 3D gesture interaction. In: D USER INTERFACES (3DUI), 2010 IEEE SYMPOSIUM ON, 3. 2010. pages 135–136.

SAWANGSRI, T.; PATANAVIJIT, V.; JITAPUNKUL, S. Face segmentation using novel skin-color map and morphological technique. In: PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY. 2005. volume 2, pages 41–44.

SEEINGMACHINES. **faceAPI**. 2012.

SHAN, C.; GONG, S.; MCOWAN, P. W. Facial expression recognition based on local binary patterns: a comprehensive study. **Image and Vision Computing**, volume 27, number 6, pages 803–816, 2009.

SHI, J. et al. Good features to track. In: COMPUTER VISION AND PATTERN RECOGNITION, 1994. PROCEEDINGS CVPR'94., 1994 IEEE COMPUTER SOCIETY CONFERENCE ON. 1994. pages 593–600.

SHOTTON, J. et al. Real-time human pose recognition in parts from single depth images. In: CVPR 2011. 2011. pages 1297–1304.

SOLAR, J. Ruiz-del; VERSCHAE, R. Robust skin segmentation using neighborhood information. In: IMAGE PROCESSING, 2004. ICIP '04. 2004 INTERNATIONAL CONFERENCE ON. 2004. volume 1, pages 207–210 Vol. 1.

SONY. **Playstation Move**. 2010.

SOUZA, P. A. **Uma técnica de rastreamento de mãos para interação natural na plataforma Web**. Recife, Brazil: Centro de Informática da Universidade Federal de Pernambuco, 2012.

STENGER, B. Template-Based hand pose recognition using multiple cues. In: PROCEEDINGS OF THE 7TH ASIAN CONFERENCE ON COMPUTER VISION - VOLUME PART II, Berlin, Heidelberg. Springer-Verlag, 2006. pages 551–560. (ACCV'06).

STENGER, B. et al. Filtering using a tree-based estimator. In: NULL. 2003. pages 1063.

STENGER, B. et al. Model-based hand tracking using a hierarchical bayesian filter. **IEEE transactions on pattern analysis and machine intelligence**, volume 28, number 9, pages 1372–1384, 2006.

SVOBODA, E.; RICHARDS, B. Compensating for anterograde amnesia: a new training method that capitalizes on emerging smartphone technologies. **Journal of the International Neuropsychological Society**, volume 15, number 4, pages 629–638, 2009.

SWELLER, J. Cognitive load during problem solving: effects on learning. **Cognitive science**, volume 12, number 2, pages 257–285, 1988.

SYSTEMS, C. **CyberGlove II**. 2010.

THAYANANTHAN, A. et al. Pose estimation and tracking using multivariate regression. **Pattern Recognition Letters**, volume 29, number 9, pages 1302–1310, 2008.

TITTERTON, D.; WESTON, J. L. **Strapdown inertial navigation technology**. IET, 2004. volume 17.

TRENDS, D. **US Smartphone Users By Age**: comparison chart. 2011.

TSEKERIDOU, S.; PITAS, I. Facial feature extraction in frontal views using biometric analogies. In: SIGNAL PROCESSING CONFERENCE (EUSIPCO 1998), 9TH EUROPEAN. 1998. pages 1–4.

TUROQUE, B.; CRANE, D. **To air is human**: one man's quest to become the world's greatest air guitarist. Penguin, 2006.

VADAKKEPAT, P. et al. Multimodal approach to human-face detection and tracking. **IEEE transactions on industrial electronics**, volume 55, number 3, pages 1385–1393, 2008.

VALLI, A. **Notes on Natural Interaction**. 2005.

VEIGL, S. et al. Two-handed direct interaction with ARTootKit. In: AUGMENTED REALITY TOOLKIT, THE FIRST IEEE INTERNATIONAL WORKSHOP. 2002. pages 2–pp.

VEZHNEVETS, V.; SAZONOV, V.; ANDREEVA, A. A Survey on Pixel-Based Skin Color Detection Techniques. In: PROCEEDINGS OF THE GRAPHICON 2003. 2003. pages 85–92.

VIOLA, P.; JONES, M. J. Robust real-time face detection. **International journal of computer vision**, volume 57, number 2, pages 137–154, 2004.

WANG, J.-G.; SUNG, E. EM enhancement of 3D head pose estimated by point at infinity. **Image and Vision Computing**, volume 25, number 12, pages 1864–1874, 2007.

WANG, R. Y.; POPOVIĆ, J. Real-time hand-tracking with a color glove. In: ACM TRANSACTIONS ON GRAPHICS (TOG). 2009. volume 28, number 3, pages 63.

WINGFIELD, N. **Time to Leave the Laptop Behind**. 2009.

WIXON, D. Guitar Hero: the inspirational story of an overnight success. **interactions**, volume 14, number 3, pages 16–17, 2007.

WORLD, B. N. **Computers reach one billion mark**. 2002.

APÊNDICES

APÊNDICE A

Este apêndice é dedicado à apresentação dos dados utilizados durante o processo de avaliação no estudo de caso. Mais a frente, também neste apêndice, se encontram conjuntos de imagens, resultantes da segmentação de cada um dos métodos testados, os quais foram utilizados para selecionar o critério de pixels iniciadores e espalhadores explanado na subseção 4.2.2. Abaixo se encontram os dados relacionados às métricas armazenadas durante as 12 sessões de jogo para cada uma das ferramentas de rastreamento testada. A legenda para os títulos da coluna segue abaixo:

- Miss Notes: o total de notas (setas) perdidas durante a sessão.
- Wrong Rail: número de notas perdidas por conta da mão esquerda estar sobre o trilho incorreto.
- Sleep Notes: o número de notas perdidas por que a mão direita não chegou a executar o gesto.
- Hit Notes: o número total de notas acertadas na sessão.
- Max Streak: o máximo de notas acertadas consecutivamente.
- Total Points: o número total de pontos conquistados ao longo da sessão de jogo.
- Percent: a porcentagem de acertos sobre o total de notas da sessão (157).

1. Série 1: barras em azul; resultados do Kinect.
2. Série 2: barras em vermelho; resultados do HAFT
3. Série 3: barras em verde; resultados da detecção de Luvas Laranja

Abaixo está ilustrado o questionário respondido pelos 12 usuários que participaram dos testes:

Abaixo se encontram conjuntos de imagens resultantes da segmentação de cor de pele utilizando os métodos apresentados neste trabalho. A legenda para os títulos de cada imagem segue abaixo:

- inputN: imagem colorida de entrada em que N representa o número da mesma.
- high: indicativo de que os limiares utilizados foram altos, com o intuito de minimizar os falso-positivos.

Rastreamento de Luvas Laranja

Miss Notes	Wrong Rail	Sleep Notes	Hit Notes	Max Streak	Total Points	Percent
22	2	20	135	30	285	85,99
23	16	7	134	80	10	85,35
30	1	29	127	29	230	80,89
61	0	61	96	20	141	61,15
25	16	9	132	29	259	84,08
21	3	18	136	23	285	86,62
3	0	3	154	96	532	98,09
23	4	19	134	23	316	85,35
10	1	9	147	50	382	93,63
12	0	12	145	56	445	92,36
63	34	29	94	14	120	59,87
79	0	79	78	8	91	49,68

Rastreamento usando o Kinect

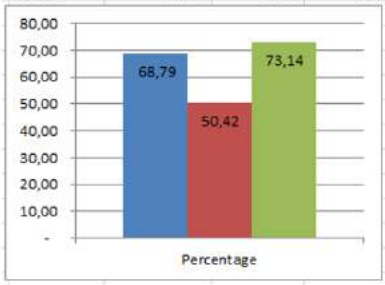
Miss Notes	Wrong Rail	Sleep Notes	Hit Notes	Max Streak	Total Points	Percent
40	8	32	96	17	139	61,15
74	6	68	83	10	99	52,87
61	11	50	96	16	149	61,15
67	6	61	90	10	119	57,32
23	16	7	134	17	244	85,35
3	0	3	154	71	519	98,09
50	13	37	107	30	195	68,15
8	2	6	149	62	392	94,90
19	16	3	138	24	255	87,90
3	1	2	154	87	513	98,09
140	56	84	17	3	17	10,83
79	8	71	78	7	86	49,68

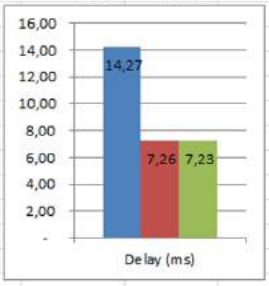
Rastreamento usando o HAFT

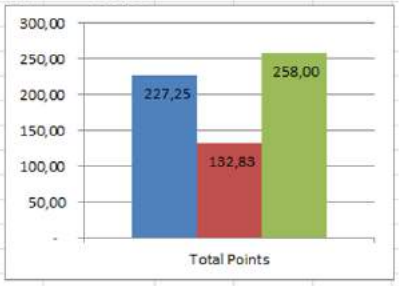
Miss Notes	Wrong Rail	Sleep Notes	Hit Notes	Max Streak	Total Points	Percent
60	7	53	97	19	144	61,78
98	0	98	59	7	64	37,58
55	3	52	102	18	172	64,97
128	14	114	29	10	36	18,47
95	20	75	62	13	96	39,49
30	0	30	127	23	269	80,89
15	15	0	142	41	340	90,45
29	12	17	128	20	249	81,53
113	8	105	44	9	52	28,03
84	2	82	73	8	81	46,50
144	12	132	13	2	13	8,28
53	10	43	74	5	78	47,13

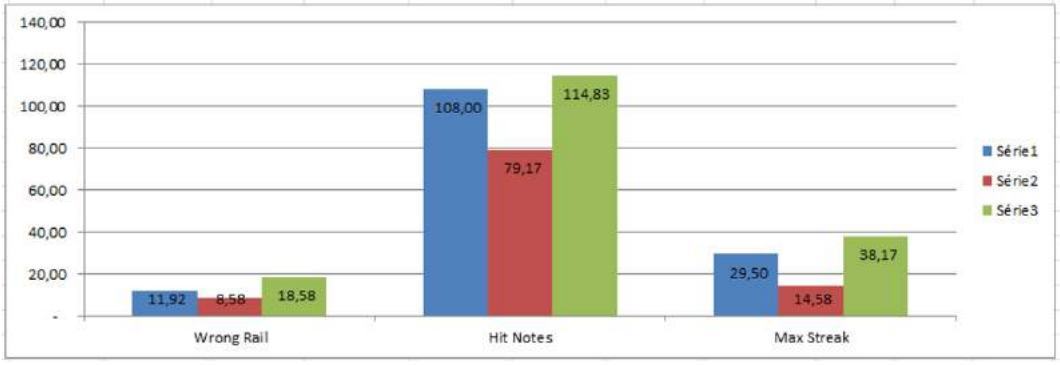
Análise Geral dos Dados Providos pelos Testes com o jogo Guitars on Air

	Percentage	Delay (ms)	Miss Notes	Wrong Rail	Hit Notes	Max Streak	Total Points
Kinect	68,79	14,27	47,25	11,92	108,00	29,50	227,25
HaFT	50,42	7,26	75,33	8,58	79,17	14,58	132,83
Glove	73,14	7,23	31,00	18,58	114,83	38,17	258,00









1. De forma geral, é fácil interpretar as representações visuais fornecidas pelo jogo

1 2 3 4 5

Não concordo de forma alguma ◯ ◯ ◯ ◯ ◯ Concordo 100%

2. O jogo apresenta uma interface clara, livre de ruídos

1 2 3 4 5

Não concordo de forma alguma ◯ ◯ ◯ ◯ ◯ Concordo 100%

3. O número de elementos visuais mostrados simultaneamente na tela é o ideal

1 2 3 4 5

Não concordo de forma alguma ◯ ◯ ◯ ◯ ◯ Concordo 100%

4. Na tela do jogo, é fácil distinguir o conteúdo interativo do não-interativo

1 2 3 4 5

Não concordo de forma alguma ◯ ◯ ◯ ◯ ◯ Concordo 100%

5. De forma geral, as sequências de comandos são fáceis de serem executadas

1 2 3 4 5

Não concordo de forma alguma ◯ ◯ ◯ ◯ ◯ Concordo 100%

6. A curva de aprendizado é suave, a adaptação e execução da tarefa apresentada evolue bem com pouco tempo de uso

1 2 3 4 5

Não concordo de forma alguma ◯ ◯ ◯ ◯ ◯ Concordo 100%

7. De forma geral, a resposta às ações do jogador são coerentes

	1	2	3	4	5
Luvas	◯	◯	◯	◯	◯
HaFT	◯	◯	◯	◯	◯
Kinext	◯	◯	◯	◯	◯

8. Todas as vezes que um comando é enviado intencionalmente pelo jogador, ele é detectado

	1	2	3	4	5
Luvas	◯	◯	◯	◯	◯
HaFT	◯	◯	◯	◯	◯
Kinect	◯	◯	◯	◯	◯

9. O jogo não enxerga movimentos inexistentes; só computa comandos realmente efetuados pelo jogador

	1	2	3	4	5
Luvas	◯	◯	◯	◯	◯
HaFT	◯	◯	◯	◯	◯
Kinect	◯	◯	◯	◯	◯

10. De forma geral, é fácil controlar as ações no jogo

	1	2	3	4	5
Luvas	◯	◯	◯	◯	◯
HaFT	◯	◯	◯	◯	◯
Kinect	◯	◯	◯	◯	◯

11. A sensibilidade do controle está ajustada corretamente

	1	2	3	4	5
Luvas	◯	◯	◯	◯	◯
HaFT	◯	◯	◯	◯	◯
Kinect	◯	◯	◯	◯	◯

12. As respostas para os comandos enviados ao jogo são fornecidas instantaneamente

	1	2	3	4	5
Luvas	◯	◯	◯	◯	◯
HaFT	◯	◯	◯	◯	◯
Kinect	◯	◯	◯	◯	◯

13. O uso da luva gerou incomodo durante a execução do jogo?

1 2 3 4 5

Nenhum ◯ ◯ ◯ ◯ ◯ Muito

14. Você é familiarizado com jogos eletrônicos?

1 2 3 4 5

Não, nem lembro a última vez que joguei alguma coisa ◯ ◯ ◯ ◯ Muito, jogo praticamente todo dia

15. Você já jogou os jogos da franquia Guitar Hero ou Rock Band?

1 2 3 4 5 6

Nunca ◯ ◯ ◯ ◯ ◯ ◯ Loucamente! e só jogo no expert!

16. Você achou a experiência cansativa, se sentiu fadigado após a execução de cada música?

1 2 3 4 5

Nem um pouco ◯ ◯ ◯ ◯ ◯ Muito

17. De forma geral, você achou a experiência divertida?

1 2 3 4 5

Nem um pouco ◯ ◯ ◯ ◯ ◯ Muito

- low: indicativo de que os limiares utilizados foram baixos, com o intuito de minimizar os falso-negativos.
- histograms: indicativo de que o método utilizado foi baseado em histogramas de treinamento
- BGR: indicativo de que o espaço de cor usado é o RGB.
- HSV: indicativo de que o espaço de cor usado é o HSV.
- YCC: indicativo de que o espaço de cor usado é o YCbCr.
- combined histograms: resultado da segmentação sobreposta dos histogramas nos três espaços de cor.
- TsekeridouPitas: indica que a técnica usada foi a apresentada em TSEKERIDOU *et al.* 1998.
- GomezMorales: indica que a técnica usada foi a apresentada em GOMEZ *et al.* 2002.
- Google: indica que a técnica usada foi a apresentada em PETRESCU *et al.* 2011.
- Kovacetal: indica que a técnica usada foi a apresentada em KOVAC *et al.* 2003.
- neighbourhood: indica que o método utilizado foi de propagação na vizinhança proposto neste trabalho.
- edges: indica que o método de propagação foi limitado por arestas extraídas com o CANNY 1986.
- .png: descreve o formato do arquivo utilizado para armazenar a imagem.


input1.png


low combined histograms.png


low histogramsBGR.png

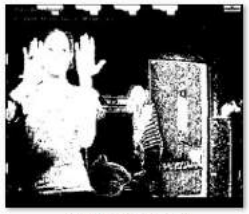

low histogramsHSV.png


low histogramsYCC.png


low HSVTsekeridouPites.png


low RGBGomezMorales.png


low RGBGoogle.png


low RGBKovacetal.png


neighbourhood edges.png


neighbourhood.png


original HSVTsekeridouPites.png


original RGBGomezMorales.png


original RGBGoogle.png


original RGBKovacetal.png


high combined histograms.png


high histogramsBGR.png


high histogramsHSV.png


high histogramsYCC.png


high HSVTsekeridouPites.png


high RGBGomezMorales.png


high RGBGoogle.png


high RGBKovacetal.png


input2.png


low combined histograms.png


low histogramsBGR.png


low histogramsHSV.png


low histogramsYCC.png


low HSVTsekeridouPitas.png


low RGBGomezMorales.png


low RGBGoogle.png


low RGBKovacetal.png


neighbourhood edges.png


neighbourhood.png


original HSVTsekeridouPitas.png


original RGBGomezMorales.png


original RGBGoogle.png


original RGBKovacetal.png


high combined histograms.png


high histogramsBGR.png


high histogramsHSV.png


high histogramsYCC.png


high HSVTsekeridouPitas.png


high RGBGomezMorales.png


high RGBGoogle.png


high RGBKovacetal.png


input3.png


low combined histograms.png


low histogramsBGR.png


low histogramsHSV.png


low histogramsYCC.png


low HSVTsekeridouPitas.png

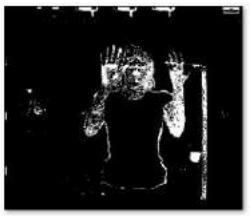

low RGBGomezMorales.png


low RGBGoogle.png


low RGBKovacetal.png


neighbourhood edges.png


neighbourhood.png


original HSVTsekeridouPitas.png


original RGBGomezMorales.png


original RGBGoogle.png


original RGBKovacetal.png


high combined histograms.png


high histogramsBGR.png


high histogramsHSV.png


high histogramsYCC.png


high HSVTsekeridouPitas.png


high RGBGomezMorales.png


high RGBGoogle.png


high RGBKovacetal.png


input4.png


low combined histograms.png


low histogramsBGR.png


low histogramsHSV.png


low histogramsYCC.png


low HSVTsekeridouPitas.png


low RGBGomezMorales.png


low RGBGoogle.png


low RGBKovacetal.png


neighbourhood edges.png


neighbourhood.png


original HSVTsekeridouPitas.png


original RGBGomezMorales.png


original RGBGoogle.png


original RGBKovacetal.png