Reconhecimento de gestos tridimensionais

Silvia Esparrachiari Ghirotti

Dissertação/Tese apresentada
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Mestre em Ciências

Programa: Ciência da Computação
Orientador: Prof. Dr. Carlos Hitoshi Morimoto

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da FAPESP

São Paulo, abril de 2010

Reconhecimento de gestos tridimensionais

Este exemplar corresponde à redação
final da dissertação defendida por
Silvia Esparrachiari Ghirotti, revisada
e aprovada pela Comissão Julgadora.

Banca Examinadora:

- Prof. Dr. Carlos Hitoshi Morimoto (orientador) - IME-USP.
- Prof. Dr. Romero Tori - EP-USP.
- Prof. Dr. Flavio Soares Correa da Silva - IME-USP.

Ao meu grande amigo e companheiro Victor,
por não me deixar desistir jamais.

Resumo

Sistemas de visualização imersiva do tipo *Cave Automatic Virtual Environment* (CAVE) ou sistemas que utilizam grandes dispositivos de alta definição (*powerwall*), são comumente utilizados para a navegação e interação em ambientes virtuais. Interfaces mais tradicionais baseadas em *mouse* e teclado, não são adequadas a estes sistemas e muitas das soluções de interação existentes apresentam desvantagens como custo elevado, conjunto de funcionalidades limitado, grande complexidade de uso, necessidade de treinamento do usuário, dentre outras. Neste trabalho, propomos uma interface baseada em gestos manuais tridimensionais que permite a navegação e interação com objetos em ambientes virtuais de forma remota, ou seja, sem que o usuário tenha contato com qualquer tipo de dispositivo.

A interface de gestos foi feita utilizando-se um sistema de câmeras em estéreo e de técnicas de visão computacional para a segmentação e rastreamento das mãos e da cabeça do usuário em tempo real. O sistema de gestos foi dividido em dois subsistemas: um subsistema de gestos simbólicos para comandos de interação com os objetos, como pegar, soltar e rotacionar, e um subsistema de gestos naturais para a navegação. Os testes do sistema foram divididos em três etapas, verificando primeiramente o desempenho dos usuários em cada um dos subsistemas de gestos e, por fim, no sistema completo. Quatro usuários, com perfis variados, participaram dos experimentos. Os resultados mostram que todos foram capazes de utilizar os subsistemas separadamente e que a combinação dos dois tipos de gestos no sistema final não acarretou num aumento significativo da complexidade de utilização pelo usuário.

Palavras-chave: reconhecimento de gestos, visão computacional, interfaces baseadas em gestos, gestos naturais, gestos simbólicos, máquinas de estado finitas, rastreamento tridimensional de objetos, tempo-real

Abstract

Immersive visualization systems such as Cave Automatic Virtual Environment (CAVE) and high definition large display (powerwall) are commonly used for navigation and interaction in virtual environments. Traditional interfaces, using mouse and keyboard are not suitable for these kind of systems and many of the existing interaction solutions present disadvantages such as high cost, limited set of features, high complexity of use, need for user training, among others. In this work, we propose a three-dimensional hand gesture based interface that allows remote navigation and interaction with objects in a virtual environment, i.e. without the need of devices in physical contact with the user.
The gesture interface was implemented using a stereo camera system and computer vision techniques for segmentation and tracking of the user's hands and head in real time. The gesture system was split into two subsystems: a symbolic gesture subsystem for interaction with objects, to catch, drop and rotate, and a natural gesture subsystem for navigation. System tests were done in three steps, first checking the users' performance in each of the subsystems, and finally, in the complete system. Four users with different profiles have collaborated with the experiments. The results show that every user was able to use the subsystems separately and that a combination of both types of gestures in the final system did not cause a significant increase in the complexity of use.

Keywords: Gesture recognition, computer vision, gesture-based interfaces, natural gestures, symbolic gestures, finite state machines, three-dimensional object tracking, real-time.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A comunicação humana é uma atividade complexa que vai muito além das palavras. Uma das principais ferramentas de suporte à comunicação são os gestos. A família de movimentos que chamamos de gestos é bastante ampla, incluindo desde expressões faciais, passando por movimentos realizados com as mãos, até atividades realizadas com o corpo inteiro. Neste trabalho, abordamos apenas o estudo dos gestos realizados com as mãos e os braços (gestos manuais).

Gestos associados à comunicação são denominados *gestos semióticos* podendo ser classificados de acordo com a sua dependência com o discurso falado. Interfaces computacionais puramente baseadas em gestos utilizam, principalmente, uma subcategoria de gestos semióticos, denominados *simbólicos*, como vocabulário de interação. Uma outra subcategoria bastante utilizada são os gestos denominados naturais, que descartam a necessidade de aprendizado e treinamento, mas possuem uma área de aplicação bastante restrita.

Computacionalmente, a estrutura de uma interface baseada em gestos pode ser dividida em três partes: rastreamento, reconhecimento e interpretação. O rastreamento realiza a identificação da pessoa na cena e a transformação da sua pose num dado momento ou com o passar do tempo. O reconhecimento interpreta as informações obtidas na etapa de rastreamento e identifica o gesto sendo realizado. A interpretação contextualiza o gesto ou uma sequência de gestos e extrai desta uma atividade de interação.

Neste trabalho, apresentamos algumas das principais técnicas de rastreamento do movimento humano e de reconhecimento de gestos, juntamente com alguns exemplos de trabalhos relacionados. Como exemplo de interpretação, construímos uma interface puramente baseada em gestos para realizar a navegação e manipulação de objetos no contexto de um quebra-cabeça virtual tridimensional.

1.1 Objetivo

O objetivo principal deste trabalho consiste no estudo e desenvolvimento de um sistema de reconhecimento de gestos manuais tridimensionais, sem marcadores. A interface permite a exploração e manipulação de objetos em um ambiente virtual tridimensional.

1.2 Principais contribuições e desafios

As principais contribuições deste trabalho são:

- levantamento bibliográfico do estado da arte dos sistemas de reconhecimento de gestos e suas aplicações como interface de interação;
- estipulação de um problema cuja utilização de interfaces baseadas em gestos seja adequada e criação de um vocabulário de gestos adequado ao problema;
- desenvolvimento de um sistema de visão computacional de tempo real para reconhecimento de poses manuais;
- desenvolvimento de um sistema de reconhecimento de gestos manuais de tempo real;
- definição de um protocólo experimental e realização de testes para avaliação da interface proposta;

1.3 Organização do trabalho

Iniciamos este trabalho com uma definição do termo "gesto" e um pequeno resumo da utilização dos gestos no dia-a-dia das pessoas, juntamente com sua taxonomia. Comentamos sobre os diversos tipos existentes de interface baseada em gestos e propomos uma metodologia de desenvolvimento destas interfaces. Nos capítulos 3 e 4, apresentamos os componentes estruturais de uma interface de gestos e as principais ferramentas utilizadas na sua construção. O capítulo 5 trata da interface desenvolvida como parte deste projeto, contextualizando-a em meio aos demais trabalhos da área e descrevendo em detalhes sua implementação. No capítulo 6, apresentamos os testes realizados para medir a usabilidade da interface e realizamos uma análise dos seus resultados. Concluímos este trabalho no capítulo 7.

Capítulo 2

Gestos

"movimento expressivo de idéias."
fonte: dicionário Priberam da língua portuguesa.

Parte do conteúdo deste capítulo foi baseado no texto ainda não publicado de Buxton [Bux10] sobre interfaces baseadas em gestos.

Nos últimos anos, termos como gestos e reconhecimento de gestos têm aparecido cada vez mais em pesquisas de interação homem-máquina (IHC). A princípio, tais termos estavam diretamente relacionados a sistemas de reconhecimento de caracteres, símbolos de edição, taquigrafia e diversos outros tipos de expressões gráficas definidas como curvas num plano. Gestos, no entanto, possuem um significado muito mais abrangente e estão relacionados a quase toda atividade física, tendo um importante papel no estabelecimento da qualidade da ação executada. Neste trabalho, nos atemos ao estudo de sistemas de reconhecimento de gestos naturais, que não utilizam qualquer tipo de dispositivo para sua expressão. Fazemos uso, então, da definição de gesto proposta por Mitra e Acharya [MA07]:

"Gestos são movimentos corporais expressivos e significativos realizados através da movimentação dos dedos, mãos, braços, cabeça, face ou corpo com o objetivo de: 1) transmitir uma informação ou 2) interagir com o ambiente."

Ou seja, pegar um objeto, correr ou indicar uma direção são atividades consideradas gestos, pois seus modos de execução possuem um papel importante na realização da atividade. Digitar um texto, no entanto, não corresponde a um gesto, pois o sinal referente a um caractere será o mesmo não importando como a tecla seja pressionada.

A execução de um gesto possui um alto grau de liberdade, o que torna a comunicação através de gestos bastante rica e complexa. Por isso, sistemas de reconhecimento de gestos contam com uma grande variedade de dispositivos utilizados na identificação e rastreamento das partes do

corpo relevantes à comunicação. A escolha do dispositivo interfere diretamente na complexidade e qualidade dos gestos analisados. Por exemplo, gestos realizados com dispositivos de ponto único (*single point devices*), como o *mouse*, limitam o vocabulário do usuário a um conjunto de símbolos planos, compostos por um ou mais traços. Enquanto que dispositivos de rastreamento 3D permitem que o usuário realize gestos mais amplos e naturais expandindo consideravelmente seu vocabulário.

Antes de analisarmos como gestos podem ser utilizados para interação com computadores, faremos algumas considerações à respeito do papel dos gestos no cotidiano e como podemos classificá-los de acordo com sua funcionalidade e dependência com outros canais de comunicação.

2.1 Taxonomia dos gestos no cotidiano

Ao colocarmos o universo dos computadores de lado por um momento e levarmos em conta apenas as interações sociais humanas, percebemos que utilizamos um amplo vocabulário de gestos para nos comunicarmos. Os gestos utilizados variam de acordo com aspectos contextuais e culturais [Kit09] e, ainda assim, estão intimamente ligados à comunicação. Por exemplo, pessoas falando ao telefone gesticulam normalmente, mesmo que seu interlocutor não seja capaz de os ver.

Gestos podem possuir significados isolados – acenar, aplaudir e apontar numa direção – ou envolvendo objetos externos – chutar uma bola, pegar e mover um objeto. Podemos, então, classificar os gestos quanto à sua funcionalidade. Cadoz [Cad94] propõe uma classificação em três grupos:

- **semióticos**: utilizados para comunicar uma informação significativa;
- **ergóticos**: utilizados para manipular o mundo físico e criar artefatos;
- **epistêmicos**: utilizados para aprender a partir do meio através da exploração tátil ou háptica.

Neste trabalho, estamos particularmente interessados em como os gestos podem ser utilizados na comunicação com sistemas computacionais, por isso, daremos um maior enfoque nos gestos semióticos que não utilizam dispositivos de interação. Rimé e Schiaratura [RS91] propõem a seguinte sub-classificação dos gestos semióticos quanto à sua funcionalidade:

- **simbólicos**: gestos cujo significado é único dentro de uma mesma cultura. Como, por exemplo, na cultura brasileira, o gesto de aprovação feito ao se exibir a mão fechada apenas com o polegar voltado para cima. Linguagens de sinais também se enquadram nesta categoria;
- **deícticos**: gestos mais comumente utilizados em IHC, pois são aqueles utilizados para apontar ou direcionar a atenção a um determinado evento ou objeto. São os gestos utilizados por Bolt [Bol80] para definir uma interface do tipo *"Coloque isto lá"*;
- **icônicos**: estes são os gestos utilizados para transmitir informações quanto ao tamanho, forma ou orientação de um objeto em questão. Quando um pescador diz: *"Eu pesquei um bagre **deste** tamanho"*, ao esticar seus braços lateralmente o máximo possível, ele está realizando um gesto semiótico icônico.

Gesticulação	→	Linguísticos	→	Pantomima	→	Emblemas	→	Linguagem de sinais
rítmicos e coesivos		icônicos		pantomímicos		deícticos		simbólicos

Tabela 2.1: Classificação dos gestos quanto à dependência de um canal de comunicação externo.

- **pantomímicos**: estes são os gestos realizados ao utilizarmos um instrumento ou objeto "invisível", como num jogo de mímica.

A esta última classificação, McNeill [McN92] adiciona mais duas:

- **rítmicos**: gestos realizados principalmente com as mãos e que acompanham o ritmo da conversação;
- **coesivos**: uma variação dos gestos icônicos, pantomímicos ou deícticos que são utilizados para conectar dois trechos de um discurso separados temporalmente.

Gestos também são fortemente associados ao discurso verbal, tanto como suporte ao canal de comunicação, facilitando a interpretação, quanto como canal próprio de transmissão. Considerando as seis sub-classificações dos gestos semióticos, apenas os simbólicos são capazes de serem interpretados sem a necessidade de um contexto. Os demais necessitam de um contexto, que pode ser providenciado sequencialmente por outros gestos, ações ou discurso verbal. Desta forma, podemos categorizar os gestos semióticos quanto à sua dependência com um discurso (verbal ou não):

- Gestos que evocam um discurso: **simbólicos** e **deícticos**;
- Gestos que retratam um discurso: **icônicos** e **pantomímicos**;
- Gestos que dizem respeito ao processo de conversação: **rítmicos** e **coesivos**;

A necessidade de um canal de comunicação externo para a compreensão varia de acordo com o tipo do gesto. Kendon [Ken88] ordena os gestos de acordo com sua dependência com o discurso conforme mostra a tabela 2.1.

Analisando a tabela da esquerda para a direita, vemos que os gestos se tornam mais comunicativos e gestos idiossincráticos são substituídos por sinais sócio-regulados. Por exemplo, linguagem de sinais possui características sintáticas e semânticas suficientemente próximas à linguagem verbal para que esta seja interpretada sem a necessidade de um canal externo. Já gestos icônicos não podem ser compreendidos se não forem acompanhados de um discurso.

Apesar de possuir uma taxonomia bastante rica, gestos ainda são pouco utilizados como interfaces de sistemas computacionais. As interfaces de gestos mais avançadas utilizam apenas gestos simbólicos ou deícticos. No entanto, podemos apontar duas principais razões para a utilização de gestos como interface de interação:

1. Pessoas utilizam normalmente um grande vocabulário de gestos no seu dia-a-dia e aprendem novos gestos fácil e rapidamente, simplesmente observando sua realização por outras pessoas;

2. Interfaces baseadas em gestos permitem a utilização natural de frases gestuais, que segmentam o diálogo em trechos com significados simples e fáceis de serem aprendidos e interpretados por sistemas computacionais. Por exemplo, a ação de mover um objeto pode ser segmentada em trechos como segurar o objeto, transladá-lo e soltá-lo.

2.2 Interfaces baseadas em gestos

Muitas interfaces baseadas em gestos combinam a utilização de gestos com a utilização de outros dispositivos de interação, como sistemas de reconhecimento da fala, *joysticks*, dispositivos de apontamento e outros. Interfaces que utilizam apenas gestos como modo de interação são conhecidas como interfaces puramente baseadas em gestos, enquanto que as demais são conhecidas como interfaces multimodais.

De um modo geral, podemos dividir a construção de uma interface baseada em gestos em três partes: rastreamento, reconhecimento e interpretação. A etapa de rastreamento diz respeito a como os gestos realizados pelo usuário serão reconhecidos pelo sistema computacional. Dentre as metodologias mais comuns estão a utilização de dispositivos físicos de rastreamento e técnicas de visão computacional. Este assunto é visto em maiores detalhes no capítulo 3. Uma vez rastreados os movimentos do usuário, na etapa de reconhecimento, são utilizados algoritmos e técnicas matemáticas com objetivo de estimar qual o gesto executado. No capítulo 4, comentamos a respeito das técnicas mais utilizadas no reconhecimento de gestos e algumas inovações. A etapa final contextualiza o gesto e o associa a uma ação única, tendo em vista a aplicação e o histórico de atividades realizadas. A seguir, abordaremos com mais detalhes esta última etapa e comentaremos sobre detalhes da sua implementação.

2.3 Interfaces puramente baseadas em gestos

Interfaces puramente baseadas em gestos são aquelas que não utilizam qualquer outro tipo de canal de comunicação além dos gestos. Os gestos reconhecidos normalmente variam desde um pequeno conjunto de gestos simbólicos até grandes vocabulários de linguagem de sinais. Os sistemas podem variar também quanto ao reconhecimento de gestos estáticos, dinâmicos ou mistos. De acordo com Mitra e Acharya [MA07], gestos estáticos são definidos por uma dada pose ou configuração espacial do usuário, enquanto que gestos dinâmicos são definidos de acordo com a movimentação do usuário num certo intervalo de tempo. Gestos mistos são aqueles compostos por elementos estáticos e dinâmicos, como por exemplo a linguagem de sinais. Em todos os casos, cada gesto apresenta um significado semântico único associado que é interpretado pela interface.

2.3.1 Interfaces de gestos naturais

O estabelecimento de um conjunto de gestos naturais diz respeito não só à aplicação como também ao usuário e ao contexto no qual se encontra. Gestos naturais são aqueles que garantem uma maior facilidade na construção do modelo mental de interação. Neste caso, definir uma interface de interação para pessoas com deficiência auditiva utilizando linguagem de sinais constitui uma interface de gestos naturais. No entanto, a mesma interface não é natural para pessoas que

não possuem habilidade no uso deste vocabulário.

Gestos naturais costumam ser muito utilizados em interfaces que utilizam o paradigma de manipulação direta, como por exemplo gestos de apontamento. Apontar para um objeto a fim de atrair a atenção de outros a ele é uma atividade que aprendemos desde cedo. Um bebê aponta para a mamadeira quando está com fome, assim como um cliente aponta para um objeto na prateleira quando pergunta seu preço numa loja de departamentos. Outro exemplo de gesto natural é a indicação de tamanho ou escala pela proximidade das mãos, como quando indicamos o tamanho de uma caixa ou pacote.

Cantoni e Garcia [CG07] permitem ao usuário, por meio da utilização de uma luva de dados (seção 3.1.1), explorar os componentes de sua instalação artística. A obra é composta por um objeto principal, uma maçã, e diversos quadros dispostos ao redor que indicam assuntos referentes à maçã. Ao fechar a mão, a pessoa é capaz de navegar entre as categorias de assuntos. Apontar numa direção permite a navegação e exploração por entre os quadros informativos.

Manders *et al.* [MFY+08] utilizam gestos naturais para controlar o movimento de uma nave espacial visualizada em um ambiente virtual através de um capacete de visualização (*head mounted display - HMD*). Os movimentos da nave são controlados pela posição relativa entre as duas mãos e um sistema de coordenadas pré-estabelecido. Por exemplo, para mover a nave para cima, o usuário dele levar ambas as mãos para cima, ao longo do eixo y, enquanto que para mover a nave para a direita o usuário deve afastar a mão esquerda para a esquerda, ao longo do eixo x.

Esta categoria de gestos possui uma importância particular no desenvolvimento de ambientes e robôs atenciosos. Problemas nesta área buscam monitorar e identificar as ações humanas sem que o usuário, necessariamente, esteja ciente da sua presença. Ambientes hospitalares podem monitorar as atividades e o estado emocional de um paciente e fornecer respostas adequadas a este comportamento como aumentar a temperatura do quarto, acionar uma medicação ou chamar um médico ou uma enfermeira. Em problemas de interface homem-robô, robôs enfermeiros ou robôs babás reconhecem atividades naturais de pessoas como andar, correr, sentar-se numa cadeira, sentar-se ou deitar-se no chão e as ajudam na sua realização [Lee06].

2.3.2 Interfaces de gestos simbólicos

Apesar de gestos naturais apresentarem uma grande facilidade de aprendizado, interfaces que utilizam um conjunto grande de comandos distintos normalmente utilizam gestos simbólicos como forma de comunicação. Apesar da linguagem de sinais poder ser classificada como simbólica e natural em determinados casos, muitos dos outros gestos simbólicos podem não se enquadrar na segunda categoria.

Gestos simbólicos são comumente utilizados em ambientes de realidade virtual imersiva, nos quais o usuário não é capaz de ver o mundo real. Nestes casos, um conjunto pré-definido de gestos é utilizado para realizar atividades de navegação e interação com o mundo virtual.

Seth *et al.* [SSSJ05] utilizam um conjunto com cinco gestos simbólicos para estabalecer uma interface com uma aplicação do tipo CAD (*Computer Aided Design* - design auxiliado por computador). A aplicação é exibida em um sistema de estéreo ativo constituído por uma tela e óculos

que propiciam a percepção da pronfundidade na cena. O usuário utiliza os gestos para manipular os objetos presentes na cena e modificar o ângulo de visão.

Kim [Kim99] utiliza gestos simbólicos para especificar comandos de controle de apresentações. Dez gestos definidos permitem ao apresentador de uma palestra controlar a sequência de imagens apresentadas voltando, avançando, reiniciando ou finalizando a apresentação, dentre outros comandos.

Baudel e Beaudouin-Lafon [BBL93] indicam algumas vantagens na utilização de gestos simbólicos para a interação:

- Interação natural: gestos podem ser definidos como uma forma natural de interação, facilitando o uso da interface;
- Conciso e poderoso: um único gesto pode definir tanto uma ação quanto seus parâmetros;
- Interação direta: a utilização de gestos elimina a necessidade de aprendizado de uso de dispositivos tradutores.

No entanto, também podemos apontar algumas desvantagens na sua utilização, como o cansaço ocasionado pela utilização contínua e prolongada e a necessidade de se aprender qual o vocabulário de gestos e como ele pode ser utilizado. O segundo problema claramente se torna pior quanto maior o vocabulário e quanto menor o grau de naturalidade da interface. Um terceiro problema ocorre ao tentarmos separar quais os gestos realizados com o propósito de interação e quais não. Por exemplo, ao posicionar os dedos de forma que apenas o dedo indicador permaneça esticado pode ser interpretado como um gesto referente ao número um, enquanto que o usuário estava apenas solicitando a alguém que fizesse silêncio (posicionando a mão próxima aos lábios). Um quarto problema é o estabelecimento do início e término de um gesto, pois muitos gestos podem corresponder a sub-unidades de outros.

Existem, no entanto, algumas medidas de implementação da interface que podem ajudar a contornar estas desvantagens. Baudel e Beaudouin-Lafon propõem que um conjunto adequado de restrições podem garantir:

- Uma detecção de intenção: gestos são reconhecidos apenas quando realizados dentro de uma região de interação ou após o sistema ter atingido um estado de ativação;
- Segmentação adequada: inícios fixos e finalizações que colaborem na separação entre gestos de comandos e gestos não considerados;
- Classificação dos gestos: gestos são classificados de acordo com sua posição inicial e dinâmica de movimentação.

Além disso, as posições iniciais devem ser distintas das finais e as posições finais dos gestos não podem ser muito distintas. Tais medidas permitem que o usuário crie sentenças com vários gestos

em sequência e permite também que um gesto seja finalizado tanto ao se atingir um ponto final ou ao se retirar a mão da região de interação.

Em seu trabalho, Baudel e Beaudouin-Lafon concluem quatro diretrizes que podem ser aplicadas no desenvolvimento de interfaces de gestos simbólicos:

- Utilizar o tensionamento da mão: em sua interface de gestos, o CHARADE, as posições iniciais envolvem o tensionamento da mão numa dada posição estática. Isto torna explícita a intenção do usuário de iniciar um comando. De modo contrário, posições finais não devem exigir tal comportamento, permitindo que o gesto seja finalizado com maior rapidez;

- Fornecer ações rápidas, incrementais e reversíveis: este é um dos princípios básicos das interfaces de manipulação direcional adaptado ao panorama das interfaces de gestos. Rapidez é essencial para evitar que o usuário se canse durante a utilização da interface. Reversibilidade permite ao usuário desfazer uma ação rapidamente e ações incrementais permitem uma resposta contínua do sistema, o que faz com que o usuário aumente sua confiança na interface;

- Favorecer um aprendizado rápido: comandos comumente utilizados devem ser associados à gestos mais naturais e de fácil aprendizado. Gestos complexos, que são mais difíceis de aprender, podem garantir ao usuário um maior controle da interface ou alguns atalhos de interação;

- Utilize os gestos apenas em tarefas apropriadas: é importante escolher com cuidado quais as tarefas (comandos) que serão realizados utilizando gestos. Enquanto gestos podem fornecer uma interface natural à atividades de navegação e exploração, estes não são adequados a sistemas que exijam alta precisão, nos quais um contato físico com um dispositivo costuma ser mais apropriado.

2.3.3 Projetando interfaces puramente baseadas em gestos

Até o momento, vimos dois tipos de interfaces puramente baseadas em gestos: interfaces de gestos naturais e interfaces de gestos simbólicos. Elas diferem entre si no tempo de treinamento necessário para a sua utilização, na expressividade e na usabilidade de suas interfaces. Interfaces de gestos naturais, apesar de apresentarem um aprendizado rápido e fácil, permitem apenas o uso de um vocabulário limitado de comandos em comparação às interfaces de gestos simbólicos. Por esta razão, projetistas de interfaces baseadas em gestos devem ser extremamente cautelosos quanto à consideração do vocabulário necessário, que servirá como parâmetro principal na escolha do tipo de interface de gesto mais apropriado.

Sturman e Zeltzer [SZ93] propõem uma metodologia de *design* iterativo para interfaces de gestos utilizando as mãos. Esta metodologia permite ao projetista avaliar a adequação deste tipo de interface e então projetar uma interface mais eficiente combinando características das tarefas às capacidades de ação das mãos e propriedades dos dispositivos (quando utilizados). O método de *design* é dividido em várias partes (figura 2.1). A primeira diz respeito à *Adequação da Aplicação* e

envolve determinar o quão bem uma interface baseada em gestos se adequa ao problema relativo à aplicação. As questões seguintes ajudam a solucionar este primeiro problema com relação às áreas de coordenação, adaptabilidade e naturalidade:

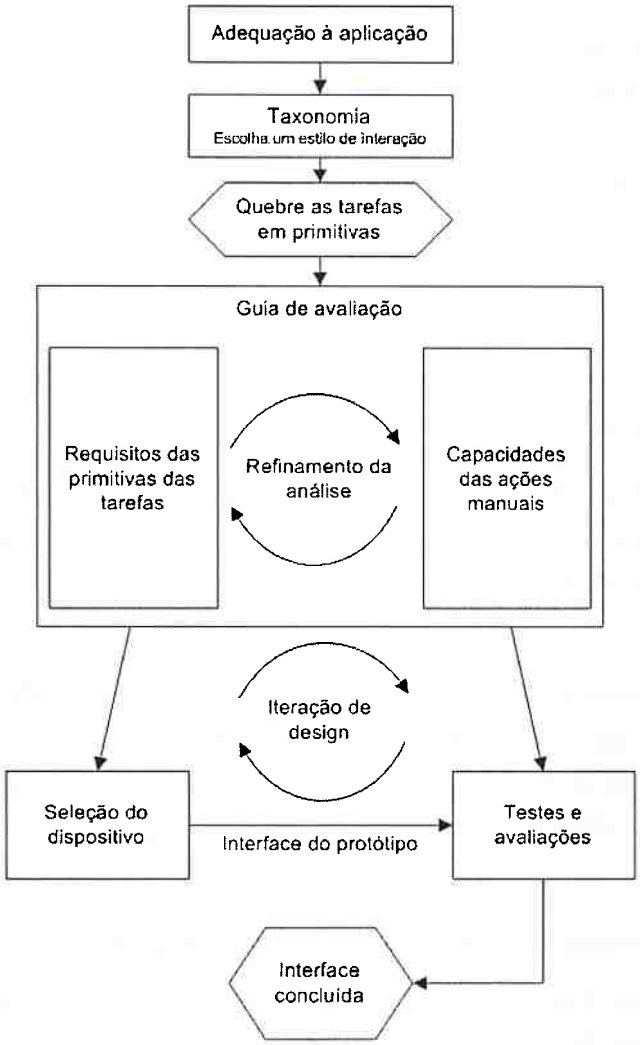

Figura 2.1: Subdivisões da metodologia de design proposta por Sturman e Zeltzer.

Coordenação: As tarefas necessitam a coordenação de muitos graus de liberdade?

Adaptabilidade: São utilizados diversos modos de controle na execução das tarefas? É importante alternar entre os diversos modos de controle rápida e suavemente?

Naturalidade: As seguintes características são úteis no controle das tarefas?

- Habilidades sensoriais e motoras pré-adquiridas;
- Sinais manuais existentes;

- Ausência de dispositivos intermediários;
- As tarefas de controle podem ser bem mapeadas em ações manuais (posicionamento e movimentação das mãos).

Quanto maior o número de respostas positivas à estas questões, mais adequada é a utilização de uma interface baseada em gestos para a aplicação. Uma vez que a interface de gestos seja definida como apropriada, é necessário desenvolver uma taxonomia que descreva como os gestos serão utilizado na interface. A taxonomia categoriza possíveis ações manuais e suas interpretações pela aplicação e também ajuda a discriminar entre os diferentes tipos de estilos de entrada de dados, o que permite ao projetista selecionar a interface de gesto mais adequada. Por exemplo, se existem muitos tipos de entrada desejados, então o reconhecimento de gestos simbólicos é mais adequado. Caso apenas alguns poucos tipos sejam necessários, uma interface baseada em gestos naturais pode ser mais vantajosa.

Em seguida, a aplicação é dividida nas suas tarefas primitivas a fim de estabelecer suas relações com as ações manuais. Em muitos casos, o projetista deve realizar uma série de experimentos até encontrar o gesto mais adequado à uma dada tarefa. O passo final do método é a escolha do dispositivo de entrada que combine com as características das tarefas e dos gestos.

Utilizando este método descrito, Sturman e Zeltzer avaliam a utilização de gestos como interface de interação em três aplicações distintas: controle de caminhada de um robô virtual, orientação de um objeto virtual e movimentação de um robô virtual ao longo de um caminho. Em todos os casos foram comparadas a utilização de uma luva de dados (seção 3.1.1) e um painel de controle com botões. A metodologia de *design* prevê que a atividade de controle de caminhada seria melhor realizada utilizando-se a luva de dados, que não haveria diferenças significativas na atividade de orientação do objeto e que a atividade de movimentação por um caminho seria melhor realizada pelo painel com botões. Os resultados dos experimentos verificaram as predições, comprovando a validade da metodologia.

Capítulo 3

Rastreando movimentos humanos

Neste capítulo veremos um pouco mais sobre os dispositivos de rastreamento e detalharemos os componentes de um sistema de visão computacional.

3.1 Dispositivos de rastreamento

Dispositivos de rastreamento acoplados também são conhecidos como dispositivos invasivos ou intrusivos, pois requererem algum tipo de contato físico com o usuário. Aplicam-se a esta categoria dispositivos como: luva de dados, mouse 3D, acelerômetros, etc. Em contra partida, técnicas se visão computacional também são referidas como técnicas ou dispositivos não-invasivos, pois não requerem nenhum tipo de contato físico com o usuário. Veremos as principais vantagens e desvantagens destas duas categorias mais adiante.

3.1.1 Luva de dados

Interfaces que reconhecem muitos gestos normalmente necessitam de informações precisas sobre posicionamento dos membros de interesse (mãos, dedos, etc.). Um dispositivo bastante comum na obtenção de dados provenientes das mãos é a luva de dados, descrita pela primeira vez por Zimmerman *et al.* [ZLB+86]. Este dispositivo é construído com uma série de pequenos cabos de fibra óptica que acompanham o formato das mãos e dos dedos. Ao executar um movimento, os cabos são flexionados, alterando a intensidade de luz transmitida pelo seu interior e fornecendo a informação sobre o ângulo formado entre cada falange dos dedos. Um acelerômetro e um rastreador eletromagnético acoplados à luva fornecem dados sobre a orientação e a posição da mão. Este dispositivo possui exemplos de aplicações que vão desde o controle remoto de sistemas robóticos (Tran *et al.* [TPD+09]), passando por sistemas de aprendizado eletrônico e treinamento em ambientes de realidade virtual (Bednarz *et al.* [BCD09]) e até mesmo em obras de arte multimídia (Cantoni e Garcia [CG07], figura 3.1).

Outra categoria de luva de dados apresenta uma tecnologia mais simplificada, que não fornece informações sobre posicionamento nem orientação. Luvas de dados como a Peregrim [Per10] foram desenvolvidas com foco na interação com jogos de computador que utilizam muitas teclas de atalho para executar comandos. Estas luvas apresentam dois tipos de pontos de contato em regiões específicas da mão que permitem ao usuário executar um comando ao conectar dois contatos de tipos distintos. No caso da Peregrim, os contatos de um tipo estão localizados na palma da mão e na

ponta do polegar, enquanto que contatos de outro tipo estão espalhados ao longo dos demais dedos (figura 3.2). O usuário pode associar atalhos de teclado às diferentes combinações de contatos, agilizando a interação. Jogadores profissionais atestam que são necessárias apenas algumas poucas horas para adquirir prática no uso de tal dispositivo e que este propicia uma interação tão natural quanto o teclado tradicional [Spe10].

Apesar de ser um bom dispositivo para a execução de comandos em jogos táticos, como jogos online massivos de múltiplos jogadores (*massive multiplayer online* - MMO) e de estratégias em tempo real (*real time strategy* - RTS), ele não é capaz de fornecer a mesma quantidade de combinações de comandos que o teclado. Em jogos baseados no paradigma de atirador em primeira pessoa (*first person shooter* - FPS), muitas vezes é necessário executar mais de um comando ao mesmo tempo, o que no teclado é possível ao pressionar múltiplas teclas, enquanto que isso não é possível utilziando a luva de dados.

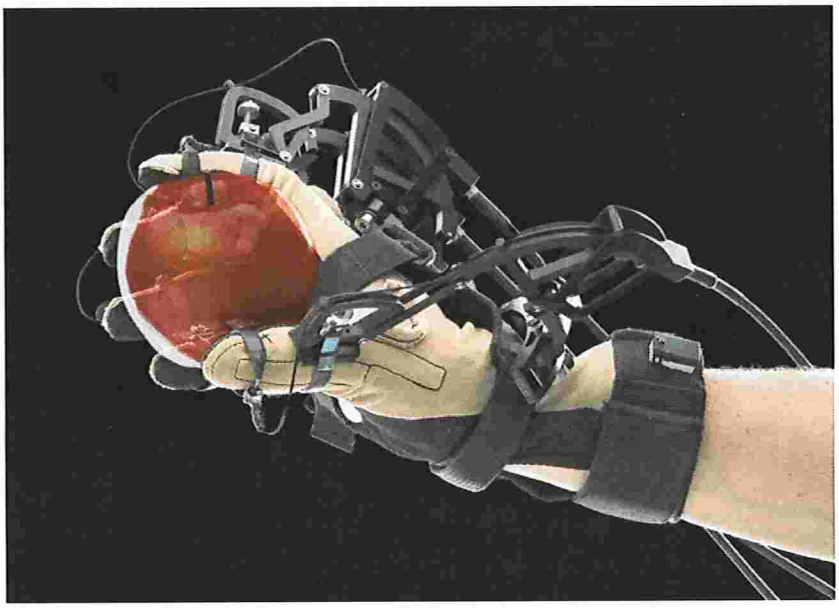

Figura 3.1: Luva de dados sendo utilizada em combinação com um dispositivo háptico na obra Hyperapple, de Cantoni e Garcia [CG07].

3.1.2 Dispositivos inerciais

Dispositivos inerciais possuem um acelerômetro capaz de informar sua orientação relativa no espaço. Um exemplo de tal dispositivo, que se tornou bastante popular, é o controle remoto do videogame *Nintendo Wii*® [Nin10], popularmente denominado *Wiimote*, figura 3.3. Este dispositivo, semelhante a um controle remoto de televisão, possui, além de um acelerômetro, uma base fixa contendo dois emissores infravermelhos e um receptor acoplado ao controle que fornecem a informação quanto ao seu posicionamento. O dispositivo também conta com alguns botões programáveis aos quais podem ser associadas funções específicas como ativar, desativar ou mudar o contexto da interface. Originalmente, o *Wiimote* foi desenvolvido como interface para um videogame, tendo

Figura 3.2: Luva de dados *Peregrine* apresentando, à esquerda, seus três pontos de contato na palma da mão e no polegar e outros pontos ao longo dos demais dedos. À direita são apresentadas algumas conexões entre os dedos e palma da mão.

como principais funções a navegação por menus e a interação com os ambientes virtuais dos jogos. Suas aplicações externas ao ambiente do console Wii variam desde a exploração e manipulação de ambientes virtuais [Cho08, RJO+09] à analise de dados médicos volumétricos [GDM08].

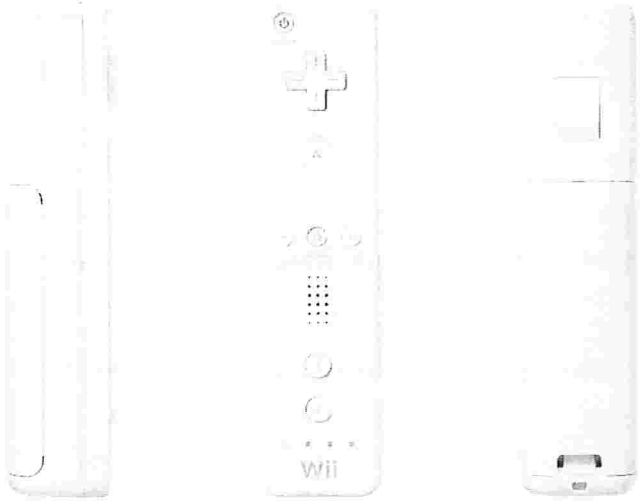

Figura 3.3: Dispositivo de controle remoto do *Nintendo Wii*®.

3.2 Rastreamento baseado em visão computacional

Apesar de precisos, os dispositivos de rastreamento invasivos podem apresentar preços elevados e por si só acabam por limitar o conjunto de gestos que podem ser executados pelo usuário, seja pela necessidade de vestí-los e carregar uma série de cabos durante a interação, ou pela necessidade de segurá-los, o que impede os dedos de moverem-se livremente. A utilização de sistemas de visão computacional corresponde a uma alternativa de rastreamento de custo reduzido que, apesar de fornecer uma precisão menor e sofrer com problemas de oclusão, não requer a utilização de

nenhum equipamento pelo usuário, garantindo um maior conforto durante a interação e permitindo a utilização de dados de cor e textura da cena como parâmetros da interface.

Estimação de pose é um termo bastante utilizado no desenvolvimento de sistemas de rastreamento baseados em visão computacional e significa estimar a pose de uma pessoa (do corpo inteiro ou apenas algumas partes) a partir dos dados provenientes de um conjunto de sensores (câmeras). Quando esta estimação leva em conta a variação da pose com o passar do tempo, o termo análise de movimento humano é empregado.

Uma observação (de uma pose) consiste numa projeção (ou um conjunto de projeções) do mundo real. Desta forma, quando uma pose é analisada via uma observação, muitas informações do mundo real são perdidas, o que ocasiona uma série de dificuldades em sua estimação. Note que a relação entre pose e observação não é unívoca pois, devido à grande variação de formas e aparências existente entre as pessoas e aos diferentes posicionamentos dos sensores, uma mesma pose pode apresentar diferentes observações. O contrário também é válido, devido à perda de informações pelas projeções, diferentes poses podem acabar por gerar observações muito parecidas. Quando apenas um sensor é utilizado, também podem ocorrer ambiguidades devido à perda da informação relativa à profundidade. Outro fator que interfere na determinação da pose é a resolução dos sensores, o que pode fazer com que pequenas variações de pose não sejam percebidas.

Poppe [Pop07] apresenta um resumo sobre os principais sistemas de análise de movimento humano, dividindo-os em dois grupos: abordagens baseada em modelos (ou generativas) ou sem modelos (ou discriminativas).

3.2.1 Soluções baseadas em modelos

As abordagens baseadas em modelos empregam um conhecimento a priori do corpo humano e a estimação da pose é feita em duas etapas: modelagem e estimativa.

Modelagem

A etapa de modelagem consiste na definição de uma função de similaridade e de seus parâmetros, levando em conta os parâmetros dos sensores, os descritores da imagem, um modelo do corpo humano, funções de semelhança e um conjunto de restrições (físicas). Alguns destes parâmetros podem ser fixos e determinados previamente, como o posicionamento dos sensores ou o tamanho real das partes do corpo. Quanto menor o número de parâmetros a serem estimados, mais tratável se torna o problema, porém, maiores se tornam suas limitações.

Modelos cinemáticos A fim de definir uma função de similaridade e seus parâmetros, soluções baseadas em modelos costumam utilizar um modelo do corpo humano, denominado *modelo cinemático*, que pode incluir suas características visuais (musculatura e epiderme) e estrutura cinemática (esqueleto). Tais modelos descrevem o corpo como uma árvore hierárquica composta por segmentos e junções. Cada junção possui um número de graus de liberdade (*degrees of freedom* - DOF), que indica em quantas direções diferentes a junção pode se mover. O conjunto de todas os DOFs representa uma pose.

Modelos cinemáticos podem ser descritos em duas ou três dimensões. Modelos em duas di-

mensões são comumente utilizados em aplicações nas quais o movimento é realizado paralelamente ao plano da imagem, como em situações de análise de caminhada. Modelos tridimensionais costumam definir um número máximo de rotações (ortogonais) para cada junta igual a três. Para cada rotação, no entanto, podem ser aplicadas mais restrições relativas ao modelo como, por exemplo, impedir que partes do corpo atravessem umas as outras.

O número total de DOFs varia de sistema para sistema. Alguns trabalhos que analisam apenas a movimentação dos membros superiores o fazem utilizando apenas 10 DOFs, enquanto que estudos de movimentação do corpo inteiro trabalham com não menos que 50 DOFs. Mesmo com um número limitado de DOFs, o número de poses possíveis é bastante alto. Para reduzir o conjunto de poses analisadas, costuma-se utilizar restrições quanto aos ângulos, velocidade e aceleração máximos e mínimos atingidos por cada junta.

Modelos de formas Além do modelo cinemático, modelos relativos ao formato da pose na imagem também podem ser empregados. Em duas dimensões, comumente são empregados formas retangulares ou trapezoidais para representar os diferentes segmentos do corpo (figura 3.4). Já em representações tridimensionais, os segmentos são representados por superfícies ou formas volumétricas, como esferas e cilindros. Ao invés de representar cada segmento do corpo por uma forma rígida, modelos baseados em superfícies costumam utilizar apenas uma superfície deformável para representar todo o corpo.

Definir as dimensões (altura, largura e comprimento) dos segmentos como fixas pode acarretar numa maior imprecisão das estimativas devido à grande variedade de dimensões entre as pessoas. Uma alternativa é definir estes valores numa etapa de inicialização, no qual a pessoa adota uma determinada pose e seus dados biométricos podem ser obtidos. Esta solução, no entanto, não é aplicável em sistemas de vigilância ou de anotação automática, nas quais muitas vezes o usuário não está ciente de que suas poses estão sendo analisadas.

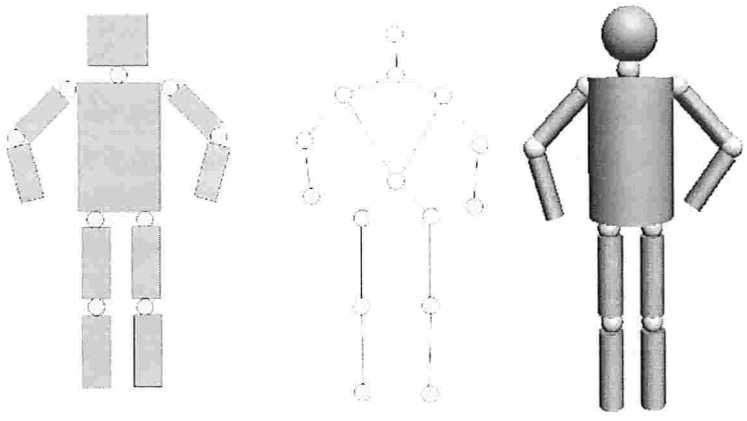

Figura 3.4: Diferentes tipos de modelos cinemáticos. À esquerda, modelo bidimensional. Ao centro, modelo bidimensional básico constituído apenas de segmentos e juntas. À direita, modelo tridimensional.

Descritores da imagem Outro fator que pode ser utilizado como parâmetro da função de similaridade são as características da imagem. A aparência das pessoas numa imagem varia devido à fatores étnicos, roupas e condições de iluminação. Uma forma de se obter dados genéricos (que apresentam pouca variação de pessoa para pessoa) a partir de uma imagem é utilizar seus descritores ao invés da imagem original. Ou seja, para identificar uma pose, não precisamos ter conhecimento sobre todos os seus dados na imagem, apenas sobre os invariantes. Descritores de imagem amplamente utilizados incluem silhuetas, contornos, bordas, mapas de profundidade, movimentação e cor.

Silhuetas Silhuetas e contornos podem ser extraídos de forma robusta de imagens com planos de fundo relativamente estáticos. A extração de silhuetas apresenta pouca sensibilidade quanto à variação de cor e textura das superfícies e fornece dados bastante relevantes para a reconstrução de poses tridimensionais. Seu desempenho, no entanto, está sujeito à presença de artefatos tais como sombras e planos de fundo complexos ou altamente variáveis. Funções de semelhança geralmente avaliam o quão bem as áreas de duas silhuetas se casam.

Bordas Bordas aparecem na imagem em regiões onde as características de cor apresentam grandes variações sendo, até certo ponto, invariantes em diferentes condições de iluminação. Podem ser extraídas de forma robusta e a um baixo custo computacional em imagens com texturas ou planos de fundo não muito complexos, por isso, bordas costumam ser identificadas dentro de um silhueta ou projeção de um modelo. Funções de semelhança costumam levar em conta a distância normalizada entre as bordas sintéticas de um modelo e a borda mais próxima encontrada na imagem.

Apesar de poderem ser extraídas de forma robusta e a um baixo custo computacional, silhuetas e bordas não fornecem dados de profundidade, dificultando a detecção de eventos como a auto-oclusão. Quando múltiplas câmeras são utilizadas, uma reconstrução 3D pode ser feita a partir de silhuetas extraídas em cada ponto de vista. Duas técnicas muito utilizadas são a intersecção volumétrica [BL01] e a abordagem baseada em *voxels* [MTHC03]. Outro modo de se obter informação sobre a profundidade é utilizando a estereometria, descrita em detalhes na seção 5.5.1.

Cor e textura A modelagem baseada nas informações de cor e textura é inspirada no fato da aparência de certas partes do corpo não variarem durante a realização do movimento. A aparência das diferentes partes do corpo pode ser descrita utilizando-se distribuições Gaussianas de cor (seção 5.5.2) ou histogramas. Modelos de cor baseados em cor-de-pele podem funcionar como boas pistas no rastreamento da cabeça e mãos [PBM04].

Movimentação Assumindo que o brilho dos *pixels* pertencentes à uma pessoa não varia, sua movimentação pode ser medida pela diferença de duas imagens consecutivas. O deslocamento dos *pixels* numa imagem é denominado *fluxo óptico*, podendo ser utilizado em combinação com parâmetros tais como borda ou silhueta a fim de atribuir um maior peso aos componente móveis (Sminchisescu e Triggs [ST03]).

Combinando descritores Funções de similaridade que levam em conta a combinação de descritores costumam apresentar uma maior robustez. Informações sobre silhuetas podem ser com-

binadas com bordas [DR05], fluxo óptico [How05] ou cor [CBK03]. Deve-se tomar cuidado, no entanto, quanto ao modo como as informações provenientes dos descritores são combinadas. A simples multiplicação dos dados pode resultar numa superfície de similaridade com muitos picos, o que diminui a eficácia da estimativa.

Considerações quanto aos sensores Apesar de apresentarem um tratamento matemático mais simples, sistemas com um único sensor sofrem com problemas de oclusão e ambiguidades quanto à profundidade. Sminchisescu e Triggs [ST03] estimam que nestes casos cerca de um terço dos DOFs não podem ser observados, principalmente aqueles relacionados a movimentos em profundidade. Tais problemas podem ser aliviados utilizando-se sistemas com múltiplos sensores. A busca por descritores pode então ser realizada de duas formas: identificando e tratando as características separadamente, em cada uma das imagens, e combinando-as num estágio posterior, a fim de solucionar ambiguidades, ou identificar as características nas imagens e combiná-las num conjunto de dados 3D, realizando o tratamento sobre este novo conjunto. Em sistemas com múltiplos sensores, sua calibração corresponde à um requisito muito importante, tendo grande influência na qualidade das estimativas. Em câmeras com projeção perspectiva, as interações são restritas à região na qual ocorre a intersecção dos campos de visão, apresentando resultados melhores próximos às câmeras e resultados piores em áreas mais distantes, nas quais os efeitos de perspectiva são menores.

Considerações quanto ao ambiente Ambientes ao ar livre ou cujo plano de fundo apresente grande variação costumam dificultar a análise do movimento pois dificultam a identificação das pessoas na cena. Condições de iluminação variáveis, tanto em relação à cor quanto à intensidade de luz também restringem as abordagens que podem ser utilizadas, inviabilizando descritores da imagem baseados em cor, textura e movimento. Ambientes internos costumam apresentar cenários mais fáceis para o reconhecimento de poses, pois os componentes do plano de fundo tendem a ser estáticos (sem árvores, nuvens ou grande movimentação de pessoas ou objetos) e possuem condições de iluminação mais controláveis.

Estimativa

A metodologia de estimação de pose está intrinsecamente associada à abordagem de modelagem utilizada. Em abordagens baseadas em modelos, estimar uma pose significa encontrar um conjunto de parâmetros para a função de similaridade que minimiza o erro entre as observações e a projeção do modelo de corpo humano, enquanto que em abordagens baseadas em aprendizado o erro é minimizado entre as observações e a função de projeção definida. Já em abordagens baseadas em exemplos, busca-se a minimização do erro entre as observações e o conjunto de exemplos.

Dentre as metodologias de estimação, podemos identificar os seguintes grupos: soluções de cima para baixo (*top-down*) e soluções de baixo para cima (*botton-up*). Soluções de cima para baixo tentam obter o melhor casamento (com o menor erro) entre uma projeção de pose do corpo humano e a imagem observada. As soluções de baixo para cima identificam as partes do corpo individualmente e as combinam posteriormente de acordo com o modelo do corpo. Algumas soluções mais recentes combinam estas duas abordagens a fim de minimizar suas desvantagens.

Mesmo apresentando vantagens e desvantagens particulares, alguns problemas atingem ambas metodologias. As funções de similaridade utilizadas costumam apresentar mais de um máximo local. Uma alternativa muito utilizada à função de similaridade é a sua conversão em uma função de custo, o que permite a busca por um mínimo local ao invés de um máximo. Outro problema é que, devido à alta dimensionalidade do espaço de busca, é extremamente necessário sempre estabelecer uma estratégia de estimação eficiente. Muitos trabalhos apresentam soluções cujo tempo de análise varia desde alguns minutos a desempenhos em tempo real.

Soluções de cima para baixo Abordagens de cima para baixo (*top-down solutions*) buscam por uma projeção do modelo de corpo humano que melhor se adapte à imagem observada. Tal método também é conhecido como "análise via síntese". Uma vez que buscas por força bruta apresentam um alto custo computacional, a busca é feita, primeiramente, em regiões da superfície de custo próximas à última pose estimada [GD96]. Em seguida, sucessivas poses *a posteriori* são encontradas seguindo-se o gradiente decrescente da superfície até que um mínimo seja atingido. Uma inconveniência da estimação de cima para baixo está na necessidade de se realizar uma inicialização (manual) no primeiro quadro da sequência de imagens, uma vez que cada pose é estimada tendo como base a pose identificada no quadro anterior. Outro problema está no custo computacional de se projetar o modelo de corpo humano em cada uma das poses analisadas e calcular a distância entre a imagem da projeção e a da observação.

Gavrila e Davis [GD96] utilizam uma abordagem de cima para baixo combinada com uma decomposição do espaço de busca. As poses são estimadas utilizando uma estratégia hierárquica de ajuste fino, estimando primeiramente o torso e a cabeça, passando então aos braços e pernas. A predição da pose inicial é baseada na aceleração constante dos ângulos entre as juntas. Uma abordagem de análise por síntese é aplicada de modo discreto, limitando o número de soluções possíveis por junta.

Este tipo de abordagem também costuma sofrer com problemas de (auto-)oclusão. Erros na estimação são propagados pela hierarquia do modelo, ou seja, caso a posição do tronco seja mal estabelecida, isso acarretará erros na estimação da pose dos braços, pernas e cabeça. Uma solução para este problema é apresentada por Drummond e Cipolla [DC01] ao introduzirem limitações entre partes do corpo conectadas pela cadeia cinemática. Isso permite que partes mais baixas na cadeia influenciem partes mais altas. Uma pose passa a ser descrita pelo deslocamento rígido de cada parte do corpo, o que gera um sistema super-parametrizado que pode ser resolvido por meio de uma técnica de mínimos quadrados ponderados.

Soluções de baixo para cima Soluções de baixo para cima (*bottom-up solutions*) são caracterizadas por encontrar partes do corpo individualmente e depois combiná-las num modelo de corpo humano. As partes são normalmente descritas por padrões 2D, que costumam gerar muitos falso-positivos, uma vez que muitas regiões da imagem podem se parecer com uma mão ou uma perna, por exemplo. Outra desvantagem é a necessidade de se construir um detector para cada parte do corpo que será rastreada.

O procedimento de juntar todos os candidatos a partes de corpo no modelo humano costuma

levar em conta restrições físicas, como proximidade das partes. Uma vantagem deste tipo de abordagem está no fato que nenhuma inicialização é necessária. Muitas vezes, abordagens de baixo para cima são utilizadas na inicialização de abordagens de cima para baixo.

Ramanan e Sminchisescu [RS06] treinam modelos que maximizam a verossimilhança para a localização conjunta de partes do corpo, ao invés de utilizar localizadores de partes do corpo individuais. O algoritmo de treinamento utilizado aprende os parâmetros de um campo condicional aleatório (*Conditional Random Field* - CRF) a partir de um pequeno grupo de amostras.

Mori *et al.* [MREM04] realizam primeiramente uma segmentação da imagem utilizando os descritores de contornos, formas e aparência. Os segmentos resultantes são classificados por um localizador de partes do corpo que realiza uma busca pelo torso e meios-membros (braço, ante-braço, etc.), cujos modelos baseados nos descritores foram construídos previamente. A partir desta configuração inicial, são encontradas as demais partes do corpo. Limitações globais incluem a proximidade entre as partes do corpo, comprimentos e larguras relativas e simetria de cor são utilizadas para reduzir o espaço de busca.

Soluções híbridas Combinar as duas abordagens é uma boa opção para contornar suas desvantagens. Trabalhos recentes buscam resolver o problema de indentificação de poses humanas em cenas complexas. Sminshisescu *et al.* [SKM06] realizam o aprendizado de funções de cima para baixo e de baixo para cima em passos alternados. Os processos de baixo para cima são ajustados a partir das amostras dos processos de cima para baixo, que são otimizados para produzir estimativas próximas àquelas previstas pelos processos de baixo para cima. Desta forma, ambos os processos acabam por convergir para um equilíbrio.

Azad *et al.* [AUAD07], utiliza informações sobre a distância das mãos à cabeça, segmentadas a partir dos descritores de cor-de-pele, como parâmetros de entrada num filtro de partículas para obter uma primeira estimativa sobre a pose do usuário acima da cintura. Informações sobre o contorno dos braços e do tronco são adicionados para ajustar o posicionamento dos ombros e cotovelos do modelo de pose.

Rastreamento baseado em hipóteses únicas ou múltiplas A estimação de poses de forma sequencial, de quadro em quadro, é denominado rastreamento. O rastreamento é realizado de forma a garantir a coerência temporal entre as poses com o passar do tempo. Ao supormos que o intervalo de tempo entre os quadros é pequeno, então também podemos supor que a distância entre as configurações do corpo em quadros subsequentes também é pequena. Sendo assim, estas pequenas diferenças de configuração podem ser rastreadas, aproximadamente, de forma linear utilizando, por exemplo, um filtro de Kalman.

Métodos de rastreamento, tradicionalmente, buscavam a manutenção de uma única hipótese ao longo da sequência, o que costumava acarretar em erros cada vez maiores da estimação com o passar do tempo. Casos de ambiguidade, tais como auto-oclusão, aumentam a possibilidade de escolha da pose incorreta e, caso esta seja feita, a estimação de pose pode se perder.

Para contornar estes problemas, podemos utilizar métodos de rastreamento baseados em múltiplas hipóteses tal como Cham e Rehg [CR99]. Eles utilizam um conjunto de filtros de Kalman para pro-

pagar suas múltiplas hipóteses, o que resulta num sistema de rastreamento menos suscetível a erros. Múltiplas hipóteses apresentam bons resultados mesmo em sequências de movimentos dinâmicos, tais como passos de dança, nos quais rastreadores de única hipótese falham.

A utilização de filtros de Kalman é adequada para o rastreamento de movimentos suaves e contínuos. Porém, muitos dos movimentos humanos não podem ser aproximados por movimentos lineares. Soluções de rastreamento baseadas em amostragem, tais como filtros de partículas [BJ98], apresentam bons resultados no rastreamento de movimentos não-lineares. De um modo geral, um conjunto de partículas e uma componente de ruído são propagados no tempo, tendo como base um modelo de dinâmica do corpo. Cada partícula possui um peso associado que é atualizado de acordo com a função de custo. Às configurações com baixo custo são associados pesos maiores. Uma vez que a soma de todos os pesos é igual a um, a pose estimada é obtida pela seleção da partícula com peso máximo.

Apesar de, em teoria, filtros de partículas funcionarem muito bem como ferramentas de rastreamento, a alta dimensionalidade característica do problema de rastreamento de poses do corpo humano exige um grande número de partículas de modo a cobrir todo o espaço de possibilidades. Cada partícula adiciona um custo computacional relativo à sua propagação e à sua avaliação pela função de custo. Para cada partícula, o modelo de corpo precisa ser projetado e analisado junto aos descritores da imagem. Outro problema é o fato das partículas tenderem a se agrupar em pequenas regiões, problema conhecido como empobrecimento [KF00], o que acarreta num número reduzido de partículas efetivas.

Atualmente, são utilizadas duas principais soluções para tornar estes problemas mais tratáveis. A primeira inclui a utilização de modelos de movimento capazes de guiar as partículas de modo mais eficiente e a redução de dimensionalidade do espaço de descrição do movimento, o que infere na diminuição do número de partículas. A segunda solução ocupa-se em espalhar as partículas mais eficientemente em regiões propícias à ocorrência de mínimos locais [ST03].

Estimando poses tridimensionais a partir de dados bidimensionais Quando a pose de uma pessoa é vista de mais de um ponto de vista distinto, é possível estimar a pose tridimensional a partir das observações bidimensionais. Liebowitz e Carlsson [LC03] realizam a reconstrução 3D a partir de um conjunto de pontos correspondentes a partir de múltiplos pontos de vista e do conhecimento prévio das dimensões das partes do corpo. Uma reconstrução geométrica linear é realizada para obter as poses de toda uma sequência de uma só vez.

Manders *et al.* [MFY+08] utilizam um sistema de câmeras em estéreo que, a partir dos descritores de cor e textura da cena, calcula o seu mapa de disparidade. A partir deste mapa, é possível obter a posição 3D de qualquer *pixel* da imagem. Utilizando descritores de cor referentes à cor-de-pele, é possível estimar a posição da cabeça, mãos e braços do usuário na cena. Técnicas como esta, que calculam a posição 3D de todos os *pixels* da cena são denominadas técnicas de "estéreo denso".

3.2.2 Soluções sem modelos

Se nenhum modelo explícito do corpo humano é definido, é necessário estabelecer uma relação direta entre observação e pose. Duas classes principais de metodologias de estimação podem ser identificadas: baseadas em aprendizado e baseadas em exemplos. Ambas as abordagens utilizam um conjunto de dados que contém informações sobre as poses e suas observações. Uma vez que as variações com relação à configuração do corpo, suas dimensões, pontos de vista e aparência são modeladas de forma implícita no conjunto de dados, tal conjunto deve ser construído de forma a generalizar os parâmetros invariantes e destacar os variantes da melhor forma possível. O conjunto de dados deve, então, conter uma grande quantidade de mapeamentos não-lineares entre o espaço de observações e o de poses, ou seja, o espaço de poses de ser densamente exemplificado por este conjunto. Abordagens livres de modelos não sofrem com problemas de (re)inicialização e muitas vezes são utilizados na inicialização de soluções de estimação de pose baseadas em modelos.

Soluções baseadas em aprendizado

Em soluções baseadas em aprendizado uma função entre o espaço de poses e o espaço de imagens é aprendido a partir de um conjunto de dados.

Rosales e Sclaroff [RS00] observam que o mapeamento inverso do espaço de imagens para o espaço de poses não pode ser modelado por uma única função. Eles, então, agrupam o espaço de poses 2D e aprendem funções especializadas dos descritores da imagem para o espaço de poses de cada grupo, utilizando redes neurais como funções de mapeamento. Em [RSAS01], seu trabalho é estendido a fim de permitir múltiplos pontos de vista. A pose é então estimada para cada ponto de vista e, posteriormente, as hipóteses são combinadas num conjunto de hipóteses 3D.

Sminchisescu *et al.* [SKLM05] modelam a característica multivalorada do mapeamento com uma mistura de modelos especialistas. Cada especialista aprende uma distribuição de estado condicional a partir do conjunto de dados, que consiste em amostras de representação de poses e modelos renderizados do corpo humano. Modelos de forma e descritores de aparências locais são utilizados como descritores de imagem. As amostras de dados envolvem atividades humanas tais como andar, correr e realizar mímicas. Demonstrações em cenas complexas vistas a partir de um único ponto de vista apresentam resultados convincentes e testes realizados com dados artificiais mostram que a estratégia proposta é melhor que abordagens baseadas em métodos como 'o vizinho mais próximo' e de regressão.

Soluções baseadas em exemplos

Soluções baseadas em exemplos, ao invés de construírem uma função de mapeamento, estabelecem um banco de dados de exemplos juntamente com os seus descritores de pose. Para uma dada imagem, é feita uma busca por similaridade e as poses candidatas são interpoladas para se obter uma melhor pose estimada. Uma desvantagem deste tipo de solução é a grande quantidade de espaço requerida para conter o banco de dados.

Mori e Malik [MM06] utilizam dados de contornos internos e externos e bordas, definidas por descritores de formas, para identificar nove juntas ao redor do corpo. Na fase de estimação, os

modelos armazenados são deformado de modo a melhor condizerem à imagem observada. Nesta deformação, a posição das juntas também é modificada e a pose estimada é obtida forçando-se a consistência das distâncias entre as partes do corpo na imagem 2D.

Em [OURH05], as poses do banco de dados são renderizadas a partir de vários pontos de vista, o que torna a abordagem relativamente invariante neste aspecto. Para cada imagem monocular, o ponto de vista é estimado utilizando-se um discriminante linear e, em seguida, a pose é obtida realizando-se uma busca por vizinhos mais próximos.

Capítulo 4

Algoritmos de reconhecimento de gestos

Reconhecimento de gestos é um exemplo ideal de pesquisa multidisciplinar. As diferentes ferramentas utilizadas no reconhecimento utilizam teorias baseadas em modelagem estatística, visão computacional, reconhecimento de padrões, processamento de imagens, etc. Muitos dos problemas são abordados com base em modelagem estatística, como análise de componentes principais (*principal component analysis* - PCA) [Lee06, KCX06], HMM [KAA05, Lee06, SPHB08], filtros de Kalman [KAA05, OP07] e filtros de partículas [STW07, OP07]. Máquinas de estado finitas (MEF) também são eficientemente empregadas na modelagem de gestos humanos [BW97, HTH00].

Conforme vimos no capítulo 3, técnicas de visão computacional e reconhecimento de padrões estão associadas à extração de características, detecção de objetos, agrupamento e classificação de componentes da imagem. Técnicas de processamento de imagens, como análise e detecção de formas, cor, textura, movimento, fluxo óptico, segmentação e identificação de contornos e bordas também representam boas ferramentas de identificação e rastreamento dos componentes da cena.

Enquanto que o reconhecimento de gestos (ou poses) estáticos pode ser realizado utilizando-se modelos de correspondência, técnicas tradicionais de reconhecimento de padrões e redes neurais, o problema de reconhecimento de gestos dinâmicos exige a utilização de técnicas mais complexas, como modelos de compressão e de deformação dinâmica do tempo, tais como modelos escondidos de Markov (*hidden Markov models* - HMM) e redes neurais de atraso de tempo (*time-delay neural networks* - TDNN). A seguir, veremos alguns detalhes sobre estas técnicas mais utilizadas.

Em seu trabalho, Mitra e Acharya*et al.* [MA07] apresentam uma revisão sobre as principais abordagens utilizadas para reconhecimento de gestos realizados com as mãos, braços, cabeça, corpo inteiro e face. Neste trabalho, no entanto, apresentaremos apenas os métodos mais genéricos, que podem ser aplicados à maioria dos problemas de reconhecimento de gestos. Não abordaremos problemas específicos de reconhecimento de poses de corpo inteiro, nem de gestos exclusivamente manuais ou que envolvam reconhecimento de face.

4.1 Modelos escondidos de Markov

Um processo dependente do tempo demonstra uma propriedade de Markov se a densidade de probabilidade condicional do evento atual, dados todos os eventos ocorridos anteriormente, depende apenas do j-ésimo evento mais recente. Se o evento atual depende exclusivamente do último evento mais recente, então o processo pode ser denominado como sendo um processo de Markov de primeira

ordem. Esta é uma suposição bastante útil a ser feita quando consideramos a posição e orientação das partes do corpo com o passar do tempo.

O HMM é um processo estocástico duplo, governado por:

- Uma cadeia de Markov com um conjunto finito de estados;
- Um conjunto de funções aleatórias, cada uma associada à um estado.

Em instantes de tempo discretos, o processo encontra-se em apenas um de seus possíveis estados e produz apenas um símbolo observável de acordo com a função aleatória relativa ao estado. Cada transição entre estados possui um par de probabilidades associadas:

1. probabilidade de transição: que indica a probabilidade de ocorrer a transição;
2. probabilidade de saída: que define a probabilidade condicional de se emitir um símbolo de um alfabeto finito quando no estado.

HMMs têm se mostrado bastante eficientes na modelagem de informações espaço-temporais. O modelo é considerado "escondido" pois a única informação que pode ser vista corresponde à uma sequência de observações. Matematicamente, um HMM pode ser expresso como $\lambda = (A, B, \Pi)$ e descrito por:

- um conjunto com N estados $S = \{s_1, \ldots, s_N\}$;
- um conjunto de sequências de observações $O = \{O_1, \ldots, O_T\}$;
- um conjunto finito e discreto de símbolos observáveis $\{v_1, \ldots, v_k\}$;
- uma matriz de transição de estados $A = \{a_{ij}\}$, na qual a_{ij} corresponde à probabilidade de ocorrer uma transição do estado i, num instante t, para o estado j no instante $t + 1$;
- uma matriz de probabilidade de emissão de símbolos $B = \{b_{jk}\}$, na qual b_{jk} indica a probabilidade de se emitir o símbolo v_k no estado s_j;
- uma distribuição de probabilidades iniciais para os estados $\Pi = \{\pi_j\}$.

Uma topologia generalizada de um HMM é uma estrutura completamente ligada conhecida como modelo *ergótico*, no qual qualquer estado pode ser atingido a partir de um dado estado. Quando aplicado ao reconhecimento de gestos dinâmicos, o índice de um estado muda apenas "da esquerda para a direita" com o passar do tempo, conforme pode ser visto na figura 4.1.

A estrutura global de um HMM é constituída de um conjunto de modelos $(\lambda_1, \ldots, \lambda_M)$, um modelo λ_m para cada um dos M gestos a serem reconhecidos, permitindo uma fácil inserção ou remoção de modelos. Para a construção de cada modelo, é necessário preencher cada elemento das matrizes e vetores de probabilidade que compõem o HMM. Tal procedimento é denominado treinamento e pode ser realizado utilizando-se o algoritmo de Baum-Welch [Rab89].

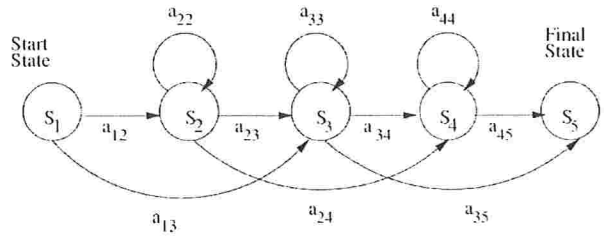

Figura 4.1: HMM com cinco estados seguindo o esquema de transição temporal da esquerda para a direita, retirado de [MA07]

Dada uma sequência de observações, podemos determinar qual a probabilidade de um dado modelo gerar a sequência (algoritmo Forward-Backward) ou mesmo qual a sequência de estados de um modelo que tem a maior probabilidade de gerar a sequência de símbolos observados (algoritmo de Viterbi). Rabiner apresenta uma descrição detalhada destes algoritmos e alguns exemplos aplicados ao reconhecimento de fala em seu trabalho [Rab89].

Um dos primeiros trabalhos a utilizar HMMs para realizar o reconhecimento de gestos foi feito por Yamato *et al* [YOI92]. Neste trabalho, um conjunto de HMMs discreto e uma sequência de vetores quantizados são utilizados para reconhecer seis movimentos característicos de uma partida de tênis. Antes de aplicar o HMM, as sequências de imagens passam por uma série de pré-processamentos tais como filtros para redução de ruídos, remoção de fundo e binarização das imagens, a fim de identificar as regiões que podem conter objetos de interesse. Tais regiões representam, de forma grosseira, a pose da pessoa e são posteriormente projetadas numa base quantizada de vetores, de forma que a sequência de imagens se torne uma sequência de vetores. Estes vetores compõem o conjunto de informações tratadas pelo HMM.

Keskin *et al.* [KAA05] utilizam HMMs para realizar o reconhecimento de oito gestos descritos em três dimensões realizados apenas com as mãos. Ele utiliza um par de câmeras em estéreo para capturar o posicionamento 3D das mãos e utiliza filtros de Kalman para rastrear as posições. Para utilizar a interface de gestos, o usuário deve vestir luvas, cuja cor é utilizada na identificação das mãos na cena. Assim como Yamamoto, Keskin utiliza valores quantizados do posicionamento das mãos para gerar a sequência que será interpretada pelo HMM.

Lee *et al.* [Lee06] aplica o reconhecimento de gestos no problema de interface homem-robô (IHR). As poses instantâneas do usuário são reconhecidas comparando-se um modelo 3D hierárquico de corpo inteiro, com 40 DOFs, à um mapa de profundidade da cena. As poses são agrupadas em regiões do espaço de características de forma que poses muito parecidas sejam representadas por uma única distribuição gaussiana. Um gesto é representado por uma trajetória através destes agrupamentos.

Schlömer *et al.* [SPHB08] identificam cinco gestos tridimensionais realizados com o dispositivo Wiimote, descrito na seção 3.1.2. A orientação do dispositivo, fornecida pelo acelerômetro, é quantizada em 14 posições de acordo com a figura 4.2. Sequências de observações, correspondentes aos gestos realizados, são fornecidas aos HMMs para identificação.

quantizada em 14 posições de acordo com a figura 4.2. Sequências de observações, correspondentes aos gestos realizados, são fornecidas aos HMMs para identificação.

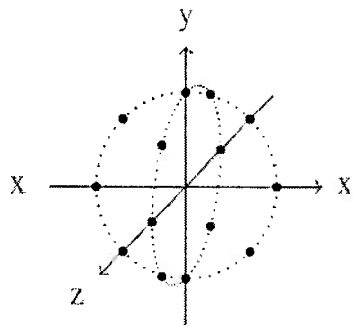

Figura 4.2: Quantização das orientações do Wiimote, retirado de [SPHB08].

4.2 Filtro de partículas

A teoria de filtros de partículas [AMGC02] pode ser aplicada tanto no rastreamento (seção 3.2.1), como no reconhecimento efetivo de gestos. Filtros de partículas são bastante eficientes na estimação do estado de sistemas dinâmicos cujas informações provêm de um conjunto de sensores. A idéia principal é representar as densidades de probabilidade por um conjunto de amostras, o que permite a estimação em tempo-real de sistemas não-lineares e não-gaussianos. Esta técnica foi originalmente desenvolvida para rastrear objetos em cenas complexas [BJ98, IB98a]. O estado de um objeto rastreado no instante t é descrito por um vetor X_t. Um vetor Y_t contém todas as amostras de observações $\{y_1, \ldots, y_t\}$. A densidade posterior $P(X_t|Y_t)$ e a densidade de observação $P(Y_t|X_t)$ são comumente não-gaussianas.

De um modo geral, filtros de partículas são uma variante do filtro de Bayes baseados em amostras. A idéia é aproximar a distribuição de densidade de probabilidade por um conjunto de amostras ponderadas $S_t = \{\langle x_t^{(i)}, w_t^{(i)} \rangle \ | i = 1, \ldots, N_p\}$. Aqui, cada amostra $x_t^{(i)}$ representa um estado hipotético do objeto e $w_t^{(i)}$ representa à correspondente probabilidade amostral discreta da amostra $x_t^{(i)}$, tal que $\sum_{i=1}^{N_p} w_t^{(i)} = 1$.

De fato, o filtro de partículas, em sua forma básica, consiste num filtro de Bayes recursivo de acordo com o processo de amostragem, também denominado amostragem de importância sequencial com reamostragem (*sequential importance sampling with resampling* - SISR) [DDFG01]. A evolução iterativa do conjunto de amostras é descrito pela propagação de cada amostra de acordo com um modelo do sistema. Cada elemento do conjunto é ponderado de acordo com as observações e N_p amostras são retiradas, com reposição, escolhendo uma amostra que tenha probabilidade posterior $w_t^{(i)} = P(y_t|X_t = x_t^{(i)})$. Em cada passo da iteração, o estado médio de um objeto é estimado por $E(S_t) = \sum_{i=1}^{N_p} w_t^{(i)} x_t^{(i)}$.

Uma vez que este tipo de filtro modela a incerteza (por meio das densidades de probabilidades

realizado por Black e Jepson [BJ98], que descrevem uma estratégia de reconhecimento incremental que constitui uma extensão do algoritmo "Condensation", proposto por Isard e Blake [IB98b]. Os gestos são modelados como trajetórias temporais de parâmetros estimados com o passar do tempo (velocidades). O algoritmo "Condensation" realiza o casamento incremental dos modelos de gestos com os dados de entrada. O método é demonstrado via um exemplo de quadro branco aumentado, no qual o usuário realiza gestos 2D simples (rabiscos) sobre o quadro para imprimir, salvar, ou limpar o seu conteúdo.

Apesar de serem bastante eficientes no rastreamento de objetos em cenas complexas, a utilização do filtro de partículas conta com algumas desvantagens, como o grande número de partículas necessárias e a necessidade de se escolher uma função de espalhamento das partículas que modele bem o comportamento do sistema. Para resolver estes problemas, Shan *et al.* [STW07] incorporam uma otimização por *mean-shift* à teoria de filtro de partículas, melhorando consideravelmente a eficiência de amostragem. O trabalho é conduzido de forma a desenvolver uma interface baseada em gestos manuais para o controle de uma cadeira de rodas robótica. O rastreamento das mãos e o reconhecimento dos gestos é realizado em tempo-real, utilizando cerca de 85% menos partículas que o algoritmo de filtro de partículas tradicional.

4.3 Máquina de estados finitos

Em soluções baseadas em máquinas de estados finitos, um gesto pode ser modelado como uma sequência ordenada de estados numa configuração espaço-temporal [BW97]. O número de estados numa MEF varia de aplicação para aplicação. O gesto é reconhecido como um protótipo de trajetória obtida a partir de uma sequência contínua e não segmentada de dados provenientes dos sensores, que descrevem, por sua vez, um aglomerado de trajetórias. As trajetórias de um gesto são representadas por um conjunto de pontos.

Normalmente, o treinamento do modelo é feito *offline*, utilizando o maior número de exemplos possíveis para cada gesto para que seja feita a inicialização dos parâmetros dos estados. O reconhecimento dos gestos é feito *online*, utilizando o MEF treinado. Quando novos dados (vetores de características, como trajetórias) são fornecidos ao identificador de gestos, baseando-se nestas novas informações, este decide se permanece no estado atual ou pula para o próximo. Uma vez que um dos estados finais é alcançado, dizemos que um gesto foi reconhecido.

A representação baseada em estados pode ser expandida de forma à acomodar diferentes gestos, ou mesmo diferentes fases de um mesmo gesto. A atribuição a um estado é determinada pelo quão bem o modelo do estado representa a observação atual. Se mais de um modelo atinge o estado final ao mesmo tempo, pode-se utilizar um critério de desempate para decidir qual o gesto mais provável de ter sido realizado.

Bobick e Wilson [BW97] utilizam máquinas de estado para reconhecer gestos 2D realizados com o *mouse* e gestos 3D realizados com as mãos, capturados por um dispositivo de posicionamento e orientação eletromagnético. Hong *et al.* [HTH00] constroem uma máquina de estados que utiliza os dados sobre o posicionamento 2D das mãos e cabeça para descrever os estados. Para ilustrar a utilização do sistema, foi construída uma aplicação do tipo "Simon diz", na qual o sistema informa

ao usuário qual o gesto que deve ser feito e o usuário tenta realizá-lo. Okkonen *et al.* [MVMJ07] utilizam MEFs para identificar os gestos realizados por uma única mão de acordo com o seu formato.

Capítulo 5

Reconhecimento de gestos 3D para navegação e interação

Conforme vimos na seção 2.3.3, interfaces puramente baseadas em gestos são apropriadas para sistemas que requerem um alto grau de liberdade de interação e utilizam diversos modos de interação, alternando rapidamente entre eles. Outras características que este sistema pode ter são: tarefas de controle bem mapeadas em ações manuais e ausência de dispositivos intermediários.

Um problema que se encaixa nesta descrição é a navegação e manipulação de objetos em ambientes virtuais. Propomos, então, um jogo de quebra-cabeças virtual em três dimensões, constituído por um cubo repartido em 27 peças de mesmo volume como aplicação para a nossa interface baseada em gestos. O objetivo do jogo é navegar pelo ambiente, analisando as peças e montando-as de forma que todas as faces do cubo sejam reconstruídas corretamente.

Trabalhos realizados até o momento para solucionar problemas de navegação e interação utilizam apenas um tipo de gesto (simbólico, natural e outros) ou apenas uma única técnica (filtro de partículas, HMM, MEF e outros) para realizar o reconhecimento dos gestos e construir a interface. Neste trabalho, visamos a construção de uma interface baseada em gestos que utiliza tanto gestos naturais quanto gestos simbólicos, uma vez que gestos naturais se mostraram bastante adequados como interfaces de navegação [CG07, MFY+08] e gestos simbólicos constituem boas ferramentas para a construção de interfaces baseadas em comandos [Kim99,SSSJ05]. A interface de gestos para esta aplicação deve satisfazer os seguintes requisitos de interação:

1. Navegação: movimentação do usuário em três dimensões pelo ambiente virtual;
2. Manipulação: movimentação e rotação de objetos em três dimensões.

Outros requisitos importantes são:

3. Baseado em gestos manuais.
4. Desempenho em tempo real;

Para satisfazer os requisitos de 1 a 3, propomos uma interface na qual as tarefas de movimentação (do usuário e dos objetos) são definidas pelos movimentos da mão direita, enquanto que a seleção e rotação de objetos é definida pelos da mão esquerda. O requisito de tempo real (4) é alcançado realizando-se pequenas otimizações de implementação nos algoritmos utilizados, descritos ao longo das seções 5.4 e 5.5.

5.1 Intenção de interação e posição de descanso

Seguindo o roteiro de desenvolvimento de interfaces proposto por Sturman e Zeltzer (seção 2.3.3), estabelecemos uma pose que indica a intenção de interação pelo usuário. Esta pose é definida no momento de inicialização do sistema de rastreamento, no qual o usuário deve se posicionar de frente para o sistema de captura, com as mãos ao lado do tronco, próximas ao peito, com as palmas da mãos voltadas para a frente. Após a inicialização, a reprodução desta pose indicará uma intenção de interação pelo usuário (figura 5.1).

Outro ponto importante indicado pelos autores é o estabelecimento de uma situação de descanso, na qual nenhuma interação ocorre. Esta posição de descanso é definida de modo que o usuário possa estabelecer momentos de descanso durante a interação, evitando seu cansaço num curto período de utilização do sistema. Para tal, definimos a posição de descanso como sendo aquela na qual o usuário posiciona as mãos relaxadas ao lado do corpo (figura 5.1).

Tendo definidas estas duas importantes características da interface, continuamos com a descrição das demais atividades de interação.

Figura 5.1: Poses associadas à intenção de interação e de descanso.

5.2 Movimentação de usuários e objetos

Conforme descrito anteriormente, a movimentação do usuário e dos objetos é regida pela movimentação da mão direita. Uma vez estabelecida a intenção de interação, mover a mão à frente a partir desta posição inicial ocasiona um movimento para a frente no ambiente virtual. Mover a mão para o lado ocasiona uma rotação para o mesmo lado no ambiente virtual. Mover a mão para cima, faz com que ocorra uma rotação para cima e, mover a mão para baixo, com que ocorra uma rotação para baixo. A combinação destes gestos também é possível, ou seja, mover a mão para frente e para o lado ocasiona um movimento para a frente à direita.

A velocidade com que o movimento no mundo virtual é realizado depende da distância da mão à posição inicial. Posicionar a mão ligeiramente à frente faz com que o movimento à frente ocorra lentamente, enquanto que esticar completamente a mão, faz com que o movimento se torne mais rápido.

Caso um objeto esteja selecionado, a movimentação deste ocorre juntamente com a movimentação do usuário, como se o usuário "carregasse" o objeto.

Um resumo da relação entre ações e interações realizadas pela mão direita é apresentada na tabela 5.1. Um exemplo de cada uma das poses é apresentado na figura 5.2.

Mão Direita	
Ação	**Interação**
Mover para frente	Move para frente
Mover para cima	Olhar para cima
Mover para baixo	Olhar para baixo
Mover para direita	Olhar para direita
Mover para esquerda	Olhar para esquerda

Tabela 5.1: Mapa de interações para a mão direita.

Figura 5.2: Associação entre poses e interações para a mão direita.

5.3 Seleção e rotação de objetos

Ao navegar pelo ambiente virtual, eventualmente, o usuário irá se deparar com um ou mais objetos (peças do cubo). Caso esteja suficientemente próximo do objeto e este se encontre próximo ao centro da imagem, o objeto será destacado dos demais, tendo sua aparência alterada, adquirindo um tom avermelhado, indicando que este pode ser selecionado. Para selecionar um objeto em destaque, o usuário deve levar sua mão esquerda à posição inicial (figura 5.3) a fim de indicar sua intenção de interação e, então, esticar sua mão esquerda totalmente à frente. O objeto em destaque será selecionado e passará a se mover junto com o usuário. A desseleção de um objeto é feita repetindo-se o movimento de seleção, isto é, esticando-se a mão esquerda totalmente à frente.

Para rotacionar um objeto, é necessário que este esteja selecionado. Novamente, o usuário deve levar a mão à posição inicial e, ao movê-la para a esquerda, o objeto realizará uma rotação no

sentido horário sobre o eixo Y (para cima). Mover a mão para a direita faz com que o objeto seja rotacionado no sentido oposto. Mover a mão para cima rotaciona o objeto no sentido horário sobre o eixo X (para a direita), enquanto que um movimento para baixo o rotaciona no sentido anti-horário sobre o mesmo eixo.

A posição de descanso da mão esquerda, na qual nenhuma interação ocorre, é definida de forma semelhante à da mão direita, ao lado do corpo.

Um resumo da relação entre ações e interações realizadas pela mão esquerda é apresentada na tabela 5.2. Um exemplo de cada uma das poses é apresentado na figura 5.3.

Mão Esquerda

Ação	Interação
Mover para frente	Seleciona / Desseleciona objetos
Mover para cima	Rotaciona no sentido horário sobre o eixo X
Mover para baixo	Rotaciona no sentido anti-horário sobre o eixo X
Mover para direita	Rotaciona no sentido anti-horário sobre o eixo Y
Mover para esquerda	Rotaciona no sentido horário sobre o eixo Y

Tabela 5.2: Mapa de interações para a mão esquerda.

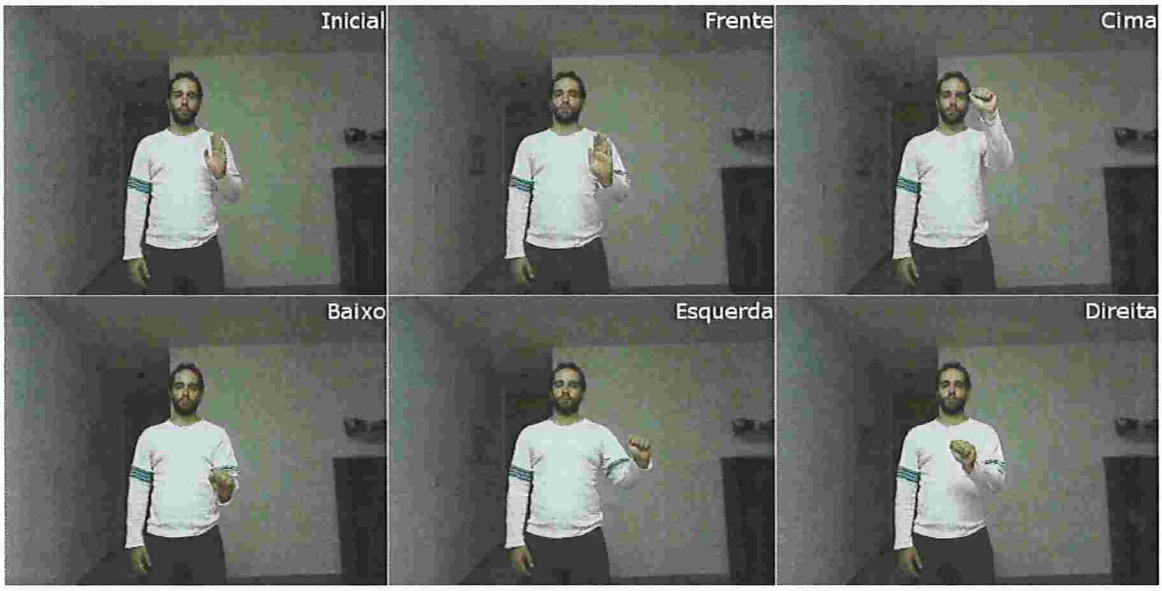

Figura 5.3: Associação entre poses e interações para a mão esquerda.

Definidas as atividades de interação, correspondentes à etapa de interpretação da nossa interface baseada em gestos, falaremos agora sobre seu sistema de reconhecimento e, posteriormente, sobre seu sistema de rastreamento.

5.4 Sistema de reconhecimento de gestos

O sistema de reconhecimento de gestos é constituído por duas partes independentes: o sistema da mão direita, baseado em gestos naturais, e o da mão esquerda, baseado na transição entre poses descritas numa máquina de estados.

5.4.1 Subinterface de gestos naturais

Gestos simbólicos já foram utilizados na construção de interfaces de navegação [YOI92]. Esta abordagem, no entanto, limita o controle de navegação, pois um vocabulário de gestos simbólicos deve ser limitado e preferivelmente não muito extenso. Desta forma, definir alguns poucos gestos como "mover à frente", "mover para a direita" e "mover para a esquerda", pode constituir uma solução de navegação válida, mas restringe bastante os movimentos do usuário. Gestos naturais já foram utilizados com sucesso na navegação em meio a ambiente virtuais [CG07] [MFY+08] de forma a permitir uma navegação mais livre e controlada. Mover a mão para a frente, para cima ou para o lado a fim de controlar sua movimentação pelo ambiente virtual pode ser considerada uma proposta de interface de gestos naturais, pois a movimentação da mão é diretamente mapeada para a movimentação pelo mundo virtual. O mapeamento é feito gerando-se vetores de movimentação que partem da posição inicial até a posição da mão. A direção e sentido destes vetores indicam a direção e sentido do movimento pelo mundo virtual, enquanto que sua norma define a velocidade.

Para cessar a movimentação, o usuário deve retornar a mão à posição inicial. Quando o usuário deseja permanecer parado, para evitar problemas de precisão, estabelecemos uma região de tolerância ao redor da posição inicial. É necessário ultrapassar o limite desta região para que alguma movimentação tenha início.

5.4.2 Subinterface de gestos simbólicos

Gestos simbólicos funcionam como uma boa interface para a execução de comandos discretos [Kim99] [SSSJ05], como rotacionar 90^o para a esquerda ou para a direita. Portanto, para controlar a manipulação dos objetos, estabelecemos uma sub-interface baseada em gestos simbólicos representados por transições entre poses estáticas descritas numa máquina de estados finita. Uma vantagem na utilização de máquinas de estado ao invés de outras técnicas (como filtros de partículas ou HMMs) é a não necessidade de treinamento ou aprendizado. O modelo de poses é de forma relativa ao sistema de coordenadas, desta forma, ele é automaticamente adaptado de usuário para usuário. A definição das poses pode ser feita manualmente, estabelecendo a posição das mãos dentro do espaço de interação, ou tendo como base as poses reais de um usuário.

Máquinas de estados

Uma máquina de estados finita (MEF), ou simplesmente máquina de estados, é um modelo comportamental abstrato composto por um conjunto finito de estados, um conjunto de transições e ações associadas a estes estados. De acordo com Hopcroft *et al.* [HUM02], uma máquina de estados finita, como as utilizadas neste trabalho, podem ser definidas formalmente por uma tupla de cinco elementos $S = \{Q, \Sigma, \delta, q_0, F\}$, sendo que:

Q é um conjunto de estados finito;

Σ é um conjunto finito de símbolos de entrada;

δ é uma função de transição que toma como parâmetros de entrada um símbolo de Σ e um estado de Q, retornando outro estado de Q;

$\mathbf{q_0}$ é o estado inicial e

F é um conjunto de estados finais.

MEFs já foram utilizadas anteriormente por Bobick e Wilson [BW97], Hong *et.al* [HTH00] e, mais recentemente, por Okkonen *et.al* [MVMJ07] para realizar o reconhecimento de gestos em duas dimensões. Neste trabalho, propomos a construção de uma MEF para realizar o reconhecimento de gestos 3D da seguinte forma: construímos um conjunto Σ com r símbolos e dividimos o espaço 3D em r regiões de forma que a cada região seja atribuído um único símbolo. Um símbolo é emitido se, e somente se, o objeto associado à MEF (mão esquerda) permanece numa mesma região dentro de um intervalo de tempo pré-definido. Caso o objeto atravesse rapidamente uma região, o símbolo referente a ela não será emitido.

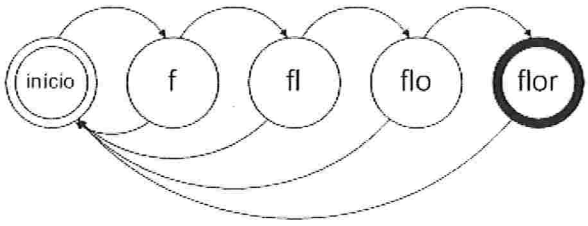

Figura 5.4: Estados e suas transições para o reconhecimento da palavra *flor*. O conjunto de símbolos é determinado pela letras $\{f, l, o, r\}$ e os estados são as subsequências apresentadas.

Os estados que compõem o conjunto Q representam subsequências válidas de símbolos, conforme ilustra a figura 5.4, na qual os símbolos são representados pelas letras da palavra *flor*. Cada máquina de estados possui apenas uma única sequência de estados válida, com um único estado inicial e um único estado final. O comportamento das funções de transição ocorre da seguinte forma: ao receber um símbolo emitido, se este for igual ao último símbolo da subsequência do estado atual, nenhuma mudança de estado ocorre. Caso o símbolo corresponda ao final da subsequência do próximo estado, ocorre uma mudança de estado e, se o próximo estado corresponder ao estado final, a ação associada àquela MEF é executada e todas as MEFs são reiniciadas. Caso o símbolo emitido não corresponda nem ao final da subsequência atual nem ao do próximo estado, a MEF retorna ao seu estado inicial. O comportamento das funções δ é ilustrado de forma resumida na figura 5.5.

No nosso caso, construímos cinco máquinas de estados, cada uma com apenas dois estados, o inicial e o final. Seguindo as propostas de desenvolvimento de interface de Baudel e Beaudouin-Lafon (seção 2.3.2), definimos o estado inicial de todas as máquinas como sendo o mesmo e igual à

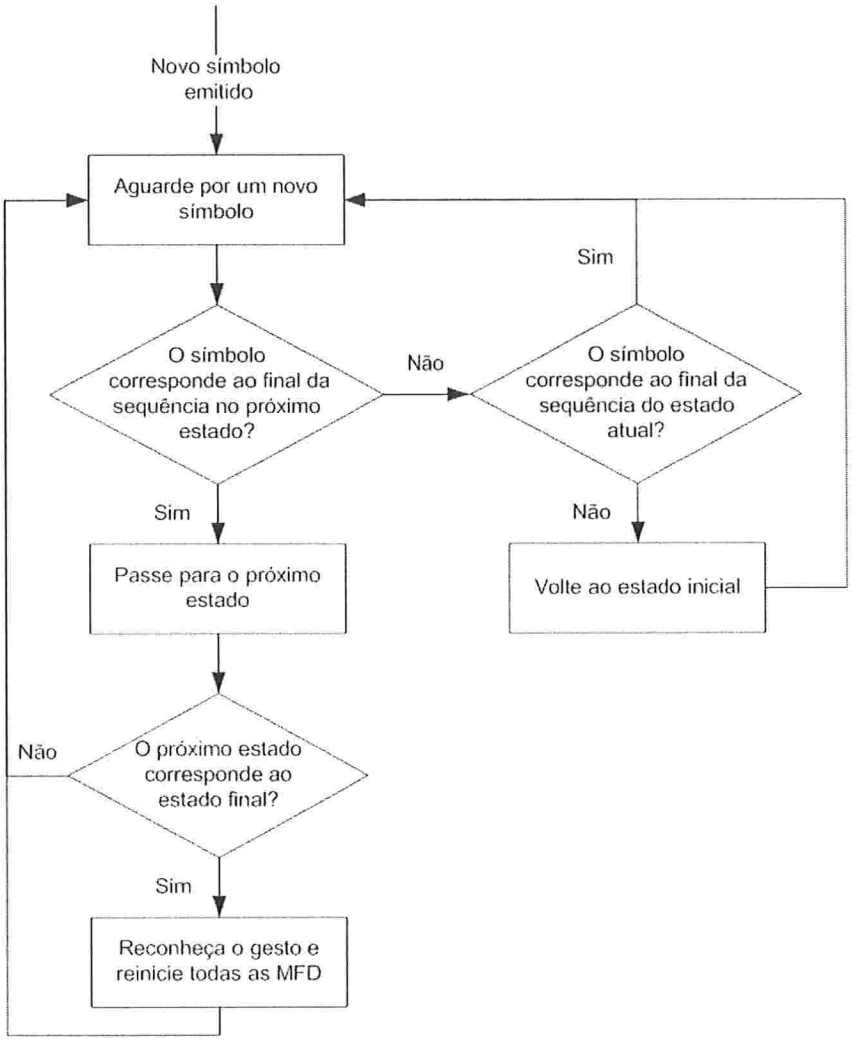

Figura 5.5: Comportamento da função δ: diagrama de decisão.

posição inicial (figura 5.3). Os estados finais são definidos pelo deslocamento da mão a partir deste ponto, conforme discutido anteriormente. Uma vez que um gesto é reconhecido, todas as máquinas são reiniciadas e, uma vez que todas possuem o estado inicial correspondente à mesma pose, um novo gesto só passa a ser reconhecido quando a mão volta à posição inicial.

Veremos agora como as poses são identificadas e rastreadas e como a posição 3D das mãos é calculada e passada para o sistema de reconhecimento.

5.5 Sistema de aquisição de dados

O sistema de aquisição de dados implementado é responsável pelo rastreamento das mãos e da cabeça do usuário durante a interação. Ele é baseado no trabalho de Keskin *et al.* [KEA03] [KAA05] e Azad *et al.* [AUAD07] no qual se utilizam dispositivos não invasivos e, a partir dos dados visuais capturados por duas câmeras em estéreo, obtém-se informações 3D sobre o posicionamento dos

objetos de interesse. Um requisito do sistema de aquisição adotado é que o usuário utilize blusa de manga comprida de cor distinta da pele, uma vez que o sistema utiliza um algoritmo de segmentação por cor de pele para identificar as três regiões de interesse. Para evitar que componentes do ambiente sejam reconhecidos erroneamente, as imagens da câmera passam por um pré-processamento (efetuado antes da segmentação por cor de pele) que remove os componentes do plano de fundo, deixando apenas o usuário na cena.

Uma vez obtidas as três regiões de interesse nas duas imagens, calcula-se o centróide de cada região e, baseando-se na disparidade dos centróides em cada imagem, obtém-se suas posições 3D. Filtros de Kalman 2 e 3D são utilizados para evitar problemas de oclusão de curta duração e melhorar a qualidade do rastreamento.

A seguir serão detalhadas as técnicas de segmentação por cor de pele e segmentação de fundo, bem como a utilização do filtro de Kalman como ferramenta de rastreamento.

5.5.1 Calibração dos dispositivos de captura

Neste trabalho utilizamos um sistema de visão em estéreo para estimar a posição 3D dos objetos presentes numa cena real. Neste sistema, tal estimativa pode ser feita utilizando-se a teoria de geometria projetiva intrínseca entre as duas câmeras, denominada *geometria epipolar*, descrita a seguir.

O conteúdo desta seção é baseado no texto do livro *Multiple View Geometry* de Hartley e Zisserman, capítulos 9 a 11 [HZ08].

Geometria Epipolar

A geometria epipolar de uma cena corresponde à geometria projetiva intrínseca de dois pontos de vista distintos. Sejam c e c' os centros ópticos de duas câmeras distintas cujas matrizes de projeção são P e P', respectivamente, a geometria epipolar de tal sistema pode ser encapsulada numa matriz F, 3×3, de posto 2, denominada *matriz fundamental*, construída de tal forma que, se X é um ponto no espaço 3D visto por ambas as câmeras e x e x' correspondem à suas projeções nos planos das imagens, então:

$$x'^T F x = 0 \tag{5.1}$$

Ou seja, apesar de ser independente da estrutura da cena, utilizando 5.1, podemos calcular F a partir da correspondência entre os pontos das imagens em estéreo sem que haja a necessidade do conhecimento das matrizes de projeção P e P'.

A figura 5.6 apresenta um resumo dos demais componentes da geometria epipolar. São estes:

linha de base (*baseline*): linha que une os dois centros ópticos c e c';

epipolos e e e': correspondem aos pontos de intersecção da linha de base com os planos das imagens das câmeras;

planos epipolares Π: planos que contém a linha de base e um outro ponto qualquer no espaço 3D;

linhas epipolares l e l': linhas formadas pela intersecção de um plano epipolar Π com os planos das imagens das câmeras. Uma linha epipolar, l', também pode ser interpretada como sendo a projeção da reta $\overline{Xc}$ sobre o plano de imagem da outra câmera (centro óptico c').

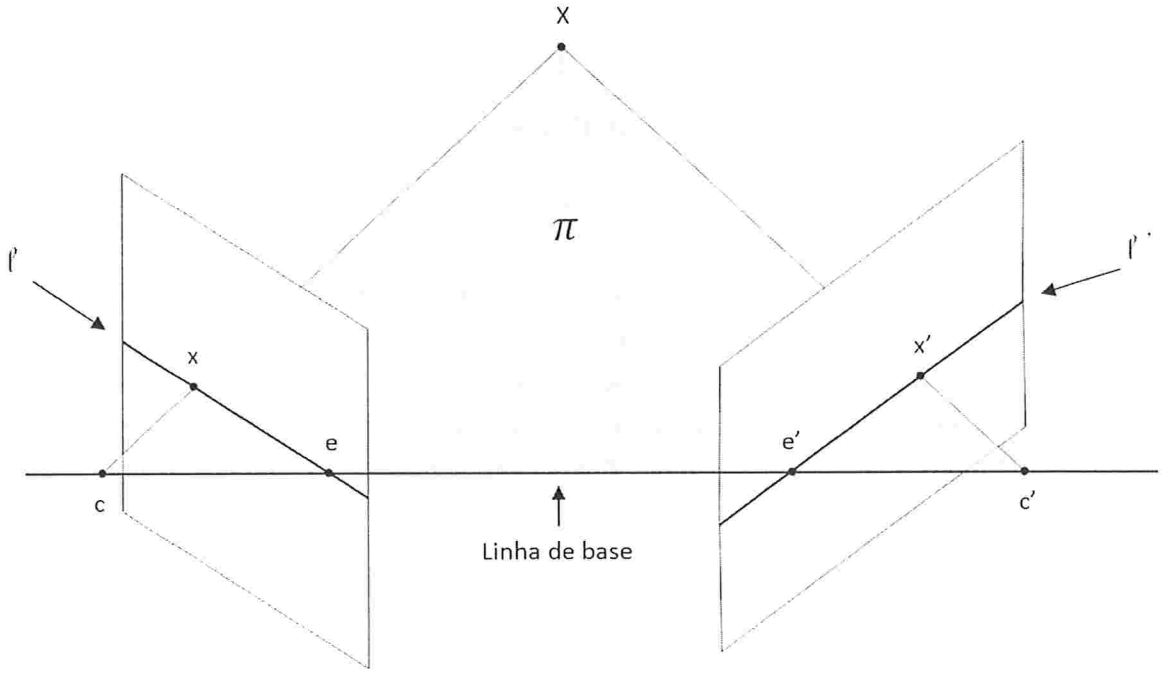

Figura 5.6: Componentes da geometria epipolar

Derivação geométrica da matriz fundamental Uma primeira interpretação da matriz fundamental pode ser feita geometricamente.

Seja π um plano no espaço 3D que não contenha nenhum dos centros ópticos das câmeras. Um raio com origem em c, que passe por x, intercepta o plano π no ponto X, ou seja, X é a projeção de x no plano π. Seja x' a projeção de X no plano da segunda imagem. Uma vez que X pertence à reta $\overline{xc}$, x' deve pertencer à projeção desta reta sobre o plano da segunda imagem, isto é, x' pertence à linha epipolar l'. Tal procedimento é denominado *transferência de ponto via plano* e, ao aplicarmos tal procedimento à cada ponto x_i na primeira imagem, obtemos um conjunto de pontos x'_i projetivamente equivalentes (uma vez que estes também correspondem à projeções dos pontos X_i no plano π). Sendo assim, existe uma matriz de homografia 2D H_π, que mapeia cada x_i em x'_i, isto é, $x'_i = H_\pi x_i$.

Dado um ponto x', a linha epipolar l' que contém x' e o epipolo $\mathbf{e}'$ pode ser descrita como

$l' = \mathbf{e}' \times x' = [\mathbf{e}']_\times x'$[1]. Uma vez que $x' = H_\pi x$, temos que $l' = [\mathbf{e}']_\times H_\pi x = Fx$, ou seja, $F = [\mathbf{e}']_\times H_\pi$.

Ou seja, geometricamente, F representa um mapeamento do plano projetivo 2D da primeira imagem num conjunto de linhas epipolares sobre o plano da segunda imagem que contém o epipolo $\mathbf{e}'$.

Derivação algébrica da matriz fundamental Conforme dito anteriormente, a matriz fundamental pode ser obtida a partir das matrizes de projeção das câmeras (P e P'). Inicialmente, projetamos um raio X a partir de x que satisfaça $PX = x$. As soluções desta equação podem ser escritas na forma:

$$X(\lambda) = P^+ x + \lambda C$$

Onde P^+ representa a pseudo-inversa de P, que satisfaz $PP^+ = I$, e C é o seu vetor nulo (centro óptico da câmera). Facilmente podemos identificar dois pontos pertencentes à reta: o centro óptico C e P^+x ($\lambda = 0$). Tais pontos são projetados na imagem da segunda câmera em $P'C$ e $P'P^+x$ respectivamente. Conforme visto anteriormente, a projeção do centro óptico da primeira câmera sobre a imagem da segunda corresponde ao epipolo da segunda ($\mathbf{e}'$) e a projeção dos pontos do raio X sobre a imagem da segunda câmera corresponde à uma linha epipolar (l'). Sendo assim, $l' = P'C \times P'P^+x = [\mathbf{e}']_\times P'P^+x = Fx$, onde F é a matriz $F = [\mathbf{e}']_\times P'P^+$.

Comparando este resultado com o da seção 5.5.1, verificamos que $H_\pi = P'P^+$.

Cálculo da matriz fundamental Alguns dos algoritmos mais utilizados para o cálculo da matriz fundamental são o 7-Pontos, 7-Pontos com consenso randômico de amostras (*RANdom SAmple Consensus (RANSAC)*) e 8-Pontos normalizado [Har97]. Descreveremos a seguir apenas o algoritmo 8-Pontos normalizado que apresenta bons resultados e cuja implementação é mais simples que as demais.

Considerando um conjunto de n pontos ($x_i \leftrightarrow x_i'$) correspondentes nas duas imagens, com pelo menos 7 correspondências, podemos utilizar a equação $x'^T F x = 0$ para calcular a matriz F. Cada par de pontos correspondentes dá origem a uma equação do tipo:

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0 \tag{5.2}$$

Se definirmos $\mathbf{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})$, podemos reescrever 5.2 como:

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1)\mathbf{f} = 0 \tag{5.3}$$

Sendo assim, a partir das n correspondências, podemos construir o seguinte sistema de equações lineares:

---

[1]Seja $a = (a_1, a_2, a_3)$ e $b = (b_1, b_2, b_3)$, então $[a]_\times$ é uma matriz singular, 3×3, tal que $[a]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$ e $a \times b = [a]_\times b = ((a^T)[b]_\times)^T$

$$Af = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathrm{f} = 0 \tag{5.4}$$

Para que uma solução do sistema exista, a matriz A deve ter um posto no máximo 8. Caso o posto seja exatamente 8, a solução é única (a menos de um fator de escala) e pode ser obtida por métodos lineares.

O algoritmo de 8-Pontos, descrito por Longuet-Higgins [LH81] em 1981, foi utilizado primeiramente para estimar os elementos da matriz essencial (uma especialização da matriz fundamental). Ele consiste em determinar f como sendo o vetor singular que corresponde ao menor valor singular de A. Em seguida, para assegurar a singularidade de F e garantir que esta tenha posto 2, substituímos F por F', tal que $(\det F' = 0)$ e F' minimiza a norma de Frobenius $\|F - F'\|$.

Apesar de ser teoricamente possível aplicar o algoritmo original de 8-Pontos para calcular a matriz fundamental, em 1997, Hartley [Har97] descreveu o seguinte problema em sua utilização: teoricamente, a matriz A deveria ter apenas um valor singular igual a zero, no entanto, experimentos práticos mostram que alguns dos valores diferentes de zero podem ser muito pequenos em relação aos maiores valores. Se utilizarmos mais que oito pares de pontos para construir A, a matriz resultante pode não conter um único valor singular bem definido que possa ser reduzido à zero. Consequentemente, a solução do sistema linear homogêneo de equações pode não ser suficientemente acurada para ser utilizado.

Como uma solução para este problema, Hartley sugeriu que o sistema de coordenadas de cada imagem fosse transformado, independentemente, para um novo sistema que apresentasse as seguintes características:

- A origem do sistema de coordenadas é posicionada no centro da imagem. Tal característica é obtida transladando-se o centro original para o centróide.
- A distância média de um ponto à origem é $\sqrt{2}$, o que pode ser obtido escalando uniformemente as coordenadas do sistema por um fator adequado.

Do resultado destas transformações, é obtido o conjunto de pontos $(\bar{x_i}, \bar{x'_i})$, tais que $\bar{x_i} = Tx_i$ e $\bar{x'_i} = T'x'_i$, onde T e T' são as transformações do sistema original para o novo sistema de coordenadas de cada imagem. O novo conjunto de correspondências também satisfaz, $\bar{x'}^T \bar{F} \bar{x} = 0$, sendo que $\bar{F} = (T'^T)^{-1} F T^{-1}$, ou seja, podemos utilizar os dados normalizados para estimar o valor de $\bar{F}$ pelo algoritmo de 8-Pontos e depois "desnormalizar" o sistema para obtermos a matriz fundamental no sistema original.

Em seu trabalho, Hartley demonstra que, em geral, matrizes fundamentais estimadas utilizando o algoritmo de normalização representam melhor o sistema que as obtidas sem a normalização.

Cálculo das matrizes das câmeras Conforme visto durante a derivação geométrica da matriz fundamental, um par, P e P', de matrizes de câmeras define unicamente uma matriz fundamental F.

Tal mapeamento, no entanto, não é injetor no sentido que uma matriz fundamental não determina unicamente um par de matrizes de câmeras, mas sim um conjunto de pares de matrizes relacionados entre si por uma transformação projetiva. Isto é, sejam P, P' e $\tilde{P}, \tilde{P}'$ dois pares de matrizes de câmeras que possuam F como matriz fundamental. Existe, então, uma matriz H não-singular, 4×4, tal que $\tilde{P} = PH$ e $\tilde{P}' = P'H$. Uma demonstração desta afirmação é apresentada no apêndice B.2.

Dada esta ambiguidade, o cálculo de um par de câmeras definidas por F pode ser simplificado utilizando-se um par de matrizes na sua *forma canônica*, ou seja:

$$P = [I|\mathbf{0}] \quad P' = [M|\mathbf{m}]$$

onde I é uma matriz identidade 3×3 e $\mathbf{0}$ é um vetor coluna nulo de dimensão 3.

Uma matriz não-nula F é a matriz fundamental correspondente ao par de matrizes de câmera P e P' se, e somente se, ${P'}^T F P$ é anti-simétrica [2]. Tal condição equivale à igualdade $\mathbf{X}^T {P'}^T F P \mathbf{X} = 0$ para todo $\mathbf{X}$. Uma vez que $x' = P'\mathbf{X}$ e $x = P\mathbf{X}$, temos que ${x'}^T F x = 0$, que corresponde à definição da matriz fundamental.

Sendo assim, seja S uma matriz anti-simétrica qualquer, e seja um par de matrizes de câmeras de F descrito como

$$P = [I|\mathbf{0}] \quad \text{e} \quad P' = [SF|\mathbf{e}']$$

onde $\mathbf{e}'$ é o epipolo tal que $\mathbf{e}'F = \mathbf{0}$ e P' é possui posto 3. Então, F é a matriz fundamental correspondente ao par (P, P').

Para demonstrar tal afirmação, verificamos que

$$[SF|\mathbf{e}']^T F [I|\mathbf{0}] = \begin{bmatrix} F^T S^T F & \mathbf{0} \\ {\mathbf{e}'}^T F & 0 \end{bmatrix} = \begin{bmatrix} F^T S^T F & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$$

que é anti-simétrica.

Podemos escrever S em termos do seu vetor nulo $S = [s]_\times$, então $P' = [[s]_\times F|\mathbf{e}']$. Luong e Viéville [LV96] sugerem que uma boa escolha para S é $S = [\mathbf{e}']_\times$, o que faz com que $P' = [[\mathbf{e}']_\times F|\mathbf{e}']$. Utilizando os resultados relacionados à ambiguidade projetiva das matrizes das câmeras apresentados no apêndice B.2, podemos descrever uma fórmula geral para um par canônico de matrizes de câmera para uma dada matriz fundamental F como:

$$P = [I|\mathbf{0}] \quad \text{e} \quad P' = [[\mathbf{e}']_\times F + \mathbf{e}'\mathbf{v}^T|\lambda \mathbf{e}']$$

onde $\mathbf{v}$ é um vetor qualquer de dimensão 3 e λ é uma constante real não nula.

[2] Uma matriz $A = a_{ij}$ é anti-simétrica se, para todo i e j, $a_{ij} = -a_{ji}$. Ou seja, uma matriz anti-simétrica é da forma $A = \begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix}$

Reconstrução 3D

De um modo geral, se não for possível estabelecer nenhum conhecimento acerca da localização e das dimensões de uma cena real, não será possível reconstruir sua posição, orientação ou escala absoluta a partir de um par de imagens (ou qualquer outro número de pontos de vista distintos). Isto significa que não é possível determinar qual a latitude, a longitude ou o tamanho real da casa na figura 5.7, mesmo tendo total conhecimento das matrizes das câmeras. Pode-se dizer então que, a partir do conhecimento apenas das matrizes das câmeras (P e P') e de um conjunto de pontos correspondentes em cada imagem, é possível realizar a reconstrução 3D da cena a menos de uma transformação similar (rotação, translação e escala).

Figura 5.7: Casa de bonecas à frente e casa real ao fundo. Sem informações adicionais sobre a dimensão e localização da cena, só é possível realizar a reconstrução a menos de uma transformação similar.

Tal problema pode ser formalmente descrito como: seja H_s uma transformação similar qualquer dada pela equação 5.5, onde R é uma matriz de rotação, $\mathbf{t}$ é um vetor de translação e λ é uma constante de escala.

$$H_s = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & \lambda \end{bmatrix} \tag{5.5}$$

Sejam P e P' um par de matrizes de câmeras e X_i um conjunto de pontos reais vistos por ambas as câmeras. Ao aplicarmos a transformação similar à cena, obteremos o novo conjunto de pontos $H_s X_i$ e as novas matrizes das câmeras PH_s^{-1} e $P'H_s^{-1}$. Note que a projeção dos novos pontos $H_s X_i$ sobre o plano de imagem da nova câmera PH_s^{-1} coincidem com as projeções dos pontos da cena original, pois $PX_i = (PH_s^{-1})(H_s X_i)$. O mesmo é válido para as projeções sobre as imagens

das câmeras P' e $P'H_s^{-1}$.

Decompondo P em $P = K[R_p|\mathbf{t}_p]$, onde K é a matriz de calibração da câmera, temos que $PH_s^{-1} = K[R_pR^{-1}|\mathbf{t}']$, ou seja, a aplicação da transformação similar não altera a matriz K. Tal ambiguidade existe mesmo para sistemas de câmeras calibradas e Longuet-Higgins [LH81] demonstrou em seu trabalho que, para tais sistemas, esta é a única ambiguidade existente na reconstrução.

Resumindo, se nenhuma informação à respeito da calibração das câmeras ou do seu posicionamento relativo é conhecida, então, a resconstrução 3D é possível a menos de uma transformação projetiva. Se a posição relativa entre as câmeras é conhecida, então a reconstrução pode ser feita à menos de uma transformação afim. Caso tanto as matrizes de calibração quanto o posicionamento relativo seja conhecido, então, a reconstrução pode ser feita à menos de uma transformação similar.

Teorema da reconstrução projetiva Seja $x_i \leftrightarrow x_i'$ um conjunto de pontos correspondentes entre duas imagens e seja F a matriz fundamental definida unicamente por estes pontos tal que $x_i'^T F x_i = 0$ para todo i. Se (P_1, P_1', X_{1i}) e (P_2, P_2', X_{2i}) são duas reconstruções das correspondências $x_i \leftrightarrow x_i'$, então, existe uma matriz H não singular tal que $P_2 = P_1H^{-1}$, $P_2' = P_1'H^{-1}$, $X_{2i} = HX_{1i}$ para todo i, exceto àqueles tais que $Fx_i = x_i'^T F = 0$.

Tal teorema implica que é possível calcular uma reconstrução projetiva de uma cena com dois pontos de vista distintos a partir de um conjunto de pontos correspondentes apenas, sem que seja necessário nenhum conhecimento a respeito do posicionamento ou da calibração das câmeras. Outro resultado importante é o fato de reconstruções similares estarem contidas neste conjunto de reconstruções projetivas, ou seja, se (P_E, P_E', X_{Ei}) é uma reconstrução similar e (P, P', X_i) é uma reconstrução projetiva qualquer, ambas estão relacionadas por uma matriz de homografia H, tal que

$$P_E = PH^{-1} \quad P_E' = P'H^{-1} \quad \text{e } X_{Ei} = HX_i$$

Calculando uma transformação similar Existem diversas formas de se calcular a matriz de homografia H, no entanto, apresentaremos aqui apenas um método conhecido como reconstrução direta [HZ08]. Neste método supõe-se a existência de um conjunto com n pontos de controle X_{Ei} cujas posições reais são conhecidas e cujas projeções sobre os planos das imagens constitui o conjunto de correspondências $x_i \leftrightarrow x_i'$.

Seja X_i o conjunto de pontos de controle obtidos a partir de uma reconstrução projetiva qualquer. Considerando que H possui 15 graus de liberdade e que cada ponto de controle fornece 3 equações linearmente independentes sobre os elementos de H, temos que, se $n \geq 5$ (com não mais que 4 pontos coplanares), então é possível obter uma solução linear sobre os elementos de H.

Retificação das imagens

O processo de retificação de imagens pode ser interpretado de diversas formas. Neste trabalho, retificar um par de imagens em estéreo significa aplicar transformações projetivas nas imagens de forma que linhas epipolares referentes a um mesmo ponto real X se tornem paralelas ao eixo x e possuam a mesma coordenada y. Ou seja, as imagens transformadas não apresentam disparidade

na direção do eixo y, apenas na direção do eixo x. Tal processo facilita a busca por pontos correlatos x e x', limitando a busca à linha epipolar de componente y em ambas as imagens.

Uma forma de transformar as linhas epipolares de uma imagem em linhas paralelas, apresentado por Hartley [Har99], é definir uma transformação afim H' que mapeie o epipolo da imagem num ponto no infinito, por exemlo $(1,0,0)^T$. Uma outra condição necessária para se obter bons resultados na escolha de H' é exigir que esta se comporte o mais próximo possível de uma transformação rígida (rotação e translação) ao redor de um dado ponto x_0 da imagem original. Uma boa escolha de x_0 é o ponto central da imagem original.

Tendo isso em mente, seja $x_0 = (0,0,0)$, $\mathbf{e}' = (f,0,1)$ e G uma transformação projetiva tal que:

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix} \tag{5.6}$$

Podemos ver que G leva o epipolo $\mathbf{e}'$ para o ponto no infinito $(f,0,0)^T$. Um ponto $(x,y,1)^T$ é mapeado por G para o ponto $(\hat{x},\hat{y},1)^T = (x,y,1-x/f)$. Se $|x/f| < 1$, então, $(\hat{x},\hat{y},1)^T = (x,y,1-x/f) = (x(1+x/f+\ldots), y(1+y/f+\ldots),1)^T$. Note que, se $x = y = 0$, então $\hat{x} = \hat{y} = 0$, ou seja, nas proximidades da origem, G se comporta aproximadamente como uma transformação identidade.

Para quaisquer x_0 e $\mathbf{e}'$, a transformação H' pode ser, então, obtida por $H' = GRT$, onde G é a transformação afim que leva o epipolo para o ponto no infinito $(0,0,1)^T$, R é a matriz de rotação que leva $\mathbf{e}'$ para um ponto no eixo x $(f,0,1)^T$ e T é uma matriz de translação que leva o ponto x_0 à origem do centro de coordenadas da imagem.

Uma vez definida a transformação H' da primeira imagem, precisamos definir a transformação H da segunda imagem de tal forma que, se l e l' são linhas epipolares correspondentes, então $H^{-T}l = H'^{-T}l'$. Uma forma de se obter H é escolher uma transformação que minimize a soma das distâncias quadráticas:

$$\sum_i d(Hx_i, H'x_i')^2 \tag{5.7}$$

Transformações H e H' que tornam linhas epipolares paralelas e coincidentes em relação à componente y são denominadas *transformações casadas* (*matching transformations*). Conforme é demonstrado no apêndice B.1, se J e J' são imagens em estéreo cuja matriz fundamental é $F = [\mathbf{e}']_\times M$ e H' é uma transformação projetiva de J' que leva o epipolo $\mathbf{e}'$ para um ponto no infinito, então, H é uma transformação projetiva de J casada com H' se, e somente se, é da forma:

$$H = (I + H'\mathbf{e}'\mathbf{a}^T)H'M \tag{5.8}$$

para algum vetor $\mathbf{a}$.

Uma vez que temos interesse apenas em transformações que levem o epipolo $\mathbf{e}'$ para algum ponto no infito, p.ex. $(1,0,0)^T$, temos que $I + H'\mathbf{e}'\mathbf{a}^T = I + (1,0,0)^T\mathbf{a} = H_A$, que é da forma:

$$H_A = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.9}$$

e constitui uma transformação afim.

Sendo assim, dado um conjunto de pontos correlatos $(\hat{x_i}, \hat{x_i}')$, onde $\hat{x_i}' = H'x_i'$ e $\hat{x_i} = H'Mx_i$, podemos definir H_A resolvendo o problema de minimização:

$$\sum_i d(H_A\hat{x_i}, \hat{x_i}')^2 \tag{5.10}$$

Note que, uma vez que H' e M são conhecidas, podemos calcular os valores de $\hat{\mathbf{x}}_i = (\hat{x_i}, \hat{y_i}, 1)^T$ e $\hat{\mathbf{x}_i'} = (\hat{x_i}', \hat{y_i}', 1)^T$. Desta forma, 5.10 se torna:

$$\sum_i (a\hat{x_i} + b\hat{y_i} + c - \hat{x_i}')^2 + (\hat{y_i} - \hat{y_i}')^2 \tag{5.11}$$

Uma vez que $(\hat{y_i} - \hat{y_i}')^2$ é constante (e igual a zero), o problema de se encontrar H_A é reduzido a um problema de minimização de um sistema linear de mínimos quadrados, conforme apresentado em 5.12.

$$\sum_i (a\hat{x_i} + b\hat{y_i} + c - \hat{x_i}')^2 \tag{5.12}$$

Finalmente, para retificar as imagens J e J', basta reamostrá-las utilizando as transformações projetivas de cada imagem, H e H' respectivamente.

Experimentos de calibração

Os experimentos de calibração foram realizados utilizando-se duas câmeras USB idênticas, cujas especificações técnicas são descritas no apêndice C.2, posicionadas de acordo com o esquema apresentado na figura 5.8, na qual:

α: é o ângulo horizontal do campo de visão da câmera;

γ: é o ângulo horizontal formado entre os campos de visão das câmeras;

B: é a distância horizontal entre as câmeras (linha de base);

d: é a distância entre as câmeras e o início da área de visão estéreo;

D_p: é a distância aproximada do padrão de calibração às câmeras;

A_p: é a largura da área estéreo próxima à posição do padrão;

Considerando que os eixos ópticos das câmeras são posicionados de forma aproximadamente paralela, temos que $\gamma \approx \alpha$. Sendo assim, os valores de d e A_p podem ser estimados pelas equações 5.13 e 5.14.

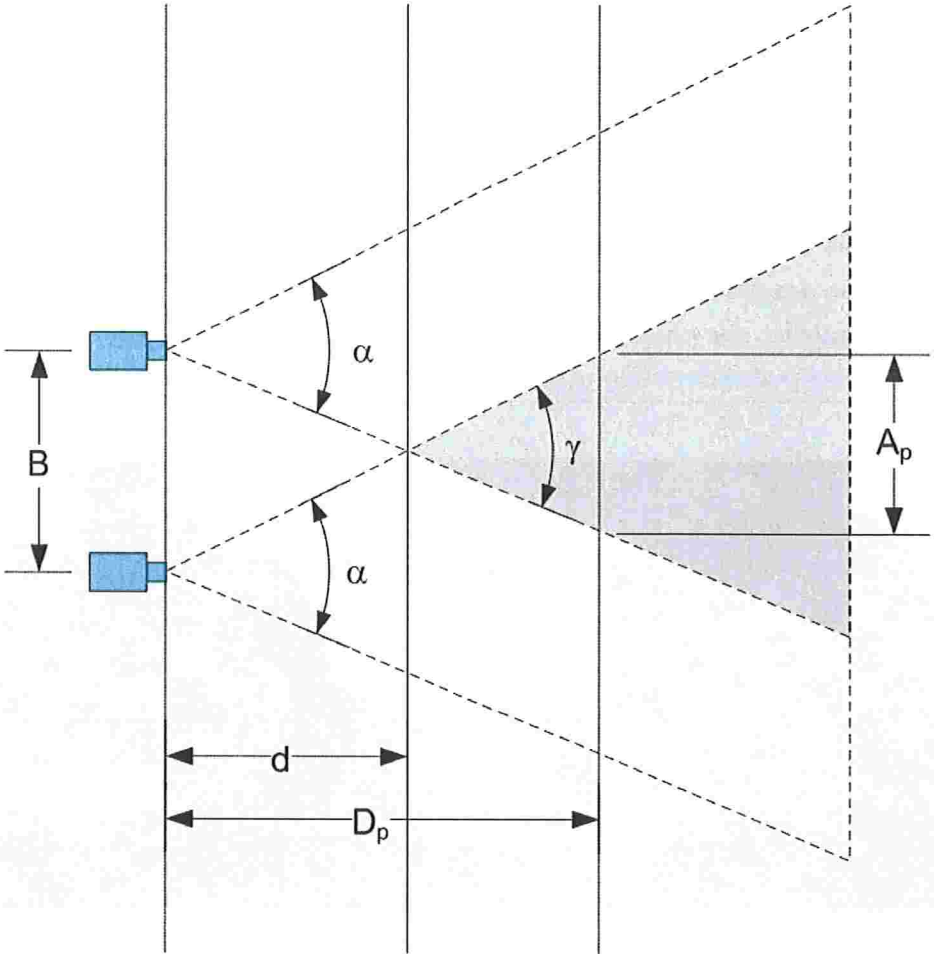

Figura 5.8: Vista superior do posicionamento das câmeras.

$$d \approx \frac{B}{2\tan(\alpha/2)} \tag{5.13}$$

$$A_p \approx 2(D_p - d)\tan(\alpha/2) \tag{5.14}$$

Para identificar os pontos correlatos em ambas as imagens, foram utilizadas as funções *cvFindChessboardCorners()* e *cvFindCornerSubPix()* da biblioteca de visão computacional *OpenCV* [Ope09]. A primeira função recebe uma imagem em tons de cinza e busca por um padrão xadrez de dimensões pré-definidas, retornando as posições dos seus cantos internos se encontrados (figura 5.9). A segunda função recebe os pontos identificados pela primeira e realiza iterações sub-pixel na imagem a fim de aumentar a precisão do posicionamento dos cantos internos do padrão xadrez. Uma vez obtido o conjunto de pontos correlatos, são calculadas as matrizes fundamental, das câmeras e de retificação das imagens de acordo com a teoria descrita anteriormente. Um exemplo

de retificação é apresentado na figura 5.10.

Detalhes de implementação: a utilização de padrões quadriculados é bastante comum dentre os algoritmos de calibração de câmeras isoladas ou em estéreo. As rotinas destinadas à calibração de câmeras isoladas visam principalmente a definição dos parâmetros intrínsecos das câmeras, enquanto que as destinadas a sistemas de visão estéreo visam tanto a definição dos parâmetros intrínsecos quanto dos extrínsecos. Sendo assim, para a correta definição dos parâmetros extrínsecos durante a calibração de sistemas em estéreo, é importante garantir que o padrão será reconhecido com a mesma orientação em ambas as câmeras, o que pode ser obtido utilizando-se padrões não simétricos ($m \times n$, $m \neq n$).

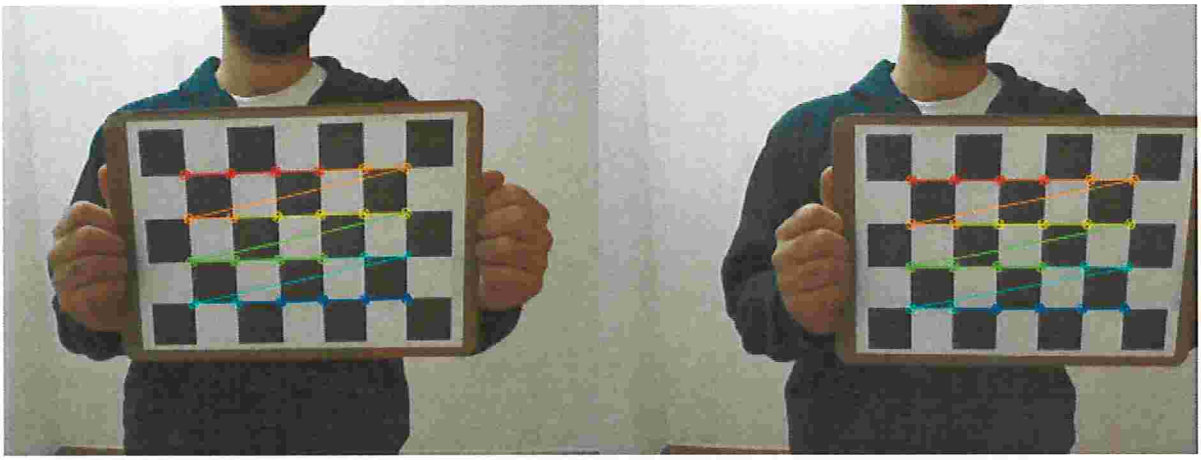

Figura 5.9: Identificação dos cantos internos do padrão xadrez.

Figura 5.10: Resultado da retificação das imagens em estéreo utilizando o algoritmo descrito.

Como uma forma de verificação da qualidade das matrizes obtidas, dados os pontos x_i numa das imagens retificadas, traçamos suas respectivas linhas epipolares na outra imagem, de acordo

B(cm)	d(cm)	A_p(cm)	$Erro_1$	$Erro_2$	$Erro_3$	$Erro_4$	$Erro_5$	Média
10	8,00	46,24	0,16	0,34	0,43	0,30	0,27	0,30
12	9,60	44,24	0,28	0,27	0,32	0,50	0,25	0,32
14	11,20	42,24	0,21	0,31	0,17	0,25	0,24	0,24
16	12,80	40,24	0,26	0,28	0,25	0,31	0,12	0,24
18	14,40	38,24	0,20	0,16	0,39	0,43	0,25	0,29
20	16,00	36,24	0,23	0,28	0,31	0,33	0,19	0,31

Tabela 5.3: Valores de d e A_p e erros obtidos (em pixels) em cada uma das 5 sequências de calibração realizadas sob valores de linha de base B distintos.

com a equação $l'_i = Fx_i$. Em seguida, para cada ponto x'_i, consideramos como medida de erro por ponto a distância à sua respectiva linha epipolar l'_i. O erro geral é dado pela média dos erros por ponto, ou seja:

$$Erro_{ret} = \frac{\sum_{i=0}^{n} dist(x'_i, l'_i)}{n} \tag{5.15}$$

De acordo com as especificações técnicas das câmeras, o ângulo α é fixo e igual a 64°. Durante os testes de calibração, a distância D_p foi mantida fixa em aproximadamente $45cm$, enquanto que a distância entre as câmeras B variou entre 10, 12, 14, 16, 18 e $20cm$. O padrão xadrez utilizado possui 5 linhas e 7 colunas (como o da figura 5.9), sendo que seus quadrados internos possuem lados iguais a $4cm$. Para cada um dos diferentes valores de B, foram realizadas 5 calibrações, cada uma com 10 imagens estéreo do padrão xadrez. A tabela 5.3 apresenta os valores calculados de d e A_p (em centímetros), os erros obtidos (em pixels) em cada uma das sequências de calibração e a média final para cada valor da linha de base.

Conforme podemos observar na tabela 5.3, mesmo variando a linha de base em 10cm, o erro obtido não ultrapassa 0.50 pixel. Durante os testes de interação, o usuário foi posicionado à 180cm do sistema de captura e a cena foi capturada em imagens de resolução de 320×240 pixels. Nestas condições, utilizando as equações 5.13 e 5.14, obtemos que a largura da área estéreo próxima ao usuário é igual a $212,95cm$, o que faz com que a definição seja de $0,66cm/pixel$. Neste caso, um erro igual a 0.50 pixel corresponde à $0.33cm$, que é pequeno em relação aos objetos de interesse.

5.5.2 Segmentação de imagens

Reconhecer objetos em uma cena constitui uma tarefa fácil, natural e cotidiana para a maior parte das pessoas, no entanto, tal atividade, apresenta diversos desafios na área de visão computacional. Neste trabalho, utilizamos uma abordagem de identificação de objetos baseada na composição de duas técnicas de segmentação de imagens para identificar as mãos e a cabeça do usuário. Primeiramente, destacamos o usuário dos demais componentes da cena por meio do algoritmo *CodeBook* de segmentação de fundo. Em seguida, identificamos as áreas da silhueta do usuário referentes às suas mãos e cabeça através de uma técnica de segmentação por cor de pele. Ambos procedimentos são descritos e detalhados nas duas próximas seções. Finalmente, para removermos quaisquer

ruídos remanescentes e identificarmos as três maiores regiões contendo cor-de-pele, utilizamos o procedimento descrito na seção 5.5.2, que é baseado no algoritmo *Union-Find* com compressão de caminhos.

Segmentação de fundo

Em algumas situações, a utilização do algoritmo de segmentação por cor-de-pele pode não ser suficiente para identificar com precisão as áreas referentes às mãos e à cabeça do usuário, como em casos nos quais objetos da cena possuam cores próximas às da pele humana. Para garantir a correta segmentação dos objetos de interesse mesmo nestas situações e para otimizar o processo de segmentação por cor-de-pele, utilizamos o algoritmo proposto por Kim *et al.* [KCHD05] de segmentação de fundo, baseado em *codebook* (CB).

Modelos unimodais, como o descrito por Wren *et al.* [WADP97], não apresentam bons resultados em cenas de ambientes dinâmicos, como cenas ao ar livre com árvores balançando ao vento. Já modelos baseados em mistura de gaussianas (MoG), como o proposto por Stauffer e Grimson [SG99], modelam muito bem fundos complexos e dinâmicos, apesar de apresentarem algumas desvantagens. De acordo com Elgammal *et al.* [EHD99], cenários que apresentam um plano de fundo bastante variável não são fácil e precisamente modelados com poucas gaussianas. Além do mais, dependendo da taxa de aprendizado e atualização do modelo de plano de fundo, podem ocorrer problemas como absorção dos objetos de interesse pelo modelo (taxa muito alta de atualização) ou a exclusão de fragmentos do plano de fundo (taxa muito baixa), conforme descrito por Toyama *et al.* [TKBM99].

O algoritmo CB adota uma técnica de quantização/agrupamento para construir um modelo do plano de fundo a partir de longas sequências de observação. Para cada pixel, é atribuído um *codebook* que consiste em um ou mais *codewords*, compostos por informações sobre a cor do pixel, sua frequência e último momento de utilização daquele *codeword*. Nem todos os pixels possuem o mesmo número de *codewords* e os agrupamentos representados por estes não necessariamente correspondem a uma única distribuição gaussiana ou qualquer outra distribuição paramétrica. Mesmo se a distribuição de cor de um pixel for normal, pode haver vários *codewords* para aquele pixel. Desta forma, o fundo é codificado numa base pixel-a-pixel.

A detecção consiste em testar a diferença entre a imagem atual e o modelo de fundo com respeito às diferenças de cor e brilho. Caso um pixel satisfaça as duas seguintes condições, então ele será classificado como pertencente ao fundo:

a) A distorção de cor referente a um dado *codeword* é menor que o limiar de detecção e

b) O brilho está contido dentro do intervalo de brilho do mesmo *codeword*.

Caso contrário, o pixel é classificado como pertencente a um objeto de interesse.

A atualização do fundo é feita adicionando-se os novos *codewords* (construídos com os dados dos pixels definidos como não pertencentes ao fundo) e removendo-se os *codewords* que possuam uma baixa frequência de utilização ou que não sejam utilizados a muito tempo.

Na figura 5.11 é apresentado um exemplo do resultado obtido utilizando-se o algoritmo CB para a segmentação dos objetos de interesse do plano de fundo.

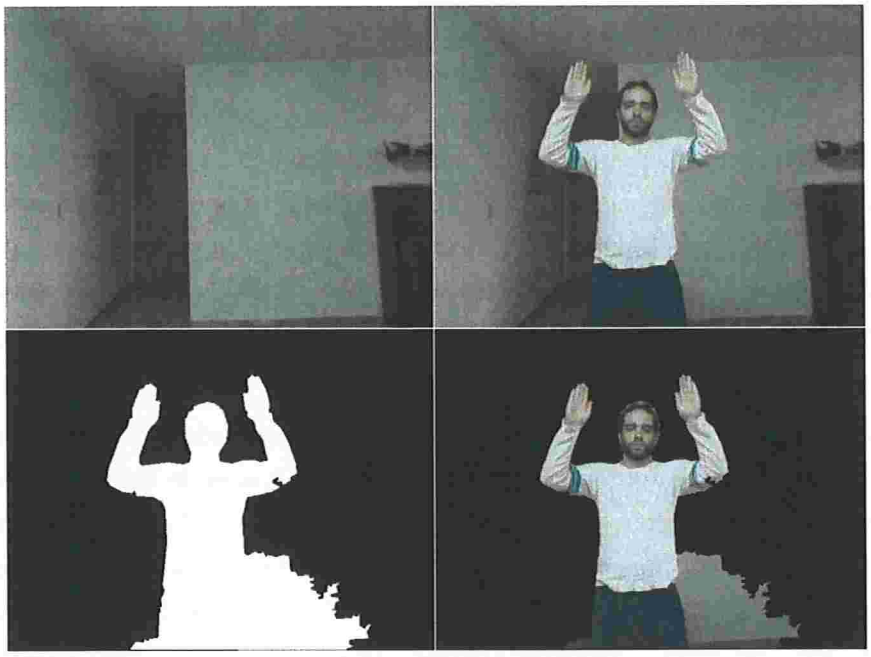

Figura 5.11: Exemplo de segmentação de fundo utilizando-se o algoritmo CB. À cima à esquerda: imagem do fundo. À cima à direita: imagem do fundo mais usuário. Abaixo à esquerda: máscara de segmentação gerada pelo algoritmo. Abaixo à direita: combinação da máscara com a imagem do usuário na cena.

Segmentação por cor de pele

Técnicas de segmentação por cor de pele são muito utilizadas para identificação e rastreamento devido à sua simplicidade [FLW+09] [BBH+07] [SP06]. No entanto, ao optar por esse tipo de segmentação, deve-se levar em conta que a informação de cor não é um fenômeno físico, mas sim uma reação entre uma onda eletromagnética e o dispositivo de captura, seja ele um filme fotográfico ou um dispositivo de carga acoplada (*charge-coupled device (CCD)*). Desta forma, a informação de cor obtida por um dispositivo pode ser influenciada por diversos fatores, dentre eles: iluminação, movimentação dos objetos na cena, tempo de exposição, sensibilidade do filme ou do *CCD*, etc. Variando-se alguns parâmetros, como iluminação ou tempo de exposição, um mesmo dispositivo é capaz de atribuir cores diferentes à um mesmo objeto numa mesma cena. Em se tratando apenas da cor de pele humana, uma mesma pessoa pode apresentar regiões de pele com variações de cor em diferentes partes do corpo, sendo que pessoas diferentes podem apresentar uma variação ainda maior entre si.

Apesar das diversas etnias humanas apresentarem cor de pele aparentemente distintas, Yang, *et al.* [YLW97] mostraram que tais diferenças são mais devido à intensidade que à tonalidade da cor. Sendo assim, a utilização de um espaço de cor normalizado colabora fortemente com a redução da diferença de cor de pele entre diferentes pessoas. Yang, *et al.* também demonstraram que sob uma dada condição de iluminação, as tonalidades da cor de pele humana podem ser representadas por uma distribuição normal.

Baseando-se no trabalho de Yang, *et al.*, Pera *et al.* [PBM04] realizaram uma análise dos resultados obtidos pela segmentação de cor de pele em diferentes espaços cromáticos, objetivando identificar o espaço que apresentasse uma distribuição que tornasse a segmentação mais robusta. A conclusão deste trabalho mostrou que a componente T do espaço de cor TSL possui uma variância muito menor que as demais componentes. Utilizando apenas a componente T ao invés de TS, foram obtidos resultados com a mesma quantidade de falsos positivos e uma menor quantidade de falsos negativos. Outra vantagem na utilização do espaço T-normalizado está na sua dimensão unitária, que reduz o número de operações nos cálculos de probabilidade, tornando este espaço adequado às utilização em sistemas que requerem processamento em tempo real.

O espaço de cor TSL No espaço de cores RGB tradicional cada cor é representada por uma tripla (R, G, B) que indica a intensidade de vermelho, verde e azul de um pixel, respectivamente. O espaço de cores rg-normalizado é um espaço de cores bidimensional que não possui a informação de luminância. Neste espaço, uma cor é representada pela proporção de vermelho, verde e azul ao invés da intensidade absoluta de cada componente. Uma vez que a soma dessas proporções é sempre igual a 1 (um), podemos considerar apenas a proporção de vermelho e verde e descartar a de azul. Consequentemente, uma cor neste espaço pode ser representada pela dupla (r, g).

Apesar de uma dupla (r, g) possuir menos informação que uma equivalente tripla (R, G, B), a primeira costuma ser mais útil em aplicações nas quais queremos justamente considerar apenas a informação de crominância e ignorar a de luminância, ou seja, em sistemas nos quais se deseja uma maior tolerância em relação à mudanças de iluminação.

O espaço TSL, proposto por Terrillon *et al.* [TSFA00], consiste em uma variação do HSV (Hue-Saturation-Value) [GW02]. Neste espaço, cada cor é representada por uma tripla (T, S, L) na qual a componente T (*Tint* - Matiz) representa a informação da cor; S (Saturação) representa a diluição da cor original com a luz branca; e L (Luminância) representa a quantidade de energia de uma fonte de luz percebida por um observador.

Dada uma tripla (R, G, B) representando a intensidade de vermelho, verde e azul, podemos determinar a equivalente tripla (T, S, L) através da seguinte relação:

$$S = \sqrt{9/5(r'^2 + g'^2)} \tag{5.16}$$

$$T = \begin{cases} arctan(r'/g')/2\pi + 1/4 & \text{se } g' > 0 \\ arctan(r'/g')/2\pi + 3/4 & \text{se } g' < 0 \\ 1/2 & \text{se } g' = 0 \end{cases} \tag{5.17}$$

$$L = 0.299R + 0.587G + 0.114B \tag{5.18}$$

Considerando que $r' = r - 1/3$, $g' = g - 1/3$, $r = R/(R + G + B)$, $g = G/(R + G + B)$ e $arctan(r'/g')$ retorna um ângulo no intervalo $[-\pi/2, \pi/2]$.

Classificando um pixel como cor de pele A partir de imagens coloridas de face ou mãos, delimitamos regiões de cor de pele para formar o banco de imagens utilizadas no treinamento. Considerando que as condições de iluminação não são variáveis, a distribuição de cor de pele pode ser representada por um modelo gaussiano, então, tal distribuição pode ser determinada pelo vetor de médias M e pela matriz de covariância C das amostras de cor de pele.

A distribuição da cor de pele obtida na fase de treinamento é, então, utilizada para atribuir uma probabilidade a cada pixel da imagem conforme a sua cor. Tal probabilidade é calculada com base na distância de Mahalanobis, utilizando a média e a covariância calculadas na fase de treinamento.

A distância do pixel x à distribuição gaussiana é dada por:

$$D_M(x) = \sqrt{(x - M)^T C^{-1} (x - M)} \tag{5.19}$$

Uma vez que apenas a componente T do espaço TSL é utilizada, a distância pode ser simplificada utilizando-se apenas a média m e a variância σ^2 do canal T:

$$D_M(x_T) = \sqrt{\frac{(x_T - m)^2}{\sigma^2}} \tag{5.20}$$

E a probabilidade pode ser obtida fazendo-se:

$$P(x) = e^{(-D_M)} \tag{5.21}$$

O resultado de tal algoritmo é uma imagem em escala de cinza cujos pixels mais claros são aqueles que possuem uma maior probabilidade de representarem um objeto com cor de pele. Um exemplo de sua utilização pode ser visto na figura 5.12.

Figura 5.12: À direita, mapa de probabilidade de cor-de-pele gerado à partir da imagem à esquerda. Regiões mais claras representam alta probabilidade. Note que alguns elementos do cenário, como a porta e o chão, apresentam alta probabilidade.

Utilizando a segmentação de fundo como máscara, podemos otimizar o processo de segmentação por cor-de-pele analisando apenas a região referente à silhueta do usuário. Aplicamos um limiar ao mapa de probabilidade obtido, fazendo com que todos os pixels que possuam probabilidade superior ao limiar adquiram a cor branca e todos os demais, cor preta, obtendo, então, uma imagem em

branco e preto. Os passos deste procedimento são ilustrados na figura 5.13, no qual o valor do limiar define como branco todos os pixels com probabilidade maior que 80%.

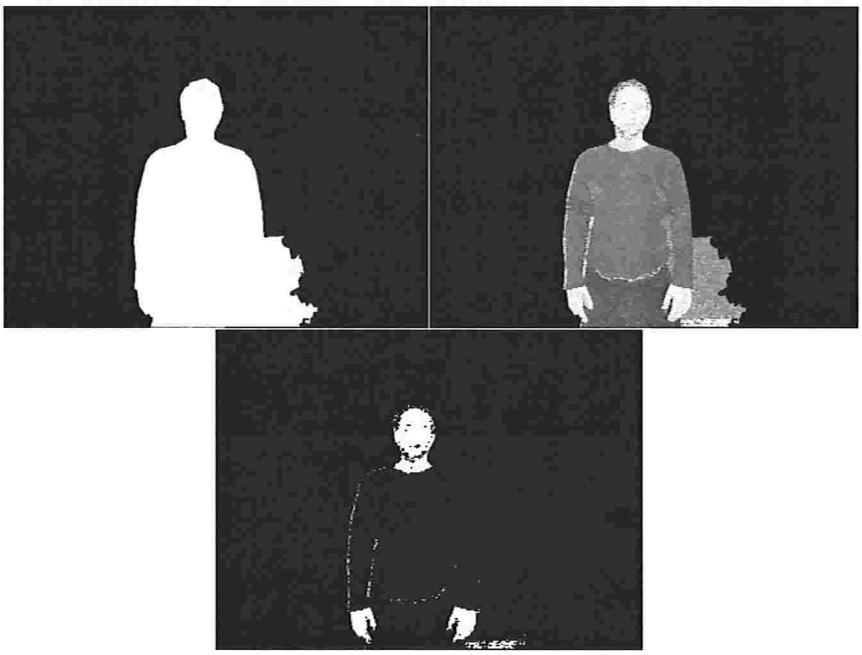

Figura 5.13: Utilização da máscara de segmentação de fundo como otimização da segmentação por cor-de-pele e utilização de um limiar para a geração de um mapa em branco e preto. Topo à esquerda: máscara da segmentação de fundo. Topo à direita: segmentação por cor-de-pele aplicada apenas na região da silhueta. Embaixo ao centro: resultado da aplicação do limiar na imagem do topo à direita.

Identificação dos componentes conexos Conforme podemos ver na figura 5.13 (embaixo ao centro), as três maiores regiões correspondem à cabeça e às mãos do usuário. No entanto, há ainda alguns pontos esparsos, não pertencentes aos objetos de interesse que ainda permanecem após a aplicação da máscara de fundo e do limiar de probabilidade. Para identificar os objetos maiores, utilizamos um procedimento para identificação de componentes conexos baseado no algoritmo *Union-Find* com compressão de caminhos [Sed04]. Uma vez identificados os componentes conexos, selecionamos os três maiores como candidatos à objetos de interesse.

O algoritmo *Union-Find* com compressão de caminhos pode ser resumido pelo pseudo-código apresentado a seguir, no qual $sz[N]$ corresponde a um vetor com N inteiros que armazena o tamanho dos componentes conexos.

```
int sz[N];

unionFind(int p, int q){

   /* Busque pela raiz do componente ao qual 'p' pertence */
   for (i = p; i != id[i]; i = id[i])
```

```
      id[i] = id[id[i]];

   /* Busque pela raiz do componente ao qual 'q' pertence */
   for (j = q; j != id[j]; j = id[j])
      id[j] = id[id[j]];

   /* Se ambos já pertencem ao mesmo componente, continue */
   if (i == j) continue;

   /* Caso contrário, una o componente menor ao maior */
   if (sz[i] < sz[j]){
      id[i] = j; sz[j] += sz[i];
   }
   else{
      id[j] = i; sz[i] += sz[j];
   }
}
```

A função *unionFind* recebe dois identificadores inteiros (p e q) que devem ser conectados. No caso da imagem binária, estes identificadores correspondem à pixels da imagem, que é percorrida de cima para baixo, da esquerda para a direita. Para cada pixel branco encontrado, verifica-se a cor de seus vizinhos diretamente adjacentes à direita e abaixo (conectividade 4). Caso um ou ambos sejam brancos, a função *unionFind* é chamada para conectar o pixel atual ao seu adjacente branco. Um exemplo do resultado da execução deste algoritmo é apresentado na figura 5.14, que constitui o resultado do algoritmo aplicado à figura 5.13 (abaixo ao centro), tendo apenas seus três maiores componentes conexos representados em branco.

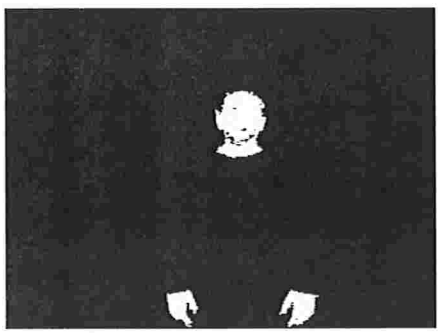

Figura 5.14: Três maiores componentes conexos resultantes.

As coordenadas dos baricentros destes componentes são utilizadas durante o rastreamento 2D dos objetos de interesse.

5.5.3 Rastreando objetos

Neste trabalho, optou-se pela utilização de filtros de Kalman para a realização do rastreamento dos objetos de interesse, pois estes apresentaram bons resultados em outros sistemas de reconhecimento de gestos [MSAG07] [KAA05] [CAHS06].

Filtros de Kalman

Filtros de Kalman são baseados em sistemas lineares dinâmicos discretos. De um modo geral, o filtro é modelado sobre uma cadeia de Markov construída sobre operadores lineares perturbados por um ruído Gaussiano. O estado de um sistema é definido por um vetor de parâmetros e, a cada intervalo discreto, um operador linear é aplicado ao estado (escondido) atual para gerar um novo estado (escondido), considerando-se o modelo de ruído (Gaussiano) e informações de controle do sistema se disponíveis. Outro operador linear é aplicado aos parâmetros do novo estado (escondido), juntamente com um modelo de ruído adequado, para gerar um estado visível.

O estado k pode ser obtido a partir do estado $k-1$ da seguinte forma:

$$x_k = F_k x_{k-1} + B_{k-1} u_{k-1} + w_{k-1} \tag{5.22}$$

Considerando-se que:

- F_k é o modelo (matriz) de transição de estado a ser aplicado ao estado x_{k-1};
- B_{k-1} é o modelo (matriz) de controle do sistema a ser aplicada aos parâmetros do sistema (u_{k-1});
- w_{k-1} é o ruído associado do modelo, construído, normalmente, em cima de uma distribuição normal multivariada ao redor do zero, cuja matriz de covariância é Q_{k-1}, ou seja, $w_{k-1} \sim N(0, Q_{k-1})$.

Num dado momento k, uma observação z_k do estado x_k é feita de acordo com:

$$z_k = H_k x_k + v_k \tag{5.23}$$

No qual H_k é o modelo (matriz) de observação que mapeia os estados escondidos no espaço de observação e v_k é o modelo de ruído relativo ao espaço de observação, construído de forma similar a w_k, ou seja, em cima de uma distribuição normal multivariada ao redor do zero ($v_k \sim N(0, R_k)$).

Tanto o estado inicial, quanto os parâmetros de ruídos a cada intervalo discreto são considerados como sendo mutuamente independentes.

Um filtro de Kalman possui duas fases distintas: predição e atualização. A fase de predição utiliza as informações do estado imediatamente anterior para gerar informações à respeito do novo estado, enquanto que a fase de atualização utiliza dados de observações para refinar o novo estado e obter uma estimativa mais acurada para o mesmo.

Os cálculos relativos a estas fases são descritos nas tabelas 5.4 e 5.5.

Predição do estado	$x_k = F_k x_{k-1} + B_{k-1} u_{k-1} + w_{k-1}$
Predição da covariância estimada	$P_k = F_k P_{k-1} F_k^T + Q_{k-1}$

Tabela 5.4: Fase de predição do filtro de Kalman.

Inovação ou medição residual	$y_k = z_k - H_k x_{k-1}$
Inovação ou covariância residual	$S_k = H_k P_{k-1} H_k^T + R_k$
Ganho ótimo de Kalman	$K_k = P_{k-1} H_k^T S_k^{-1}$
Estado estimado	$x_k = x_{k-1} + K_k y_k$
Covariância estimada	$P_k = (I - K_k H_k) P_{k-1}$

Tabela 5.5: Fase de atualização do Filtro de Kalman.

Rastreando objetos utilizando filtros de Kalman

Neste trabalho utilizamos duas implementações diferentes do filtro de Kalman, uma para o rastreamento das posições 2D dos objetos em cada uma das imagens e outra para o rastreamento das posições 3D calculadas.

Rastreamento em 2 dimensões No rastreamento 2D, são utilizadas as coordenadas dos baricentros dos três maiores componentes conexos para realizar a inicialização e atualização das posições dos objetos em cada uma das imagens. Inicialmente, cada objeto é identificado por sua configuração em relação ao usuário:

Cabeça: objeto acima dos demais;

Mão esquerda: objeto mais à direita na imagem;

Mão direita: objeto mais à esquerda na imagem;

Nos demais instantes, a nova posição de um objeto é primeiramente prevista pelo filtro, levando-se em conta sua direção e sentido de movimentação. Esta previsão é corrigida utilizando-se a posição do componente conexo mais próximo e este dado corrigido é então utilizado como a nova posição do objeto. A associação entre um componente conexo e uma posição prevista é limitada a uma distância de 20 pixels a fim de evitar associações incorretas no caso de um objeto sofrer oclusão ou alguma falha temporária de segmentação. Isto é, caso não haja nenhum componente conexo à uma distância máxima de 20 pixels da posição prevista pelo filtro, utilizamos a posição prevista como próxima posição do objeto. Cada componente conexo é associado a apenas um único objeto. Uma ilustração do rastreamento 2D dos objetos é apresentada na figura 5.15.

Figura 5.15: Rastreando os objetos de interesse em uma imagem 2D. O centro da cabeça é representado por um círculo azul, o da mão esquerda, por um círculo vermelho e o da mão direita, por um círculo verde.

Num dado instante k, o estado x do filtro de Kalman 2D é definido pela posição p e a velocidade v de um objeto em uma das imagens, ou seja, $x_k = (p_x^{(k)}, p_y^{(k)}, v_x^{(k)}, v_y^{(k)})$. A velocidade instantânea do objeto é definida como sendo a diferença entre a nova posição medida e a sua posição anterior. Uma vez que consideramos que entre dois quadros a velocidade do objeto é constante, a matriz de mudança de estado F_k é fixa e dada por:

$$F_k = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.24}$$

Os valores de covariância do ruído associado ao modelo (w_k) e do modelo de ruído relativo ao espaço de observação (v_k) são ajustados de acordo com as condições do ambiente, enquanto que o modelo de observação H_k, que mapeia os estados escondidos no espaço de observação, é definido como uma matriz identidade. Os componentes de controle do sistema B_k e u_k não são considerados, pois não se pode dizer nada à respeito da movimentação do usuário.

Rastreamento em 3 dimensões O rastreamento 3D é feito de forma bastante similar ao 2D. Durante a inicialização, a posição 3D dos objetos é calculada a partir da posição 2D inicial utilizando o algoritmo descrito na seção 5.5.1. Nos demais instantes, as posições 3D são previstas pelo filtro e corrigidas utilizando-se as posições 3D calculadas a partir das posições 2D corrigidas.

Um estado do filtro de Kalman 3D, num dado instante k, é dado por $x_k = (p_x^{(k)}, p_y^{(k)}, p_z^{(k)}, v_x^{(k)}, v_y^{(k)}, v_z^{(k)})$, enquanto que a matriz de mudança de estado F_k é definida como:

$$F_k = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.25}$$

Assim como no caso 2D, os valores de w_k e v_k são ajustados de acordo com as condições do ambiente, e o modelo de observação H_k é definido como uma matriz identidade. Os componentes de controle do sistema B_k e u_k não são considerados.

Em ambos os casos, o posicionamento da mão é calculado relativamente à posição da cabeça. Desta forma, a posição inicial acompanha a posição do usuário na cena, fazendo o sistema mais robusto à pequenas variações de posição paralelas ao plano da imagem.

Os gestos executados pelo usuário por meio da mão esquerda são reconhecidos utilizando-se uma máquina de estados finitos para cada gesto. O posicionamento da mão é dado de acordo com um sistema de coordenadas local definido na seção 5.5.4. Esta mudança de coordenadas torna o sistema mais robusto a pequenas variações no deslocamento lateral e variação na altura do usuário durante a interação. Os gestos são constituídos por uma série de mudanças de estado, que correspondem a mudanças de regiões no espaço 3D local. O funcionamento das máquinas de estado é detalhado na seção 5.4.2.

5.5.4 Mudança de coordenadas

A fim de tornar o sistema mais robusto a pequenas variações de posicionamento do usuário na cena, definimos um sistema de coordenadas local para cada mão $S = \{C, (E_S, E_U, E_F)\}$, no qual C é o centro do sistema de coordenadas, E_S é o vetor correspondente aos eixo lateral, E_U é o vetor que aponta para cima e E_F é o vetor que aponta para a frente (relativamente ao usuário) (figura 5.16). O centro do sistema de coordenadas é dado por $C = P_C + \mathbf{t}$, aonde P_C é a posição real da cabeça do usuário num dado instante e t é um vetor de translação definido pelo usuário antes do início da interação. Desta forma, o sistema de coordenadas move-se sempre relativamente à cabeça do usuário, permitindo que este realize pequenas translações dentro do campo de visão do sistema estéreo sem que o sistema deixe de funcionar.

Os eixos (E_S, E_U, E_F) também são definidos pelo usuário antes do início da interação, representando os principais sentidos de movimentação das mãos: para o lado (E_S), para cima (E_U) e para a frente (E_F). A atividade de definição de S é realizada da seguinte forma: pede-se ao usuário para posicionar ambas as mãos numa posição próxima à cintura, com a palma da mão voltada para a frente (figura 5.17). As distâncias entre as posições reais das mãos (P_E e P_D) e a posição real da cabeça P_C do usuário neste instante definem os vetores de translação t^E e t^D nas equações $C^E = P_C + \mathbf{t}^E$ e $C^D = P_C + \mathbf{t}^D$, nas quais C^E corresponde ao centro do sistema de coordenadas da mão esquerda (S_E) e C_D, ao centro do sistema de coordenadas da mão direita (S_D).

Definidos os centros, pede-se ao usuário para posicionar suas mãos à frente, ao lado e à cima a

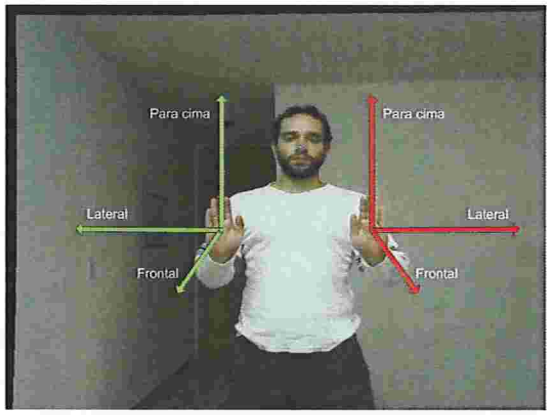

Figura 5.16: Sistemas de coordenadas locais para a mão esquerda (vermelho) e direita (verde).

fim de definir os eixos frontais, laterais e superiores respectivamente (figura 5.17). Note que, desta forma, os conjuntos de eixos E_S^E, E_U^E, E_F^E e E_S^D, E_U^D, E_F^D não são necessariamente ortogonais, mas se adaptam melhor à noção de movimentação do usuário nas direções requeridas.

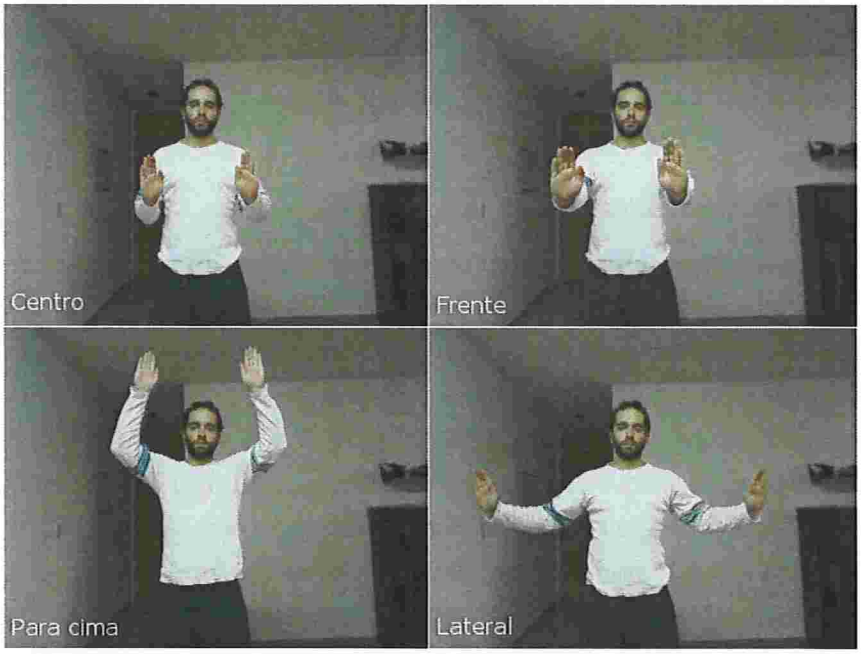

Figura 5.17: Definição do sistema de coordenadas local pelo usuário.

Definido seu sistema de coordenadas local, todas as demais posições das mãos são convertidas e normalizadas com base nos seu centros e eixos.

Capítulo 6

Resultados

Para avaliarmos a viabilidade de utilização da interface de gestos proposta como ferramenta de navegação e manipulação de objetos no cenário do quebra-cabeça 3D, realizamos 3 séries de experimentos:

1. Avaliação da utilização da subinterface de gestos naturais como ferramenta de navegação;
2. Avaliação da utilização da subinterface de gestos simbólicos como ferramenta de interação;
3. Avaliação da viabilidade de integração das duas subinterfaces como solução de interface para o problema do quebra-cabeças 3D.

Para a realização dos testes, utilizamos o sistema computacional e câmeras especificados nos apêndices C.1 e C.2, respectivamente. O roteiro de preparação do ambiente e da execução dos experimentos pode ser visto em detalhes no apêndice A.

6.1 Preparação do sistema

Antes do início de cada seção de testes, seguindo o protocolo do apêndice A, preparamos o ambiente da seguinte forma: fixamos as câmeras do sistema de visão estéreo sobre um tripé com uma distância de $12,0cm$ entre elas. As câmeras foram posicionadas a uma altura de $1,35m$ do chão (figura 6.1), na qual o ângulo de abertura da lente permite que uma pessoa de $1,75m$ se posicione de braços abertos a cerca de $1,80m$ das câmeras e seja visto por completo dos joelhos para cima.

Fixadas as câmeras, foi feita a calibração utilizando um padrão xadrez como o da figura 5.9. Foram adquiridas 10 imagens do padrão em posições ligeiramente diferentes e os pontos referentes aos seus cantos internos foram utilizados como conjunto de pontos correspondentes no procedimento de calibração descrito na seção 5.5.1. Realizamos 5 sequências de calibrações e utilizamos aquela que apresentou o menor erro de reprojeção dos pontos correspondentes ($0,17$ *pixel*).

Calibrado o sistema estéreo, fizemos a captura do modelo de fundo da cena. Para tal, desligaram-se as opções de ajuste automático de brilho, contraste e cor das câmeras para que estes não variassem durante as demais fases de preparação e de interação com o sistema. Para construir o modelo de fundo conforme o algoritmo *CodeBook*, seção 5.5.2, utilizamos uma sequência de 10 segundos de vídeo a 15 quadros por segundo, num total de 150 quadros.

Figura 6.1: Montagem do sistema de captura.

Uma vez construído o modelo de fundo, fizemos a aquisição das amostras de cor-de-pele. Pedimos a cada um dos usuários para se posicionar de frente para o sistema de visão com as palmas das mão voltadas para a frente 6.2. As regiões da testa e das palmas são então selecionadas e as informações de cor dos seus *pixels* são utilizadas na construção do modelo conforme o procedimento descrito na seção 5.5.2.

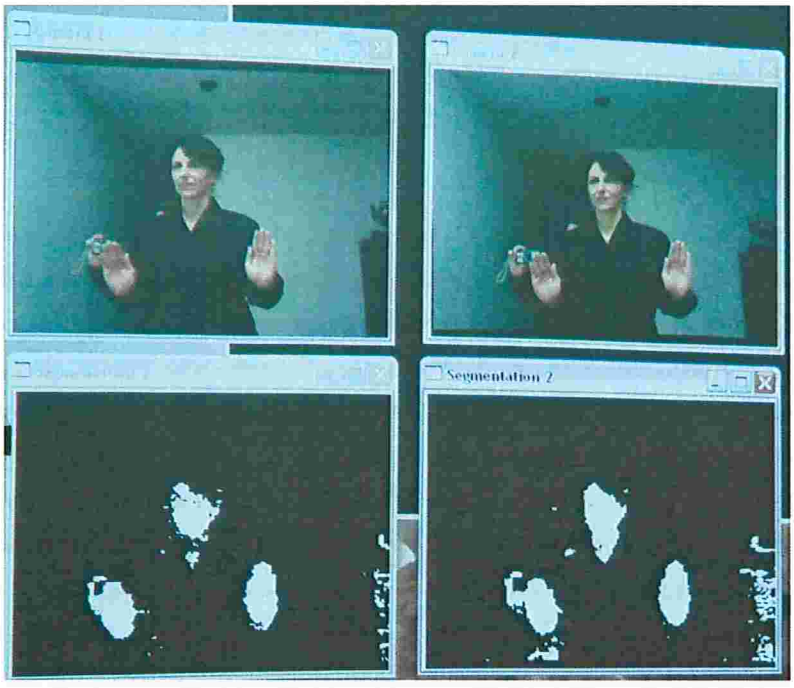

Figura 6.2: Segmentação por cor-de-pele durante os testes de avaliação.

Feita a calibração do sistema estéreo e a inicialização dos filtros de segmentação, os objetos de interesse (mãos e cabeça) passam a ser rastreados em 2 e 3 dimensões. Cada usuário estabelece seu

sistema de coordenadas local pela reprodução de quatro poses pré-definidas:

1. posicionando as mãos de frente para o sistema de captura, com as palmas das mãos voltadas para a frente, à altura do peito. As posições das mãos nesta configuração são definidas como sendo o centro dos seus sistemas de coordenadas;
2. movendo as mãos para a frente, definindo os eixos e limites frontais do sistema;
3. movendo as mãos para cima, definindo os eixos e limites superiores do sistema;
4. movendo as mãos para as laterias, definindo os eixos e limites laterais do sistema.

Esta atividade de definição do sistema de coordenadas é ilustrada na figura 5.17. Definido o sistema de coordenadas local, o sistema está pronto para reconhecer gestos.

De um modo geral, não é necessário repetir as atividades de calibração e construção do modelos de fundo aos se trocar de usuário. Já as atividades de construção do modelo de cor-de-pele e definição do sistema de coordanadas local devem ser refeitas sempre que haja a troca de usuário. No sistema computacional utilizado, o rastreamento dos objetos foi feita a uma taxa de 14 quadros por segundo.

6.2 Perfis dos usuários de teste

Participaram da avaliação do subsistema de gestos naturais 4 usuários cujos perfis eram os seguintes:

Usuário	Idade	Altura	A	B	C
1	25	1,63m	Sim	Sim. Wii, luva de dados e rastreador eletromagnético.	Sim
2	31	1,73m	Sim	Sim. Wii, luva de dados e rastreador eletromagnético.	Sim
3	40	1,60m	Sim	Não	Não
4	55	1,54m	Não	Não	Não

Tabela 6.1: Perfil dos usuários de teste.

Os campos A, B e C da tabela correspondem às respostas das seguintes perguntas:

A: Você utiliza computadores no seu dia-a-dia?

B: Já utilizou alguma interface baseada em gestos? Se sim, quais?

C: Já teve contato com ambientes virtuais tridimensionais?

Conforme podemos ver pelos resultados do questionário, os usuários se dividem em dois grupos (I e II): os usuário 1 e 2 possuem uma maior familiarização com ambientes virtuais e com a utilização de interfaces baseadas em gestos, enquanto que os usuário 3 e 4 possuem pouca familiarização com tais tecnologias.

6.3 Avaliação do subsistema de gestos naturais

A avaliação da interface de gestos naturais é feita medindo-se os tempos de execução de tarefas de movimentação em meio ao ambiente virtual. A movimentação é feita de modo a se alcançar cinco pontos de controle no espaço 3D demarcados ao longo de um caminho por cubos. Duas sequências de caminhos foram utilizadas, nos quais os pontos estavam dispostos da seguinte forma:

Caminho 1:

Ponto 1) 100 unidades à frente da posição inicial;

Ponto 2) 50 unidades à esquerda e 50 unidades à frente do ponto 1;

Ponto 3) 50 unidades à direita e 50 unidades à frente do ponto 2;

Ponto 4) 100 unidades acima do ponto 3;

Ponto 5) 100 unidades abaixo do ponto 4;

Caminho 2:

Ponto 1) 100 unidades acima da posição inicial;

Ponto 2) 100 unidades à frente do ponto 1;

Ponto 3) 50 unidades à esquerda e 50 unidades à frente do ponto 2;

Ponto 4) 100 unidades abaixo do ponto 3;

Ponto 5) 50 unidades à direita e 50 unidades à frente do ponto 4;

Cada usuário repetiu o experimento 3 vezes e os tempos de execução estão dispostos na tabela a seguir em segundos:

	Caminho 1				Caminho 2			
Usuário	1	2	3	Média	1	2	3	Média
1	52	47	39	46	58	51	56	55
2	41	37	33	37	53	59	48	53
3	57	51	55	54	68	72	63	68
4	59	55	64	59	61	59	63	61

Tabela 6.2: Tempos dos testes de interação com o subsistema de gestos naturais.

Analisando a tabela, primeiramente, vemos que todos os usuário, tanto do grupo I quanto do grupo II foram capazes de utilizar o subsistema de gestos naturais para percorrer os caminhos. Os usuários do grupo I apresentaram um tempo médio de execução do Caminho 1 cerca de $16s$ menor que os usuários do grupo II e, para o Caminho 2, cerca de $10s$ menor. Isso mostra que uma maior familiariadade com a navegação em ambientes tridimensionais e a utilização de interfaces baseadas em gestos facilita o uso da interface proposta. Vemos também que, para os usuários do grupo I, os tempos obtidos na 3^a repetição são ligeiramente menores que as da primeira, o que mostra que mesmo com pouco tempo de utilização, é possível adquirir maior prática na utilização da subinterface.

6.4 Avaliação do subsistema de gestos simbólicos

Os quatro usuários descritos na tabela 6.1 também participaram da avaliação do subsistema de gestos simbólicos. Antes de dar início aos testes, os usuários foram apresentados ao vocabulário de gestos e os foram concedidos cinco minutos de prática. As tarefas de avaliação deste subsistema consistiram na execução de cinco sequências de dez ações (tabela 5.2). Para cada sequência, foram contabilizados o número total de ações realizadas pelo usuário, apresentados na tabela 6.4. O número mínimo de ações (e considerado ideal) é 10. Valores acima deste estão associados à imprecisão do sistema e erros do usuário.

	Sequências					
Usuário	1	2	3	4	5	Média
1	15	17	16	15	12	15
2	18	16	12	10	11	13
3	15	18	16	13	16	16
4	16	14	14	14	15	15

Tabela 6.3: Contagem de gestos durante os testes de interação com o subsistema de gestos simbólicos.

Nesta subinterface, ambos os grupos apresentaram um desempenho semelhante, o que aponta para o fato de uma experiência prévia com interfaces de gestos e ambientes virtuais não favorecer seu desempenho durante a utilização.

Durante a execução das tarefas, percebemos que a maior parte dos erros cometidos pelos usuários eram devido a uma confusão entre pares de gestos complementares, como "Rotacionar para a direita" e "Rotacionar para a esquerda". Ao ser solicitado que executasse um gesto, o usuário acabava por executar o seu oposto.

Demais erros ocorridos estão associados com a precisão do sistema. Durante os experimentos, o gesto que apresentou maior taxa de reconhecimento incorreto foi o "Rotacionar para baixo", no qual o usuário deve posicionar a mão esquerda abaixo da posição inicial para que o objeto na tela seja rotacionado no sentido anti-horário ao redor do eixo X. Ao fazê-lo, muitas vezes a mão não permanecia por tempo suficiente na região do espaço referente ao estado "para baixo", indo direto à região de descanso, o que ocasionava a mudança indesejada do estado do sistema para "descanso".

No entanto, ao seguirmos o roteiro de desenvolvimento de interfaces sigerido por Sturman e Zeltzer (seção 2.3.3) e criarmos um vocabulário de gestos que inclui ações opostas e complementares, permitimos ao usuário corrigir tais erros e prosseguir com a interação.

6.5 Avaliação da interface de gestos

Participaram da avaliação da interface de gestos apenas os usuários do grupo I. A tarefa realizada durante a avaliação consiste em montar o quebra-cabeça 3D utilizando ambas as subinterfaces ao mesmo tempo. Para diminuir o tempo necessário para completar a tarefa e facilitar a visualização do ambiente e dos objetos, a cada interação, era retirada apenas uma peça do cubo e posicionada numa orientação aleatória longe de sua posição correta. Com o mesmo propósito de facilitar a

interação foi estabelecido uma região próxima em torno da posição correta da peça que, uma vez atingida, ajustava automaticamente a posição da peça para a correta. Foram realizadas três seções de interação com cada usuário e, a cada interação, uma peça diferente era removida (figura 6.3). Foram medidos os tempos de interação (em segundos) e o número de comandos executados em cada interação, apresentados nas tabelas 6.4 e 6.5.

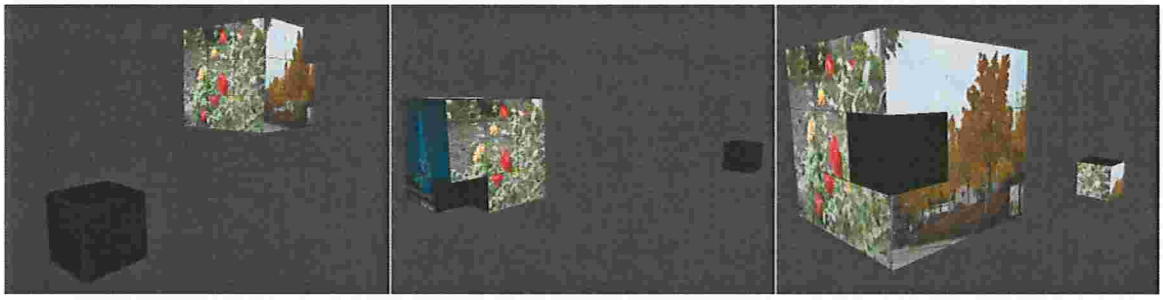

Figura 6.3: Diferentes cenários de interação para avaliação da interface de gestos (1, 2 e 3).

	Tempos							
Cenários	Usuário 1			Média	Usuário 2			Média
1	55	44	40	46	79	51	46	59
2	48	19	25	31	23	32	31	29
3	62	38	53	51	40	62	29	44

Tabela 6.4: Tempos (em segundos) obtidos pelos usuários durante os testes da interface de gestos.

	Número de comandos							
Cenários	Usuário 1			Média	Usuário 2			Média
1	4	16	4	8	26	9	4	13
2	2	2	2	2	2	2	2	2
3	4	4	8	5	6	4	4	5

Tabela 6.5: Número de comandos emitidos pelos usuários durante os testes da interface de gestos.

Conforme podemos observar pelos dados das tabela, o cenário 1 foi o que apresentou uma dificuldade maior em ser completado. Em seguida está o cenário 3 e, por fim, o cenário 2. Nos cenários 1 e 2 a peça faltante encontrava-se não apenas fora da sua posição correta, mas também fora de sua orientação ideal. Já no cenário 3, ela estava apenas fora de lugar.

Fora a primeira interação do usuário 2 com o primeiro cenário, ambos os usuários não tiveram grandes dificuldades em utilizar ambas as interfaces ao mesmo tempo. Comparando os tempos de utilização da subinterface de gestos naturais com a interface final, não observamos tempos maiores de navegação (exceto pelo caso o caso do usuário 2). Levando em consideração que o número ideal de comandos para voltar as peças deslocadas à rua orientação correta é 2, 2, 0, para os

cenários 1, 2 e 3, respectivamente, observamos que um maior número de comandos foi utilizado na primeira interação. Uma vez descoberta a orientação correta do cubo, os usuários foram capazes de aproximar bastante o número de comandos executados ao ideal.

Podemos dizer então que a interface proposta constitui uma interface baseada em gestos em tempo-real que une a liberdade de movimentos dos gestos naturais à praticidade dos gestos simbólicos. Uma outra vantagem estabelecida pela interface é a não-necessidade de treinamento e adaptação aos diferentes perfis de usuário a partir da realização de 3 poses de controle.

Capítulo 7

Conclusão

Neste trabalho de dissertação de mestrado realizamos um estudo a respeito de interfaces homem-máquina baseadas em gestos. Foi feito um levantamento bibliográfico e um resumo do estado da arte, bem como a descrição dos trabalhos mais recentes e relevantes na área.

O estudo e desenvolvimento de interfaces baseadas em gestos mostrou-se um trabalho bastante multidisciplinar envolvendo desde questões psico-sociais da utilização de gestos como ferramentas e suporte à comunicação, passando por técnicas de visão computacional para rastreamento de pessoas e identificação de suas poses, até a utilização de ferramentas matemáticas como filtros de partículas, HMMs e máquinas de estado finitas para a realização efetiva do reconhecimento de gestos.

Outra principal contribuição deste trabalho foi a construção de uma ferramenta de rastreamento de cabeça e mãos em tempo real, baseada em filtros de cor-de-pele, de modelo de fundo e visão computacional estéreo.

Por fim, este trabalho gerou uma interface baseada em gestos que utiliza tanto gestos naturais quanto gestos simbólicos para permitir a navegação e a interação em meio a ambientes virtuais. O sistema foi utilizado e testado por quatro usuários com diferentes perfis de idade, altura e de afinidade com a tecnologia de interfaces de gestos. Todos os usuários de teste foram capazes de realizar as atividades de avaliação das subinterfaces de gestos naturais e simbólicos sem grandes dificuldades. Também foi realizado um teste de interação com as duas subinterfaces sendo utilizadas ao mesmo tempo. Os resultados não apresentaram um aumento significativo de dificuldade, o que mostra que a integração das duas subinterfaces é vantajosa, pois une a liberdade de movimentos dos gestos naturais à praticidade dos gestos simbólicos sem que haja a necessidade de treinamento. A interface também apresentou uma boa adaptação aos diferentes perfis de usuário sendo necessária apenas a realização de 4 poses para determinação do sistema de coordenadas de cada um.

Apêndice A

Protocolo dos experimentos

Este protocolo descreve passo a passo o procedimento de interação com o jogo de quebra-cabeças descrito no capítulo 5, que utiliza o sistema de reconhecimento de gestos proposto.

A.1 Configuração do ambiente

Num ambiente com pelo menos $(3 \times 3)m^2$ de área livre, monte o sistema de captura da seguinte forma:

1. Fixe as câmeras sobre um tripé deixando uma distância de 10 a $12cm$ entre elas. Ajuste a orientação das câmeras de forma a maximizar o campo de visão estéreo do sistema;
2. Posicione o tripé a uma distância de 1.80 a $2.00m$ do local onde o usuário ficará durante a interação;
3. Conecte as câmeras ao computador;
4. Para um melhor funcionamento do sistema, certifique-se de que não haja fontes de luz muito fortes ou variáveis no local e que a maior parte do ambiente e dos objetos visíveis possua cor diferente da cor-de-pele.
5. Posicione o projetor ou monitor de forma que o usuário possa estar sempre, ao mesmo tempo, de frente para o sistema de câmeras e para tela de visualização do jogo.

A.2 Configuração do sistema de reconhecimento de gestos

Após o posicionamento das câmeras, execute o *script* "puzzle3D.bat" presente no diretório "D:/Puzzle3D/". Os próximos passos a serem executados são a calibração das câmeras e a construção do modelo de fundo.

A.2.1 Calibração das câmeras

Para que os pontos tridimensionais tenham suas coordenadas calculadas corretamente, é necessário realizar o procedimento de calibração das câmeras em estéreo. Para tal:

1. Escolha a opção "[1] Configure" dentre a lista de opções apresentadas logo após o início do programa;

2. Escolha, então, a opção "[2] Calibrate" para iniciar a rotina de calibração;

3. Posicione-se à frente das câmeras segurando o padrão xadrez de modo que este ocupe a maior área possível e apareça por completo em ambas as imagens;

4. Uma vez posicionado o padrão, pressione a barra de espaço para realizar a aquisição da imagem;

5. Repita os itens 3 e 4 até que 10 imagens do padrão tenham sido capturadas. Certifique-se de mover ligeiramente o padrão ao redor da imagem entre uma captura e outra, lembrando-se sempre que o padrão deve aparecer por completo em ambas as imagens.

6. Terminada a captura das imagens, o programa realizará o cálculo das matrizes das câmeras e apresentará o erro de reprojeção calculado e o resultado de distorção das imagens. Caso o erro seja maior que 0.5 *pixel* ou as imagens apresentem-se muito distorcidas (bordas abauladas ou singularidades), repita a calibração.

A.2.2 Construção do modelo de fundo

Após a calibração das câmeras, realize o procedimento de construção do modelo de fundo da seguinte forma:

1. Certifíque-se de que nenhum usuário seja visto por nenhuma das câmeras e que as condições do ambiente são as mesmas que no momento da interação do usuário.

2. Ainda nas opções de configuração, escolha a opção "[2] Background". Esta opção fará a aquisição de uma sequência de $10s$ de vídeo que será utilizada para a construção do modelo de fundo. Durante este período, caso o ambiente seja modificado pela introdução de um usuário ou objeto, repita este procedimento.

Este passo conclui o conjunto de estapas de configuração do sistema que podem ser realizadas sem a presença do usuário. Caso as condições do ambiente se mantenham as mesmas entre diferentes seções de interação, não é necessário repetir as etapas descritas até aqui.

A.2.3 Construção do modelo de cor-de-pele

O modelo de cor-de-pele deve ser construído para cada usuário da seguinte forma:

1. Nas opções de configuração (opção "[1] Configure" do menu inicial), escolha a opção "[3] Skin color";

2. Posicione o usuário em frente ao sistema de captura estéreo e peça a este para permanecer imóvel;

3. Utilizando o mouse, selecione uma região de uma imagem que contenha apenas cor-de-pele;

4. Pressione [R] para confirmar a seleção;

5. Repita o procedimento dos itens 2 e 3 para selecionar uma região de cor-de-pele na outra imagem;

6. Uma vez selecionadas as regiões de amostra de cor-de-pele, o modelo será construído e o mapa de probabilidade será exibido. Com base neste mapa, ajuste o limiar de segmentação até que as regiões referentes à cabeça e à mão do usuário estejam bem definidas. É normal que a cena apresente algumas regiões pequenas (ruídos) como contendo cor-de-pele, portanto, tente minimizar os ruídos sem fazer com que as regiões das mãos e cabeça sejam por demais reduzidas.

A.2.4 Definição dos limites de interação

Após a construção do modelo de cor-de-pele, o sistema passará a rastrear as mãos e a cabeça do usuário na cena. Precisamos, então, definir seus limites de interação, para isso:

1. Posicione o usuário à frente do sistema de captura;

2. No menu de configuração (opção "[1] Configure" do menu inicial), escolha a opção "[4] User limits". O sistema mostrará então os pontos centrais das mãos e da cabeça sendo rastreados em ambas as imagens;

3. Peça ao usuário para posicionar suas mãos ao lado da cintura, um pouco à frente do peito, com as palmas voltadas para a frente e, então, clique [I] para definir a posição inicial;

4. Peça ao usuário para estender suas mãos à frente e, então, clique [1] para definir o limite frontal;

5. Peça ao usuário para estender as mãos para cima e, então, clique [2] para definir o limite superior;

6. Peça ao usuário para estender as mãos para os lados e, então, clique [3] para definir os limites laterais;

A.3 Realização do experimento

Finalizada a configuração do sistema de reconhecimento de gestos, o usuário deve realizar ao menos 5 (cinco) seções de avaliação da subinterface de gestos naturais, 5 (cinco) seções de avaliação da subinterface de gestos simbólicos e 5 (cinco) seções de avaliação da interface de gestos.

Para acessar o menu de testes, escolha a opção "[2] Tests" no menu principal.

A.3.1 Subinterface de navegação

No menu de testes, escolha a opção "[1] Navegation" para dar início à atividade de avaliação da subinterface de navegação. Os gestos de navegação são realizados pela mão direita para controlar a navegação pelo ambiente virtual. O objetivo desta atividade é alcançar os 5 pontos de controle demarcados pelos cubos vistos na cena. O próximo ponto de controle a ser atingido é sempre

demarcado por um cubo verde. Pontos de controle já atingidos são demarcados por cubos pretos e pontos ainda não atingidos, por cubos vermelhos.

Para se mover em meio ao ambiente e atingir os pontos de controle, você pode realizar os seguintes movimentos com a mão direita:

mão à frente: mover-se à frente;

mão à direita: rotacionar a câmera para a direita;

mão à esquerda: rotacionar a câmera para a esquerda;

mão para cima: mover a câmera para cima;

mão para baixo: mover a câmera para baixo;

A composição destes movimentos também é possivel, ou seja, você pode mover a mão para frente e para a direita, o que fará com que a câmera se mova para a frente rotacionando à direita.

Ao término de uma seção, defina o nome do arquivo de resultados da seguinte forma: [DDMMAAAA]-[Nome]-Navigation-[Numero].txt, no qual [DDMMAAAA], corresponde à data do experimento, [Nome] deve ser substituído pelo seu nome e [Numero] corresponde ao número de repetição da atividade (1, 2, 3, ...).

A.3.2 Subinterface de interação

No menu de testes, escolha a opção "[2] Interaction" para dar início à atividade de avaliação da subinterface de interação. Os gestos de interação são realizados pela mão esquerda para controlar a interação com os objetos contidos no ambiente virtual. O objetivo desta atividade é realizar os comandos que aparecem na tela realizando gestos com a mão esquerda. Serão requisitadas a execução de 10 comandos em cada atividade. A relação entre gestos e comandos é estabelecida da seguinte forma:

mão à frente: seleciona/desseleciona uma peça em destaque;

mão à direita: rotacionar o objeto para a direita;

mão à esquerda: rotacionar o objeto para a esquerda;

mão para cima: rotacionar o objeto para cima;

mão para baixo: rotacionar o objeto para baixo;

A composição destes comandos não é possível, ou seja, não é possivel rotacionar um objeto para a direita e para cima ao mesmo tempo, portanto, caso você deseje fazê-lo, realize primeiro um comando depois o outro.

Ao término de uma seção, defina o nome do arquivo de resultados da seguinte forma: [DDMMAAAA]-[Nome]-Interaction-[Numero].txt, no qual [DDMMAAAA], corresponde à data do experimento, [Nome] deve ser substituído pelo seu nome e [Numero] corresponde ao número de repetição da atividade (1, 2, 3, ...).

A.3.3 Interface de gestos

No menu de testes, escolha a opção "[3] Puzzle 3D" para dar início à atividade de avaliação da interface de gestos. Os gestos de interação são realizados pela mão esquerda para controlar a interação com os objetos contidos no ambiente virtual, enquanto que os gestos de navegação são realizados pela mão direita para controlar a navegação pelo ambiente virtual. O objetivo desta atividade é completar o quebra-cabeças 3D posicionando as peças faltantes nas suas posições corretas e com a devida orientação. Os gestos a serem utilizados são os mesmo descritos nas seções A.3.1 e A.3.2. Uma seção de avaliação é composta por 3 cenários e em cada um deles estará faltando uma peça diferente.

Ao término de uma seção, defina o nome do arquivo de resultados da seguinte forma: [DDMMAAAA]-[Nome]-Puzzle3D-[Numero].txt, no qual [DDMMAAAA], corresponde à data do experimento, [Nome] deve ser substituído pelo seu nome e [Numero] corresponde ao número de repetição da atividade (1, 2, 3, ...).

A.4 Finalização do experimento

Para finalizar qualquer atividade em qualquer momento, pressione [ESQ].

Muito obrigada pela sua colaboração!

Apêndice B

Provas e demonstrações

B.1 Teorema das transformações casadas

Seja J e J' duas imagens cuja matriz fundamental é $F = [\mathbf{e}']_\times M$, e seja H' uma transformação projetiva de J'. Uma transformação projetiva H de J é casada com H' se, e somente se H é da forma

$$H = (I + H'\mathbf{e}'\mathbf{a}^T)H'M \tag{B.1}$$

para algum vetor $\mathbf{a}$.

Prova: Se $\mathbf{x}$ é um ponto em J, então $\mathbf{e} \times \mathbf{x}$ é a linha epipolar da primeira imagem e $F\mathbf{x}$ é a linha epipolar da segunda imagem. As transformações H e H' constituem um par casado se, e somente se, $H^{-T}(\mathbf{e} \times \mathbf{x}) = H'^{-T}F\mathbf{x}$. Uma vez que tal igualdade deve ser verdadeira para todo $\mathbf{x}$, podemos escrever reescrever a equação B.1, de forma equivalente como

$$H^{-T}[\mathbf{e}]_\times = H'^{-T}F = H'^{-T}[\mathbf{e}']_\times M \tag{B.2}$$

Dado que, para um vetor qualquer $\mathbf{t}$ e uma matriz não-singular M, a seguinte regra de comutação é válida

$$[\mathbf{t}]_\times M = M^*[M^{-1}\mathbf{t}]_\times = M^{-T}[M^{-1}\mathbf{t}]$$

reescrevemos a equação B.2 como $[H\mathbf{e}]_\times H = [H'\mathbf{e}']_\times H'M$. Finalmente, utilizando o teorema visto na seção seguinte (B.2), chegamos ao resultado da equação B.1.

Para provar o contrário, caso a equação B.1 valha, então

$$\begin{aligned} H\mathbf{e} &= (I + H'\mathbf{e}'\mathbf{a}^T)H'M\mathbf{e} = (I + H'\mathbf{e}'\mathbf{a}^T)H'\mathbf{e}' \\ &= (1 + \mathbf{a}^T H'\mathbf{e}')H'\mathbf{e}' = H'\mathbf{e}' \end{aligned}$$

O que implica em H e H' serem transformações casadas.

B.2 Teorema da ambiguidade de transformações projetivas

Para facilitar a demonstração deste problema, podemos considerar que ambos os pares estão na sua forma canônica, ou seja, $P = \tilde{P} = [I|\mathbf{0}]$, $P' = [A|\mathbf{a}]$ e $\tilde{P}' = [\tilde{A}|\tilde{\mathbf{a}}]$. A matriz fundamental F

pode, então, ser escrita como $F = [\mathbf{a}]_\times A = [\tilde{a}]_\times \tilde{A}$, o que implica em $\tilde{\mathbf{a}} = ka$ e $\tilde{A} = k^{-1}(A + \mathbf{a}\mathbf{v}^T)$, para alguma constante k diferente de zero e algum vetor $\mathbf{v}$ de dimensão 3.

Seja H a matriz dada por:

$$H = \begin{bmatrix} k^{-1}I & \mathbf{0} \\ k^{-1}\mathbf{v}^T & k \end{bmatrix}$$

Temos que $PH = k^{-1}[I|\mathbf{0}] = k^{-1}\tilde{P}$ e

$$P'H = [A|\mathbf{a}]H = [k^{-1}(A + \mathbf{a}\mathbf{v}^T)|k\mathbf{a}] = [\tilde{A}|\tilde{\mathbf{a}}] = \tilde{P}'$$

.

o que mostra que os pares P, P' e $\tilde{P}, \tilde{P}'$ são, de fato, relacionados projetivamente.

Apêndice C

Especificações técnicas

C.1 Sistema computacional

Todos os testes de utilização do sistema de rastreamento de gestos foram realizados utilizando-se um computador com as especificações padrão Dell Precision 690, que correspondem à:

- processador Intel Xeon 2GHz (x4);
- 4GB de memória RAM;
- adaptador de vídeo NVidia Quadro FX 3500;

C.2 Câmeras USB

Como dispositivo de captura do sistema estéreo, foram utilizadas duas câmeras Logitech QuickCam Pro 9000 com as seguintes características:

- óptica Carl Zeiss Tessar 2.0/3.7 com autofocus;
- velocidade de captura de vídeo de 30 quadros por segundo (porém, quando duas câmeras estão conectadas, a taxa de captura cai para 15 quadros por segundo);
- resolução de vídeo de até $960 \times 720 pixels$ (nos testes foi utilizada apenas a resolução de $320 \times 240 pixels$);
- conexão USB 2.0.

Referências Bibliográficas

[AMGC02] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[AUAD07] P. Azad, A. Ude, T. Asfour, and R. Dillmann. Stereo-based markerless human motion capture for humanoid robot systems. *Robotics and Automation, 2007 IEEE International Conference on*, pages 3951–3956, April 2007.

[BBH+07] M.E. Bhuiyan, M.A.and Islam, N. Begum, M. Hasanuzzaman, Liu C.H., and H. Ueno. Vision based gesture recognition for human-robot symbiosis. *Computer and information technology, 2007. iccit 2007. 10th international conference on*, pages 1–6, Dec. 2007.

[BBL93] T. Baudel and M. Beaudouin-Lafon. Charade: remote control of objects using free-hand gestures. *Communications of the ACM*, 36(7):28–35, 1993.

[BCD09] T. P. Bednarz, C. Caris, and O. Dranga. Human-computer interaction experiments in an immersive virtual reality environment for e-learning applications. *Engineering Education*, (1992):834–839, 2009.

[BJ98] M.J. Black and A.D. Jepson. Recognizing temporal trajectories using the condensation algorithm. *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 16–21, Apr 1998.

[BL01] A Bottino and A. Laurentini. A silhouette based technique for the reconstruction of human movement. *Computer Vision and Image Understanding*, 83(1):79–95, 2001.

[Bol80] R.A. Bolt. "put-that-there": Voice and gesture at the graphics interface. *of the 7th annual conference on Computer graphics*, pages 262–270, 1980.

[Bux10] Bill Buxton. Gesture based interaction. Chapter 14, 2010.

[BW97] A.F. Bobick and A.D. Wilson. A state-based approach to the representation and recognition of gesture. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(12):1325–1337, Dec 1997.

[Cad94] C. Cadoz. Le geste canal de communication homme/machine: la communication «instrumentale». *TSI. Technique et science informatiques*, 13(1):31–61, 1994.

[CAHS06] T. Coogan, G. Awad, J. Han, and A. Sutherland. *Real Time Hand Gesture Recognition Including Hand Segmentation and Tracking*, volume 4291, pages 495–504. Springer Berlin / Heidelberg, 2006.

[CBK03] K.M.G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, pages I–77–I–84, 2003.

[CG07] R. Cantoni and W. Garcia. Hyperapple. http://www.rejanecantoni.com/hyperapple.html, 2007.

[Cho08] Y. Chow. The wii remote as an input device for 3d interaction in immersive head-mounted display virtual reality. *IADIS International Conference Gaming*, pages 85–92, 2008.

[CR99] Tat-Jen Cham and J.M. Rehg. A multiple hypothes is approach to figure tracking. volume 2, page 244 Vol. 2, 1999.

[DC01] T Drummond and R. Cipolla. *Real-time tracking of highly articulated structures in the presence of noisy measurements.* Citeseer, 2001.

[DDFG01] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice.* Springer Verlag, 2001.

[DR05] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, 2005.

[EHD99] A. Elgammal, D. Harwood, and L.S. Davis. Non-parametric model for background subtraction. *European Conference on Computer Vision*, 2:751–767, 1999.

[FLW+09] Q. Fei, X. Li, T. Wang, X. Zhang, and G. Liu. Real-time hand gesture recognition system based on q6455 dsp board. *Intelligent Systems, 2009. GCIS '09. WRI Global Congress on*, 2:139–144, May 2009.

[GD96] D. Gavrila and L. Davis. *Tracking of humans in action: a 3-d model-based approach*, page 737–746. Citeseer, 1996.

[GDM08] L. Gallo, G. DePietro, and I. Marra. 3d interaction with volumetric medical data: experiencing the wiimote. *Proceedings of the First International Conference on Ambient Media and Systems*, pages 1–6, 2008.

[GW02] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (2rd Edition).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2002.

[Har97] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transaction on Pattern Recognition and Machine Intelligence*, 19(6):580–593, June 1997.

[Har99] R.I. Hartley. Theory and practice of projective rectification. *Int. Journal of Computer Vision*, 35:115–127, 1999.

[How05] N.R. Howe. *Flow lookup and biological motion perception*, volume 1, page 3. 2005.

[HTH00] P. Hong, M. Turk, and T.S. Huang. Gesture modeling and recognition using finite state machines. *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 410–415, 2000.

[HUM02] J.E. Hopcroft, J.D. Ullman, and R. Motwani. *Automatos finitos determinísticos*, pages 48–56. Editora Campus, 2a. edição edition, 2002.

[HZ08] R. Hartley and A. Zisserman. *Multiple View Geometry (2nd Edition)*, chapter 9 - 11. Cambridge University Press, Cambridge, CB2 8RU, UK, 2008.

[IB98a] M. Isard and A. Blake. Condensation—conditional density propagation for visual tracking. *International journal of computer vision*, 29:5–28, 1998.

[IB98b] M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. *Computer Vision, 1998. Sixth International Conference on*, pages 107–112, Jan 1998.

[KAA05] C. Keskin, O. Aran, and L. Akarun. Real time gestural interface for generic applications. In *European Signal Processing Conference*, 2005.

[KCHD05] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(Video Object Processing):172–185, July 2005.

[KCX06] M. Kato, Y.W. Chen, and G. Xu. *Articulated hand tracking by PCA-ICA approach*, page 329–334. 2006.

[KEA03] C. Keskin., A. Erkan, and L. Akarun. Real time tracking and 3d gesture recognition for interactive interfaces using hmm. In *ICANN/ICONIP*, 2003.

[Ken88] A. Kendon. How gestures can become like words. *Cross-cultural perspectives in non-verbal communication*, page 131–141, 1988.

[KF00] O. King and D. Forsyth. How does condensation behave with a finite number of samples? *Lecture Notes in Computer Science*, 1842:695–709, 2000.

[Kim99] J.H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1010):961–973, 1999.

[Kit09] Sotaro Kita. Cross-cultural variation of speech-accompanying gesture: A review. *Language and Cognitive Processes*, 24(2):145–167, 2009.

[LC03] D. Liebowitz and S. Carlsson. Uncalibrated motion capture exploiting articulated structure constraints. *International Journal of Computer Vision*, 51(3):171–187, 2003.

[Lee06] S. Lee. Automatic gesture recognition for intelligent human-robot interaction. *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 645–650, 2006.

[LH81] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, (293):133–135, September 1981.

[LV96] Q.T. Luong and T. Viéville. Canonical representations for the geometries of multiple projective views. *Computer Vision and Image Understanding*, 64(2):193–229, september 1996.

[MA07] S. Mitra and T. Acharya. Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3):311–324, May 2007.

[McN92] D. McNeill. *Hand and mind: what gestures reveal about thought.* University of Chicago Press, 1992.

[MFY+08] C. Manders, F. Farbiz, T.K. Yin, Y. Miaolong, B. Chong, and C.G. Guan. *Interacting with 3D objects in a virtual environment using an intuitive gesture system*, volume 1. ACM New York, NY, USA, 2008.

[MM06] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1052–1062, 2006.

[MREM04] G. Mori, X. Ren, A.A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. *Proc. Comp. Vis. and Pattern Rec*, 2:326–333, 2004.

[MSAG07] R. Munoz-Salinas, E. Aguirre, and M. Garciasilvente. People detection and tracking using stereo vision and color. *Image and Vision Computing*, 25(6):995–1007, 2007.

[MTHC03] I. Mikić, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53(3):199–223, 2003.

[MVMJ07] Okkonen M., Kellokumpu V., Pietikäinen M., and Heikkilä J. A visual system for hand gesture recognition in human-computer interaction. In *In: Image Analysis, SCIA 2007 Proceedings, Lecture Notes in Computer Science 4522, 709-718.*, 2007.

[Nin10] Nintendo wii. http://www.nintendo.com/wii, 2010.

[OP07] A. Oikonomopoulos and M. Pantic. Human body gesture recognition using adapted auxiliary particle filtering. *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 441–446, 2007.

[Ope09] Open computer vision library - opencv. http://sourceforge.net/projects/opencvlibrary/, 11 2009.

[OURH05] C. Orrite-Urunuela, JM Del Rincon, and JE Herrero. 2d silhouette and 3d skeletal models for human detection and tracking. *Proceedings of the International Conference on Pattern Recognition*, 4:244–247, 2005.

[PBM04] B. Pera, R.A. Barbosa, and C.H. Morimoto. Análise comparativa de diferentes espaços cromáticos para detecção de cor de pele. In *SVR 2004: Procedins of the VII Symposium on Virtual Reality*, 2004.

[Per10] The peregrine. http://theperegrine.com/, 2010.

[Pop07] Ronald Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1-21-2):4–18, 2007.

[Rab89] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.

[RJO+09] S. Robertson, B. Jones, T. O'Quinn, P. Presti, J. Wilson, and M. Gandy. *Virtual and Mixed Reality*, volume 5622 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2009.

[RS91] B. Rimé and L. Schiaratura. *Gesture and speech*, pages 239–281. Editions de la Maison des Sciences de l'Homme, 1991.

[RS00] R. Rosales and Stan Sclaroff. *Inferring body pose without tracking body parts*, volume 2. IEEE Computer Society; 1999, 2000.

[RS06] D. Ramanan and C. Sminchisescu. *Training deformable models for localization*, volume 1. 2006.

[RSAS01] R. Rosales, M. Siddiqui, J. Alon, and S. Sclaroff. *Estimating 3D body pose using uncalibrated cameras*, volume 1. IEEE Computer Society; 1999, 2001.

[Sed04] *Union-Find Algorithms*, pages 11–19. Addison-Wesley, 3rd edition, 2004.

[SG99] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 2:–252 Vol. 2, 1999.

[SKLM05] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. *Discriminative density propagation for 3d human motion estimation*, volume 1, page 390. 2005.

[SKM06] C. Sminchisescu, A. Kanaujia, and D. Metaxas. *Learning joint top-down and bottom-up processes for 3d visual inference*, volume 2. 2006.

[SP06] E. Stergiopoulou and N. Papamarkos. A new technique for hand gesture recognition. *Image Processing, 2006 IEEE International Conference on*, pages 2657–2660, Oct. 2006.

[Spe10] Ieee:spectrum: A gaming glove that's fast enough for pros. http://spectrum.ieee.org/video/consumer-electronics/gaming/a-gaming-glove-thats-fast-enough-for-pros, 03 2010.

[SPHB08] T. Schlömer, B. Poppinga, N. Henze, and S. Boll. Gesture recognition with a wii controller. *Proceedings of the 2nd international conference on Tangible and embedded interaction - TEI '08*, page 11, 2008.

[SSSJ05] A. Seth, S.S. Smith, M. Shelley, and Q. Jiang. A low cost virtual reality human computer interface for cad model manipulation. *The Engineering Design Graphics Journal*, 69(2):31–38, 2005.

[ST03] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *The International Journal of Robotics Research*, 22(6):371–391, 2003.

[STW07] C Shan, T Tan, and Y Wei. Real-time hand tracking using a mean shift embedded particle filter. *Pattern Recognition*, 40(7):1958–1970, 2007.

[SZ93] D.J. Sturman and David Zeltzer. A design method for "whole-hand" human-computer interaction. *ACM Transactions on Information Systems (TOIS)*, 11(3):219–238, 1993.

[TKBM99] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1:255–261 vol.1, 1999.

[TPD+09] N.X. Tran, H. Phan, V.V. Dinh, J. Ellen, B. Berg, J. Lum, E. Alcantara, M.G. Bruch, M. an d Ceruti, C. Kao, and et al. *Wireless Data Glove for Gesture-Based Robotic Control*, page 280. Springer, 2009.

[TSFA00] J.-C. Terrillon, M.N. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 54–61, 2000.

[WADP97] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, Jul 1997.

[YLW97] J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaptation. *Lecture Notes in Computer Science*, page 687–694, 1997.

[YOI92] J. Yamato, J. Ohya, and K. Ishii. *Recognizing human action in time-sequential images using hidden Markov model*, page 379–385. 1992.

[ZLB+86] T.G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. A hand gesture interface device. *ACM SIGCHI Bulletin*, 17:189–192, 1986.