Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Computação e Automação Industrial

Alinhamento Imagem-Modelo Baseada na Visão Estéreo de Regiões Planares Arbitrárias

por

Carlos Henrique Quartucci Forster

sob orientação de

Clésio Luis Tozzi

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação, FEEC-UNICAMP, como requisito parcial para obtenção do título de DOUTOR EM ENGENHARIA ELÉTRICA.

Junho – 2004

Campinas – SP

Universidade Estadual de Campinas

Faculdade de Engenharia Elétrica e de Computação

Departamento de Engenharia de Computação e Automação Industrial

Alinhamento Imagem-Modelo Baseada na Visão Estéreo de Regiões Planares Arbitrárias

por

Carlos Henrique Quartucci Forster

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação, FEEC-UNICAMP, como requisito parcial para obtenção do título de DOUTOR EM ENGENHARIA ELÉTRICA.

Aprovada em 29/06/2004 por

Clésio Luis Tozzi (orientador)	Marcelo Knorich Zuffo
Hélio Pedrini	Wu Shin-Ting
Léo Pini Magalhães	Marco Aurélio Amaral Henriques

AGRADECIMENTOS

Agradeço meu orientador Clésio Luis Tozzi por me conduzir em mais este trabalho, dando-me bastante liberdade, mas também me ajudando a manter os pés no chão.

Agradeço Diego Alberto Aracena Pizarro por ter despertado minha atenção ao tema da Realidade Aumentada. O tema já era de meu conhecimento desde uma palestra de José Luis Encarnação no SIBGRAPI de 1999. Agradeço Marçal dos Santos, do Centro de Computação da Unicamp pelo auxílio enquanto eu ainda procurava o tema para a pesquisa.

Agradeço André Luiz Vasconcelos Coelho por despertar meu interesse em participar da representação discente. Agradeço demais companheiros que participam da representação e da APOGEEU, nossa associação de pós-graduandos. Agradeço aos colegas representantes de todas as categorias na Congregação da FEEC e aos membros executivos pela convivência e pelo aprendizado que adquiri nesse conselho.

Agradeço com muito carinho meus pais e familiares que abriram mão de muitas conveniências a favor de me facilitar desenvolver este trabalho.

Agradeço demais colegas e professores que contribuíram direta ou indiretamente para a minha pesquisa.

Aos meus pais e

em memória de Dalva Pacheco Forster

RESUMO

Este trabalho propõe uma solução para determinação automática da transformação rígida 3D que alinha um modelo geométrico previamente conhecido com um modelo observado a partir de imagens, atentando para precisão, robustez e custo computacional. A abordagem proposta contorna diversas dificuldades de métodos similares descritos na literatura. É assumida pela solução a existência de uma fase de preparação *off-line* da representação da geometria conhecida e de uma fase *on-line* de reconhecimento onde imagens são adquiridas por um par estéreo de câmeras com orientação relativa conhecida. A solução proposta se fundamenta no estabelecimento de correspondências imagem-modelo a partir do reconhecimento de regiões planares e na determinação de invariantes entre o modelo conhecido e o modelo reconstruído com base no par de imagens estéreo. A redução da complexidade do algoritmo de reconhecimento foi considerada como principal critério para determinação do conjunto de invariantes utilizados. A determinação das correspondências entre as feições do modelo conhecido e as feições do modelo reconstruído é realizada através do estabelecimento de um consenso por votação quanto à compatibilidade de posições e orientações relativas de pares de feições. A consolidação do alinhamento é feita com base nas múltiplas feições identificadas a partir do processo de votação. Como forma adicional de redução do custo computacional, um processo de indexação por *hashing* é utilizado para representação do conhecimento do modelo necessário para a implementação do processo de votação. A robustez da abordagem quanto aos efeitos de oclusão, movimento abrupto, variações de iluminação e distração é discutida. A viabilidade da proposta é verificada através de experimentos com imagens de uma cena física.

Palavras-chave: Reconhecimento de Objetos, Visão Estéreo, Feições Regionais, Invariantes, Estimação de Pose, Correspondência, Processamento de Imagens.

ABSTRACT

This thesis proposes a solution to automatic determination of a rigid 3D transformation that aligns a previously known geometric model to a model observed from images, allowing for accuracy, robustness and low computational cost. The proposed approach avoids several difficulties present in similar methods reported in the literature. The solution assumes the existence of an *off-line* preparation phase of the geometry representation and an *on-line* phase where images are acquired by a stereo pair of cameras with known relative orientation. The solution is based on the establishment of image-model correspondences given recognized planar regions and on measured invariants between the known model and the model reconstructed from stereo. Complexity reduction for the recognition algorithm was taken as the main criterion for the selection of the invariants. The determination of correspondences between features from the known model and features from the reconstructed model is achieved through the establishment of a consensus by voting regarding the compatibility of relative position and orientation of feature pairs. Camera pose consolidation is obtained based on multiple identified features from the voting process. The approach robustness considering the effects of occlusion, abrupt motion, lighting variations and distraction is discussed. The viability of the proposal is verified by experiments with images from a physical scene.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

TABELA DE SÍMBOLOS

$\bullet^{T}$	transposição de matriz
$\bullet^{-T}$	inversão da matriz transposta
$\tilde{\bullet}$	vetor em coordenadas homogêneas
$\bullet^{*}$	forma inversível de uma matriz (removendo ou acrescentando linhas e colunas)
$\bullet^{-}$	valor intermediário (predição)
$\hat{\bullet}$	valor estimado
$\bullet_{(1)}$	coordenadas em relação ao referencial (1)
$\lvert\bullet\rvert$	cardinalidade
$\varnothing$	símbolo para ausência de correspondência
$\leftarrow$	atribuição
$\rightarrow$	domínio e contra-domínio de função
$\rightarrow$	implicação
$\vee$	disjunção
$\wedge$	conjunção
$\neg$	negação
$\times$	produto vetorial
$\propto$	proporcional
$\mathbf{0}$	vetor nulo
α_i	coeficientes que regulam a forma da imagem retificada
Γ	curva genérica
ε	ângulo genérico
θ	inclinação entre eixos x e y idealmente 90°
θ	ângulo genérico
λ	multiplicador de Lagrange
ρ	razão cruzada
$\mathbf{A}$	deformação afim do plano imagem
$\mathbf{A}$	domínio dos atributos individuais de feições

$[a\ b\ c]^T$	vetor na direção de mínima dispersão
A_i	atributos de I_i
a_{ij}	coeficientes de uma matriz de transformação afim **A**
B	coordenadas do centróide
B	matriz de calibração de câmera
b	número de elementos na base
B_m	atributos de M_m
$b(X,Y,Z)$	representação binária implícita de um volume
C	centro de projeção
C	matriz de calibração de câmera
C_i	centro de projeção da câmera i
C_{im}	verdadeiro se houver correspondência entre I_i e M_m
C_x, C_y	valores intermediários para determinar o centróide
$d(a,b)$	métrica genérica
$\vec{d}(A,B)$	distância de Hausdorff orientada
E	função de energia genérica
E	matriz essencial
e_1, e_2	epipolos
E_1, E_2	epipolos no espaço, mas sobre o plano-imagem
F	matriz fundamental
f	distância focal
$\mathbf{F}_R$	matriz fundamental para imagens retificadas
$g(a,b,c)$	restrição não-linear
H	custo de acesso à tabela *hash*
$H(A,B)$	distância de Hausdorff
$h(\bullet)$	função de *hash*
I	número de feições-imagem
I	identidade
I_i	feição-imagem
K	tolerância no RANSAC
K	ganho de Kalman
k_i	coeficientes de uma transformação projetiva planar
$[K_u, K_v]$	densidade de sensores no eixo horizontal e no vertical
L	número de tentativas no RANSAC
$\mathbf{L}_1, \mathbf{L}_2$	transformações de retificação
M	número de ítens disponíveis para o RANSAC
M	conjunto de feições que forma o modelo
M	número de feições-modelo
M	número de feições em cada um dos m modelos
M	matriz de transformação geométrica

m	número de modelos
$\mathbf{M}_{12}$	transformação do referencial 1 para o referencial 2
M_m	feição-modelo
m_{pqr}	um dos momentos estatísticos de ordem $p+q+r$
N	número de ítens suficiente para o RANSAC
$\mathbf{N}$	matriz projeção perspectiva
n	distância ao eixo X
$N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	distribuição normal
P	ponto no espaço
P	soma do número de vértices dos polígonos que representam as feições de uma imagem
p	ponto no plano (plano-imagem)
$P(a \mid b)$	probabilidade condicional de a dado b
$\mathbf{Q}$	inversão da deformação afim no plano imagem
R, G, B ou r, g, b	intensidade para cada cor
$\mathbf{R}$	matriz de rotação
$\mathbf{R}$	domínio dos atributos de pares de feições
r	medida dos atributos de um par de feições-imagem ou de feições-modelo
R_{ij}	atributos do par (I_i, I_j)
r_{ij}	parâmetros de rotação
$\mathbf{S}$	matriz produto vetorial
S_{mn}	atributos do par (M_m, M_n)
$\mathbf{T}$	matriz de translação
$\mathbf{t}$	linha de base
t_i	parâmetros de translação
$\mathbf{u}$	vetor perpendicular ao plano epipolar
$[u_0 \; v_0]^T$	ponto principal da imagem
$\mathbf{v}$	vetor que define o eixo óptico
$[X_V \; Y_V \; Z_V]^T$	componentes do vetor $\mathbf{v}$
$[X \; Y]^T$	coordenadas de um ponto sobre a imagem
$[X \; Y \; Z]^T$	coordenadas de um ponto no espaço 3D
$[x \; y]^T$	coordenadas de um ponto sobre a imagem
$[x \; y \; z]^T$	coordenadas de um ponto no espaço 3D
W	coordenada homogênea
w	coordenada homogênea

Capítulo 1 - INTRODUÇÃO

1.1 O problema de alinhamento

Alinhamento é a determinação da transformação pela qual se pode relacionar e converter medidas entre dois sistemas de referências. Um caso especial de alinhamento é o alinhamento visual, onde a transformação considerada ocorre entre os referenciais de objeto e de câmera. Da relação entre esses sistemas de referência depende a formação da imagem do objeto na câmera.

O termo “alinhamento baseado em visão” se refere à técnica utilizada para alinhamento que se baseia em informações adquiridas através do processamento e da análise de imagens. O alinhamento visual pode ser ou não baseado em visão da mesma forma que o alinhamento baseado em visão pode ter finalidades diversas.

O problema da determinação do alinhamento automático baseado em imagens de intensidade, que é o objetivo desta tese, se enquadra no problema mais geral de visão por computador. Visão é um processo complexo que parte de imagens e realiza medições diretas e indiretas, bem como interpretação e inferência de significados. Visão Computacional é a disciplina que trata desse processo quando realizado por um computador.

Imagens de intensidade são conjuntos de medidas de intensidade luminosa sobre uma superfície. Sem perda de generalidade, consideram-se amostras pontuais (pixels) espaçadas regularmente sobre um suporte retangular definido sobre uma superfície plana. Além do

referencial do objeto e do referencial da câmera, que são sistemas de referência do espaço 3D, deve ser considerado também o referencial 2D da imagem. A relação entre o referencial 2D da imagem e o referencial 3D da câmera é definida pelos modelos de câmera, dentre os quais o mais comum é o modelo de câmera de orifício, que é adotado nesta tese.

A relação entre referenciais de câmera e de objeto é representada por uma transformação rígida (isométrica), que é composta por uma translação e uma rotação, com seis graus de liberdade e representada por seis parâmetros. A relação entre referenciais de objeto e câmera é, dessa forma, definida por uma medida de posição, que corresponde à representação da translação, e por uma medida de orientação, que corresponde à representação da rotação. O conjunto formado pelas medidas de posição e de orientação é chamado pose. Determinar a pose relativa entre câmera e objeto, ou seja suas relações de posição e orientação é equivalente a determinar a transformação entre os referenciais, tomando como entrada pontos conhecidos no modelo geométrico da cena e seus pontos correspondentes numa imagem ou conjunto de imagens de intensidade obtidas simultaneamente por um sistema de câmeras. Dadas as correspondências em número suficiente, o problema de determinar os parâmetros da pose é bem conhecido, vide apêndices B e C. Ao invés de buscar a solução baseada em pontos, é comum, para reduzir a complexidade do processo de determinação de pontos homólogos, considerar entidades geométricas salientes, chamadas feições, dentre as quais, cantos, bordas e regiões uniformes são exemplos típicos.

1.2 Aplicações e seus requisitos

O problema de alinhamento automático está presente em diversas aplicações. Em avanços tecnológicos recentes nas áreas de ambientes de trabalho, edição de vídeo, vigilância, monitoramento e navegação de veículos, tem-se acentuado a necessidade de determinar automaticamente a localização e a orientação de objetos no ambiente. Para ambientes de trabalho, por exemplo, tecnologias que permitem localizar objetos têm sido utilizadas em projetos inovadores de interface homem-máquina, e, de fato, o campo de possibilidades abertas é grande. Já, para edição de vídeos, a posição e a orientação de objetos no espaço vem sendo utilizada para acrescentar gráficos sintéticos de forma alinhada com as imagens dos objetos reais. Nas áreas de vigilância e monitoramento, é importante registrar as trajetórias dos

movimentos de pessoas ou objetos. Apresentamos a seguir, de forma breve, a relação de três exemplos de aplicações (navegação autônoma, interfaces modernas e realidade aumentada) com o problema de alinhamento.

No caso da navegação autônoma de veículos, usualmente tem-se um robô dotado de câmeras com um objetivo a cumprir e mecanismos para auto-localização permitem a esse robô planejar seus movimentos até atingir seu objetivo. Como no problema de alinhamento visual, quer-se determinar a pose (auto-localização) conhecendo um modelo do ambiente (mapa) com base nas informações extraídas de imagens. É importante no problema de navegação autônoma que não haja necessidade de intervenção do usuário. Por exemplo, a capacidade de recuperar a localização do robô sem dados do passado quando o robô é ligado numa localização arbitrária dentro da região modelada por seu mapa.

Dentre o escopo das interfaces modernas, destaca-se o problema de rastreio de *phicons* (ícones físicos). Ícones físicos são objetos manipuláveis cujo estado é monitorado e serve como entrada de dados para uma aplicação. O rastreio de ícones físicos enquadra-se no problema de alinhamento, pois se quer determinar as poses dos objetos físicos que são definidos na fase de projeto de forma que seus modelos são previamente conhecidos. Os requisitos dessa aplicação são a automaticidade (não se deve solicitar intervenção do usuário na determinação de correspondências), a robustez ao movimento abrupto (é importante saber sempre o estado de cada *phicon*) e obtenção de resposta num prazo compatível com a atividade de interação.

O terceiro exemplo de aplicação é a realidade aumentada. Muitos sistemas de realidade aumentada se fundamentam na capacidade de rastrear *displays*, sensores (por exemplo, câmeras) e objetos físicos para que se possa combinar *on-line* imagens do mundo real com imagens de objetos sintéticos. Novamente, há consistência com o problema de alinhamento considerado dado que se quer o alinhamento entre a imagem adquirida e a imagem sintética, que pode ser obtido uma vez determinada a pose relativa entre câmera e objeto. Os principais requisitos de sistemas de realidade aumentada são a robustez ao movimento abrupto e à oclusão parcial, a reinicialização automática, a precisão e o respeito a restrições de tempo de execução. Dependendo da aplicação, requer-se *throughput* alto e latência baixa, pois o olho

humano é muito sensível a variações e desrespeitar esses requisitos de tempo em um procedimento *on-line* implica efeitos indesejáveis.

Para criar uma solução para o alinhamento automático aplicável às diferentes situações encontradas, procura-se, em geral, atender a requisitos de tempo, robustez e precisão e condições de flexibilidade, como por exemplo:

- limites de *throughput* e latência;
- robustez a movimentos abruptos de câmeras e de objetos;
- robustez a variações de iluminação;
- robustez a oclusão parcial, distratores e imagens com *clutter*, isto é, a presença de muitos objetos irrelevantes na imagem ou no caso de uma cena muito complexa;
- independência da qualidade da segmentação;
- erro pequeno de alinhamento medido em número de pixels sobre a imagem;
- exclusividade do uso de câmeras como sensores;
- possibilidade de incluir no processo de visão feições naturais além de feições artificialmente inseridas na cena.

1.3 Abordagens e dificuldades na solução do problema de alinhamento

O problema de alinhamento baseado em visão pode ser abordado de diferentes formas. Uma dessas formas são as abordagens híbridas que complementam o sensor óptico com outro tipo de sensor de forma que um sensor compensa os pontos fracos do outro e vice-versa. Por exemplo, um sensor óptico pode corrigir os erros acumulados por um sensor mecânico inercial. Por sua vez, o sensor inercial não é influenciado por oclusão ou por variações de iluminação, que afetam o sensor óptico. Uma abordagem que seja baseada no emprego exclusivo de câmeras como sensores para localização de objetos é bastante conveniente por

muitos motivos, entre eles, a liberação da necessidade de acoplamento físico que existe em algumas técnicas de localização que utilizam outros tipos de sensores. Além disso, a precisão de medida por câmeras é comparativamente melhor que outros tipos de sensores e a câmera pode ser vista como um dispositivo “passivo” no sentido de que, apesar de necessitar de um ambiente iluminado, não atua no ambiente emitindo sinais, como por exemplo no caso de um sonar. Outro motivo importante é que câmeras têm-se tornado produtos de baixo custo, popularizadas em produtos de consumo de massa como é o caso recente dos telefones celulares.

Evidentemente, também existem problemas relacionados à opção por utilizar câmeras. As maiores críticas são quanto à robustez e ao custo computacional. Técnicas baseadas em câmeras estão sujeitas aos problemas de campo de visão limitado, oclusão, amostragem e dependência das fontes de iluminação. Quando se utilizam câmeras para coletar informação do ambiente, é necessária a aplicação de um conjunto de técnicas para análise das imagens adquiridas. Quanto ao custo computacional, dependendo de quão automático é o sistema projetado, algoritmos iterativos muito pesados de busca e otimização podem ser necessários.

Para realizar o alinhamento baseado em feições extraídas de imagens, divide-se o problema nas seguintes 3 etapas: (1) extração da informação das feições da imagem; (2) correspondência entre feições da imagem e do modelo; (3) estimação da pose. A dificuldade para automação do processo de alinhamento não se encontra efetivamente na etapa de estimação de pose, mas sim nas etapas de determinação das correspondências que, de certa forma, depende do processo de segmentação.

A correspondência entre feições da imagem e do modelo consiste em uma busca num espaço combinatório de hipóteses, que se torna mais complexa sob a presença de ambigüidades. Enumeramos, a seguir, as formas de abordagem mais comuns ao problema de estabelecimento de correspondências:

- A abordagem por intervenção do usuário consiste em solicitar, sempre que for necessário, para o usuário interativamente marcar as correspondências. Essa abordagem se baseia em que, para o cérebro humano, o problema de

correspondência é trivial. No entanto, a entrada dos dados é problemática por ser lenta, cansativa e sujeita a erros. Quando se tem muitas imagens para marcar ou tempo restrito, recomenda-se descartar essa abordagem.

- A abordagem por solução incremental é muito utilizada, principalmente em sistemas com restrições severas de tempo, como é o caso da realidade aumentada (discutida no item E.3 do apêndice E). Esse tipo de abordagem considera uma seqüência de imagens com pouca variação e o conhecimento da correspondência no quadro anterior. Assim, essa abordagem produz métodos pouco robustos por serem sensíveis ao movimento abrupto da câmera e ao *clutter* (caso de imagens repletas de objetos irrelevantes). Além disso, esse tipo de abordagem é incapaz de atuar por si só por necessitar da solução para um quadro inicial. A automaticidade e a robustez ao movimento abrupto são características muito importantes. Sistemas atuais de rastreio apresentam grande dificuldade de inicialização e muitas vezes recaem no caso do item anterior, requerendo do usuário o estado inicial para o processo de rastreio. Abordagens incrementais costumam falhar em casos de movimento abrupto, havendo necessidade de reinicialização sempre que esses casos ocorrerem. Um sistema que possa se reinicializar automaticamente na ocorrência desses casos é de grande valia.

- A abordagem por marcação considera um ambiente controlado no sentido que podem ser colocadas marcas nos objetos. Essas marcas são planejadas de forma a serem facilmente detectadas e identificadas pelo sistema de visão. As aplicações dessa abordagem são limitadas a esse tipo de ambiente. No caso de feições naturais, dado que há necessidade do conhecimento prévio do modelo, este deve ser construído durante uma fase de autoria. A possibilidade de exploração de marcas que aparecem de forma natural nos ambientes de trabalho permite que a abordagem se aplique a um número maior de situações.

Dividimos as abordagens automáticas do processo de estabelecimento de correspondências nos dois seguintes grupos:

- A abordagem automática por exploração combinatória do espaço de possíveis correspondências se apóia, geralmente, em algoritmos de busca, nos quais hipóteses de correspondências são levantadas e testadas. Esse espaço cresce exponencialmente com o número de feições do modelo ou da imagem, sendo necessárias restrições ou heurísticas para que o problema seja tratável. A restrição mais empregada, quando o problema permite, consiste em limitar o espaço de possibilidades de acordo com o tipo de transformação geométrica que mapeia do espaço do modelo ao espaço da imagem.

- A abordagem automática por exploração do espaço de possíveis transformações geométricas assume que há uma forma fixa para a transformação geométrica (paramétrica, por exemplo) que relaciona o espaço-modelo com o espaço-imagem. Hipóteses de transformações são geradas e é testado se estas produzem correspondências compatíveis entre as feições da imagem e do modelo. Esse processo de geração e teste pode ter a forma de uma busca, que acontece nos métodos de correlação, ou de um agrupamento, como em métodos baseados na transformada de Hough. Para transformações representadas por um número finito de parâmetros, essa abordagem tem custo polinomial. Por exemplo, para a transformação rígida, representada por seis parâmetros, a busca é feita num espaço de dimensão seis.

É possível considerar estas duas últimas formas de determinação da correspondência como abordagens por exploração do espaço de hipóteses. Uma diferença notável entre elas é que o espaço de transformações geométricas (por exemplo, o espaço de parâmetros que define uma classe de transformações) é contínuo enquanto que o espaço de correspondências é discreto. Na verdade, há um compromisso entre esses dois espaços visto que hipóteses de correspondência são validadas pela compatibilidade com uma transformação geométrica. Da mesma forma, os parâmetros de uma transformação geométrica são válidos apenas se da transformação que representam for possível gerar um conjunto de correspondências entre as feições. Métodos que utilizam esse fato se beneficiam da redução de dimensão do espaço de hipóteses e de dois conceitos importantes para lidar com esses espaços de grande dimensão: (1) o conceito de base e (2) o conceito de invariância.

Uma base consiste em uma tupla ordenada de tamanho mínimo de feições do modelo correspondidas a feições da imagem que é suficiente para determinar a transformação geométrica que mapeia as feições do modelo nas feições da imagem. O tamanho da base depende do tipo de transformação e do tipo de feição adotados. Com esse conceito em mente, a busca por correspondências se restringe às tuplas de feições da imagem do tamanho da base, pois uma vez determinados os parâmetros da transformação geométrica, as demais correspondências entre feições do modelo e da imagem ficam estabelecidas. Para bases de tamanho b, o número de possibilidades para correspondências é $O(I^b M^b)$, para um total de M feições no modelo e I feições na imagem.

O outro conceito importante que simplifica a análise do espaço de correspondências e do espaço de transformações é a invariância. Uma medida sobre tuplas de feições é considerada invariante a determinado tipo de transformação geométrica se esta medida permanece constante independentemente dos parâmetros da transformação geométrica que se aplicou às feições da tupla. A invariância de um atributo é definida para um determinado espaço de transformações. Nota-se que a invariância é um conceito muito útil para comparar objetos em sistemas de referência dos quais se desconhece a transformação que os relaciona.

É possível construir invariantes considerando grupos de $b+1$ feições do modelo, onde b representa o tamanho da base. Como observado, as primeiras b feições são suficientes para determinação da transformação e, conseqüentemente, a aplicação da transformação obtida à feição restante produz uma medida invariante aos parâmetros da transformação. Em outras palavras, as coordenadas das primeiras b feições são utilizadas para normalizar as coordenadas da última feição. Outros mecanismos menos genéricos para construir invariantes dependem do tipo de transformação e do tipo de feição envolvidos.

Como o modelo da cena é previamente conhecido, pode-se indexar os conjuntos de $b+1$ feições-modelo utilizando os atributos invariantes como chave numa fase *off-line*. Faz-se isso para todo possível arranjo de feições-modelo, ocupando espaço em memória da ordem $O\left(M^{b+1}\right)$. Durante a fase *on-line*, após extração das feições da imagem, para um grupo de $b+1$ feições-imagem computam-se os invariantes e se utilizam estes para encontrar possíveis arranjos de feições-modelo com os mesmos atributos. Sendo o custo computacional para

determinar os possíveis arranjos da ordem $O(H)$, o custo de processamento no pior caso, onde $O(I^{b+1})$ arranjos de feições-imagem são testados, é de $O(I^{b+1}H)$. Essa técnica é conhecida como reconhecimento ou correspondência por *hashing*.

Assim, observa-se do acima exposto que a redução da complexidade do processo de estabelecimento de correspondências decorre da determinação de uma classe de transformações geométricas que modele a variação da cena e um conjunto de atributos a se extrair das feições de forma a reduzir a complexidade da determinação de correspondências imagem-modelo.

A forma mais natural de modelar o alinhamento consiste na utilização de uma composição da transformação rígida em 3D com uma transformação de projeção perspectiva que mapeia pontos do espaço 3D em pontos de um plano. Entretanto, por ser a transformação resultante dessa composição uma transformação entre espaços diferentes, torna-se imprópria a determinação de invariantes. Assim, a modelagem é de fato baseada em casos particulares dessa transformação. Dois desses casos particulares tem maior importância: a transformação projetiva planar e a transformação rígida em 3D.

Uma transformação projetiva planar é um mapa de plano para plano que modela o efeito da perspectiva. Para utilizá-la, é assumido que o objeto apresenta forma plana. Essa transformação é não-linear e preserva poucas propriedades. Por exemplo, distâncias e ângulos sobre o plano não são preservadas e nem as relações de paralelismo. Além disso, as propriedades preservadas são um tanto complexas, como é o exemplo da razão cruzada. No caso de feições pontuais, a base que determina sem ambigüidades os parâmetros dessa transformação tem tamanho $b = 4$. No caso de curvas e regiões, há deformação, pela perspectiva, dificultando a análise pela forma.

Um outro caso particular é a transformação rígida em 3D, considerando, para isso, a transformação de projeção perspectiva conhecida. A classe das transformações rígidas em 3D é capaz de modelar o alinhamento e não apresenta os mesmos problemas encontrados com a transformação projetiva planar. É fácil obter invariantes a essa classe de transformações por estas serem lineares e preservarem medidas de distância e de ângulos. No caso de feições

pontuais, a base formada considerando essa classe de transformações tem tamanho $b = 3$. Assim, o número de possibilidades de correspondências é da ordem de $O\left(I^3 M^3\right)$. Para feições na forma de cônicas, a base tem tamanho $b = 2$.

Entretanto, no caso do alinhamento por visão, para utilizar a transformação rígida 3D, é necessário um mecanismo de aquisição de coordenadas tridimensionais dos objetos no sistema de referência de câmera, o que a princípio não é dado pelas imagens de intensidade. Porém, dado que se admite utilizar mais de uma câmera para adquirir imagens da cena, um método de visão estéreo pode ser utilizado para reconstruir um modelo tridimensional a partir de pares de imagens de intensidade. Por outro lado, é sabido que o problema de reconstrução estéreo depende da solução de um usualmente difícil problema de determinação de correspondências entre elementos das imagens direita e esquerda.

Dado que uma abordagem automática de alinhamento por visão pode ser construída escolhendo-se adequadamente a classe de transformações e a classe de feições de forma a se estabelecer um conjunto de invariantes para determinação de correspondências, a escolha pela transformação rígida em 3D é justificável desde que se possa reconstruir as feições em 3D. Além disso, deve-se complementar a abordagem com soluções para os demais problemas envolvidos, como a segmentação das feições, a descrição das feições e a estimação consolidada da pose. Feitas essas considerações sobre as diversas abordagens e tendo apresentado os conceitos utilizados para construção da solução, passamos a descrever nossa proposta.

1.4 Proposta e justificativa

O principal objetivo desta tese é a proposta de uma abordagem ao problema de alinhamento automático onde se procura atender os requisitos apresentados ao mesmo tempo que se evita repetir as deficiências das abordagens relacionadas no item 1.3. Nossa proposta consiste em considerar, como fonte de informação para o alinhamento, feições regionais, obtidas por um sistema binocular de aquisição de imagens com orientação relativa entre câmeras conhecida. Para a abordagem por nós proposta, é suposto desconhecimento de estados anteriores o que confere à abordagem um caráter global, em contraposição às abordagens incrementais locais.

A combinação do uso de visão estéreo e feições regionais, na forma que propomos, é sinergística por apresentar as seguintes vantagens:

- **correspondência imagem-modelo simplificada**

A consideração de uma etapa de reconstrução 3D permite-nos utilizar a transformação rígida como modelo da variação da cena sob movimento da câmera ou de objetos rígidos. Os parâmetros da transformação rígida são diretamente relacionados aos 6 graus de liberdade da pose. A adoção da transformação rígida permite também a escolha de invariantes mais simples resultando em bases de tamanho menor e em algoritmos de menor complexidade para determinação de correspondências.

Dado que a informação devida a cada feição regional estabelece pelo menos 5 graus de liberdade da pose, são consideradas bases de tamanho $b = 2$, de forma que a estimação de pose é mais simples do que no caso de pontos e pode ser realizada com a partir de 2 correspondências. Uma conseqüência adicional é que o número de hipóteses de correspondências é reduzida para $O(I^2 M^2)$.

Invariantes podem ser determinados para conjuntos de 2 feições, o que reduz a complexidade em tempo de processamento e espaço em memória do algoritmo de correspondência, que passa a ter como limite superior $O\left(I^2 H\right)$ para tempo de processamento e $O\left(M^2\right)$ para espaço em memória.

- **correspondência estéreo simplificada**

Para a determinação de correspondências entre feições das imagens direita e esquerda de um par estéreo, pode-se lançar mão da restrição epipolar, que garante que para um determinado ponto da imagem direita, o ponto correspondente da imagem esquerda pertence a uma reta (reta epipolar) que pode ser determinada a partir das coordenadas do ponto da imagem direita se for conhecida a orientação relativa entre as câmeras. Um problema surge devido a imprecisões das coordenadas dos pontos sobre a imagem e dos parâmetros que representam a orientação relativa, pois um ponto homólogo a outro ponto dado pode não pertencer exatamente à reta epipolar estimada. Outro problema é a ambigüidade, pois sem

informação adicional é impossível discernir uma correspondência correta dadas várias feições pontuais sobre a reta epipolar.

Ao contrário da correspondência estéreo de feições pontuais, a correspondência estéreo de regiões pode ser determinada de forma mais simples quando se supõe o conhecimento da orientação relativa entre câmeras. Para um par estéreo retificado, resulta da restrição epipolar que dois pontos homólogos possuem a mesma ordenada *y*. Embora a retificação não seja uma operação absolutamente necessária para a solução do problema, simplifica muito o seu tratamento, ao contrário de como é comumente apresentada, sua aplicação pode se restringir aos vértices dos polígonos extraídos das imagens ao invés de considerar todos pixels da imagem. Estendendo a idéia para o caso de regiões, duas regiões homólogas possuem pontos cujos pares de ordenadas *y* máxima e mínima são idênticos. A correspondência é realizada casando-se regiões homólogas que apresentem pares de valores próximos. Dado que se consideram restrições sobre dois atributos ao invés de sobre apenas um, como no caso de pontos, reduz-se o problema de ambigüidade, mesmo sob condições de imprecisão.

- **possibilidade de um tratamento independente da forma da feição**

A determinação de correspondências e de atributos invariantes pode ser feito sem a necessidade de considerar a forma da região. Isso ocorre tanto para o problema de determinação de correspondências imagem-modelo quanto para o problema de determinação de feições homólogas no par estéreo de imagens, o que nos leva às seguintes vantagens adicionais:

- Evita-se o problema da ambigüidade no caso de regiões com forma simétrica, que ocorre, por exemplo, para um círculo ou um triângulo eqüilátero, dos quais não se pode determinar a rotação no plano de forma única;

- É possível estabelecer um procedimento para análise das imagens que tolera segmentação de baixa qualidade visto que uma abordagem baseada na forma da região exigiria um processo de segmentação mais cuidadoso;

- Permite-se a substituição do contorno da região pelo contorno do fecho convexo para representar a feição, o que simplifica os processos de reconstrução 3D e descrição da feição por atributos.

♦ **simplificações adicionais decorrentes das condições adotadas**

As hipóteses assumidas permitem implementações mais simples e eficientes para a abordagem.

A utilização de um algoritmo de *scanlines* para reconstruir o contorno do fecho convexo da feição regional plana é adotada ao invés da reconstrução do contorno da própria feição. Embora determinada a correspondência entre regiões, a reconstrução das feições no espaço 3D ainda depende do estabelecimento de correspondência entre pontos da imagem. A determinação dessas correspondências entre pontos é simplificada considerando que o contorno do fecho convexo de uma região plana encontra uma linha reta em no máximo dois pontos. Sabendo-se a orientação relativa das câmeras e a correspondência de um ponto da imagem direita com um ponto da imagem esquerda, determina-se a posição de um ponto em 3D e se aplica esse procedimento para todos os pontos considerados do contorno do fecho convexo.

A utilização da técnica de correspondência por *hashing*, na qual se cria uma tabela para indexar as feições pelos seus atributos, simplifica ainda mais os algoritmos de determinação da correspondência e é aplicável tanto no caso da determinação de correspondências estéreo de feições regionais como também na determinação das correspondências entre as feições reconstruídas a partir das imagens e as feições do modelo conhecido.

A utilização das posições dos centróides dos fechos convexos reconstruídos em 3D das feições e das direções normais a seus planos como medidas de posição e orientação extraídas das feições das imagens é bastante simples por se basear no cálculo dos momentos de Hu. Essas medidas são utilizadas para o estabelecimento de correspondência entre feições-imagem e feições-modelo e para a estimação da pose. Invariantes à transformação rígida são obtidos para pares de feições, como a distância entre dois centróides ou o ângulo formado pelos planos de duas feições.

A utilização de um esquema de consenso por votação para determinação da correspondência entre feições da imagem e feições do modelo é uma forma adequada para tratar a informação proveniente das várias feições detectadas e modeladas. Considerando a multiplicidade de feições, há necessidade de levar em conta as informações devidas a cada feição presente para obter um algoritmo de determinação das correspondências mais robusto. No esquema de votação, de cada combinação de 2 feições-imagem se extraem atributos invariantes à transformação rígida e se atribuem votos às feições-modelo compatíveis com essas medidas de feições-imagem. Uma vez que se tenham histogramas de votos para cada feição-imagem, esses histogramas são analisados, associando cada feição-imagem àquela feição-modelo com mais votos. Uma medida de qualidade dessa correspondência também pode ser obtida na análise do histograma.

A utilização de um método de quadrados mínimos com poda é uma forma simples para consolidar uma estimação mais precisa e robusta da pose a partir de um número maior de correspondências, embora o estabelecimento das correspondências de 2 feições-imagem com as respectivas 2 feições-modelo já seja suficiente para se estimar a pose. Na prática, podem existir erros nas medidas obtidas das feições que podem ter sua influência no resultado final atenuada pela estimação utilizando maior quantidade de medidas. Apenas um número determinado das melhores correspondências, dadas suas medidas de qualidade, são utilizadas para estimação de pose. Assim, a ausência de uma feição devido à oclusão ou a presença de ruído na imagem têm menor influência no resultado final.

1.5 Organização da tese

Além deste capítulo, a tese conta com 5 outros capítulos e 6 apêndices. No capítulo 2, é revista a literatura a fim de mostrar o posicionamento do trabalho frente outras soluções propostas na literatura. No capítulo 3, são revistos os algoritmos e as técnicas utilizadas em Visão Computacional para estimação de pose, reconhecimento e reconstrução a partir de imagens estéreo. No capítulo 4, detalham-se a nossa proposta e os algoritmos utilizados para sua implementação. No capítulo 5, apresentam-se a metodologia de testes e os resultados obtidos. No capítulo 6, apresentam-se as conclusões e as sugestões para extensões do trabalho.

Material complementar, relacionado com temas considerados na solução ou implementação da abordagem proposta é apresentado nos apêndices A a F. O apêndice A é uma introdução ao raciocínio probabilístico utilizado no item 4.4. O apêndice B esclarece quanto às transformações geométricas discutidas na tese e descreve algoritmos para estimação da pose a partir de feições pontuais. O apêndice C descreve de forma rápida o paradigma RANSAC para estimação robusta. O apêndice D contém uma descrição compacta do método de *hashing* geométrico. O apêndice E apresenta as questões de navegação autônoma de veículos, interfaces homem-máquina modernas e realidade aumentada como aplicações recentes para o alinhamento 3D. O apêndice F mostra resultados da validação dos processos de calibração de câmeras e de retificação com fins mais ilustrativos.

Capítulo 2 - ABORDAGENS AO PROBLEMA DE ALINHAMENTO

Queremos neste capítulo contextualizar o trabalho realizado revisando soluções propostas para o problemas similares ao aqui tratado e que consiste no alinhamento geométrico da projeção de um modelo 3D com uma imagem. Abordagens para o problema tratado nesta tese podem utilizar as idéias de reconhecimento de objetos para realizar a correspondência entre feições da imagem e feições do modelo automaticamente. Também as abordagens ao problema de rastreio, que consiste em manter as correspondências ao longo de uma seqüência temporal de imagens, devem ser levadas em conta visto que o alinhamento para seqüências temporais de imagens de intensidade é um dos nossos objetivos.

2.1 Reconhecimento e rastreio

Muitos dos métodos apresentados na revisão bibliográfica do item 2.2 são encontrados como abordagens ao problema de reconhecimento de objetos. Esse problema difere do problema de alinhamento pelo fato de que, no caso do reconhecimento, tem-se uma base de dados de modelos de muitos objetos conhecidos e se deseja explicar a imagem através da determinação de uma hipótese que consiste no posicionamento de projeções desses modelos sobre a imagem. No problema de alinhamento, há apenas um modelo ou um número muito reduzido de modelos, e o modelo é representado parcialmente na imagem.

Nas principais aplicações do reconhecimento, é necessário analisar um número grande de imagens. Assim, nestes casos, as soluções para o problema de reconhecimento precisam ter custo computacional reduzido. Conceitos encontrados nas abordagens ao problema de reconhecimento, como os conceitos de base e invariância mencionados no capítulo 1, podem ser reutilizados para o problema da determinação automática de correspondências entre feições da imagem e do modelo.

O outro problema para o qual citamos algumas abordagens no item 2.2 é o rastreio (*tracking*). Quando o alinhamento deve ser mantido ao longo de uma seqüência de imagens, temos um problema de rastreio. As feições da imagem devem ser perseguidas ao longo da seqüência a fim de se manter as correspondências com feições do modelo.

O termo rastreio não é preciso, sendo utilizado em Visão Computacional como denominação para mais de um problema que envolve seqüências de imagens. Dentre esses problemas, mencionamos os exemplos seguintes.

- Em aplicações com um sistema de câmeras observando uma mesma cena onde é desejado obter a orientação relativa entre as câmeras a partir de seqüências de imagens, o rastreio é um problema de auto-calibração.

- No caso de rastreio de regiões ou contornos como proposto por Blake e Isard [1998], há outro tipo de rastreio, em que se deseja manter identificada ao longo do tempo uma estrutura presente na seqüência de imagens. Em geral, os contornos são perseguidos sobre a imagem considerando movimento não-rígido em 2D e desconsiderando a estrutura 3D.

- O rastreio de movimento de câmera (*egomotion*) por fluxo óptico é um outro possível caso, veja por exemplo [Horn, 1989]. Neste caso, o rastreio corresponde à estimação da trajetória da câmera.

- Rastreio de planos por textura, às vezes chamado rastreio de regiões, considera a diferença entre a imagem observada e a textura projetada com os parâmetros

estimados, permitindo corrigir os parâmetros posteriores. Este caso de rastreio é encontrado, por exemplo, em [Hager e Belhumeur, 1998].

- O rastreio também pode consistir na busca de uma região de interesse (ROI). Neste caso, deve-se perseguir um objeto apenas se delimitando um retângulo em cada quadro onde o objeto é visível.

Se for considerado que há pouca variação entre os quadros consecutivos da seqüência de imagens, é possível adotar uma abordagem diferencial ou incremental ao rastreio, de forma que as feições no quadro seguinte sejam procuradas apenas em posições da imagem próximas às posições que ocupavam no quadro corrente. Por esta razão, este tipo de abordagem é considerada local, não sendo necessário percorrer a imagem por inteiro a fim de encontrar as feições.

O uso de algoritmos incrementais de rastreio é uma abordagem apropriada para resolver os problemas de tempo, todavia tais algoritmos podem ser sensíveis aos seguintes fenômenos. Em seqüências de imagens com movimento abrupto, o algoritmo de rastreio pode prever erroneamente a posição futura do objeto rastreado produzindo como efeito variações ainda mais abruptas, o que pode ser chamado *jitter*. O movimento abrupto também pode levar a uma perda da seqüência do rastreio, devendo-se reiniciá-lo. Em longas seqüências de imagens, algoritmos incrementais podem acumular o erro resultante de cada quadro, podendo haver um deslocamento notável entre o objeto real e a estimação de sua localização no final da seqüência, este é o problema de *drift*. Para o caso específico de imagens, ainda há problemas de detecção incorreta de feições, que pode acontecer de duas formas. Feições podem ser totalmente ou parcialmente obstruídas por um objeto opaco ou estarem fora do campo de visão, não sendo identificadas na imagem. As feições podem deixar de ser localizadas também por variação da iluminação. Além disso, há o problema de *clutter* em imagens muito poluídas ou cenas muito complexas onde objetos que não são de interesse são confundidos com os objetos procurados. Uma forma de atenuar esse problema é a fusão de dados com outros sensores, discutida no item 2.2. Outras formas, que discutimos a seguir, são o rastreio preditivo através de filtragem e o rastreio global.

Técnicas de rastreio preditivo são utilizadas para tornar mais eficiente a análise dos dados sensoriais em cada quadro, determinando, de forma mais comprometida com um modelo da dinâmica da cena, as janelas da imagem para se realizar a busca por feições. Blake e Isard [1998] perseguem os pontos do contorno de uma região da imagem considerando as janelas de busca estimadas conforme um espaço de formas e uma dinâmica associada à deformação do contorno. Pode não ser necessário buscar feições por todo o suporte da imagem, mas apenas nessas janelas, o que torna o processamento mais eficiente, podendo ser utilizado em aplicações de tempo real. Além de auxiliar a obtenção de um melhor *throughput*, a predição é um mecanismo importante para se reduzir o problema da latência, sendo possível se ter um resultado estimado simultaneamente com a leitura dos sensores. A leitura dos sensores é então utilizada para corrigir o estado atual e predizer o estado nos quadros seguintes.

O rastreio global analisa a imagem como um todo e não depende unicamente de variações, sendo o principal objetivo resolver o problema de correspondência automaticamente. Com isso, os métodos globais são mais robustos, podendo lidar com problemas de *clutter*, oclusão, variações abruptas e variações de iluminação. Ao contrário dos métodos locais, métodos globais consideram múltiplas hipóteses para estimar o melhor conjunto de correspondências entre feições da imagem e do modelo geométrico.

O rastreio por métodos globais também pode ser incremental, de forma que a proximidade do estado anterior influencie a decisão pelo melhor conjunto de correspondências. Dessa forma, é possível construir um método global de rastreio com predição de estado. É também possível desenvolver métodos que sejam um compromisso entre o global e o local, equilibrando robustez e custo computacional, de forma que o número de hipóteses consideradas é limitado pela localidade.

É importante perceber também que o rastreio incremental é dependente do conhecimento do estado inicial, de forma que um algoritmo incremental não é capaz de resolver o problema por si só, sendo necessária uma técnica complementar. Exemplos dessas técnicas são as abordagens semi-automática com alguma intervenção do usuário, o rastreio híbrido por fusão de dados e o rastreio global.

2.2 Revisão de abordagens

Uma vez discutida a relação entre o alinhamento e os problemas de rastreio e de reconhecimento de objetos, passamos à revisão bibliográfica, onde esses problemas são freqüentemente mencionados. As abordagens revistas, apresentadas a seguir, estão agrupadas nos seguintes ítens: (1) atributos invariantes, (2) feições invariantes, (3) aparência, (4) espaço de poses, (5) estimação e (6) outras abordagens, incluindo aproximação por perspectiva fraca, marcação artificial da cena, fusão de dados sensoriais e o problema de correspondência de formas.

2.2.1 Abordagens por atributos invariantes

Muitas abordagens para o reconhecimento de objetos são baseadas em medidas invariantes, isto é, conjuntos de atributos independentes dos parâmetros das transformações entre a geometria conhecida do objeto e a geometria observada. Algumas abordagens utilizam invariantes, mas não explicitamente, através de alguma construção geométrica que resulte em invariantes, como a transformação para um referencial canônico definido pela estrutura das feições. Uma vez que se tenham invariantes, uma tabela de espalhamento (*hash*) é uma forma de implementar reconhecimento rápido. A tabela é indexada pelos invariantes e preenchida *off-line* com os atributos dos modelos dos objetos conhecidos. Na fase *on-line*, um objeto observado que possua o mesmo conjunto de atributos de um objeto conhecido vai colidir na mesma posição da tabela. Veja a figura 2-1. Não necessariamente existe uma função de *hash* para computar os índices e, como são esperadas colisões, cada célula de uma tabela destas contém uma lista de itens.

Forsyth *et al*. [1991] investigam a teoria de invariantes aplicada ao reconhecimento de objetos, introduzindo vários exemplos de invariantes. A seguinte definição de invariante é apresentada: um invariante $I(\mathbf{p})$ de uma função $f(\mathbf{x},\mathbf{p})$ sujeito ao grupo $\mathcal{G}$ de transformações nas coordenadas $\mathbf{x}$ é transformado de acordo com $I(\mathbf{q}) = I(\mathbf{p})h(g)$, onde $g \in \mathcal{G}$ e $h(g)$ é função apenas dos parâmetros da transformação, sendo independente de $\mathbf{x}$ e de $\mathbf{p}$. $I(\mathbf{p})$ depende apenas dos parâmetros $\mathbf{p}$. Para $I(\mathbf{q}) = I(\mathbf{p})$, diz-se de $I(\mathbf{p})$ um

invariante escalar. O termo “invariante escalar” normalmente é encontrado apenas como “invariante”.

Um grupo, segundo Forsyth *et al.* [1991] é uma coleção de transformações que podem ser compostas e invertidas. Grupos modelam adequadamente movimento rígido e também as transformações projetivas planares.

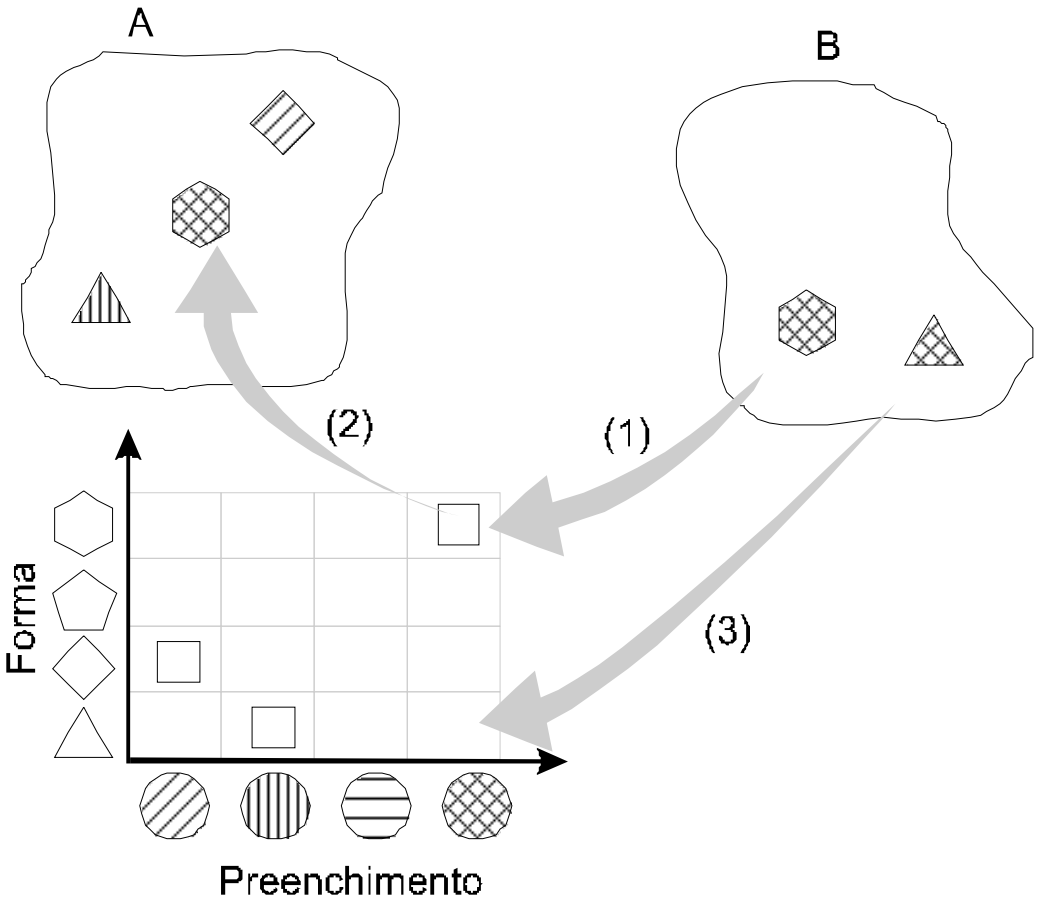


Figura 2-1 - Correspondência por tabela *hash*.

A tabela foi construída para o conjunto A, considerando os atributos de forma e preenchimento como índices. O elemento correspondente a cada elemento do conjunto B é obtido através de um único acesso à tabela para os índices definidos pelos atributos. Em (1) o objeto hexagonal com preenchimento cruzado é procurado na tabela e se encontra a referência em (2) para o objeto correspondente do conjunto A. Em (3), é procurado um objeto triangular com preenchimento cruzado, mas não é encontrado na tabela porque não há elemento correspondente no conjunto A.

Invariantes à transformação projetiva podem ser construídos a partir de 5 pontos. Uma das formas de fazê-lo é utilizando 4 pontos para fixar um referencial projetivo, sendo as coordenadas do quinto ponto, transformado para esse novo referencial, invariantes [Forsyth *et al.*, 1991].

Outra possível construção é baseada na razão cruzada. A razão cruzada é um exemplo clássico e útil de invariância à projeção. Consiste em uma razão entre os comprimentos de segmentos de retas definidos pelos pontos de intersecção de uma reta com um feixe de retas como na equação (2-1)

$$\rho = \frac{AC \cdot BD}{BC \cdot AD}. \qquad \textbf{(2-1)}$$

Os pontos A, B, C, D são definidos na figura 2-2. O valor de ρ não se altera se substituirmos esses pontos A, B, C, D por A', B', C', D' respectivamente. Seis valores invariantes a projeção perspectiva planar podem ser obtidos permutando-se os nomes desses pontos e aplicando a equação (2-1).

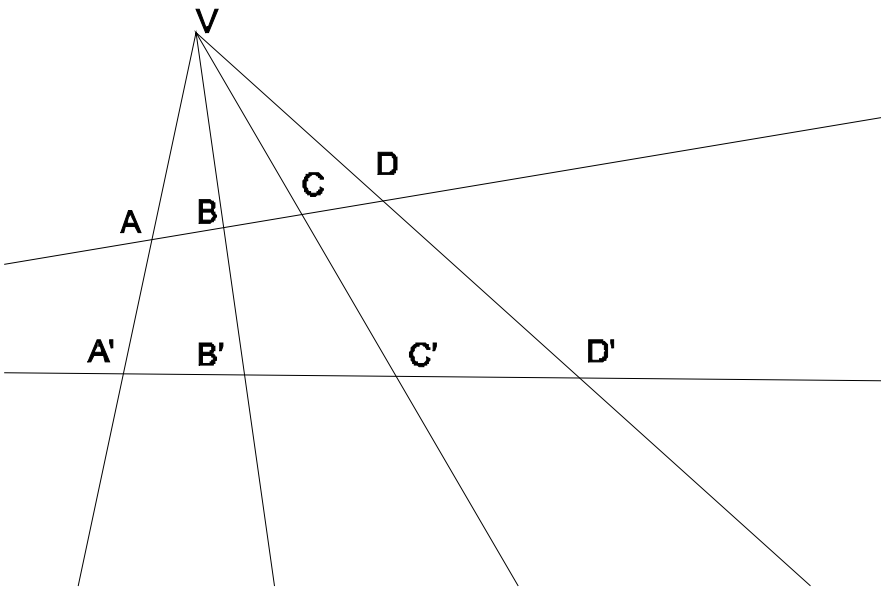
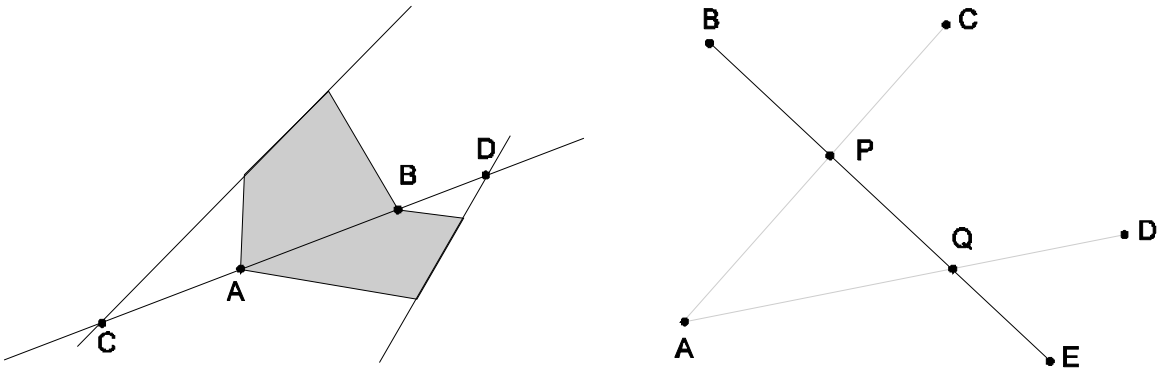


Figura 2-2 – Razão cruzada.

Apresentamos na figura 2-3 dois exemplos de esquemas para construção de razões cruzadas. Mundy [1994] mostra como obter medidas invariantes baseadas na razão cruzada para 6 pontos não colineares (figura 2-3a) através do chamado esquema de “borboleta” que é obtido estendendo os lados opostos do polígono de 6 lados e determinando a intersecção com a reta que passa pelos vértices restantes. Reiss [1993] mostra um esquema para 5 pontos (figura 2-3b).



(a) (b)

Figura 2-3 – Esquemas para obter invariantes projetivos a partir de pontos não colineares.

Forsyth *et al*. [1991] exploram os invariantes para par de cônicas co-planares. Uma forma de construir invariante para pares de cônicas consiste em considerar 4 pontos de intersecção entre as duas cônicas e construir uma razão cruzada utilizando um outro ponto de uma das cônicas. Outro invariante é também obtido utilizando um outro ponto da outra cônica. A razão cruzada é independente do ponto escolhido da cônica.

Uma outra forma de construir invariantes para par de cônicas explorada em [Forsyth *et al.*, 1991] considera a forma algébrica para a cônica $\mathbf{c}$ dada por $\mathbf{x}^T\mathbf{c}\mathbf{x} = 0$, onde $\mathbf{x} = [x\ y\ 1]^T$ e $\mathbf{c}$ é matriz 3×3 simétrica de norma unitária. Os invariantes são dados pela equação (2-2). A transformação projetiva de $\mathbf{c}_1$ e $\mathbf{c}_2$ corresponde a uma transformação de similaridade de $\mathbf{c}_1^{-1}\mathbf{c}_2$, que preserva os autovalores e, conseqüentemente, o traço (soma dos valores da diagonal).

$$I_{12} = \text{traço}\left(\mathbf{c}_1^{-1}\mathbf{c}_2\right) \qquad \textbf{(2-2)}$$
$$I_{21} = \text{traço}\left(\mathbf{c}_2^{-1}\mathbf{c}_1\right)$$

As medidas sobre as curvas têm o mesmo valor independentemente dos parâmetros da transformação projetiva.

Forsyth *et al*. [1991] apresentam uma solução para reconhecimento de objetos 3D baseados em feições na forma de círculos. A partir de pares de círculos deformados pela perspectiva, os seguintes invariantes são medidos: os ângulos entre seus planos em 3D, a razão do raio do círculo em relação à distância entre os centros dos círculos e o vetor unindo o centro dos círculos. Desse vetor, três medidas são obtidas: seu comprimento e os ângulos em relação às normais dos planos do primeiro e do segundo círculo. O processo de reconhecimento e estimação de pose segue as etapas de extração dos contornos das imagens, encadeamento das curvas, ajuste de cônica (e descarte de curvas que não se aproximam de cônicas), cálculo dos descritores de forma, e determinação de correspondências com o modelo utilizando uma tabela *hash*. Os autores ainda propõem extensão para curvas algébricas planas através da aproximação por cônicas.

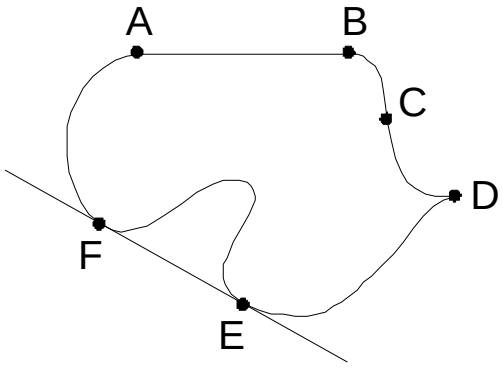


Figura 2-4 – Feições invariantes projetivas de uma curva.

A e B são extremidades de um segmento de reta, C é um ponto de inflexão, D é um vértice ou uma cúspide e F e E são pontos bitangentes.

2.2.2 Abordagens por feições invariantes

Algumas características locais e globais de contornos são invariantes projetivos (invariantes à projeção perspectiva planar) e, da mesma forma, os elementos geométricos correspondentes formam feições invariantes do contorno, que podem ser detectadas tanto no modelo como na imagem observada independente da transformação. Reiss [1993] discute diversos tipos de feições do contorno que podem ser detectadas de forma invariante à transformação projetiva planar (figura 2-4). Segmentos do contorno de curvatura nula podem ter seu início e fim detectados. Pontos isolados sem curvatura são pontos de inflexão, essa propriedade é invariante à transformação projetiva, uma vez que o ponto transformado é também um ponto de inflexão na curva transformada. Vértices e cúspides, pontos em que não se define a direção tangente à curva são também detectáveis e mantêm suas propriedades. Pontos em que a curva se "desprende" de seu fecho convexo são chamados pontos bitangentes. O fecho convexo de uma região transformada por projeção perspectiva planar equivale à transformação projetiva do fecho convexo da região original. Assim, esses pontos são preservados. Existem outras variações de pontos bitangentes segundo Reiss [1993] que podem ser vistas na figura 2-5.

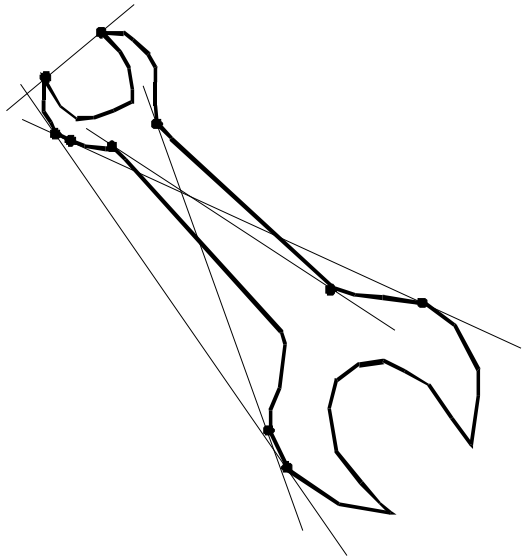


Figura 2-5 – Variações de pontos bitangentes.

Na figura se observam pontos bitangentes externo-externo, interno-interno e externo-interno como discutido em [Reiss, 1993].

Rothwell *et al*.[1995] propõem a construção de uma função de índices que, a partir de um conjunto de feições da imagem, retorna o índice para a feição do modelo que melhor corresponde a cada feição da imagem. Essa função deve ser construída invariante à transformação. A obtenção de invariantes proposta por Rothwell *et al*.[1995] considera a transformação para um referencial canônico. Assim, se algumas feições formam um referencial, medidas sobre as demais feições transformadas para esse referencial são invariantes à transformação. A fim de determinar um conjunto de atributos para curvas planares que possam ser mapeados sobre um referencial canônico, foram considerados pontos bitangentes e pontos dentro da concavidade que são tangentes a retas que partem dos bitangentes (figura 2-6). Uma vez obtidas as medidas invariantes para curvas do modelo, a função de índice é construída através do treinamento de um discriminador linear.

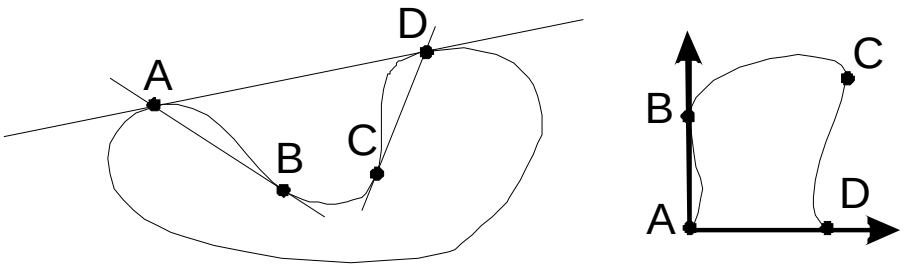


Figura 2-6 – Normalização de uma concavidade.

Os pontos A e D são bitangentes e os pontos B e C são intersecções da curva com retas tangentes passando por A e D respectivamente. Os pontos A, B e D da curva são mapeados para um referencial canônico (à direita) de forma que as coordenadas do ponto C mapeado constituem medidas invariantes à transformação projetiva.

As técnicas baseadas em invariantes e feições invariantes têm bastante sucesso desde que se possam extrair das feições a informação específica para a construção de invariantes ao grupo de transformações considerado.

2.2.3 Abordagens por aparência

Métodos baseados em aparência utilizam as informações da vizinhança dos pixels para comparar a imagem observada com o modelo. Em geral, deve-se partir de um conjunto de

vistas do objeto procurado para se construir o modelo. As correspondências estabelecidas podem conter mais informação que uma correspondência entre feições pontuais e o número de feições extraído pode ser superior ao caso de detectores de cantos.

Lowe [1999] propõe as chaves SIFT (*scale-invariant feature transform*) que são vetores característicos de localidades da imagem, que podem ser detectados de modo invariante às transformações de translação, escala e rotação, a variações de iluminação e às transformações afim e projetiva. As chaves SIFT são consideradas melhores do que detectores de cantos uma vez que contém um número maior de informações e sua obtenção considera a escala da imagem. As chaves SIFT são construídas num processo de multi-escala, tirando proveito do processo da construção da pirâmide com várias escalas da imagem. Na construção de cada novo nível da pirâmide é aplicada uma filtragem com um núcleo gaussiano. Para obter as chaves SIFT, faz-se a diferença entre cada dois níveis filtrados consecutivos e localizam-se os máximos e mínimos locais considerando 3×3 pixels. Assim, são obtidas as posições das chaves SIFT e suas escalas, outros atributos são a orientação e a magnitude do gradiente, que são computadas localmente por diferenças de pixels.

A discriminação das chaves pode ser realizada por indexação. Lowe [1999] considera a indexação por *k-d tree* ou pelo espaço de Hough utilizando uma tabela *hash* com as seguintes dimensões: *x*, *y*, ângulo, escala (para transformações de similaridade). Devido a alta dimensionalidade, é necessário recorrer a métodos de indexação baseado em *k-d trees* [Beis e Lowe, 1997] ou então alargar o tamanho de cada célula da tabela *hash*. É considerado também que a inserção na tabela *hash* não deve ser pontual, mas deve se espalhar pelas células vizinhas, o que é um complicador no caso de dimensões mais altas.

Outros problemas são o número excessivo de chaves que podem resultar de algumas imagens e a dificuldade com imagens muito deformadas pela projeção perspectiva. Esses problemas são tratados pelo SIFT estéreo, que possibilita a eliminação de chaves quando não houver consenso entre as vistas. Se *et al.* [2002] consideram um sistema trinocular de câmeras.

Se *et al.* [2002] aplicam as chaves SIFT estéreo ao problema de navegação robótica, considerando apenas 3 graus de liberdade do robô: *x*, *z* e θ. Além disso, dados de odometria

do robô são utilizados para simplificar a correspondência das chaves. Dois métodos de determinação da correspondência são considerados, um baseado em acumulação no espaço de Hough e outro baseado em estimação utilizando o paradigma RANSAC (*random sample consensus* [Fischler e Bolles, 1981]).

No método baseado em espaços de Hough, para cada hipótese de correspondência entre uma chave SIFT da imagem e uma marca SIFT do modelo, é acumulado um voto em cada célula que represente uma pose compatível. Como uma só correspondência não é suficiente para determinar a pose, faz-se então uma votação através da seleção das células mais votadas e a medida da pose é refinada por quadrados mínimos considerando poda.

O método baseado no RANSAC consiste em, a cada iteração, selecionar 2 correspondências aleatórias, determinar o alinhamento a partir delas e testar as demais correspondências quanto à compatibilidade com esse alinhamento. Para o melhor alinhamento obtido (isto é, a hipótese com mais suporte), refina-se a medida da pose utilizando quadrados mínimos.

Em [Se *et al.* 2001], os autores relatam obter a localização do robô em 2,5 segundos num processador Pentium de 700MHZ. Eles justificam que o método só é aplicado esporadicamente para localização global inicial e que se deve proceder com correção incremental rápida. Em [Se *et al.*, 2002] além da técnica baseada em Hough, é também avaliada a técnica baseada no paradigma RANSAC. Para três graus de liberdade são reportados resultados de 0.725 segundos para o método baseado em Hough e de 0.021 segundos a 0.296 segundos para o método baseado em RANSAC dependendo do nível de contaminação das amostras.

Uenohara e Kanade [1995] propõem um método de rastreio incremental em que pontos característicos são detectados pela aparência, isto é, a partir do reconhecimento pela análise de suas vizinhanças na imagem do objeto. Devem ser conhecidas imagens de diversas vistas do objeto e sob diferentes condições de iluminação. A construção do modelo para o reconhecimento é feita por um método de auto-espaços. Variações das imagens das feições são representadas como combinações lineares de auto-vetores, no caso chamados auto-imagens.

Através de correlações efetuadas em paralelo por processadores de sinais (DSP), essa representação paramétrica pode ser determinada para regiões da imagem, encontrando as feições de forma robusta.

Uenohara e Kanade [1995] consideram apenas o caso incremental para execução em tempo-real com o objetivo de sobrepor imagens sintéticas a imagens reais de forma alinhada. Para iniciar o rastreio, o usuário deve posicionar interativamente o objeto ou a câmera de forma que a imagem em *wire-frame* sobreposta à imagem da câmera esteja bem próxima da geometria do objeto e o algoritmo incremental de rastreio tenha condições de efetuar a primeira correção da pose, que é feita com um algoritmo de otimização, mais lento que o algoritmo utilizado na fase de rastreio. Isto revela a necessidade de métodos globais de alinhamento eficientes.

Outro exemplo de rastreio incremental baseado em aparência é encontrado em [Hager e Belhumeur, 1998], onde uma imagem é rastreada com determinação de pose utilizando um padrão de características locais da superfície dado por uma amostragem de pixels. O algoritmo de Hager e Belhumeur permite modelar a deformação da superfície de forma paramétrica e corrigi-la ao longo de uma seqüência de imagens. Variações das imagens em função de cada parâmetro são armazenadas de forma a se conseguir um algoritmo rápido.

Peters [2000] apresenta um método inspirado biologicamente para reconhecimento e estimação de pose dado um conjunto esparso de vistas utilizando *wavelets* de Gabor. A inspiração biológica é defendida pelo seguinte fato: sabe-se que é fácil distinguir um triângulo apenas a partir de uma descrição em linguagem natural, mas esse tipo de descrição é incapaz de ensinar alguém a reconhecer o rosto de uma pessoa específica.

2.2.4 Abordagens pelo espaço de poses

Feições podem muitas vezes ser reconhecidas individualmente por seus atributos. Quando isso não é possível, é necessário, para identificá-las, utilizar a estrutura formada pelo relacionamento entre feições. Uma possível estrutura é dada pela restrição de rigidez, pela qual distâncias e ângulos se preservam e a pose do objeto é o conjunto de parâmetros que determina a transformação rígida que alinha feições observadas com feições correspondentes do modelo.

O espaço das poses pode ser explorado através da determinação de um consenso das relações entre feições. Métodos que realizam uma votação no espaço das possíveis poses são chamados métodos de *pose clustering*, isto é aglomeração de poses. O método de Haustler e Ritter [1999] ilustra esse tipo de abordagem. É proposto determinar a pose com seis graus de liberdade a partir de segmentos de reta como feições. Para cada par de pontos da imagem e para cada par de pontos do modelo (ligados por uma aresta), são acumulados votos numa tabela de Hough para poses. Devido a alta dimensão, usa-se uma projeção 2D do espaço das poses que considera dois ângulos de Euler para a orientação. Cada hipótese de correspondência de pares de pontos desenha uma trajetória nesse plano parametrizada pelo terceiro ângulo que determina a orientação. As células da tabela que se encontram sob essas trajetória recebem votos. As células mais votadas correspondem a pontos de aglomeração que suportam as melhores hipóteses e a pose pode ser estimada pelos índices dessas células no espaço de Hough.

Outros métodos que utilizam uma estratégia de consenso de hipóteses compatíveis através de aglomeração são o método de Olson [1994] e o método de espalhamento geométrico (*hashing*) de Wolfson e Rigoutsos [1997]. Esses métodos são discutidos no item 3.5.

Outra forma de estimar a pose simultaneamente com o reconhecimento é através da subdivisão do espaço de poses. Esse tipo de abordagem descarta intervalos (retângulos ou caixas) do espaço de poses em que não há hipóteses suficientemente fortes para a pose, repetindo o processo até que se tenha um intervalo de tamanho pequeno de onde se extrai uma estimação pontual. Jurie [1998] propõe um método de reconhecimento que pode ser utilizado para rastrear objetos 3D já modelados e em cenas com *clutter*. O método pode ser aplicado para solução do problema global, para o qual é relatado um resultado de reconhecimento robusto em 10 a 30 segundos. Foi utilizado um modelo baseado em arestas e um *framework* probabilístico para as correspondências. É proposto como técnica de rastreio uma vez que um método de predição pode delimitar uma caixa inicial no espaço das poses para busca baseando-se no intervalo da pose estimado para o quadro anterior e na dinâmica de movimentação do objeto.

2.2.5 Abordagens por estimação

Uma outra maneira de realizar estimação de pose em conjunto com o reconhecimento é através de métodos de estimação de parâmetros. Destacam-se os métodos de filtragem, que podem ser aplicados tanto a seqüências de imagens, como a múltiplas fontes de dados. A dimensão do conjunto de parâmetros a estimar não é tão limitadora como no caso da exploração do espaço de poses, podendo haver de estimação de mais parâmetros além da pose. Outra vantagem das técnicas baseadas em filtragem é o suporte a feições não registradas previamente.

Koller [1993] propõe um método para reconhecer, rastrear e classificar carros de uma seqüência de imagens, estimando parâmetros de um modelo. Diversas condições simplificadoras são consideradas: a câmera é fixa e calibrada, portanto a imagem de fundo também é fixa, a pose do carro é considerada sobre o plano com 2 graus de liberdade e é assumido um modelo de movimento do carro tratando como ruído qualquer variação da trajetória. A estimação de 12 parâmetros da dimensão e forma do carro é realizada utilizando um protótipo genérico para o modelo geométrico de carros e um modelo da sombra projetada do carro. A perseguição é feita com um filtro de Kalman estendido iterado. O sistema implementado ainda classifica carros segundo as categorias *sedan*, *hatchback*, *station wagon*, *mini van* e *pick-up*.

Enumeramos a seguir alguns métodos baseados em estimação aplicados a seqüências de imagens que podem aproveitar feições naturais do sistema não registradas previamente no modelo. Uma vez que se estabeleça a geometria das feições não declaradas, elas podem ser registradas no modelo e empregadas para estimar a pose com maior precisão e robustez.

Neumann e Park [1998] utilizam discos coloridos na forma de círculos e triângulos de seis cores diferentes para criar doze tipos diferentes de marcas. As marcas conhecidas são rastreadas de forma incremental utilizando um filtro de Kalman estendido. O método descrito por Neumann e Park permite ainda a adição de novas marcas fiduciais durante a execução do rastreio, registrando a posição de novas marcas encontradas conforme se estima o referencial geométrico da cena.

Kanbara *et al*. [2002] propõem iniciar o sistema com um par estéreo de câmeras observando um objeto calibrador com marcas fiduciais. A partir daí marcas naturais podem ser capturadas ao longo da seqüência de imagens e incorporadas ao modelo para manter o alinhamento. Um sensor inercial ainda é utilizado para melhorar a robustez do sistema.

Simon e Berger [1997] utilizam curvas tridimensionais, pontos e retas para alinhamento. O sistema é inicializado com parâmetros de câmera conhecidos e o usuário deve apontar a correspondência de quatro pontos da imagem com o modelo 3D e as marcas naturais encontradas no quadro inicial são utilizadas no processo incremental de alinhamento. Utilizando estimação robusta da pose, o método é capaz de compensar erros de rastreio das feições. Conforme feições se tornam visíveis ou oclusas ao longo da seqüência de imagens, o método proposto por Simon e Berger atualiza a lista de feições a rastrear.

Alguns métodos para processamento *off-line* de vídeo, mas que se beneficiam da continuidade de seqüências para processar rapidamente muitas imagens, realizam o rastreio por técnicas de *structure-from-motion*, que é um problema mais amplo que o problema de alinhamento e inclui reconstrução tridimensional, necessitando de soluções ainda mais sofisticadas. Este problema consiste em, a partir de um vídeo obtido sem calibração de câmera capturando imagens de uma cena, reconstruir simultaneamente a estrutura tridimensional da cena e a trajetória da câmera. Cornelis *et al*. [2000] propõem criar realidade aumentada *off-line*, detectando quinas. Pelos pontos detectados e a aplicação do paradigma RANSAC, as vistas são auto-calibradas. Utilizando a restrição epipolar em duas vistas razoavelmente distantes, estabelecem uma única referência tridimensional para todo o segmento do vídeo. Os parâmetros intrínsecos da câmera são fatorados por serem considerados constantes e a trajetória da câmera é determinada. Uma estrutura para a cena é, então, gerada.

Zisserman *et al.* [2000] resolvem o problema de criar uma referência em vídeo para o caso não calibrado através de uma rápida intervenção do usuário, que deve selecionar uma região plana da imagem. Utilizam o detector de cantos de Haris e o paradigma RANSAC para selecionar os cantos que representam a região plana ao longo da seqüência de imagens.

A estimação por métodos locais é limitada por testar um número muito pequeno de hipóteses. Para se construir sistemas mais robustos, capazes de lidar com quebras de contexto ou movimentos abruptos em seqüências de imagens, com cenas muito simétricas que geram ambigüidades e com o *clutter*, são necessários métodos globais ou de múltiplas hipóteses. Uma enumeração de possíveis abordagens é apresentada em [Fox *et al*.,1999].

Fox *et al.* [1999] utilizam filtros de partículas para resolver o problema de localização global para um robô com 3 graus de liberdade que conhece a estrutura do ambiente em que deve-se locomover. O espaço de pose é amostrado e a probabilidade condicional da pose é atualizada em função das observações. Esse método espera o movimento do robô e as observações em novas posições para resolver as ambigüidades e localizar globalmente o robô. Filtros de partículas são utilizados também por Isard e Blake [1998] através do algoritmo *Condensation* no contexto de rastreio por contornos ativos só que não para o problema global e sim incremental sob presença de *clutter*.

2.2.6 Outras abordagens

Há ainda muitas formas de abordagens para o problema de reconhecimento e rastreio. Koller [1993] enumera métodos mais antigos, especialmente baseados em representações poliédricas, para estimar pose e reconhecer objetos numa imagem dado o modelo 3D para objetos rígidos.

Discutimos, a seguir, algumas outras abordagens ao problema de alinhamento com considerações adicionais. (1) O caso da perspectiva fraca é uma simplificação do problema aproximando por sistema linear. (2) O método de marcação artificial assume uma codificação das feições para determinar as correspondências de forma imediata. (3) A fusão sensorial complementa o alinhamento com dados de outras formas sensoriais. (4) A abordagem por correspondência de curvas considera a forma de curvas do modelo e da imagem.

- **Perspectiva fraca**

Para escapar das dificuldades de se trabalhar com a transformação projetiva, muitas abordagens baseiam-se numa aproximação por perspectiva fraca. A perspectiva fraca é um modelo de aproximação por transformações lineares para lidar com a deformação devido à

projeção perspectiva. Uma outra alternativa poderia ser, por exemplo, a aproximação por similaridades euclidianas.

O modelo de perspectiva com centro de projeção na origem e eixo óptico sobre o eixo Z é dado pela equação (2-3)

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix}, \qquad \textbf{(2-3)}$$

onde *x*, *y* são coordenadas de imagem, *f* é a distância focal e *X*, *Y* e *Z* coordenadas 3D.

Utilizando a aproximação pela série de Taylor dada por (2-4),

$$\frac{1}{z} \approx \frac{1}{z_0} - \frac{(z - z_0)}{{z_0}^2} \qquad \textbf{(2-4)}$$

é obtido o modelo de perspectiva fraca dado pela equação (2-5)

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z_C}\left(\begin{bmatrix} X_C \\ Y_C \end{bmatrix} + \begin{bmatrix} X \\ Y \end{bmatrix} - \frac{Z}{Z_C}\begin{bmatrix} X_C \\ Y_C \end{bmatrix}\right), \qquad \textbf{(2-5)}$$

onde X_C, Y_C, Z_C são as coordenadas de um ponto próximo ao objeto. Essa formulação foi extraída de [Blake e Isard, 1998], que dizem ser uma boa aproximação para até 1 radiano de campo de visão.

A aproximação por perspectiva fraca é utilizada, por exemplo, em [Kutulakos e Vallino, 1995]. É proposto por esses autores um sistema que não faz uso da calibração, tendo como característica ser capaz de alinhar objetos reais e virtuais sobre a imagem ao invés do espaço 3D, evitando erros do processo de calibração e de estimação de pose. Kutulakos e Vallino [1995] utilizam marcas fiduciais que consistem em pontos coloridos e regiões poligonais. O sistema persegue 4 pontos fiduciais não co-planares que definem um sistema de referência afim. É considerado que este rastreio ocorre em tempo real. O problema dessa abordagem é não ser capaz de tratar a estrutura tridimensional do mundo real e que objetos virtuais devem ser representados num sistema de referência afim. Outras limitações relatadas pelos autores é

que o sistema confia apenas no processo de rastreio de pontos no vídeo e gera um atraso de 4 a 5 quadros numa implementação que roda a 30 quadros por segundo.

- **Marcação artificial**

Marcas artificiais são comuns para a identificação rápida das feições. No *Illuminating Light* [Underkoffler e Ishii, 1998], ícones físicos sobre uma superfície plana são marcados com pontos de material retrorefletivo. Rekimoto utiliza marcas codificadas com barras coloridas ou matrizes binárias de quadrados, como no Navicam [Rekimoto e Nagao, 1995] e no *Augmented Surfaces* [Rekimoto e Saitoh, 1999], tendo um espaço de rótulos grande para representar muitos objetos. Em realidade aumentada, Bajura e Neumann [1995], por exemplo, utilizam LED (*light emitter diode*) como marca fiducial por ser extremamente simples a segmentação que consiste em capturar as regiões da imagem de maior intensidade de vermelho.

- **Fusão sensorial**

Fusão sensorial é uma estratégia comum no rastreio aplicado à realidade aumentada. Segundo Azuma [1995], vários tipos de rastreadores podem ser utilizados em realidade aumentada, mas cada tecnologia tem suas deficiências. Alguns rastreadores mecânicos são suficientemente precisos, mas tem um volume de trabalho limitado por necessitar de uma conexão mecânica como um braço articulado por exemplo. Rastreadores magnéticos, capazes de medir os seis graus de liberdade, são vulneráveis à distorção por metais no ambiente, que podem existir em muitas aplicações. Além disso, para os rastreadores magnéticos é necessário um acoplamento por cabos da referência até o objeto rastreado. Rastreadores por ultra-som sofrem com ruído e dificilmente funcionam a longas distâncias devido a variações de temperatura no meio. Sabe-se que sensores ópticos são os mais precisos para determinar posições e orientações, mas são pouco robustos a efeitos como a oclusão, o campo de visão limitado e as variações de iluminação. Alguns sensores diferenciais, como os sensores mecânicos inerciais, podem apresentar erro por *drift*, isto é, acúmulo do erro ao longo de uma seqüência prolongada.

Azuma [1999] discute sobre o desafio de se produzir realidade aumentada em ambientes externos. As seguintes dificuldades são enumeradas: a portabilidade de dispositivos,

considerando potência, tamanho e peso, a falta de controle sobre a iluminação, o menor controle sobre o ambiente e a redução de recursos de computação e sensores. Os sensores para ambiente externo incluem o GPS, sensores inerciais, bússola eletrônica e sensores de inclinação. O GPS (*Global positioning system*) tem resolução global de 30 metros e diferencial de 3 metros na medida de posição. Necessita de uma área aberta para funcionar pois o sensor recebe dados de vários satélites. O GPS pode falhar em ambientes fechados e receber interferência. A bússola eletrônica é sugerida para medir a orientação, é um dispositivo barato e acurado (até meio grau), porém sensível a distorções do campo magnético. Os sensores inerciais de velocidade rotação e aceleração podem complementar o sistema, não podendo ser utilizados independentemente por acumularem erro ao longo do tempo.

Sistemas híbridos são os mais promissores, pois uma tecnologia compensa a deficiência da outra. Por exemplo, um sensor pode ser mais preciso enquanto o outro é mais robusto, ou então, um sensor pode medir dados referenciados de forma global ou absoluta e o outro de forma local ou diferencial. No caso da combinação de sensores ópticos e mecânicos inerciais, o deslocamento e a rotação da câmera podem ser estimados de forma robusta por sensores inerciais (giroscópios e acelerômetros), insensíveis a variações de iluminação, oclusão e distração por objetos da cena, enquanto o rastreio por câmera é uma forma precisa de se obter o alinhamento [Azuma 1995].

É comum o uso de sensores de tipos diversos para rastreio híbrido em sistemas de realidade aumentada. Neste caso é necessário resolver o problema de fusão de dados. Uma introdução ao problema geral de fusão de dados de múltiplos sensores pode ser encontrado em [Hall e Llinas, 1997]. O processo de fusão dos dados consiste em alinhamento e associação dos dados, correlação dos dados, inferência e interpretação. O alinhamento consiste em criar um referencial único para os dados e ajustá-los a esse referencial. A associação e a correlação refinam o resultado do alinhamento. A fusão dos dados é obtida por inferência, baseada em diversos métodos como *clustering* ou métodos de votação, redes neurais, inferência bayesiana e método de Dempster-Shafer. Para informações cinemáticas no tempo, utilizam-se também técnicas de predição como filtragem de Kalman. A interpretação consiste principalmente em um classificador. O processo de fusão de dados pode contar também com uma base de dados para auxiliar nas diversas etapas.

Para o problema da localização global relacionada à navegação autônoma, Lu e Milios [1994] utilizam rastreio de marcas fiduciais efetuando correspondência de modelos 3D com imagem 2D através de algoritmos de Visão e uso de outros sensores e algoritmos como a correspondência de varreduras de profundidade (*range scans*). Essa varredura é um sinal unidimensional definido pela distância da cena ao sensor em função do ângulo de observação.

Na literatura referente a realidade aumentada, encontram-se muitas abordagens baseadas em outras formas sensoriais, principalmente as abordagens iniciais. Bajura *et al.* [1992], numa aplicação médica, utilizam apenas rastreio magnético, assim como Feiner *et al.* [1993], num caso de manutenção. São casos em que o alinhamento é global, mas poderia ser melhorado com a fusão de dados de dispositivos ópticos, diminuindo erros de precisão e de distorção do campo magnético.

- ***Shape-matching***

O problema de reconhecimento quando se tem feições que consistem em curvas e regiões e se tem confiança no processo de segmentação pode ser abordado por correspondência ou reconhecimento de formas. Há várias dificuldades para a solução do problema do reconhecimento de formas (*shape-matching*). Veltkamp e Hagedoorn [1999] enumeram as seguintes dificuldades: ambigüidades, imprecisão da definição do problema, ruídos, eventuais alterações na topologia da forma reconstruída a partir de uma amostragem. De modo geral, a correspondência de formas já é um problema difícil e não seria recomendado para casos com número de graus de liberdade elevado (6 graus para a pose) e restrições severas de tempo. Além da aplicação no problema de alinhamento, as soluções para o problema de *shape-matching* podem ser aplicadas, segundo [Gdalyahu e Weinshall], para reconhecimento, *depth-from-stereo* e categorização de bancos de dados de imagens.

A solução da correspondência por minimização da distância é chamado de *matching* por proximidade. Segundo Veltkamp e Hagedoorn [1999], esse tipo de solução é útil quando a correspondência, um a um, foi perdida devido a ruído e oclusão, ou seja, em casos que se tenha uma previsão dos parâmetros da curva. Para minimizar a distância, uma busca global por todo espaço de parâmetros pode ser feita como em [Rucklidge 1996] utilizando subdivisão do espaço. As distâncias entre formas também são úteis para verificação: os parâmetros da

transformação do objeto são estimados, o objeto é reprojetado sobre a imagem e a distância é medida servindo como uma estimação do erro.

Para medir a dissimilaridade entre conjuntos de pontos, a distância de Hausdorff é freqüentemente utilizada por avaliar uma relação de correspondência com o ponto mais próximo e considerar os extremos ao invés de um caso médio. Isso também se torna um problema, na medida que a distância de Hausdorff se torna muito sensível ao ruído, sendo substituída pela distância parcial de Hausdorff, menos sensível (veja [Rucklidge 1996] ou [Veltkamp e Hagedoorn, 1999]).

Segundo Veltkamp [2001], atributos invariantes podem ser obtidos da forma da curva. Uma função diretriz (*turning function*) corresponde ao ângulo da reta tangente a uma curva parametrizada por comprimento de arco. Propriedades interessantes dessas funções são a invariância à translação e transformação da rotação em deslocamento. Dissimilaridades baseadas nessas funções são fáceis de se utilizar para processar linhas poligonais, mas também podem ser utilizadas em curvas suaves. É fácil realizar o *matching* ponto a ponto de duas curvas através do *matching* de funções diretrizes para similaridades euclidianas no plano.

Uma forma de obter invariantes é através da normalização da forma de curvas. Assinatura, segundo Gonzalez e Woods [1992], é uma função unidimensional descrevendo uma curva bidimensional. Um exemplo é a medida da distância do centróide até o ponto da curva em função do ângulo. A função gerada é uma medida invariante à translação. Pode ser feita invariante à escala se for normalizada em amplitude. A rotação corresponde a um deslocamento do parâmetro da função.

Operações simples sobre a forma de uma curva podem ser utilizadas para estabelecer um referencial normalizado dependente da forma da curva, que por sua vez, pode ser utilizado na construção de invariantes ou na determinação direta da transformação. Descritores de Fourier são outra possibilidade para representação da curva para *matching*. Segundo Gonzalez e Woods [1992], descritores de Fourier perdem as propriedades de invariância, porém transformações de translação, rotação e escala são relacionadas a transformações simples dos descritores no campo complexo.

Seguindo essa mesma idéia, há o método dos momentos estatísticos. Momentos estatísticos descrevem a distribuição dos pontos pertencentes à região como se a posição do ponto fosse uma variável aleatória. Por exemplo, o momento de Hu de ordem zero mede a área da região, os momentos de Hu de ordem um medem a posição do baricentro e os modelos de ordem maior caracterizam a forma ou dispersão. Podem ser obtidos momentos invariantes para descrever a forma e também podem se utilizar os momentos para estimar a pose de regiões transformadas. No item 3.4 descrevemos como utilizar os momentos como parte do processo de descrição de feições regionais em 3D.

2.3 Discussão

As seguintes dificuldades são encontradas nos vários métodos estudados.

- Nos métodos de inserção de marcas artificiais, as desvantagens são que, em geral, é necessário modificar o ambiente de trabalho e é necessário projetar uma forma de código para a marcação que garanta o discernimento entre as marcas.

- Métodos que exploram marcas naturais não declaradas na autoria do modelo não têm a garantia de que essas marcas preservam suas propriedades.

- O uso de múltiplos tipos de sensores sujeita o projetista às condições impostas pela presença de cada tipo de sensor. Uma grande inconveniência pode ser a necessidade de acoplamento físico que ocorre por exemplo em sensores magnéticos ou sensores baseados em articulações mecânicas. A distorção do campo magnético pode mesmo inviabilizar o uso de sensores magnéticos.

- Métodos que consideram a transformação projetiva devem lidar com a deformação, que torna difícil a análise da forma geométrica dos elementos da imagem em tempo restrito. Métodos que consideram aproximação por perspectiva fraca podem ser utilizados para alinhamento 2D, mas não são capazes de suprir dados 3D para a aplicação.

- Técnicas incrementais precisam de um estado inicial conhecido para prosseguir com o alinhamento quadro a quadro. Essas técnicas podem, por exemplo, solicitar do usuário uma inicialização. Essa forma de solução claramente não resolve o problema da descontinuidade em tempo real, mas pode ser utilizada por exemplo na edição de vídeo *off-line* ou desde que se garanta ou se assuma ausência de descontinuidades e oclusões.

- A intervenção do usuário é geralmente utilizada porque o problema de correspondência é facilmente resolvido pelo usuário enquanto que uma solução computacional precisa ser muito sofisticada para obter resultados equivalentes. Entretanto, para seqüências grandes de imagens ou sistemas de tempo real, a intervenção do usuário não é uma opção factível por consumir muito tempo e consistir em processos cansativos e sujeitos a erros.

- O reconhecimento da forma das curvas dificulta o processo de autoria, faz o sistema muito dependente do desempenho do algoritmo de segmentação e pode exigir análise de casos de ambigüidade para formas muito simétricas.

- Métodos baseados em aparência podem produzir um número muito grande de feições e a autoria do modelo é mais complexa e pode ser difícil para um usuário comum.

Assim, frente a esses problemas, podemos levantar algumas características importantes no desenvolvimento de uma abordagem ao problema de alinhamento. (1) É importante considerar marcas naturais, porém aquelas que sejam conhecidas ou incluídas no modelo numa fase de autoria. (2) O uso de apenas câmeras para o alinhamento permite um projeto de sistema com menores restrições. (3) A consideração de transformações rígidas em 3D permite a construção de invariantes. Em contraste com a transformação projetiva planar, não há distorção da forma nesse caso. É, também, importante que as aplicações possam contar com dados 3D da cena e portanto deve-se evitar aproximações para o modelo de projeção perspectiva. (4) O emprego de uma técnica global que considere cada quadro individualmente permite a automatização do processo de alinhamento e acrescenta robustez a movimentos abruptos e trocas de contexto na

seqüência de imagens. (5) O tipo de feição considerado deve ser escolhido levando em conta a facilidade do processo de autoria do modelo e a eficiência do processo de determinação de correspondências. (6) É desejável um método que seja independente da forma da feição, utilizando apenas posições e orientações, impondo requisitos mais fracos aos algoritmos de segmentação das feições.

Nos capítulos seguintes desenvolvemos nossa abordagem baseada nos princípios discutidos. No capítulo 3, descrevemos métodos para trabalhar com os conceitos de Visão Computacional relacionados aos problemas de estimação de pose, descrição e correspondência. A abordagem proposta é, então, descrita no capítulo 4.

Capítulo 3 - Métodos da Visão Computacional para Construção de uma Solução para o Alinhamento

Nossa proposta de solução demanda o conhecimento de um conjunto de métodos da Visão Computacional, enumerados a seguir, para lidar com o problema de alinhamento geométrico através da análise de imagens. Com o objetivo de facilitar o entendimento da solução proposta, apresentamos a seguir uma revisão destes conceitos. Primeiro, descrevemos o modelo de câmera considerado e também um método para sua calibração a partir de correspondências entre pontos da imagem e pontos da cena. O resultado desse processo é uma matriz de calibração cujo produto por um vetor que representa um ponto no espaço em coordenadas homogêneas é um vetor que representa o ponto projetado na imagem, dado em coordenadas homogêneas. Fornecemos então um método para fatoração dos parâmetros intrínsecos e dos parâmetros extrínsecos da câmera a partir da matriz de calibração, que pode ser utilizado para extração do componente de movimento rígido. Depois disso, discutimos a questão da geometria epipolar de um par de câmeras, equacionando a restrição epipolar e, através da retificação das imagens, simplificamos esse equacionamento. Mostramos como reconstruir um ponto 3D a partir de dois pontos correspondentes em um par de imagens estéreo. Apresentamos o método dos momentos estatísticos de Hu para determinar área, orientação e centro de massa de um pedaço de superfície aproximadamente planar. E, finalmente, apresentamos um estudo de métodos de aglomeração para realizar reconhecimento de objetos e estimação de pose simultaneamente a partir de feições.

3.1 Modelo de câmera

O modelo de câmera relaciona a geometria da cena com a geometria da imagem, permitindo-nos lidar com informação de ambos contextos. Definimos, a seguir, o modelo de câmera que é utilizado no nosso tratamento do problema de alinhamento e apresentamos mecanismos para sua calibração e sua construção a partir de parâmetros.

3.1.1 Modelo de uma câmera

Consideramos apenas o modelo de câmera de orifício (figura 3-1), que é o mais utilizado. Neste modelo, tem-se um ponto chamado centro de projeção e o plano imagem ou retinal. A imagem de um ponto no espaço se forma pela intersecção do plano imagem com a reta que une o ponto do espaço com o centro de projeção.

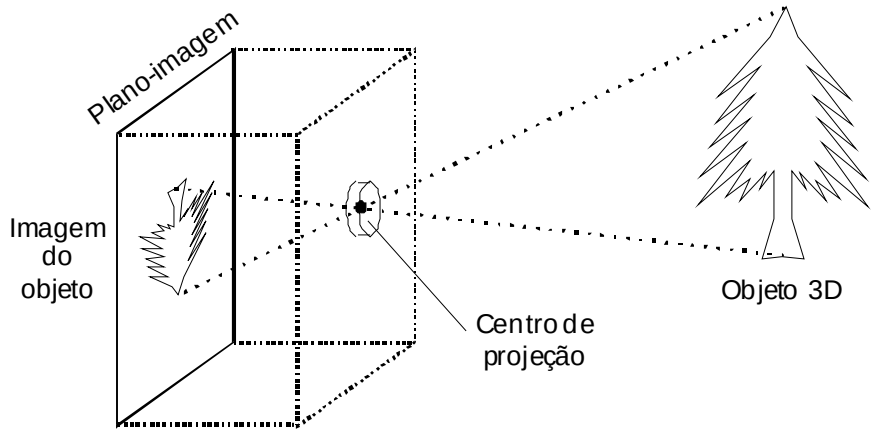


Figura 3-1 – Modelo de câmera de orifício

A maioria das câmeras comerciais podem ser descritas satisfatoriamente por este modelo. Entretanto, é muito comum haver distorção do plano imagem devido a problemas do sistema de lentes, deslocamento do filme ou dos sensores (retina) e variações de fabricação de câmera para câmera. O pior problema de distorção consiste em uma não-linearidade que ocorre pelo fato da projeção ser esférica ou elipsoidal ao invés de plana. Utilizamos um modelo mais simples em que a projeção é suposta plana e a distorção no plano imagem é uma transformação afim (Figura 3-2).

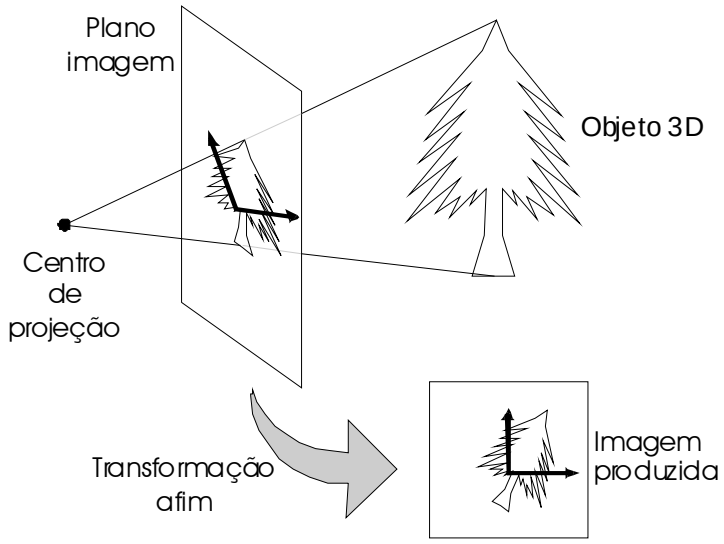


Figura 3-2 – Modelo de câmera de orifício com distorção afim sobre o plano imagem.

Descrevemos, a seguir, de maneira formal o modelo de câmera. A posição do centro de projeção é dada pelo vetor $C = [X_c \quad Y_c \quad Z_c]^T$ e a direção do eixo óptico (eixo z perpendicular ao plano imagem) é definido pelo vetor $\mathbf{v} = [X_V \quad Y_V \quad Z_V]^T$ que também define o plano-imagem em relação ao centro de projeção, pois seu comprimento é a distância do centro de projeção ao plano-imagem e sua direção é normal ao plano. Apesar de ser comum definir a distância focal como a norma deste vetor, fixamos a norma deste vetor em 1 e consideramos a distância focal como parte da distorção no plano. Assim, o centro de projeção define a translação da câmera, com 3 graus de liberdade, e o vetor $\mathbf{v}$ define 2 graus de liberdade para a orientação da câmera. Esses elementos estão ilustrados na figura 3-3.

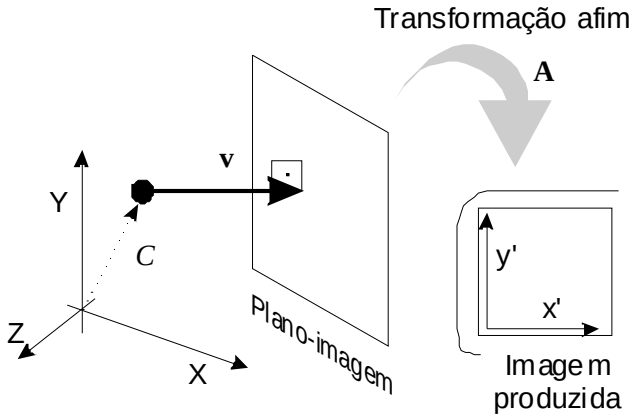


Figura 3-3 – Elementos para definição da câmera

A transformação afim no plano é dada pela matriz $[a_{ij}]$ de dimensões 2×3 como mostra a equação (3-1), onde x', y' são coordenadas da imagem com deformação e x, y são coordenadas da projeção do ponto sobre o plano imagem.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}_{2\times3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \textbf{(3-1)}$$

Neste ponto, introduzimos coordenadas homogêneas para facilitar o trabalho com geometria projetiva. Um ponto $P = [X \quad Y \quad Z]^T$ no espaço euclidiano corresponde aos múltiplos de $[X \quad Y \quad Z \quad 1]^T$ no espaço projetivo. Assim, referimo-nos a este ponto como $\tilde{P} = [W \cdot X \quad W \cdot Y \quad W \cdot Z \quad W]^T$ para qualquer $W > 0$. O caso em que a quarta coordenada é nula ($W = 0$) consiste um caso especial em que o ponto é impróprio, representando a direção de um feixe de retas ou planos paralelos. Um estudo mais aprofundado sobre a geometria projetiva e coordenadas homogêneas pode ser encontrado em [Stolfi, 91].

Um ponto na imagem é representado como $p = [x \quad y]^T$, cujo correspondente em coordenadas homogêneas é $\tilde{p} = [wx \quad wy \quad w]^T$. Um ponto P no espaço é projetado sobre um ponto p da imagem segundo a equação (3-2)

$$\tilde{p} = \mathbf{B}\tilde{P}\,, \tag{3-2}$$

onde definimos $\mathbf{B}$ como a matriz de calibração dada na equação (3-3)

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix}. \tag{3-3}$$

A matriz de calibração é o produto das matrizes de parâmetros intrínsecos e de parâmetros extrínsecos que descrevemos nas seções seguintes.

3.1.2 Construção da matriz de parâmetros intrínsecos

A matriz de projeção, uma vez que o ponto P esteja definido no sistema de coordenadas da câmera é dada por $\mathbf{N}$ na equação (3-4) para distância focal f.

$$\mathbf{N} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{3\times 4} \tag{3-4}$$

$$\tilde{p} = \mathbf{N}\tilde{P} \tag{3-5}$$

Este modelo de projeção considera o plano imagem em $Z = f$ e o centro de projeção na origem. Fixamos $f = 1$, pois em nosso modelo o vetor $\mathbf{v}$ é unitário e a focal é considerada como parâmetro da distorção afim da imagem. Assim, adotamos a matriz de projeção perspectiva da equação (3-6)

$$\mathbf{N} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3-6}$$

A matriz $\mathbf{N}$ define uma projeção plana com centro de projeção na origem, de forma que alterar o valor da distância focal é equivalente a uma transformação de escala. Considerando que uma transformação de escala pode ser diferente em cada eixo, alterando o aspecto (*aspect*

ratio) da imagem, fica difícil estabelecer uma definição para a distância focal sem conhecimento das dimensões do pixel. Assim, englobamos a distância focal e o aspecto nos parâmetros da distorção do plano imagem como transformações de escala, como pode ser observado na discussão que se segue.

A distorção no plano imagem é descrita pelo conjunto de parâmetros intrínsecos da câmera, isto é, os parâmetros que não mudam com o deslocamento da câmera no espaço. Já, os parâmetros que descrevem apenas o deslocamento da câmera no espaço são os parâmetros extrínsecos.

A matriz dos parâmetros intrínsecos é normalmente escrita da forma da expressão (3-7)

$$\mathbf{A} = \begin{bmatrix} -fK_u & fK_u \cot\theta & u_0 & 0 \\ 0 & -f\dfrac{K_v}{\operatorname{sen}\theta} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{3\times 4} . \qquad \textbf{(3-7)}$$

Nessa expressão, f é a distância focal. K_u e K_v são a densidade de sensores nos eixos horizontal e vertical respectivamente, de forma a definir o aspecto e a escala. O ângulo θ é o *skew*, inclinação entre os eixos x e y, que pode ser diferente de $\frac{\pi}{2}$ radianos. O vetor $\begin{bmatrix} u_0 & v_0 \end{bmatrix}^T$ corresponde ao ponto principal da imagem, as coordenadas do ponto em que o plano imagem é interceptado pelo eixo óptico. Esses parâmetros $f, K_u, K_v, \theta, u_0, v_0$ são os parâmetros intrínsecos da câmera.

Em nosso modelo, os parâmetros extrínsecos consistem no centro de projeção, na direção do eixo óptico e em uma rotação em torno do eixo óptico. Os parâmetros intrínsecos consistem dos coeficientes da matriz de transformação afim **A** descrita na equação (3-8), uma vez eliminada a rotação em torno do eixo óptico, isto é, uma rotação sobre o plano imagem. Utilizamos a forma geral (3-8) da matriz para qualquer transformação afim, incorporando a focal, o aspecto e a rotação em torno do eixo óptico. Essa rotação no plano pode ser facilmente fatorada, bastando determinar o ângulo entre o vetor $\begin{bmatrix} a_{11} & a_{21} \end{bmatrix}^T$ com o eixo *x*.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{3\times 4} \quad \textbf{(3-8)}$$

Definimos a forma inversível (3-9) da matriz para $\mathbf{a}_1 = \begin{bmatrix} a_{11} & a_{21} \end{bmatrix}^T$ e $\mathbf{a}_2$ independentes.

$$\mathbf{A}^* = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}_{3\times 3} \quad \textbf{(3-9)}$$

3.1.3 Construção da matriz de parâmetros extrínsecos

Considerando como parâmetros extrínsecos a posição C do centro de projeção e a direção $\mathbf{v}$ do eixo óptico, descrevemos, a seguir, como construir a matriz dos parâmetros extrínsecos, $\mathbf{M}$. Essa construção se dá pela composição de transformações. Essa matriz é chamada matriz de câmera porque representa uma transformação do referencial absoluto para um referencial baseado nos parâmetros da câmera: o centro de projeção C é mapeado na origem e o vetor $\mathbf{v}$ que define o eixo óptico é mapeado sobre o eixo *z*.

Inicialmente, todo ponto definido no referencial absoluto deve ser transladado para que C coincida com a origem. Uma matriz de translação com as coordenadas opostas do centro de projeção permite realizar esta tarefa

$$\mathbf{T}_{-C} = \begin{bmatrix} 1 & 0 & 0 & -X_C \\ 0 & 1 & 0 & -Y_C \\ 0 & 0 & 1 & -Z_C \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad \textbf{(3-10)}$$

Em seguida, rodamos o eixo óptico para que coincida com o eixo Z. Decompomos essa rotação em duas. Em primeiro lugar, rodamos $\mathbf{v}$ em torno do eixo X até o plano XZ. Veja a figura 3-4a.

Definimos n na equação (3-11)

$$n = \sqrt{Y_v^{\,2} + Z_v^{\,2}}\ . \tag{3-11}$$

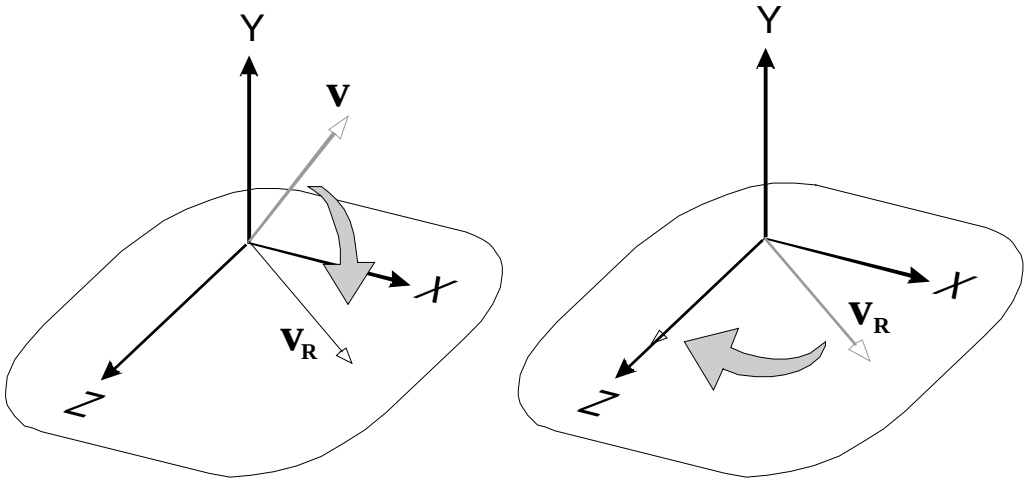
A rotação em torno do eixo X é dada pela equação (3-12)

$$\mathbf{R}_X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \dfrac{Z_v}{n} & \dfrac{-Y_v}{n} & 0 \\ 0 & \dfrac{Y_v}{n} & \dfrac{Z_v}{n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3-12}$$

O resultado da rotação em X é o vetor $\mathbf{v}_R$ definido em (3-13)

$$\mathbf{v}_R = \mathbf{R}_X \mathbf{v}\,. \tag{3-13}$$

Em seguida, rodamos este em torno de Y até que coincida com o eixo Z conforme a figura 3-4b.



(a) (b)

Figura 3-4 – Rotação do eixo óptico ao plano XZ e em seguida ao eixo Z.

A matriz de rotação em torno de Y é dada por (3-14)

$$\mathbf{R}_Y = \begin{bmatrix} \frac{n}{|\mathbf{v}|} & 0 & \frac{-X_v}{|\mathbf{v}|} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{X_v}{|\mathbf{v}|} & 0 & \frac{n}{|\mathbf{v}|} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3-14}$$

A composição das rotações resulta na equação (3-15)

$$\mathbf{R}_\mathbf{v} = \begin{bmatrix} \frac{n}{|\mathbf{v}|} & \frac{-X_v Y_v}{n|\mathbf{v}|} & \frac{-X_v Z_v}{n|\mathbf{v}|} & 0 \\ 0 & \frac{Z_v}{n} & \frac{-Y_v}{n} & 0 \\ \frac{X_v}{|\mathbf{v}|} & \frac{Y_v}{|\mathbf{v}|} & \frac{Z_v}{|\mathbf{v}|} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3-15}$$

Uma rotação em torno do eixo *z* pode ainda ser aplicada, dependente de uma parâmetro extrínseco θ. É difícil estabelecer uma referência para $\theta = 0$ sem ter antes fixado a direção do eixo óptico. Assim, essa rotação é considerada parte da matriz $\mathbf{A}$ definida em (3-8) a fim de simplificar o processo de fatoração dos parâmetros.

A composição dos parâmetros extrínsecos é representada pela matriz $\mathbf{M}$, definida em (3-16)

$$\mathbf{M} = \mathbf{R}_v \mathbf{T}_{-C}. \tag{3-16}$$

Na forma geral, esta matriz e seus elementos se escrevem como na equação (3-17)

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3-17}$$

Se for interessante ter apenas as rotações representadas, utiliza-se a matriz definida em (3-18)

$$\mathbf{M}^{*} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}_{3\times 3} . \qquad \textbf{(3-18)}$$

A matriz $\mathbf{M}$, dos parâmetros extrínsecos da câmera, define a relação entre o referencial tridimensional absoluto e o referencial da câmera. Multiplicando-se o vetor coluna de coordenadas homogêneas de um ponto no referencial absoluto por essa matriz, são obtidas coordenadas homogêneas desse ponto no referencial de câmera, conforme equação (3-19)

$$\tilde{p} = \mathbf{AM}\tilde{P} . \qquad \textbf{(3-19)}$$

3.1.4 Calibração unificada

A calibração de câmeras consiste da determinação dos parâmetros do modelo de câmera dadas as coordenadas de pontos no referencial absoluto e as coordenadas de imagem da projeção desses pontos. A calibração de parâmetros intrínsecos e extrínsecos pode ser feita de forma unificada pela equivalência com uma transformação de coordenadas homogêneas representada por uma matriz 3×4 de calibração. Um processo de fatoração dessa matriz separa seus parâmetros intrínsecos e extrínsecos. Este método de calibração de câmera nos é particularmente útil porque o procedimento para estimação de pose que utilizamos é uma generalização dele.

Partindo da equação (3-19) e definindo a matriz $\mathbf{B}$ na equação (3-20) como a matriz de, calibração, construímos em (3-21) a equação que modela a formação da imagem

$$\mathbf{B} = \mathbf{AM} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix}_{3\times 4} , \qquad \textbf{(3-20)}$$

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix}_{3x4} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \qquad \textbf{(3-21)}$$

Assim, o produto de um vetor coluna das coordenadas homogêneas de um ponto por **B** produz as coordenadas homogêneas de um ponto da imagem resultado do posicionamento da câmera, da projeção do ponto do espaço no plano imagem e da distorção afim da imagem.

A determinação dos coeficientes b_{ij} produz toda a informação que consideramos no modelo da câmera. A determinação da pose da câmera pode ser feita pela determinação da matriz de calibração e da fatoração dos parâmetros extrínsecos a partir desta matriz. Tratamos, agora, a determinação da matriz de calibração. Tendo obtido a imagem de um objeto conhecido, é necessário saber as coordenadas no espaço de pelo menos seis pontos desse objeto e relacioná-las com as coordenadas da imagem destes pontos para determinar os parâmetros. A equação (3-21) aplicada aos diferentes pontos gera um sistema de equações homogêneas, onde as variáveis são os elementos b_{ij} da matriz de calibração, cujas soluções estão numa reta que contém a origem, em casos não degenerados. Conhecendo a forma de **B**, podemos fixar (3-22) sem problemas, uma vez que o valor de w pode ser arbitrado.

$$b_{34} = 1 \qquad \textbf{(3-22)}$$

O objeto calibrador deve consistir de pelo menos seis pontos[1] de coordenadas conhecidas, porque são necessárias pelo menos onze equações para resolver o sistema para as onze variáveis remanescentes. O sistema de equações relaciona coordenada da tela $\begin{bmatrix} x_i & y_i \end{bmatrix}^T$ com coordenada do ponto no espaço $\begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix}^T$. Para seis pontos, a equação para a calibração é dada por (3-23)

[1] Excetuando-se configurações degeneradas, como no caso de pontos co-planares.

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -X_1x_1 & -Y_1x_1 & -Z_1x_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -X_2x_2 & -Y_2x_2 & -Z_2x_2 \\ X_3 & Y_3 & Z_3 & 1 & 0 & 0 & 0 & 0 & -X_3x_3 & -Y_3x_3 & -Z_3x_3 \\ X_4 & Y_4 & Z_4 & 1 & 0 & 0 & 0 & 0 & -X_4x_4 & -Y_4x_4 & -Z_4x_4 \\ X_5 & Y_5 & Z_5 & 1 & 0 & 0 & 0 & 0 & -X_5x_5 & -Y_5x_5 & -Z_5x_5 \\ X_6 & Y_6 & Z_6 & 1 & 0 & 0 & 0 & 0 & -X_6x_6 & -Y_6x_6 & -Z_6x_6 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -X_1y_1 & -Y_1y_1 & -Z_1y_1 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & -X_2y_2 & -Y_2y_2 & -Z_2y_2 \\ 0 & 0 & 0 & 0 & X_3 & Y_3 & Z_3 & 1 & -X_3y_3 & -Y_3y_3 & -Z_3y_3 \\ 0 & 0 & 0 & 0 & X_4 & Y_4 & Z_4 & 1 & -X_4y_4 & -Y_4y_4 & -Z_4y_4 \\ 0 & 0 & 0 & 0 & X_5 & Y_5 & Z_5 & 1 & -X_5y_5 & -Y_5y_5 & -Z_5y_5 \end{bmatrix}_{11x11} \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{14} \\ b_{21} \\ b_{22} \\ b_{23} \\ b_{24} \\ b_{31} \\ b_{32} \\ b_{33} \end{bmatrix}_{11x1} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}_{11x1} \quad . \quad \textbf{(3-23)}$$

Para um número arbitrário de pontos, a equação para calibração é dada por (3-24) e pode ser resolvida utilizando-se a matriz pseudo-inversa para aproximação por mínimos quadrados

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -X_ix_i & -Y_ix_i & -Z_ix_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -X_iy_i & -Y_iy_i & -Z_iy_i \\ \cdots & & & & & & & & & & \end{bmatrix}_{2N\ x11} \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{14} \\ b_{21} \\ b_{22} \\ b_{23} \\ b_{24} \\ b_{31} \\ b_{32} \\ b_{33} \end{bmatrix}_{11x1} = \begin{bmatrix} x_i \\ y_i \\ \vdots \end{bmatrix}_{2N\ x1} \quad . \quad \textbf{(3-24)}$$

Este método obtém uma calibração com 11 graus de liberdade, 3 correspondem à posição do centro de projeção, 2 à direção do eixo óptico e 6 aos parâmetros da distorção afim no plano conforme nosso modelo descrito na equação (3-8), dos quais pode ser extraído o ângulo de rotação em torno do eixo óptico.

3.1.5 Fatoração de parâmetros intrínsecos e extrínsecos

Para determinar a matriz de parâmetros de distorção afim $\mathbf{A}$, a partir da matriz de calibração $\mathbf{B}$, primeiramente determinamos os parâmetros extrínsecos e reconstruímos a matriz de câmera $\mathbf{M}$. A construção da matriz de câmera é abordada no item 3.1.3 e a determinação dos parâmetros extrínsecos é mostrada na seqüência.

O plano focal é um plano paralelo ao plano imagem que contém o centro de projeção. O centro de projeção C projeta em $\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$, que é considerado uma indefinição em coordenadas homogêneas. Além disso, todo ponto do eixo óptico deve se projetar no ponto principal (origem do sistema de coordenadas do plano-imagem), exceto pelo centro de projeção que não tem projeção definida. Assim, a equação (3-25) pode ser utilizada na determinação das coordenadas de C

$$\mathbf{B}\tilde{C} = \begin{bmatrix}0\\0\\0\end{bmatrix}. \qquad \textbf{(3-25)}$$

Definimos a matriz $\mathbf{B}^*$ em (3-26)

$$\mathbf{B}^* = \begin{bmatrix}b_{11} & b_{12} & b_{13}\\ b_{21} & b_{22} & b_{23}\\ b_{31} & b_{32} & b_{33}\end{bmatrix}. \qquad \textbf{(3-26)}$$

Escrevendo (3-25) por extenso e reagrupando na forma matricial, escrevemos um sistema para solução das coordenadas do centro de projeção no referencial global, descrito na equação (3-27)

$$\mathbf{B}^* C = \begin{bmatrix}-b_{14}\\ -b_{24}\\ -1\end{bmatrix}. \qquad \textbf{(3-27)}$$

O vetor $\mathbf{v}$ que define o eixo óptico e o plano imagem pode ser facilmente determinado por ser normal ao plano focal. A equação (3-28) é a equação do plano focal, formada a partir de

(3-21) pelas projeções em $w = 0$. O vetor normal a esse plano é o vetor dos coeficientes $[b_{31} \quad b_{32} \quad b_{33}]^T$

$$[b_{31} \quad b_{32} \quad b_{33} \quad 1]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0 . \tag{3-28}$$

Assim, calculamos o vetor $\mathbf{v}$ unitário por (3-29)

$$\mathbf{v} = \frac{1}{\sqrt{{b_{31}}^2 + {b_{32}}^2 + {b_{33}}^2}}[b_{31} \quad b_{32} \quad b_{33}]^T . \tag{3-29}$$

Conhecendo estes parâmetros, é possível reconstruir a matriz $\mathbf{M}$ de parâmetros extrínsecos da câmera pela equação (3-30), conforme o item 3.1.3

$$\mathbf{M} = \mathbf{R}_v \mathbf{T}_{-C} . \tag{3-30}$$

Com isso, é possível obter a matriz de parâmetros intrínsecos $\mathbf{A}$ pela equação (3-31)

$$\mathbf{A} = \mathbf{B}\mathbf{M}^{-1} . \tag{3-31}$$

Há outras maneiras de fatorar os parâmetros extrínsecos dos intrínsecos. Por exemplo, pode-se assumir que os parâmetros intrínsecos não se alteram numa seqüência de imagens com movimento de câmera. O método que apresentamos é mais apropriado para um quadro individual quando se desconhece os parâmetros intrínsecos.

3.2 Geometria epipolar

Num sistema com mais câmeras, ilustrado na figura 3-5, objetos definidos no referencial de uma câmera podem ser relacionados a pontos da imagem formada na outra câmera. Esse relacionamento constitui a geometria epipolar (figura 3-6). A principal conseqüência de se determinar a geometria epipolar é a chamada restrição epipolar, sob a qual, um ponto de uma imagem só pode ser correspondido aos pontos da outra imagem que estiverem sobre uma reta

determinada a partir das coordenadas do primeiro ponto. A restrição epipolar é uma ferramenta bastante forte para auxiliar na determinação da correspondência entre pontos de duas vistas.

3.2.1 Sistemas de referência

Para trabalhar com múltiplas vistas, utilizamos sistemas de referência em cada câmera e um sistema de referência absoluto. A relação entre sistemas de referência é descrita por matrizes da forma (3-17) que guardam a restrição de rigidez do movimento.

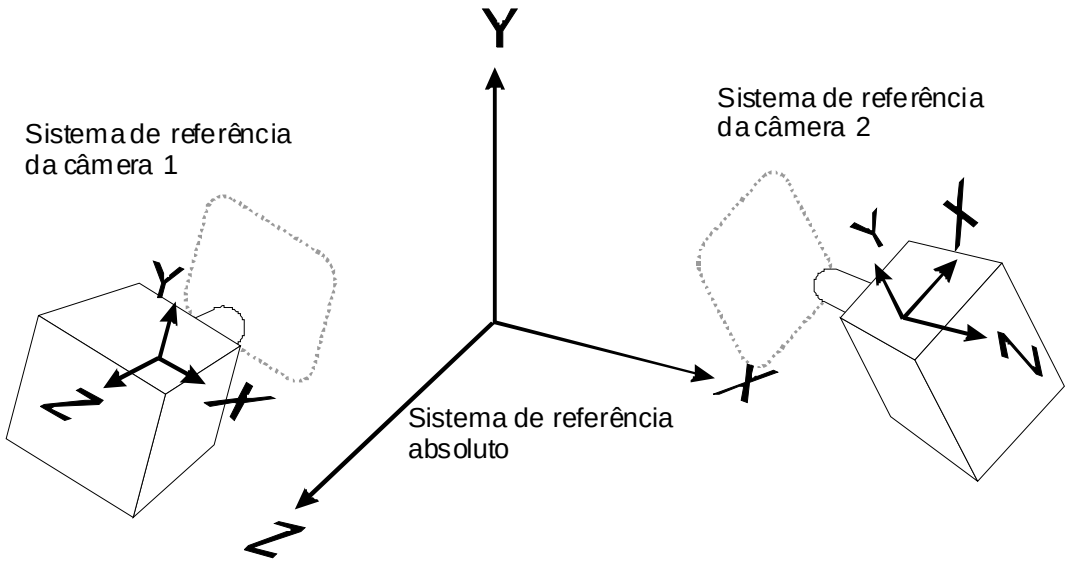


Figura 3-5 – Sistemas de referência.

Utilizamos a notação de subscrito entre parênteses para denotar o sistema de coordenadas utilizado. Assim, nas equações (3-32), definimos a transformação de coordenadas absolutas do ponto P para coordenadas no sistema de referência da imagem 1, representadas por $P_{(1)}$ e da imagem 2, por $P_{(2)}$. O til se refere ao espaço projetivo, isto é, o ponto é representado em coordenadas homogêneas

$$\tilde{P}_{(1)} = \mathbf{M}_1 \tilde{P} \qquad \textbf{(3-32)}$$
$$\tilde{P}_{(2)} = \mathbf{M}_2 \tilde{P}.$$

$\mathbf{M}_i$ é a matriz de câmera da câmera i. A transformação entre coordenadas de câmeras, ao invés de utilizar o referencial absoluto, é representada pela multiplicação de matrizes como na equação (3-33)

$$\mathbf{M}_{12} = \mathbf{M}_2 \mathbf{M}_1^{-1}. \qquad \textbf{(3-33)}$$

Dessa forma, a equação (3-34) descreve como converter as coordenadas de um ponto descrito no referencial da câmera 1 para coordenadas do referencial da câmera 2

$$\tilde{P}_{(2)} = \mathbf{M}_{12} \tilde{P}_{(1)}. \qquad \textbf{(3-34)}$$

Da matriz $\mathbf{M}_{12}$, muita informação sobre o par estéreo pode ser extraída. Destaca-se a linha de base C_1C_2 que é a distância entre os centros de projeção dada pela norma do vetor $\mathbf{t}$, que corresponde à posição do centro de projeção da câmera 2 no sistema de referência da câmera 1 descrito na equação (3-35)

$$\mathbf{t} = C_{2(1)}. \qquad \textbf{(3-35)}$$

3.2.2 Modelo de par estéreo

O equacionamento apresentado a seguir estabelece a restrição epipolar dada a orientação relativa entre as câmeras. Dadas duas câmeras e um ponto no espaço, define-se o plano epipolar como o plano que contém os dois centros de projeção e o ponto objeto no espaço. As projeções deste ponto nos planos imagens também pertencem a este plano. Todo plano epipolar contém os dois centros de projeção, contendo também a reta que os une. A intersecção entre essa reta e cada um dos planos imagens consiste nos dois pontos epipolares ou epipolos. No sentido da geometria projetiva, estes pontos podem estar no infinito. Veja a figura 3-6.

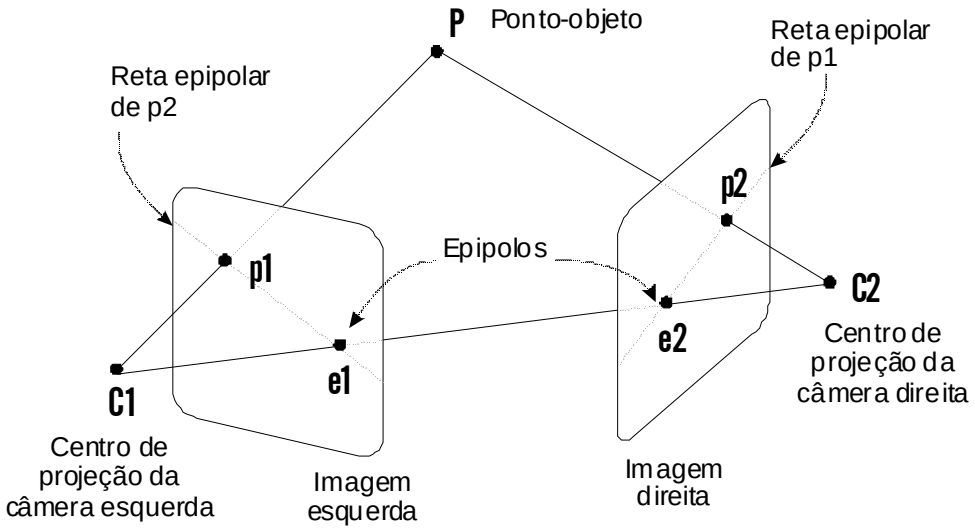


Figura 3-6 – Geometria epipolar.

Os pontos epipolares consistem na projeção do centro de projeção de uma câmera sobre o plano imagem da outra câmera. As coordenadas de imagem dos pontos epipolares $\mathbf{e}_1$ e $\mathbf{e}_2$ são dadas pela equação (3-36)

$$\begin{aligned} \mathbf{e}_1 &= \mathbf{A}_1\mathbf{M}_1 C_2 \\ \mathbf{e}_2 &= \mathbf{A}_2\mathbf{M}_2 C_1 \end{aligned}, \qquad \textbf{(3-36)}$$

onde $\mathbf{A}_i$ é a matriz de parâmetros intrínsecos da câmera i.

É conveniente que possamos expressar as coordenadas do ponto no espaço projetado sobre o plano imagem. Assim, para um ponto P desconhecido, cuja projeção na imagem tem coordenadas p_1, a projeção no plano imagem no espaço no sistema de referência da câmera 1 é $P'_{1(1)}$ que pode ser obtido pela equação (3-37)

$$\tilde{P}'_{1(1)} = \mathbf{Q}_1 \tilde{p}_1 . \qquad \textbf{(3-37)}$$

Onde, a matriz $\mathbf{Q}_1$ corresponde à definida na equação (3-38)

$$\mathbf{Q}_1 = \begin{bmatrix} \left(\mathbf{A}^{*-1}\right)_{3\times3} \\ \cdots\cdots\cdots \\ 0 \quad 0 \quad 1 \end{bmatrix}_{4\times3}. \tag{3-38}$$

As coordenadas desse ponto no sistema de referência absoluto podem ser obtidas por (3-39)

$$\tilde{P}_1' = \mathbf{M}_1^{-1}\tilde{P}_{1(1)}'. \tag{3-39}$$

Seja $\mathbf{u}$ o vetor perpendicular ao plano epipolar obtido por (3-40)

$$\mathbf{u} = \left(C_2 - C_1\right)\times\left(P_1' - C_1\right). \tag{3-40}$$

Nas coordenadas da câmera 1, é escrito como (3-41)

$$\mathbf{u}_{(1)} = C_{2(1)} \times P_{1(1)}'. \tag{3-41}$$

Que é equivalente a (3-42)

$$\mathbf{u}_{(1)} = \mathbf{t} \times P_{1(1)}'. \tag{3-42}$$

Esse produto pode ser escrito na forma matricial, definindo-se a matriz $\mathbf{S}$ como em (3-43)

$$\mathbf{S} = \begin{bmatrix} 0 & -Z_t & Y_t \\ Z_t & 0 & -X_t \\ -Y_t & X_t & 0 \end{bmatrix}. \tag{3-43}$$

O produto vetorial é então escrito da forma de multiplicação de matrizes em (3-44)

$$\mathbf{u}_{(1)} = \mathbf{S}P_{1(1)}'. \tag{3-44}$$

Seja $\mathbf{R}$ a matriz 3×3 que corresponde apenas às rotações de $\mathbf{M}_{12}$. O vetor $\mathbf{u}$ nas coordenadas da câmera 2 é dado pela equação (3-45)

$$\mathbf{u}_{(2)} = \mathbf{E}P'_{1(1)}, \tag{3-45}$$

onde $\mathbf{E}$ é a matriz essencial 3×3 definida por (3-46)

$$\mathbf{E} = \mathbf{RS}. \tag{3-46}$$

A equação (3-45) é válida para qualquer escala de $\mathbf{u}$, importando apenas sua direção. Assim a matriz $\mathbf{E}$ é dependente de um fator de escala. A matriz essencial não é inversível uma vez que a matriz $\mathbf{S}$ é singular de posto 2. No caso da degeneração de $\mathbf{u} = 0$, ocorre que P'_1 é um epipolo. Os epipolos podem ser determinados pelas equações (3-47) e (3-48), onde E_1 é o epipolo da vista 1 no espaço

$$\mathbf{E}E_{1(1)} = \mathbf{0}, \tag{3-47}$$

$$\mathbf{E}^T E_{2(2)} = \mathbf{0}. \tag{3-48}$$

Para resolver o sistema (3-47) pode-se fixar $Z_{E1(1)} = 1$, pois não haverá solução para um epipolo no infinito. A matriz essencial estabelece o plano epipolar no espaço euclidiano e não no espaço projetivo, de forma que não se pode representar pontos no infinito.

Um ponto P'_2 pertence ao plano epipolar definido por $\mathbf{u}$ e que passa por C_1, C_2 e P'_1 se $P'_2 - C_2$ for perpendicular a $\mathbf{u}$. Igualando o produto escalar a zero nas coordenadas da câmera 2, obtemos (3-49)

$$P'^{T}_{2(2)}\mathbf{u}_{(2)} = 0\ (escalar). \tag{3-49}$$

Substituindo (3-45) e (3-46), obtém-se a equação de Longett-Higgins (3-50) para a restrição epipolar

$$P'^{T}_{2(2)}\mathbf{E}\ P'_{1(1)} = 0\ (escalar). \tag{3-50}$$

A equação (3-51) permite determinar a reta dos possíveis pontos P'_1 dado um ponto P'_2 e o mesmo pode ser feito no caso inverso a partir de (3-50)

$$P'_{2(2)}{}^{T}\mathbf{E} = \mathbf{u}_{(1)}{}^{T}. \qquad \textbf{(3-51)}$$

Esse equacionamento, entretanto, não é muito conveniente por não considerar coordenadas de imagem e nem coordenadas homogêneas. É preciso que, dado um ponto em uma imagem, encontre-se a reta epipolar sobre a outra imagem. Lembrando que as coordenadas homogêneas da imagem são dadas por (3-52),

$$\begin{aligned} \tilde{p}_1 &= \mathbf{A}_1{}^{*} P'_{1(1)} \\ \tilde{p}_2 &= \mathbf{A}_2{}^{*} P'_{2(2)} \end{aligned} \qquad \textbf{(3-52)}$$

onde $\mathbf{A}_i^{*}$ são definidas em (3-9) para cada câmera i, as coordenadas espaciais euclidianas dos pontos projetados do plano imagem são dadas por (3-53)

$$\begin{aligned} P'_{1(1)} &= \mathbf{A}_1{}^{*-1}\tilde{p}_1 \\ P'_{2(2)}{}^{T} &= \tilde{p}_2{}^{T}\mathbf{A}_2{}^{*-T}. \end{aligned} \qquad \textbf{(3-53)}$$

A matriz fundamental $\mathbf{F}$ é definida pela equação (3-54)

$$\mathbf{F} = \mathbf{A}_2{}^{*-T}\mathbf{E}\mathbf{A}_1{}^{*-1}. \qquad \textbf{(3-54)}$$

Substituindo em (3-50), obtém-se (3-55)

$$\tilde{p}_2{}^{T}\mathbf{F}\ \tilde{p}_1 = 0. \qquad \textbf{(3-55)}$$

Esta equação relaciona a um ponto de uma imagem a equação da reta epipolar correspondente sobre a outra imagem. Enquanto a matriz essencial é utilizada no espaço tridimensional euclidiano, a matriz fundamental é utilizada no espaço das coordenadas homogêneas do plano.

Os epipolos nas duas imagens são obtidos diretamente da matriz fundamental através das equações (3-56), permitindo representá-los também no infinito

$$\begin{matrix} \mathbf{F}^T \tilde{e}_2 = \mathbf{0} \\ \mathbf{F}\, \tilde{e}_1 = \mathbf{0} \end{matrix} . \qquad \textbf{(3-56)}$$

A matriz fundamental é 3×3, não inversível, com posto 2, de forma que sua escala não interfere na equação (3-55).

Auto-calibração é a determinação da orientação relativa entre câmeras sem o conhecimento de coordenadas no espaço tridimensional. A matriz fundamental deve ser determinada a partir da correspondência de pontos nas duas imagens, dadas suas coordenadas 2D.

A determinação da matriz fundamental pode ser feita pelo método dos 8 pontos. A partir da associação de 8 pontos de cada imagem, utilizando a equação (3-55), constrói-se um sistema de equações tendo os elementos da matriz fundamental como variáveis. Cada ponto corresponde a uma equação. Entretanto, $\mathbf{F}$ é dependente de escala e qualquer múltiplo de uma solução é solução para o sistema homogêneo, assim há apenas 8 graus de liberdade para a matriz fundamental e não 9. Veja uma algoritmo para solução do método dos 8 pontos em, por exemplo, [Trucco e Verri, 1998].

A restrição de que a matriz fundamental tem posto 2, e portanto seu determinante é zero, pode ser utilizada na solução. Assim, uma equação não linear que force a anulação do determinante permite realizar a calibração com apenas 7 pontos e portanto dando origem ao método dos 7 pontos. Em ambos os métodos pode haver configurações degeneradas dos pontos, que devem ser evitadas.

3.2.3 Retificação

A retificação, ilustrada na figura 3-7, consiste em aplicar transformações sobre as coordenadas das imagens de um par estéreo de geometria epipolar conhecida de forma que as linhas epipolares se tornem paralelas ao eixo x e o valor em y de um ponto da imagem modificada da câmera 1 corresponda necessariamente a um ponto de mesmo valor em y na imagem modificada da câmera 2. Essa operação tem como finalidade simplificar os algoritmos que trabalham sobre pares estéreo de imagens. Essas transformações consistem em um par de homografias, por isso os centros de projeção não se alteram.

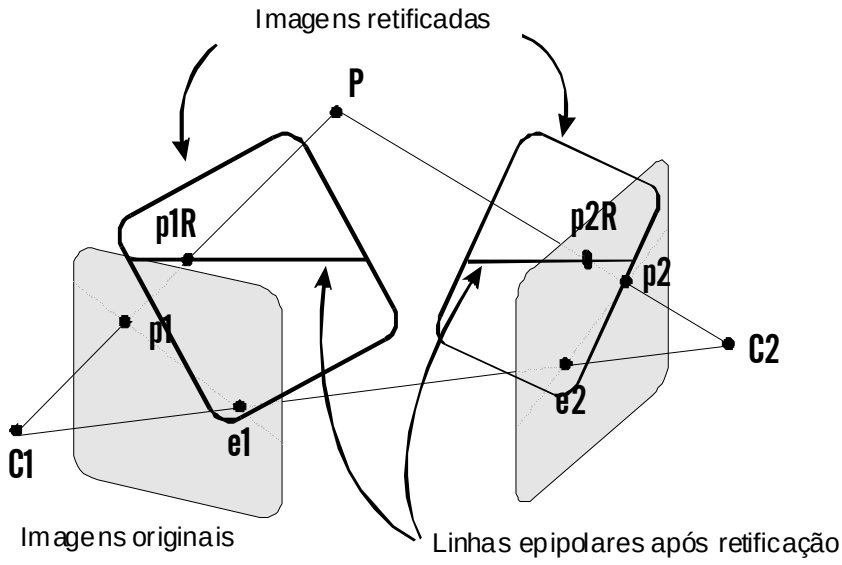


Figura 3-7 – Retificação.

Desenvolvemos uma forma simples de se fazer a retificação conhecendo apenas a matriz fundamental. Por utilizar apenas a matriz fundamental, essa abordagem à retificação é conveniente tanto no caso calibrado como no caso não calibrado (isto é, onde houve auto-calibração). Outras abordagens lineares à retificação podem ser encontradas por exemplo em [Fusiello, *et al.*, 2000] e [Robert *et al.*, 1993].

Vamos supor que as transformações de retificação sejam uma homografia para cada imagem e consideraremos implicitamente demonstrado no nosso desenvolvimento a seguir que esta transformação é suficiente. As transformações para retificação são representadas pelas homografias $\mathbf{L}_1$ e $\mathbf{L}_2$ nas equações (3-57)

$$\begin{aligned} \tilde{p}_{1R} &= \mathbf{L}_1 \tilde{p}_1 \\ \tilde{p}_{2R} &= \mathbf{L}_2 \tilde{p}_2 \end{aligned}. \qquad \textbf{(3-57)}$$

Queremos que as coordenadas resultantes respeitem a restrição (3-58), onde $\mathbf{F}_R$ é a matriz fundamental modificada pela retificação

$$\tilde{p}_{2R}{}^T \mathbf{F}_R \tilde{p}_{1R} = 0\,. \qquad \textbf{(3-58)}$$

Esta condição deve ser cumprida para quaisquer pares de pontos homólogos da forma (3-59), para quaisquer x_1, x_2 e y

$$\begin{aligned} \tilde{p}_{1R} &= [x_1 \quad y \quad 1]^T \\ \tilde{p}_{2R} &= [x_2 \quad y \quad 1]^T . \end{aligned} \tag{3-59}$$

Escrevemos $\mathbf{F}_R$ da forma (3-60)

$$\mathbf{F}_R = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}. \tag{3-60}$$

Substituindo na equação (3-58) obtemos (3-61)

$$[x_2 \quad y \quad 1] \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y \\ 1 \end{bmatrix} = 0 . \tag{3-61}$$

Expandimos essa equação em (3-62)

$$f_{22}y^2 + (f_{12}x_2 + f_{21}x_1 + f_{23} + f_{32})y + f_{11}x_1x_2 + f_{13}x_2 + f_{31}x_1 + f_{33} = 0 . \tag{3-62}$$

Como (3-62) deve ser válida para quaisquer x_1, x_2 e y, obtemos o seguinte sistema de equações (3-63).

$$\begin{aligned} & f_{22} = 0 \\ & f_{12} = f_{21} = 0 \\ & f_{11} = f_{13} = f_{31} = f_{33} = 0 \\ & \text{e} \\ & f_{23} + f_{32} = 0, \text{ visto que } (f_{23} + f_{32})y = 0, \forall y \end{aligned} \tag{3-63}$$

Como a matriz fundamental é dependente de escala, então podemos escolher uma solução para representar todos seus múltiplos. Assim arbitramos (3-64)

$$\mathbf{F}_R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \qquad \textbf{(3-64)}$$

Substituindo em (3-58)

$$\tilde{p}_2{}^T \mathbf{L}_2{}^T \mathbf{F}_R \mathbf{L}_1 \tilde{p}_1 = 0\,. \qquad \textbf{(3-65)}$$

Devemos achar $\mathbf{L}_1$ e $\mathbf{L}_2$ tais que (3-66) se cumpra

$$\mathbf{L}_2{}^T \mathbf{F}_R \mathbf{L}_1 = \mathbf{F}\,. \qquad \textbf{(3-66)}$$

Sugerimos normalizar a matriz fundamental, para que se tenham matrizes de retificação normalizadas (apesar de trabalharem no espaço projetivo).

Decompondo $\mathbf{F}_R$ em valores singulares (SVD), obtemos (3-67)

$$\mathbf{F}_R = \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{\mathbf{V}^T}. \qquad \textbf{(3-67)}$$

Definimos as matrizes auxiliares $\mathbf{G}$ e $\mathbf{H}$ inversíveis como em (3-68)

$$\begin{aligned} \mathbf{G}^{-1} &= \mathbf{L}_2{}^T \mathbf{U} \qquad (\textit{inversível}) \\ \mathbf{H} &= \mathbf{V}^T \mathbf{L}_1 \qquad (\textit{inversível}) \end{aligned}\,. \qquad \textbf{(3-68)}$$

A equação (3-66) é reescrita como (3-69)

$$\mathbf{G}^{-1}\mathbf{D}\mathbf{H} = \mathbf{F}\,. \qquad \textbf{(3-69)}$$

E, aplicando o fato de $\mathbf{G}$ ser inversível, obtemos (3-70)

$$\mathbf{D}\mathbf{H} = \mathbf{G}\mathbf{F}\,. \qquad \textbf{(3-70)}$$

Expandindo $\mathbf{DH}$ em seus elementos, temos (3-71)

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{GF}\,. \tag{3-71}$$

As seguintes restrições (3-72) são obtidas a fim de cumprir os requisitos da retificação,

$$\begin{cases} \mathbf{g}_3{}^T\mathbf{F} = \mathbf{0} \\ \mathbf{h}_1{}^T = \mathbf{g}_1{}^T\mathbf{F} \\ \mathbf{h}_2{}^T = \mathbf{g}_2{}^T\mathbf{F} \\ \mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3 \text{ independentes} \\ \mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3 \text{ independentes} \end{cases} \tag{3-72}$$

onde os vetores $\mathbf{h}_i$ e $\mathbf{g}_i$ são definidos em (3-73)

$$\begin{aligned} \mathbf{h}_i &= \begin{bmatrix} h_{i1} & h_{i2} & h_{i3} \end{bmatrix}^T \\ \mathbf{g}_i &= \begin{bmatrix} g_{i1} & g_{i2} & g_{i3} \end{bmatrix}^T . \end{aligned} \tag{3-73}$$

Se $\mathbf{g}_1$, $\mathbf{g}_2$ e $\mathbf{g}_3$ forem independentes, $\mathbf{h}_1$ e $\mathbf{h}_2$ também o são, visto que $\mathbf{F}^T\mathbf{g}_3 = 0$ e que $\mathbf{F}$ tem posto 2.

Lembrando que o epipolo da segunda imagem é o espaço nulo de $\mathbf{F}^T$, observa-se (3-74)

$$\mathbf{g}_3 \propto \tilde{e}_2\,. \tag{3-74}$$

Há 9 graus de liberdade a se arbitrar, que não são limitados pelas restrições de retificação (3-72). A liberdade corresponde à escolha de $\mathbf{g}_1$, $\mathbf{g}_2$ e $\mathbf{h}_3$. Neste ponto, é possível introduzir restrições que minimizem a distorção. Na ausência de informação adicional e considerando que a distorção não tem importância se a imagem não for reamostrada em pixels, limitamo-nos a arbitrar um procedimento para a escolha desses parâmetros livres de forma a respeitar as condições de independência linear em (3-72). Supondo-se que o epipolo não está sobre o ponto principal, o seguinte esquema preenche as matrizes $\mathbf{G}$ e $\mathbf{H}$,

$$\begin{aligned}
&\mathbf{g}_3 \leftarrow \tilde{e}_2 \\
&\mathbf{g}_1 \leftarrow \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T \times \mathbf{g}_3 \\
&\mathbf{g}_2 \leftarrow \mathbf{g}_3 \times \mathbf{g}_1 \\
&\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3 \leftarrow \frac{\mathbf{g}_1}{\alpha_1}, \frac{\mathbf{g}_2}{\alpha_2}, \frac{\mathbf{g}_3}{\alpha_3}
\end{aligned} \tag{3-75}$$

onde α_i são coeficientes arbitrados dependendo do formato final desejado para a imagem. Para a matriz $\mathbf{H}$, usamos (3-76)

$$\begin{aligned}
&\mathbf{h}_1 \leftarrow \mathbf{F}^T \mathbf{g}_1 \\
&\mathbf{h}_2 \leftarrow \mathbf{F}^T \mathbf{g}_2 \quad . \\
&\mathbf{h}_3 \leftarrow \mathbf{h}_1 \times \mathbf{h}_2
\end{aligned} \tag{3-76}$$

Finalmente, as matrizes de retificação podem ser obtidas através de (3-77)

$$\begin{aligned}
&\mathbf{L}_1 = \mathbf{V}^{-T} \mathbf{H} \\
&\mathbf{L}_2 = \mathbf{U}^{-T} \mathbf{G}^{-T} .
\end{aligned} \tag{3-77}$$

Observar as propriedades das matrizes $\mathbf{U}$ e $\mathbf{V}$ em (3-78).

$$\begin{aligned}
&\mathbf{V} = \mathbf{V}^{-1} = \mathbf{V}^T = \mathbf{V}^{-T} \\
&\mathbf{U}^{-1} = \mathbf{U}^T \\
&\mathbf{U} = \mathbf{U}^{-T}
\end{aligned} \tag{3-78}$$

Obtivemos resultados adequados com este algoritmo aplicado a coordenadas reais dos pontos das imagens. Entretanto, as imagens resultantes podem ficar muito distorcidas, de forma que, se forem reamostradas na grade de pixels, haverá perda de informação. Por outro lado, se estruturas geométricas forem extraídas da imagem e suas coordenadas forem retificadas, não há problema com a distorção, pois não há reamostragem. Assim, a representação de regiões da imagem por polígonos pode ser retificada através da retificação das coordenadas dos vértices. Além disso, é um processo mais rápido do que a retificação para todo o conjunto de pixels.

3.3 Reconstrução de pontos em 3D

Uma vez estabelecidas as coordenadas de dois pontos homólogos nas imagens, é necessário determinar a coordenada espacial do ponto objeto que gerou essas projeções. No caso calibrado em que se conhece a matriz $\mathbf{M}_{12}$, é possível reconstruí-lo por completo, enquanto que, no caso não calibrado, onde se conhece apenas a matriz fundamental e as matrizes $\mathbf{A}$ de parâmetros intrínsecos, a solução é dependente de um fator de escala. Este fator de escala pode ser obtido, por exemplo, a partir do conhecimento do comprimento de qualquer aresta da cena.

3.3.1 Caso calibrado

No caso calibrado, as matrizes de calibração são conhecidas. A solução da equação (3-79) representa a reconstrução para matrizes de calibração $\mathbf{B}$ e $\mathbf{C}$ de tamanho 3×4 e índices b_{ij} , c_{ij} . As coordenadas do ponto p_1 são (x_1, y_1) e do ponto p_2, seu homólogo, são (x_2, y_2). O ponto P reconstruído tem coordenadas (X, Y, Z)

$$\begin{bmatrix} b_{31}x_1 - b_{11} & b_{32}x_1 - b_{12} & b_{33}x_1 - b_{13} \\ b_{31}y_1 - b_{21} & b_{32}y_1 - b_{22} & b_{33}y_1 - b_{23} \\ c_{31}x_2 - c_{11} & c_{32}x_2 - c_{12} & c_{33}x_2 - c_{13} \\ c_{31}y_2 - c_{21} & c_{32}y_2 - c_{22} & c_{32}y_2 - c_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} b_{14} - b_{34}x_1 \\ b_{24} - b_{34}y_1 \\ c_{14} - c_{34}x_2 \\ c_{24} - c_{34}y_2 \end{bmatrix}. \qquad \textbf{(3-79)}$$

É importante lembrar que, em imagens retificadas, tem-se que $y_1 = y_2$ e as matrizes $\mathbf{B}$ e $\mathbf{C}$ devem incluir as homografias de retificação.

3.3.2 Caso não calibrado

Apesar de não considerá-lo na abordagem proposta nesta tese, discutimos brevemente o caso não calibrado para não deixar o texto incompleto. No caso não calibrado, devemos estimar a matriz $\mathbf{M}_{12}$ a partir da matriz essencial. Cipolla e Giblin [1999] recomendam uma técnica na qual se decompõe a matriz essencial em valores singulares na forma (3-80)

$$\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^T. \qquad \textbf{(3-80)}$$

O menor valor singular deve ser nulo (posto 2) e os demais devem ser idênticos. A matriz essencial mais próxima da original e que preserva as condições impostas é obtida forçando os novos dois maiores auto-valores terem o valor da média dos dois maiores auto-valores anteriores e o menor ser nulo. A decomposição da matriz essencial em termos de (3-46) pode ser obtida utilizando (3-81).

$$\mathbf{S} = \mathbf{U}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{U}^T \qquad \textbf{(3-81)}$$

$$\mathbf{R} = \mathbf{U}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\mathbf{V}^T$$

A matriz $\mathbf{M}_{12}$ é então obtida diretamente por (3-82), sendo desconhecida a norma de $\mathbf{t}$, isto é, a linha de base.

$$\mathbf{M}_{12} = \left[\begin{array}{ccc:c} & \mathbf{R}_{3\times3} & & \mathbf{t} \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]_{4\times4} \qquad \textbf{(3-82)}$$

São possíveis quatro soluções para $\pm\mathbf{t}$ e $\mathbf{R}$ ou $\mathbf{R}^T$, cuja ambigüidade pode ser facilmente resolvida assegurando-se que os pontos reconstruídos estejam à frente das câmeras.

3.4 Estimação de pose de um contorno pelo método dos momentos

O método dos momentos é uma forma de descrever a geometria de feições regionais de forma livre. Baseado em agregações dos valores das coordenadas dos pontos, os momentos podem ser rapidamente calculados. Há uma relação direta destas medidas com a pose da curva sob transformações de similaridade. Uma vez normalizadas, podem ser determinados descritores invariantes da curva baseados nestas medidas.

3.4.1 Momentos de Hu

Os momentos estatísticos de Hu, conforme descrito em [Gonzalez e Woods, 1992] são dados pela equação (3-83). Podem ser generalizados para dimensões menores ou maiores que três. A ordem dos momentos corresponde ao grau da expressão do integrando de (3-83), ou seja, à soma dos índices p, q e r. Assim, o momento de ordem zero corresponde ao volume do sólido definido pela função binária $b(X,Y,Z)$. Os momentos de ordem um correspondem às coordenadas do baricentro do sólido. Os momentos de ordem dois estão relacionados à dispersão. Normalmente se utilizam momentos até grau dois. Não se recomenda utilizar momentos de ordem maior por serem mais sensíveis ao ruído.

$$m_{pqr} = \iiint b(X,Y,Z) X^p Y^q Z^r \, dX dY dZ \qquad \textbf{(3-83)}$$

Existem outros tipos de momentos que são utilizados no problema de alinhamento e de reconhecimento, entre eles citamos momentos centrais e momentos ortogonais. Apesar de serem medidas mais estáveis numericamente e muito utilizadas para computar invariantes não são úteis para medir a pose justamente por serem normalizados e independentes da pose.

Para regiões planas, é possível se utilizar o teorema de Green para calcular os momentos do interior baseado em integrais ao longo do contorno. No caso de contornos poligonais, essa integração resulta num somatório para os lados do polígono. A área de uma região pode ser calculada pela equação (3-84) ao longo do seu contorno, para coordenadas 2D

$$m_{00} = \oint x \, dy \, . \qquad \textbf{(3-84)}$$

O centróide é dado por (3-85)

$$B = \frac{1}{m_{00}} \left(C_x, C_y \right), \quad \text{onde} \qquad \textbf{(3-85)}$$

$$C_x = \oint \frac{x^2}{2} dy = -\oint xy \, dx \ .$$

$$C_y = \oint xy \, dy = -\oint \frac{y^2}{2} dx$$

A forma geral dos momentos de área sobre um plano calculados pela integral ao longo do contorno é dada pela equação (3-86)

$$m_{pq} = \frac{1}{p+1}\oint x^{p+1} y^{q} dy = -\frac{1}{q+1}\oint x^{p} y^{q+1} dx . \qquad \textbf{(3-86)}$$

No caso de uma curva que não é plana, ajusta-se um plano pelos pontos da curva e se projeta a curva sobre esse plano para poder aplicar essa técnica. Outra forma de tratar curvas não-planas é descrita a seguir.

3.4.2 Momentos modificados

Com a finalidade de estender em trabalhos futuros a aplicação dos momentos ao contorno, e para curvas que não delimitam regiões planas, redefinimos os momentos de Hu para efetuar medidas sobre o comprimento da curva ao invés de seu interior. Por exemplo, o momento de ordem zero é normalmente considerado a área do interior da curva, mas na nossa definição é o perímetro da região. Assim, os momentos que utilizamos são descritos por (3-87)

$$m_{pqr} = \int_{\Gamma} X^{p} Y^{q} Z^{r} d\gamma . \qquad \textbf{(3-87)}$$

Essa forma para os momentos é especialmente importante pelo fato de que um contorno não define unicamente um pedaço de superfície no espaço.

3.4.3 Determinação de um sistema de coordenadas baseado na geometria da curva tridimensional

A normalização da curva tridimensional é uma seqüência de transformações que leva os pontos da curva para um sistema de coordenadas padrão definido a partir da forma da curva. Para transformações rígidas, determinar esse sistema de coordenadas equivale a determinar a pose da curva de modo que curvas com formas semelhantes estarão definidas num sistema de coordenadas semelhante. Uma vez normalizadas as curvas, seus momentos de ordem mais alta podem ser recalculados a fim de prover uma descrição. Efetivamente, outros métodos de descrição também podem ser utilizados após a normalização. Descrevemos nos sub-itens, a

seguir, o processo de normalização da curva, apresentando, uma a uma, as transformações geométricas.

- **Transladar baricentro à origem**

Os momentos de primeira ordem fornecem as coordenadas do baricentro da curva. Uma boa estratégia de normalização é transferir a origem do sistema de referência para o baricentro, criando assim, um sistema independente da posição da curva ou invariante à translação. A transformação para esse caso é dada pela matriz de translação da equação (3-88)

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & \dfrac{-m_{100}}{m_{000}} \\ 0 & 1 & 0 & \dfrac{-m_{010}}{m_{000}} \\ 0 & 0 & 1 & \dfrac{-m_{001}}{m_{000}} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \qquad \textbf{(3-88)}$$

- **Direção de mínima dispersão no espaço**

Para fins de simplificação, consideramos recalculados os momentos com o baricentro na origem, desejando agora determinar a orientação do plano da curva. Os momentos de segunda ordem são uma medida de inércia ou de dispersão de modo que procuramos a direção na qual a dispersão seja mínima. A expressão da dispersão é dada pela função energia definida na equação (3-89)

$$E = \int_{\Gamma} (aX + bY + cZ)^2 \, d\gamma . \qquad \textbf{(3-89)}$$

Se minimizarmos E em relação aos coeficientes do vetor $[a \quad b \quad c]^T$, obteremos a direção da normal ao plano de dispersão mínima. Para isso, calculamos o gradiente de E. Entretanto não basta igualá-lo a zero porque nosso problema de otimização é restrito dado que a equação (3-89) é definida para vetores $[a \quad b \quad c]^T$ unitários. Em (3-90), temos a derivada parcial de E em relação ao coeficiente a. Para os demais coeficientes, a expressão é análoga.

$$\begin{aligned}\frac{\partial E}{\partial a} &= \int_{\Gamma} 2\left(aX + bY + cZ\right)Xd\gamma \\ &= 2a\int_{\Gamma} X^2 d\gamma + 2b\int_{\Gamma} XYd\gamma + 2c\int_{\Gamma} ZXd\gamma \\ &= 2am_{200} + 2bm_{110} + 2cm_{101}\end{aligned} \tag{3-90}$$

O gradiente de E pode ser escrito utilizando-se a matriz dos momentos de segunda ordem, como mostramos em (3-91)

$$\nabla E = 2\begin{bmatrix} m_{200} & m_{110} & m_{101} \\ m_{110} & m_{020} & m_{011} \\ m_{101} & m_{011} & m_{002} \end{bmatrix}\begin{bmatrix} a \\ b \\ c \end{bmatrix}. \tag{3-91}$$

A restrição à minimização para termos um vetor unitário é dada por (3-92)

$$a^2 + b^2 + c^2 = 1. \tag{3-92}$$

A função g é definida de forma que seus zeros correspondam à satisfação da restrição (3-92)

$$g(a,b,c) = a^2 + b^2 + c^2 - 1 = 0. \tag{3-93}$$

Pelo método dos multiplicadores de Lagrange, os mínimos e máximos de E para vetores unitários $\begin{bmatrix} a & b & c \end{bmatrix}^T$ são obtidos pelo sistema (3-94)

$$\begin{cases} g(a,b,c) = 0 \\ \nabla E = \lambda \cdot \nabla g(a,b,c) \end{cases}. \tag{3-94}$$

O gradiente de g é dado por (3-95)

$$\nabla g = \begin{bmatrix} 2a \\ 2b \\ 2c \end{bmatrix}. \tag{3-95}$$

O sistema resultante, substituindo (3-95) e (3-91) é dado por (3-96)

$$\begin{cases} a^2 + b^2 + c^2 = 1 \\ \begin{bmatrix} m_{200} & m_{110} & m_{101} \\ m_{110} & m_{020} & m_{011} \\ m_{101} & m_{011} & m_{002} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \lambda \begin{bmatrix} a \\ b \\ c \end{bmatrix} \end{cases}. \quad \textbf{(3-96)}$$

Temos, assim, um problema de determinação de auto-vetores. Os resultados para λ correspondem aos autovalores da matriz dos momentos. Para esse sistema, é possível a existência de até 3 soluções que consistem dos múltiplos dos auto-vetores normalizados.

As matrizes de rotação para que a normal ao plano de mínima dispersão coincida com o eixo Z são dadas por (3-97) e (3-96). Em (3-97), efetuamos a rotação da normal ao plano de mínima dispersão até que coincida com o plano XZ

$$\mathbf{R}_1 = \mathbf{R}_Z(-\theta) \quad \textbf{(3-97)}$$
$$\cos\theta = \frac{a}{\sqrt{a^2 + b^2}}.$$
$$\text{sen}\,\theta = \frac{b}{\sqrt{a^2 + b^2}}$$

Em (3-98) rodamos o vetor obtido até que coincida com o eixo Z

$$\mathbf{R}_2 = \mathbf{R}_Y(-\varepsilon) \quad \textbf{(3-98)}$$
$$\cos\varepsilon = c \quad .$$
$$\text{sen}\,\varepsilon = \sqrt{a^2 + b^2}$$

- **Eixo de mínima inércia no plano**

De forma equivalente, procuramos um eixo em que a projeção no plano de mínima dispersão tenha mínima inércia. Para isso, utilizamos uma generalização do método que acabamos de descrever a fim de aplicá-lo em duas coordenadas. Consideramos que os momentos calculados correspondem aos momentos no plano XY, que agora é o plano de mínima dispersão. A solução é dada pela equação (3-99). Este método é equivalente ao método apresentado por Horn [1989]

$$\begin{cases} a^2 + b^2 = 1 \\ \begin{bmatrix} m_{20} & m_{11} \\ m_{11} & m_{02} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \lambda \begin{bmatrix} a \\ b \end{bmatrix}. \end{cases} \quad \textbf{(3-99)}$$

Novamente, temos um problema de determinação de auto-vetores. Neste caso, há possibilidade de duas soluções, que são os auto-vetores normalizados. A matriz de rotação é dada por (3-100)

$$\mathbf{R}_3 = \mathbf{R}_Z\left(\frac{\pi}{2} - \theta\right) \quad \textbf{(3-100)}$$
$$\begin{matrix} \cos\theta = a \\ \text{sen}\,\theta = b \end{matrix} \quad .$$

3.4.4 Resultados do método dos momentos

Para estimar a pose da curva em relação ao referencial absoluto, deve-se compor a transformação devida à normalização da curva observada com a transformação inversa da normalização da curva do modelo. Vê-se que a normalização de uma curva é uma transformação do sistema de coordenadas da câmera para o sistema de coordenadas da curva. Assim, se duas curvas obtidas de duas vistas são reconhecidas idênticas, a transformação entre as duas vistas corresponde à composição da transformação de uma vista ao sistema do objeto com a transformação do sistema do objeto até a outra vista, como na figura 3-8.

Os momentos são também úteis no problema de *shape-matching*, onde são utilizados para normalizar uma feição antes de descrevê-la por outro método de descrição. A descrição da forma das feições pode ser feita pelo próprio método dos momentos, utilizando momentos de ordem mais alta, mas, como esses momentos são ruidosos, é aconselhável utilizar algum outro método de descrição e identificação, como descritores de Fourier.

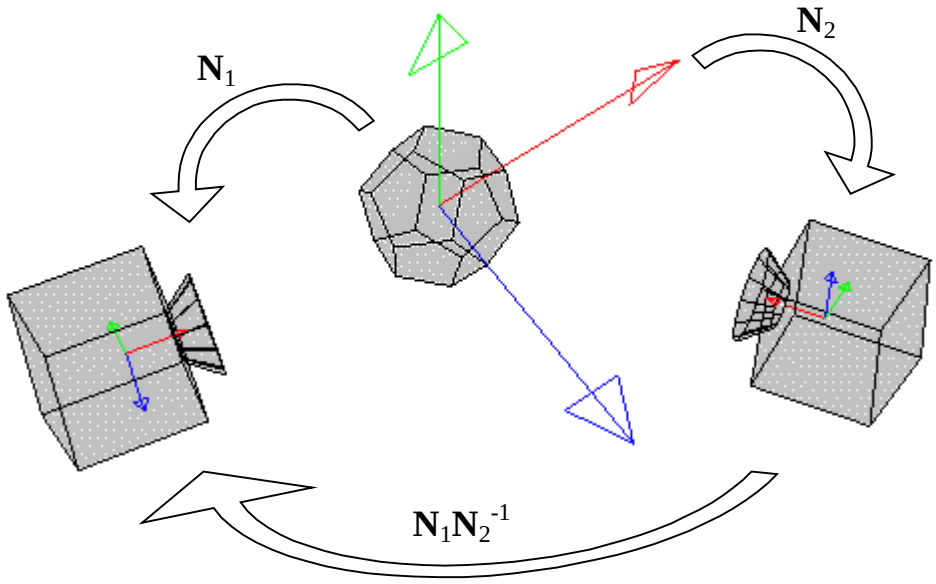


Figura 3-8 – Composição de transformações.

3.5 Reconhecimento e pose por métodos de aglomeração

Para reconhecer feições da imagem, isto é, determinar suas correspondências com feições do modelo, é necessário determinar atributos que não dependem da transformação geométrica do modelo para a imagem. Esses atributos podem ser individuais ou estruturais. Atributos individuais de uma feição são, por exemplo, sua cor e sua forma. Os métodos apresentados nesta seção exploram atributos estruturais, isto é, atributos de coleções de feições. Exemplos de atributos estruturais são distâncias e ângulos entre feições.

Todas coleções não degeneradas de correspondências entre feições da imagem e do modelo, possuem uma propriedade comum que consiste na própria transformação geométrica da formação da imagem. Assim, há uma relação de compatibilidade que é respeitada pelas correspondências corretas, permitindo que se avalie uma hipótese de correspondências quanto à compatibilidade com as demais hipóteses levantadas. Por exemplo, se a transformação procurada preserva distâncias e se conhece a distância de A para B na imagem, a distância entre as feições correspondentes a A e a B no modelo deve ser a mesma a não ser por imprecisão.

Para cada tipo de transformação, existe um número mínimo de feições de um objeto transformado que pode ser utilizado para se determinar unicamente a transformação. Cada conjunto de pontos com número mínimo e configuração adequada para se determinar a transformação dadas suas correspondências é chamado de base. Consideremos alguns casos. Para feições pontuais em duas dimensões, sujeitas apenas à transformação de translação, são necessários dois parâmetros para definir de forma única essa transformação: x e y. Assim, basta saber as coordenadas de um dos pontos para se determinar a transformação. Pela restrição de compatibilidade, independentemente dos pontos utilizados, o resultado é a mesma transformação.

Considerando agora que as feições pontuais no plano são sujeitas às transformações rígidas, são necessários três parâmetros – dois para translação (x e y) e um para rotação (θ) – para se determinar completamente a transformação. Assim, é necessário estabelecer a correspondência de um conjunto de pelo menos dois pontos para que se possa determinar a transformação. Todas as correspondências dos demais pontos mantém uma relação de compatibilidade com a correspondência desse conjunto de pontos.

A base para feições pontuais sob transformações de similaridades no plano tem tamanho 2. Para transformações afins, a base tem tamanho 3. Para a transformação projetiva planar, a base tem tamanho 4. Um algoritmo de busca exaustiva dada uma base de tamanho b considera $O(I^b M^b)$ hipóteses, para I feições da imagem e M feições do modelo. Assim, enquanto a complexidade para a transformação projetiva é pelo menos $O(I^4 M^4)$, para a transformação rígida em 2D é pelo menos $O(I^2 M^2)$.

Para correspondências de pontos no espaço tridimensional, a base para transformações rígidas tem tamanho 3. Uma possível base de tamanho 2 para a transformação rígida em 3D é formada por duas retas não paralelas [Wolfson e Rigoutsos, 1997]. Para similaridades em 3D, uma base pode ser formada por um ponto e uma reta ou então por 3 pontos.

Os métodos apresentados a seguir utilizam os conceitos e propriedades mencionados e realizam simultaneamente reconhecimento das feições e estimação de pose. A correspondência imprecisa, com feições faltando por não detecção ou por oclusão e com feições espúrias

detectadas erroneamente devido ao ruído ou à distração, é resolvida nesses métodos através de um processo de votação. O consenso das múltiplas medidas dos atributos das feições vai revelar a solução mais provável, ou seja, o conjunto de hipóteses com melhor sustentação. Medidas refinadas da pose são obtidas, selecionando das feições segmentadas, o conjunto que resulta no melhor alinhamento.

3.5.1 Aglomeração de poses

Aglomeração de poses (*pose clustering*), que é uma técnica inspirada nos métodos baseados no espaço de Hough, consiste de um esquema de votação para determinar a pose e a identidade de objetos baseado em modelos. Para cada conjunto de hipóteses de correspondências, computa-se a pose e acrescenta-se um voto para aquela pose. Para isso, o espaço de poses é dividido em um número finito de células. Para transformações rígidas em 3D, é uma técnica custosa se considerar as $O(I^3M^3)$ hipóteses de correspondência onde M corresponde ao número de feições do modelo e I ao número de feições detectadas na imagem. Olson [1994] apresenta um método eficiente que toma tempo $O(I^3M)$ e em espaço de memória $O(IM)$, baseado em consenso de amostras aleatórias (RANSAC).

Partindo da correspondência de 3 pontos da imagem com 3 pontos do modelo, é possível determinar a pose, mas não unicamente, dado que existem 4 possíveis soluções para a pose nessas condições (veja apêndice B).

Seguindo a nomenclatura de Olson [1994], chamamos um conjunto de 3 feições do modelo $\{\mu_1, \mu_2, \mu_3\}$ de um grupo-modelo e um conjunto de três pontos da imagem $\{\nu_1, \nu_2, \nu_3\}$ de um grupo-imagem. Uma correspondência hipotética entre uma única feição-modelo e uma única feição-imagem $\pi = (\mu, \nu)$ é chamada correspondência pontual, um conjunto de três correspondências pontuais com feições distintas tanto da imagem quanto do modelo $\gamma = \{ (\mu_1, \nu_1), (\mu_2, \nu_2), (\mu_3, \nu_3) \}$ é chamada correspondência de grupo.

Para m feições-modelo e n feições-imagem, há $6\binom{m}{3}\binom{n}{3}$ correspondências de grupo distintas. Cada correspondência está relacionada a duas ou quatro possíveis transformações. Fazendo-se um histograma no espaço das poses, é esperado que poses com número mais

freqüente de ocorrências, formando agrupamentos (*clusters*) no espaço de pose, correspondam às reais transformações dos objetos no espaço.

Há o problema de que o espaço de pose é hexadimensional, de forma que para obter os *clusters* é necessário ponderar precisão, robustez e custo no projeto da tabela de votos. Através das técnicas de histogramas com seções largas, de representação por multi-escala hierárquica e de projeções em sub-espaços ortogonais, por exemplo, é possível criar tabelas de votos factíveis considerando espaço em memória. No caso de seções largas ou projeções, o problema da presença de um número grande de elementos da cena não relacionados ao objeto (*clutter*) pode levar a imprecisões quanto à pose e ao reconhecimento, podendo ser necessário examinar um número muito grande de seções para remediar esse problema, o que, por sua vez, implica um custo computacional maior. Representações hierárquicas podem ser compactas e dinâmicas, mas por outro lado implicam maior custo computacional.

O algoritmo eficiente proposto por Olson [1994] é apresentado a seguir:

Repita os seguintes passos determinado número de iterações
  Escolha duas feições aleatórias ν_1 e ν_2 da imagem.
  Para toda feição μ_1 do modelo,
    Para toda feição μ_2 do modelo diferente de μ_1,
      Para toda feição μ_3 do modelo diferente de μ_1 e μ_2,
        Para toda feição ν_3 da imagem
          Determine as poses que alinham a correspondência de grupo $\gamma = \{ (\mu_1, \nu_1), (\mu_2, \nu_2), (\mu_3, \nu_3) \}$
      Encontre agrupamentos de clusters e retorne-os
Fim

Algoritmo 3-1 - Método de Olson

Analisando o algoritmo 3-1, a complexidade de cada iteração é $O(IM^3)$, entretanto, pelo método de Fischler e Bolles [1981], são necessárias $O\left(\frac{I^2}{M^2}\right)$ interações, de forma que a complexidade total do algoritmo é $O(I^3M)$.

3.5.2 Método de espalhamento geométrico

O método de espalhamento geométrico (*geometric hashing*) é conhecido como um método de indexação porque representa identificações indexadas por seus atributos. O método é descrito detalhadamente e com exemplos em [Wolfson e Rigoutsos, 1997]. A grande vantagem do método é utilizar um maior espaço de memória para permitir reconhecimento muito rápido. Além disso, este método é robusto a oclusões e a feições espúrias. Se uma feição falsamente detectada for escolhida para formar a base das feições da imagem, o método deve produzir uma votação pouco conclusiva.

Alguns problemas do método incluem a imprecisão das medidas e a dimensionalidade de algumas transformações. Para medidas muito imprecisas, o intervalo de abrangência dos índices da tabela deve ser maior, o que resulta em um número maior de colisões. Para transformações que dependem de bases com maior dimensão, o número de itens da tabela pode crescer muito.

O método de espalhamento geométrico conta com as seguintes estruturas: uma lista de modelos de objetos, uma coleção de feições detectadas na imagem, uma tabela de espalhamento (*hash*) e uma tabela de votos. Cada modelo da lista de modelos de objetos contém uma coleção de feições-modelo. A lista de feições-imagem consiste das feições a serem reconhecidas. A tabela de espalhamento é normalmente bidimensional ou de dimensão mais elevada e pode haver colisões de forma que várias entradas da tabela podem ser indexadas pela mesma chave. A tabela de votos é um histograma para a identificação da base de feições-imagem.

O método é dividido em duas fases: preparação e reconhecimento. Na fase de preparação, uma tabela *hash* é preenchida, indexando rótulos que identificam feições do modelo a partir de atributos invariantes ao tipo de transformação geométrica considerado, que são construídos a partir de coordenadas normalizadas das feições. Cada possível base no modelo é transformada para um sistema de referência intermediário e às demais feições se aplica a mesma transformação. As coordenadas dessas outras feições transformadas consistem em atributos invariantes e são utilizadas para indexar a tabela *hash*. Para cada feição fora da base,

armazena-se, na posição da tabela definida por seus atributos, a identificação da base considerada.

Na fase de reconhecimento, os mesmos atributos para o modelo são medidos para as feições da imagem. Escolhem-se algumas feições detectadas para a formação de uma base e aplica-se a transformação para o sistema intermediário a todas as feições detectadas. O índice correspondente a esses atributos na tabela *hash* aponta para a lista das bases do modelo que mais provavelmente correspondem à base considerada da imagem. O consenso de informação proveniente das feições da imagem é feito através de uma votação. Para cada hipótese encontrada de identificação da base produz-se um voto para a base candidata. A base encontrada mais freqüentemente é a mais provável de ser a identificação procurada. Compõe-se, então, a transformação da base do modelo para o sistema intermediário com a inversa da transformação da base da imagem para o sistema intermediário, obtendo a transformação do objeto e, conseqüentemente, sua pose. Essa medida da pose pode ser refinada considerando um número maior de feições além da base para estimação por quadrados mínimos. Se a votação não for conclusiva, é selecionada uma nova base na imagem, repetindo o procedimento.

Olhemos mais a fundo a descrição do método de espalhamento geométrico de Wolfson e Rigoutsos [1997], ilustrado nas figuras 3-9 e 3-10. Vamos supor inicialmente um modelo por pontos bidimensionais e que a classe de transformações possíveis seja a classe das similaridades euclidianas no plano. A base é uma tupla ordenada de feições de um modelo com mínimo número de elementos e suficiente para definir unicamente um sistema de referência. O tamanho da base depende da classe de transformações e no caso é 2. Se o modelo M de um objeto é constituído pelos pontos p_1, p_2, p_3, p_4, p_5, há 20 possíveis bases. Tomando a base (p_1, p_4) por exemplo, uma única transformação T_{14} leva o ponto médio de $\overline{p_1 p_4}$ à origem e orienta $\overrightarrow{p_1 p_4}$ com o eixo x. Dessa forma, $T_{14}(p_1) = \left(-\frac{1}{2}, 0\right)$ e $T_{14}(p_4) = \left(\frac{1}{2}, 0\right)$. As transformações dos demais pontos $\{T_{14}(p_2), T_{14}(p_3), T_{14}(p_5)\}$ caracteriza o modelo M, desde que se conheça qual ponto corresponde a p_1 e qual a p_4.

O conjunto de todas as transformações T_{ij} e os pontos transformados que não pertencem às bases caracterizam M independentemente da transformação e da escolha da base.

O algoritmo que faz uso desse fato pode ser construído da seguinte forma. Para cada possível base de M dada pelos pontos p_i, p_j computa-se a transformação T_{ij} e a aplica aos demais pontos p_k. As coordenadas de p_k transformadas são mapeadas num índice para uma tabela *hash* $h(x, y)$ ou $h(T_{ij}(p_k))$. Nessa tabela pode haver colisões. Na entrada indexada da tabela, armazena-se a tupla (M, i, j) conforme (3-101)

$$h(T_{ij}(p_k)) \leftarrow h(T_{ij}(p_k)) \cup \{(M, i, j)\}. \qquad \textbf{(3-101)}$$

Isso é feito para cada base e cada ponto fora da base. Esse é um processo *off-line* de complexidade $O(M^{b+1}H)$, onde M é o número de feições de cada modelo, b é o tamanho da base e H é o custo de acesso à tabela *hash*.

O reconhecimento do modelo a partir de feições da imagem é feita eficientemente *on-line*. Escolhem-se dois pontos p_u, p_v da imagem como base, determina-se a transformação T_{uv} normalizadora e se aplica a transformação aos demais pontos p_w da imagem. A tabela de *hash* é indexada por $h(T_{uv}(p_w))$. Compõe-se um histograma de votos para toda entrada obtida da tabela.

O resultado do histograma de votos permite selecionar as melhores hipóteses para a base do modelo correspondente à base selecionada da imagem. É necessário proceder com uma etapa de verificação [Wolfson e Rigoutsos, 1997]. Uma vez selecionada uma base e um modelo mais prováveis, determina-se uma aproximação para a transformação por mínimos quadrados. Se a transformação obtida sobre a imagem não for adequada, deve-se escolher um novo par de pontos da imagem como base e reiniciar o processo de identificação.

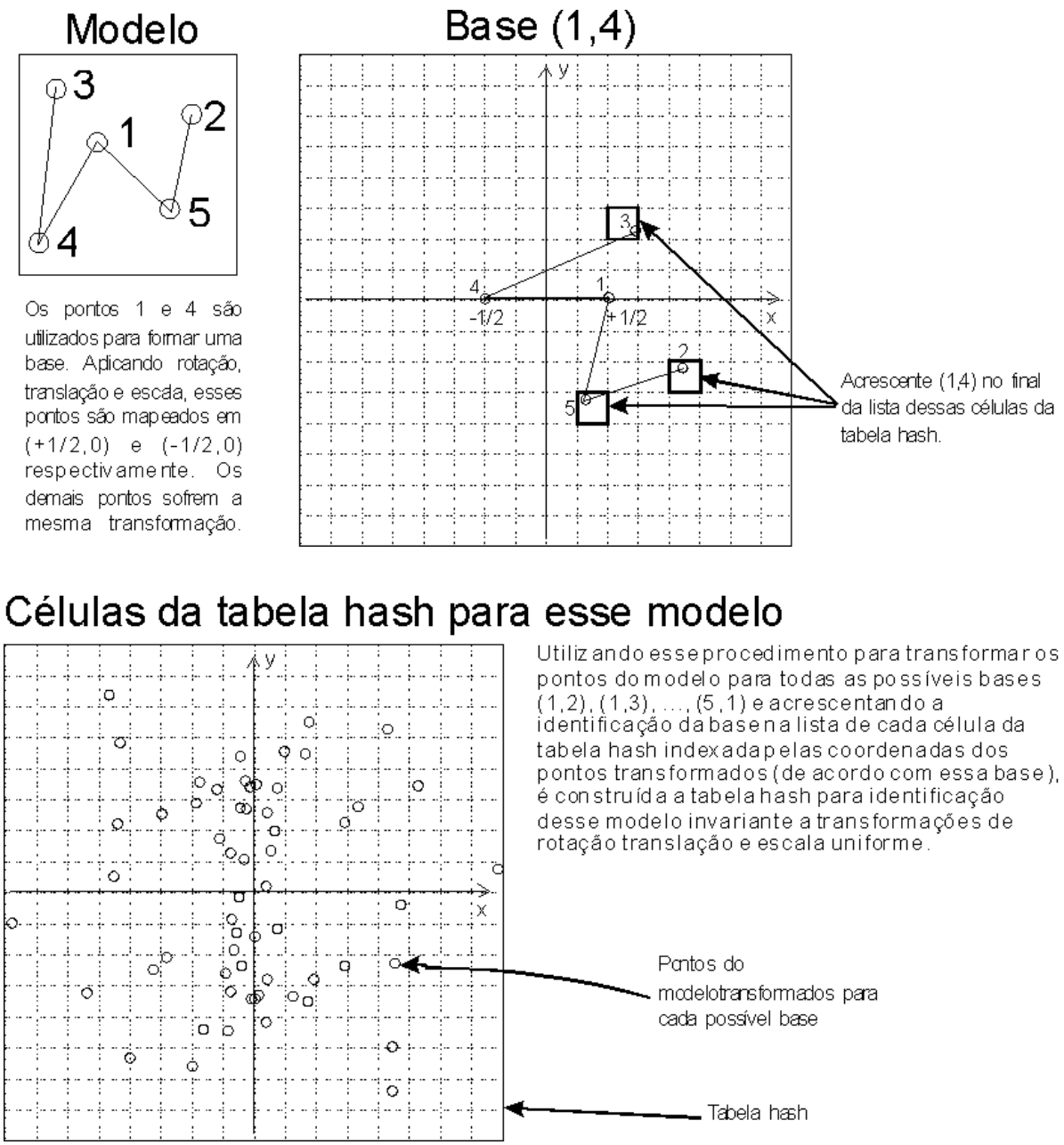


Figura 3-9 – *Hashing* geométrico – construção da tabela *hash*.

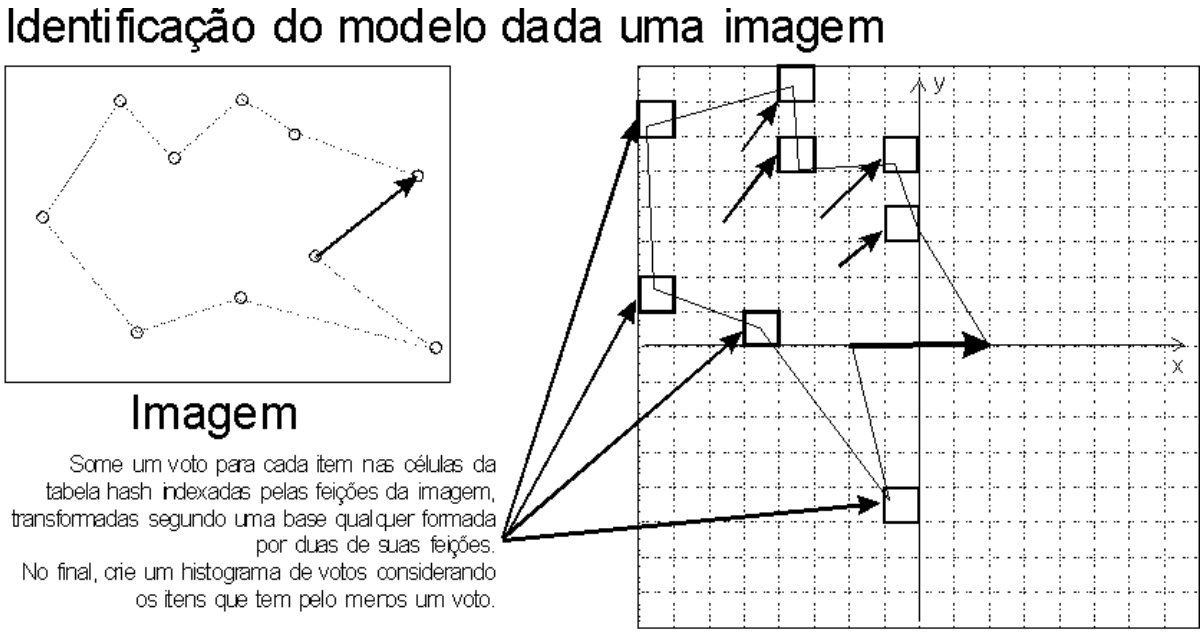


Figura 3-10 – *Hashing* geométrico – identificação.

A complexidade do algoritmo de *hashing* geométrico é $O(I^{b+1}H)$ no pior caso, onde I é o número de feições da imagem, H é o custo de manipulação da tabela *hash* e b é o tamanho da base. São testadas $O(I^b)$ possíveis bases, havendo eventual possibilidade de se obter o resultado esperado já para a primeira base testada. Para cada uma das $O(I)$ feições da imagem, são gerados $O(H)$ votos a partir da tabela *hash*. Estimamos que o custo da votação no pior caso seja $O(IH\lg(IH))$, para um histograma estruturado em árvore, onde H seria o número médio de colisões na tabela. No caso de seleção por *threshold* de votos, a complexidade da votação é $O(IH)$. A independência do número de feições-modelo faria esse método poderoso para bancos de dados com muitos modelos de objetos. Porém, na verdade, o número de colisões cresce proporcionalmente ao número de modelos, aumentando o custo.

3.5.3 Espalhamento geométrico e abordagem probabilística

A abordagem probabilística brevemente explicada em [Wolfson e Rigoutsos, 1997] permite um maior compromisso com a questão da imprecisão. No item anterior, a imprecisão é tratada considerando intervalos largos na tabela de espalhamento, aumentando o número de colisões.

Seria também adequado replicar entradas nas células vizinhas da tabela. Pela abordagem probabilística, um modelo de erro pode ser utilizado para determinar quais células da tabela devem ser preenchidas e os votos podem ter pesos. O esquema de votação pode ser visto como uma estimação de parâmetros por máxima verossimilhança.

Sejam os pontos da imagem modelados pelo conjunto $S = \{p_l\}_{l=1}^{I}$ (onde I é o número de feições-imagem) e a base $B = \{p_u, p_v\}$ e cada classe de objeto é representada por um modelo M_k. As feições fora da base compõem o conjunto $S' = S - B$. Desejamos computar o modelo e a base que sejam mais plausíveis dados os atributos das demais feições da imagem. Assim, procuramos a tupla (M_k, i, j, B) (onde i, j são índices para feições-modelo) que maximize a verossimilhança $P(S' | (M_k, i, j, B))$, isto é, a probabilidade das coordenadas das feições fora da base coincidirem com as coordenadas das feições do modelo transformadas de acordo com o conjunto de correspondências determinado pela tupla (M_k, i, j, B). Supondo independentes as feições p_l, podemos maximizar (3-102)

$$P(S' | (M_k, i, j, B)) = \prod_l P(p_l | (M_k, i, j, B)). \quad \textbf{(3-102)}$$

Trabalhando com log-probabilidades, temos (3-103)

$$\max \sum_l \log(P(p_l | (M_k, i, j, B))). \quad \textbf{(3-103)}$$

Para uma base B específica, a identificação procurada é dada por (3-104)

$$\arg\max_{(M_k, i, j)} \sum_l \log(P(p_l | (M_k, i, j))). \quad \textbf{(3-104)}$$

Isso é equivalente a encontrar o modelo e os pontos correspondentes à base para os quais existe o maior número de votos. Essa formulação por máxima verossimilhança permite tratar imprecisão através de votos com valores ponderados. Para isso, é necessário computar os votos para cada entrada da tabela de *hash* em uma vizinhança. O algoritmo torna-se mais robusto ao custo de eficiência.

3.6 Discussão

Os métodos da Visão Computacional apresentados neste capítulo são o fundamento da solução que propomos para o problema do alinhamento baseado na análise de imagens. O método de calibração de câmeras apresentado é necessário para a calibração do sistema e particularmente para construir os modelos 3D na fase de autoria. Esse método apresentado é também a base para o método de estimação de pose utilizado na nossa abordagem. Dado que o resultado da estimação de pose que propomos é uma matriz de calibração, que é útil para compor imagens reais com objetos virtuais, a fatoração dos parâmetros extrínsecos para o nosso modelo de câmera é importante para que a aplicação tenha a informação dos parâmetros da transformação rígida de fato. Veja em [Tsai, 1987] uma discussão mais profunda do problema de calibração.

A geometria epipolar é importante porque reconstruímos as feições em 3D a partir de um par estéreo de câmeras. Para isso, é necessário determinar as correspondências entre feições do par de imagens, o que, em geral, só é possível se for considerada a restrição epipolar. Além disso, a retificação é vista em nosso trabalho não só como um artifício para simplificação, mas permite o desenvolvimento de um método muito eficiente de correspondência entre feições. Dos contornos reconstruídos em 3D, uma descrição para as feições é formada pelo método dos momentos. O algoritmo proposto para a determinação de correspondências entre feições do modelo e da imagem segue o mesmo tipo de raciocínio dos métodos de reconhecimento de objetos apresentados, baseando-se em acumulação de votos.

Tendo descrito os conceitos e modelos fundamentais e métodos da Visão Computacional necessários, podemos, no capítulo 4, construir a nossa proposta de solução para o problema de alinhamento.

Capítulo 4 - UMA PROPOSTA DE SOLUÇÃO PARA O PROBLEMA DE ALINHAMENTO

Apresentamos, neste capítulo, a proposta de solução para o problema descrito no capítulo 1: o alinhamento automático de um modelo de uma cena real 3D com um modelo 3D observado a partir de imagens desta cena. Essa solução tem compromisso com requisitos de precisão, robustez e tempo e assume algumas condições como é descrito no item 4.1. A solução é composta por procedimentos para pré-processamento das imagens, análise do par estéreo, reconstrução 3D das feições, descrição das feições, reconhecimento das feições e estimação da pose. Apresentamos algoritmos para as principais etapas e avaliamos a viabilidade da abordagem em relação aos requisitos.

4.1 Detalhes das condições da solução

Antes de apresentar a solução proposta, é preciso deixar claro qual o contexto considerado no desenvolvimento dessa solução. Apresentamos em detalhes, a seguir, os requisitos considerados na construção da solução e as condições assumidas.

4.1.1 Definição dos Requisitos

Um dos requisitos é a **precisão**. Há aplicações para as quais não só a precisão do alinhamento no espaço tridimensional, mas a precisão sobre a imagem projetada medida em pixels é considerada. Por exemplo, existem aplicações como em edição de vídeo ou em realidade aumentada, em que a imagem adquirida precisa ser editada de forma coerente com a

geometria tridimensional da cena acrescentando objetos virtuais. Se utilizarmos o modelo de câmera para criar uma projeção do modelo tridimensional da cena sobre um filme bidimensional, os elementos geométricos da projeção devem coincidir com os elementos da imagem adquirida por uma câmera real. O uso de um número de feições maior do que o mínimo necessário para a determinação do alinhamento permite refinar a estimação, obtendo melhor precisão.

Outro requisito é a **execução em tempo restrito**. O requisito de tempo restrito é presente em muitas das aplicações principalmente relacionadas ao problema de rastreio. Também é necessário restringir o tempo para analisar longas seqüências de imagens. Para aplicações com realimentação visual, é necessário se produzir uma resposta, se possível, numa taxa de quadros que seja aceitável para o sistema de visão humano, que pode variar em ordem de 10 a 100 Hz. Nosso objetivo compreende propor a redução da complexidade do algoritmo de alinhamento para que tenha custo comparável ou inferior ao processo de segmentação de feições.

Consideramos, também, como requisito que a solução proposta seja uma solução automática, dispensando intervenções do usuário nas fases *on-line*. A intervenção do usuário é uma estratégia inconveniente e incompatível com a condição de tempo restrito ou no caso de um grande volume de dados. Entretanto, o requisito de **automação** cria dificuldades principalmente no processo de determinação de correspondências entre feições da imagem e do modelo.

Por fim, é necessário que a abordagem proposta resulte em sistemas **robustos**. Por um sistema robusto entendemos que ele deva ser insensível ou pouco sensível a efeitos externos, isto é, efeitos não modelados que ocorrem numa situação real. Num sistema de rastreio óptico, a robustez pode estar ligada, por exemplo, a problemas de variação de iluminação, oclusão, campo de visão, distração e amostragem. Descrevemos esses problemas a seguir.

Variações de iluminação podem modificar atributos e dificultar a segmentação da imagem e o reconhecimento de seus elementos. Nesse caso, o rastreador pode perder a pista do objeto porque sua aparência não é mais semelhante com a original. Variações de luminosidade

tornam mais difícil identificar feições, por exemplo, pois os atributos medidos diferem muito dos atributos modelados.

Oclusão pode ocorrer se um objeto opaco impedir a visibilidade de um objeto de interesse da cena. Partes importantes de objetos rastreados podem ser escondidas e enganar facilmente o rastreador. Um sistema robusto à oclusão é capaz de lidar com objetos parcialmente visíveis. Da mesma forma, a limitação do campo de visão da câmera pode tornar inviável a análise de alguns objetos mesmo que estes estejam parcialmente representados na imagem.

O problema de **distração** consiste em objetos não relevantes que causam confusão a algoritmos que interpretam a imagem por serem, esses objetos, semelhantes aos objetos de importância. Em rastreadores baseados em cor esse efeito é muito comum, se um objeto rastreado ultrapassa um objeto estático de mesmo atributo, é bem possível que o rastreador pare de perseguir o objeto em movimento, aderindo ao objeto estático.

Muitas vezes, em cenas muito poluídas visualmente ou em situações com ruído na imagem, o problema de distração não depende da existência de um objeto semelhante ao objeto perseguido. Acontece que a informação extraída do conjunto desses poluentes visuais tem chance de constituir uma configuração verossímil capaz de confundir o rastreador numa espécie de ilusão de óptica. A chance disso acontecer decresce com a quantidade de informação utilizada para estabelecer um consenso de forma que métodos locais são mais suscetíveis a esse efeito. É, então, necessário um método global no sentido de analisar a imagem como um todo e não apenas suas variações. Diz-se deste problema de distratores em extrema quantidade que as imagens apresentam ***clutter***.

Efeitos da amostragem também causam problemas. Considerando que o número de quadros capturados num intervalo de tempo é limitado por uma taxa de amostragem, um movimento abrupto de câmera pode impedir a análise correta de uma seqüência de imagens. Da mesma forma, a amostragem espacial aliada ao ruído pode gerar distorções nas formas ou alterar a topologia, por exemplo, unindo objetos distintos em uma única região conexa da imagem, dificultando a comparação com um modelo geométrico.

4.1.2 Hipóteses assumidas

A fim de cumprir os requisitos mencionados no item 4.1.1, assumimos na nossa abordagem um número de condições que são descritas a seguir.

- **Tipo de feição**

É importante na escolha do tipo de feição e na delimitação dos conjunto de transformações geométricas considerar a complexidade do algoritmo de correspondência resultante. Assim, feições regionais de forma livre são uma boa opção por ser possível determinar unicamente uma transformação rígida a partir de um agrupamento de apenas duas feições. Na representação que propomos, a detecção das feições se dá através da identificação de regiões conexas salientes da imagem consideradas facilmente segmentáveis.

Conforme mostramos no item 3.5, a escolha do uso de feições não-pontuais objetiva a obtenção de um método de reconhecimento com complexidade reduzida e pré-processamento simplificado. Devido ao menor tamanho da base, algoritmos envolvendo feições não-pontuais têm menor complexidade.

Os dados na tabela 4-1 justificam nossa opção de projeto. Nessa tabela, as duas primeiras colunas são opções do projeto e as demais são suas conseqüências. A opção por feições pontuais implica num algoritmo de maior complexidade pela necessidade de bases com maior número de feições. Para feições extensas, onde cada feição contribui com pelo menos 5 valores independentes para a determinação da pose, uma complexidade quadrática pode ser conseguida, analisando feições aos pares.

Trabalhar com feições extensas (isto é, não-pontuais) no caso monocular para produzir resultados em tempo real é proibitivo devido à deformação das feições projetadas, pois nesse caso, são necessários algoritmos que consideram a forma das feições. O problema do reconhecimento de regiões individualmente através de sua forma é difícil por constituir o problema de *shape-matching,* discutido no item 2.2.6. As principais dificuldades de trabalhar com a forma são os problemas de amostragem e os casos de ambigüidade, onde o processamento simultâneo de múltiplas regiões é claramente mais apropriado. Além disso,

para considerar feições naturais, não se pode esperar que qualquer região apresente vértices e pontos bitangentes.

tipo de feição	número de vistas	transformação	tamanho da base	complexidade
pontual	1	projetiva	4	$O(N^4)$
extensa	1	projetiva	1 ou mais	difícil tratar a feição deformada
pontual	2	rígida em 3D	3	$O(N^3)$, *matching* difícil entre vistas.
pontual	3 ou mais	rígida em 3D	3	$O(N^3)$, *matching* fácil entre vistas.
extensa	2	rígida em 3D	2	$O(N^2)$, *matching* fácil entre vistas.

Tabela 4-1 – Complexidade para tipos de feições e transformações.

Nesta tabela *O(N)* corresponde ao número de feições. Para feições da imagem, essa complexidade se reflete no tempo computacional. Para feições do modelo, essa complexidade influi principalmente o espaço em memória e indiretamente o tempo computacional.

Nota-se também que realizar o *matching* entre feições não-pontuais sob duas vistas é bem mais fácil que entre feições pontuais. Uma das vantagens de se utilizar feições não pontuais é que um maior número de atributos pode ser reconhecido para uma feição tomada individualmente, simplificando a identificação da feição.

- **Vistas**

Esta mesma análise justifica a opção por um sistema estéreo de câmeras. Com visão estéreo é possível estimar a profundidade, de maneira que se simplifica o processo de reconhecimento e se remove a deformação das feições. A consideração de apenas uma vista implica lidar com

a transformação projetiva, enquanto que com mais vistas é possível se trabalhar com a transformação rígida no espaço 3D visto que se tem uma estimação da profundidade. Algoritmos que consideram a deformação são mais custosos que o algoritmo de reconstrução 3D. O *matching* entre vistas para as feições regionais, que é o principal problema para empregar visão estéreo, é bastante simples para vistas retificadas ou com orientação relativa conhecida. A opção por um sistema estéreo ainda é justificada, no caso de realidade aumentada, pelo fato de que sistemas do tipo VST (*video see-through*) em que ambos os olhos têm sua percepção mediada empregam normalmente uma câmera para cada olho. Neste caso, a profundidade é também importante para efetuar operações de composição de imagens reais e sintéticas com realismo.

- **Câmera de orifício e orientação relativa conhecida**

O modelo de câmera de orifício é amplamente aceito para aproximar a formação de imagens nas câmeras mais usuais como máquinas fotográficas e câmeras de vídeo. Se considerarmos que os parâmetros internos da câmera não se alteram, a relação entre coordenadas da imagem e do modelo geométrico fica dependente apenas da pose da câmera no espaço.

A orientação relativa entre câmeras é uma informação importante para estabelecer a correspondência entre feições em duas vistas através da restrição epipolar. Dada a orientação relativa é possível aplicar transformações de retificação, o que simplifica muito o problema de correspondência estéreo. O conhecimento da orientação relativa pode ser obtido por um aparato mecânico ou simplesmente mantido constante por acoplamento rígido. Dessa forma, é ainda possível posicionar uma câmera em relação à outra de forma que as imagens obtidas sejam já retificadas.

- **Natureza do ambiente e da feição**

Nem sempre é possível modificar o ambiente e inserir marcas artificiais. É interessante tentar aproveitar marcas naturais da cena. Nesse caso, imagens da cena podem ser adquiridas por câmeras calibradas numa fase de autoria e, combinando as informações de várias fontes, o modelo que representa a cena física é construído. Acreditamos que as feições regionais planas

são muito comuns e podem aparecer naturalmente nos ambientes de trabalho para muitas aplicações.

- **Fase de preparação**

Assumimos que existe uma fase de preparação, permitindo particionar os algoritmos em fases *off-line* e *on-line*. A fase de preparação se constitui numa fase de autoria quando o modelo da cena não é conhecido inicialmente e deve ser formado a partir de imagens calibradas. Isso implica que os algoritmos devem ser eficientes para a fase *on-line*, enquanto que os algoritmos utilizados na fase *off-line* ou na fase de autoria podem ser mais complexos e tomar mais tempo.

O conhecimento do modelo geométrico da cena é necessário porque é a partir dele que serão obtidas correspondências com elementos das imagens. O modelo dos objetos da cena pode ser criado a partir da aquisição de imagens da cena das quais são selecionadas as feições utilizadas para reconhecimento. Os métodos apresentados aqui para analisar as imagens, reconstruindo e descrevendo as feições, também podem ser utilizados para formar uma ferramenta de autoria.

Uma implicação da necessidade de conhecimento prévio da cena é a impossibilidade do uso em aplicações em que não se tem acesso prévio ao ambiente. Para a maioria das aplicações, entretanto, é possível utilizar marcas naturais ou inserir marcas artificiais.

É necessário manter em memória uma representação da cena. Uma forma de representação deve ser procurada para que o processo de autoria seja simplificado para o usuário e que ao mesmo tempo permita uma execução eficiente e robusta. As feições regionais são aparentemente adequadas ao processo de autoria porque podem ser facilmente reconhecidas e apontadas pelo usuário.

- **Outras Restrições**

Levantamos também o problema de inversão de ordem em visão estéreo. Devido à paralaxe, pontos da curva mais próximos da câmera tem um deslocamento maior, podendo alterar a forma da curva e a seqüência em que as feições da curva são encontradas na

reconstrução 3D. Assim, é interessante acrescentar a hipótese de que não há auto-oclusão das regiões ou, de forma menos genérica, que as regiões são planares e de que não possa ser visto o verso de uma feição (figura 4-1). Sem essa restrição, a reconstrução tridimensional das feições é dificultada pois a possibilidade de inversão de ordem deve ser avaliada. Outra vantagem de considerar as regiões planares consiste da possibilidade de cálculo dos momentos de área a partir da integral no contorno, como é discutido no item 3.4 e da possibilidade de determinar fechos convexos planares para as regiões.

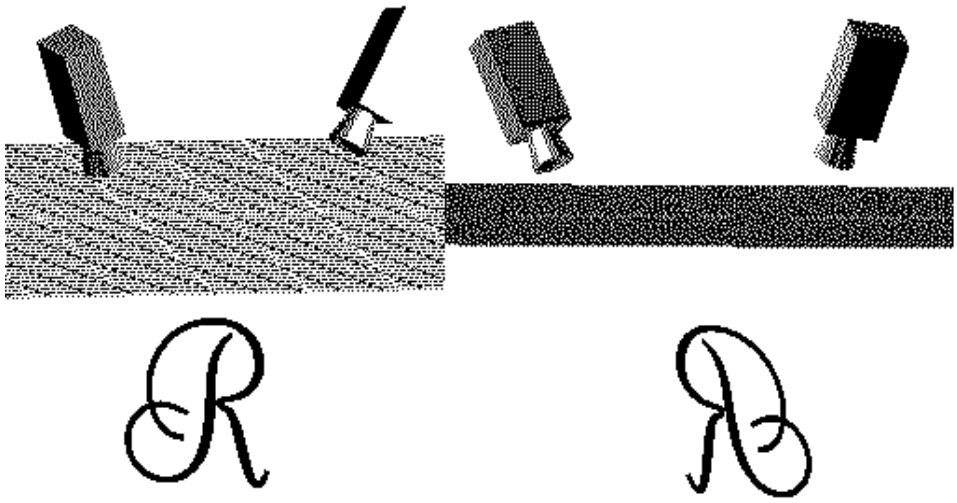


Figura 4-1 – Inversão de ordem em visão estéreo.

A feição, podendo ser vista dos dois lados, não preserva a ordem no eixo *x*, quando se alterna entre a imagem da câmera direita e da câmera esquerda.

Desconsideramos casos com cenas muito simétricas porque criam ambigüidades que não podem ser resolvidas pelo algoritmo de reconhecimento sem informação adicional. Ambigüidades podem ser resolvidas com experiências colaterais ao longo do tempo, como no problema da navegação robótica, embora não exploremos a questão da ambigüidade mais a fundo.

É esperado que as feições selecionadas sejam suficientes para caracterizar os objetos da cena permitindo discernimento entre os objetos a partir da percepção dessas feições. As feições de um mesmo objeto são caracterizadas pela rigidez, de forma que ângulos e distâncias

das feições de um objeto não podem ser alterados em relação ao modelo. Dessa forma, tratamos cenas rígidas ou objetos rígidos pelo menos no que concerne a feições.

O processo de segmentação é uma gargalo para a nossa abordagem. A suposição de feições facilmente segmentáveis é importante para garantir o desempenho do sistema proposto. Nesta tese, é feito um esforço no sentido de reduzir os requerimentos do processo de segmentação. Assim, no item 5.2, descrevemos um método bastante simples de segmentação por cor que é adotado nos experimentos. Esse método de segmentação também resolve, em parte, o problema de variações de iluminação.

4.2 Estrutura da Solução

Propomos uma abordagem dividida em várias etapas apresentadas na figura 4-2 numa forma hierárquica. O alinhamento é dividido em extração de informação das feições, correspondência entre modelo e imagem e consolidação da pose. A extração de informação das feições é, por sua vez, dividida em pré-processamento das imagens, reconstrução e descrição da pose. As fases do pré-processamento são segmentação, rotulação, traçado das bordas e determinação dos fechos convexos. As fases de reconstrução são correspondência estéreo das feições das imagens, correspondência estéreo dos pontos e reconstrução por estéreo.

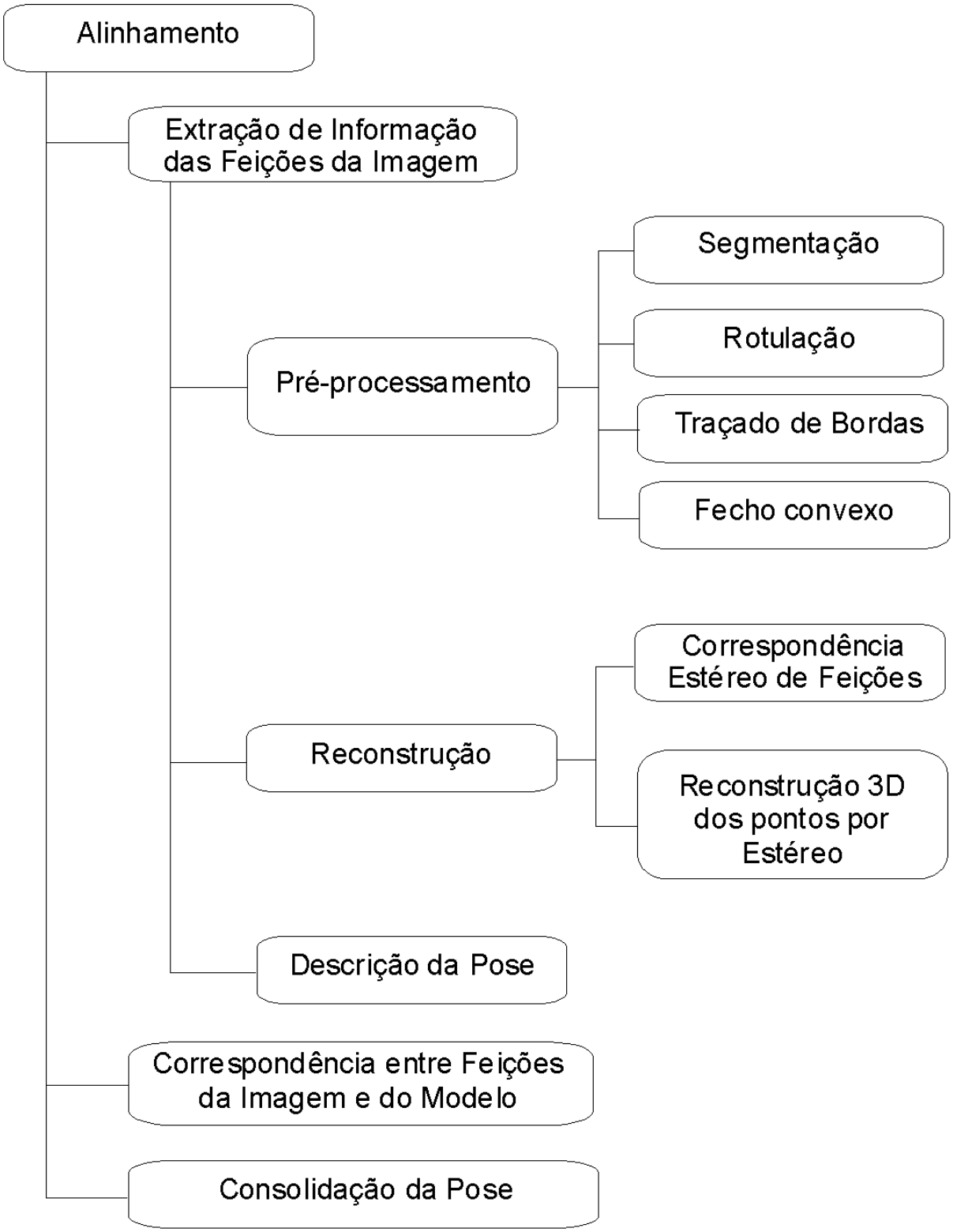


Figura 4-2 – Estrutura da solução.

- **Pré-processamento**

Como consideramos feições regionais, é necessário um processo para segmentar a imagem em regiões conexas e identificar quais regiões devem ser analisadas. O processo ideal de segmentação da imagem consiste na decomposição da imagem em regiões com significados distintos na cena correspondente. Por outro lado, a função do processo de segmentação pode ser bastante simples, classificando pixels de acordo com seus atributos locais. Entretanto, nem sempre o agrupamento de pixels semelhantes corresponde a pontos de um mesmo objeto da cena nem delimitam esse objeto, de forma que são muitas vezes necessários algoritmos mais sofisticados capazes de lidar com a imagem de forma global, por exemplo, em objetos com textura.

Uma vez que os pixels da imagem são classificados, é necessário delimitar regiões. Esse processo é chamado rotulação e considera uma relação de vizinhança entre pixels para reclassificá-los em função da conectividade. Pixels vizinhos que pertencem à mesma classe também pertencem à mesma região e, portanto, recebem o mesmo rótulo. Os rótulos atribuídos também devem garantir que pixels pertencem à mesma região se, e somente se, contêm valor de rótulo idêntico. Tendo cada região um rótulo único, esses rótulos podem identificar as regiões e, por exemplo, indexar uma tabela de atributos das regiões.

Tendo o conjunto de regiões delimitado, encadeamos os pixels do contorno de cada região e determinamos o fecho convexo que é o menor polígono convexo que contém todos os pontos da região. Representamos as regiões extraídas das imagens na forma de polígonos. Utilizar o fecho convexo facilita a reconstrução 3D e preserva as propriedades de invariância para o processo de reconhecimento.

- **Reconstrução 3D**

Um par estéreo de imagens consiste em duas vistas obtidas por câmeras de orifício com centros de projeção distintos. As duas imagens devem ser adquiridas simultaneamente, no sentido que não deve haver alteração da cena retratada nelas. O problema de correspondência

de feições entre vistas, ou *matching* estéreo, é o problema de relacionar feições presentes em vistas distintas da mesma cena.

A solução do *matching* estéreo permite fundir a informação de múltiplas câmeras obtendo uma medida de profundidade. A solução para este caso de *matching* usualmente adota restrições que limitam as possibilidades de correspondência, notavelmente a restrição epipolar. Para imagens retificadas, a restrição epipolar significa que pontos correspondentes estão na mesma coordenada y.

Uma operação importante, que visa simplificar a correspondência estéreo, é a retificação. A retificação consiste em determinar, para um par estéreo de imagens, um par de transformações projetivas a serem aplicadas na respectiva imagem de forma que os pontos homólogos das imagens transformadas se encontrem na mesma coordenada y, como visto no item 3.2.3. É possível gerar um par de imagens retificadas a partir de qualquer par de imagens estéreo quando se conhece a orientação relativa entre câmeras.

Em geral, o problema de correspondência é um problema mal-condicionado devido à falta de informação para a solução, de forma que há algoritmos que são dedicados a procurar a melhor solução, dados critérios de consistência e suavidade. Quando há informação suficiente, por exemplo, com o acréscimo de novas vistas, o problema é simplificado [Kanade *et al.*, 1995]. Mostramos no item 4.3.1 que o problema de correspondência estéreo de feições regionais é bastante simples no caso de imagens retificadas.

Reconstrução 3D significa recuperar a forma tridimensional da cena a partir de projeções. As feições regionais planares são reconstruídas em 3D de forma a se obter a informação de posição e orientação devida a cada feição. Para isso, é necessário apenas a determinação do contorno em 3D de seus fechos convexos.

Existem dispositivos capazes de obter diretamente a profundidade da cena. Existem também diversos métodos em Visão Computacional para resolver este problema, inclusive métodos ativos, que permitem estimar a profundidade da cena a partir de imagens. Cada método explora um tipo diferente de pista visual para a forma 3D. Adotamos o método de reconstrução por visão estéreo.

Utilizando uma estratégia de percorrer *scanlines* com y constante reconstruímos o contorno do fecho convexo para cada par de feições regionais homólogas cuja correspondência foi determinada no passo anterior. Cada linha intercepta os contornos dos fechos convexos em até dois pontos, que, por um processo de triangulação, têm sua profundidade determinada.

- **Descrição da pose**

A última etapa da extração de feições consiste na descrição da pose das feições. A descrição associa a cada feição um conjunto de atributos medidos. Esses atributos são importantes para auxiliar a determinação, num modelo tridimensional, da correspondência entre feições-imagem e feições-modelo. Além disso, contêm informação que é utilizada para a estimação da pose.

A escolha dos atributos depende do conjunto de possíveis transformações geométricas e radiométricas no processo de formação da imagem. Atributos adequados para resolver o problema de correspondência devem ser invariantes a essas transformações, enquanto que atributos para resolver a estimação de pose devem ser dependentes da transformação geométrica. Assim, cor, área, perímetro e centro de massa de regiões são atributos candidatos para descrição da cena (veja discussão no item 4.4).

Utilizamos as coordenadas 3D dos centros de massa e a direção normal a cada região como informação para a estimação de pose. Esses dados podem ser obtidos a partir dos contornos em 3D das regiões através do método dos momentos descrito no item 3.4. Para a determinação das correspondências, utilizamos atributos de pares de feições ao invés de atributos de feições individuais. Por exemplo, a distância entre os centros de massa ou o ângulo entre as normais de duas feições não se modificam sob transformações rígidas e podem ser utilizados para determinação das correspondências.

- **Correspondência entre feições da imagem e do modelo**

Seguindo o caso geral para métodos de alinhamento baseado em feições, uma vez extraídas informações das feições da imagem, é necessário identificá-las, correspondendo-as com feições do modelo. Abordamos este problema através de técnicas de reconhecimento de

objetos. Uma diferença entre correspondência e reconhecimento é que o primeiro relaciona um objeto observado e um modelo enquanto o segundo relaciona um objeto observado à sua classe de objetos. Outra diferença é que técnicas de reconhecimento consideram uma grande base de modelos enquanto apenas um modelo é utilizado no problema de correspondência.

É necessário desenvolver uma técnica de reconhecimento que leve em consideração os requisitos de automação, robustez e tempo impostos ao problema de alinhamento sem limitar a gama de aplicações. Construímos um método que explora a estrutura de compatibilidade mútua das posições e orientações das feições considerando as transformações de rigidez.

A construção do método de reconhecimento parte de especificar uma representação adequada tanto à fase de preparação ou autoria *off-line* como à fase de reconhecimento *on-line*. Na representação adotada, as feições são descritas por atributos individuais (cor) e por um conjunto de atributos de pares, como a distância entre centróides. Uma hipótese de identificação de um par de feições determina uma hipótese para a pose que pode ser avaliada testando se a transformação rígida definida por essa pose é coerente com as demais feições da imagem e do modelo.

Entretanto, não se deve fazer uma busca exaustiva por respeito às restrições de tempo. Sendo assim, adotamos um método de indexação no qual, a partir dos atributos de um par de feições, são obtidas as hipóteses de identificação dessas feições, isto é, quais são as possíveis feições do modelo que correspondem ao par de feições-imagem. É uma técnica baseada nos métodos de *hashing* (espalhamento) geométrico.

Para consolidar a informação das múltiplas feições encontradas na imagem e descartar as hipóteses que causam uma incompatibilidade com a restrição de rigidez, propomos um esquema de votação. Pelo consenso das hipóteses de identificação geradas pela análise de cada par de feições-imagem, é esperado obter a melhor hipótese de correspondência com as feições do modelo (isto é, a hipótese com melhor sustentação).

- **Consolidação da pose**

O reconhecimento é naturalmente associado à estimação de pose, como acontece, por exemplo, nos métodos de *hashing* geométrico. Isto acontece porque o resultado da

correspondência e o resultado da estimação de pose são intimamente relacionados. Uma hipótese para o conjunto de correspondências pode ou não gerar uma pose compatível com as demais evidências observadas. A hipótese correta para o conjunto de correspondências imagem-modelo deve levar a uma estimação correta da pose.

Preferimos, por outro lado, considerar a identificação das feições em uma etapa anterior à estimação da pose ordenando as hipóteses de correspondência segundo o número de evidências que as suportam. Uma vez determinada a feição do modelo correspondente a cada uma das feições da imagem, a estimação da pose é feita pela solução por quadrados mínimos de um sistema linear com coeficientes computados a partir das coordenadas dos centróides e direções normais das feições correspondentes. Métodos robustos como o RANSAC, que admite poda de amostras, podem ser utilizados para maior proteção e refinamento da estimação.

4.3 Da extração à descrição de feições

Consideramos que a fase de pré-processamento das imagens é dependente da aplicação e, portanto, é abordada no capítulo 5, que trata os experimentos realizados. No item 5.2, os algoritmos utilizados para pré-processamento das imagens são descritos.

Nos itens 3.2 e 3.3, mostramos os métodos básicos para análise de imagens estéreo. Neste item trabalhamos a análise de imagens, supondo conhecida a orientação relativa das câmeras e, portanto, a geometria epipolar. Consideramos que a informação de orientação relativa entre câmeras é dada na forma de uma matriz de transformação e se conhecem os parâmetros intrínsecos das câmeras. A partir dessa informação retificamos as coordenadas das imagens para determinar mais facilmente a correspondência entre as feições de ambas imagens.

Para cada imagem se tem uma lista de polígonos que representam o fecho convexo de cada região. Os elementos dessas listas são indexados por rótulos numéricos. A retificação é aplicada nas coordenadas dos vértices desses polígonos ao invés de cada pixel da imagem.

4.3.1 Correspondência estéreo das feições

Dadas duas imagens estéreo rotuladas é preciso determinar qual região de uma imagem corresponde a cada uma das regiões da outra, isto é, quais regiões do par de imagens representam o mesmo elemento da cena. Essa associação é necessária para o processo de reconstrução do contorno em 3D das feições. O resultado dessa etapa consiste em uma lista de pares de números cardinais que representam associações entre os rótulos das regiões da primeira imagem e os rótulos das regiões da segunda imagem. Essas correspondências ainda podem receber uma medida de qualidade e a lista pode ser ordenada a partir dessa medida.

A restrição epipolar diminui significantemente o espaço de busca. A partir de um par de imagens retificadas, a restrição epipolar ao *matching* de regiões é aplicada de forma eficiente. No caso de feições regionais, a restrição epipolar é mais reveladora por restringir as associações entre feições de acordo com pelo menos dois atributos.

Para regiões homólogas, os máximos e mínimos tanto globais quanto locais são pontos homólogos em imagens retificadas. No par de vistas retificadas, os pontos de máximo e os pontos mínimos em y tanto locais quanto globais de duas regiões homólogas podem ser pontos homólogos apenas se tiverem valor y idêntico. Veja figura 4-3.

Os extremantes globais são utilizados para determinar rapidamente as correspondências de regiões entre vistas, caracterizando cada feição conforme mínimo e máximo valor de y para a região.

A localização de máximos e mínimos pode ser imprecisa. Assim, é interessante estabelecer uma medida de compatibilidade para o *matching* baseado nos valores de y máximo e mínimo para cada feição, como por exemplo na equação (4-1), para a qual quanto menor o valor, melhor a compatibilidade

$$D_{12} = \sqrt{\left(y_{(1)\min} - y_{(2)\min}\right)^2 + \left(y_{(1)\max} - y_{(2)\max}\right)^2} \,. \qquad \textbf{(4-1)}$$

Uma possível implementação para o *matching* consiste em preencher uma matriz que relaciona os rótulos das duas imagens. Assim, para o rótulo i_1 da imagem 1 e para o rótulo i_2

da imagem 2, a medida de qualidade da correspondência é colocada na linha i_1 e coluna i_2. Assim, para cada linha pode se escolher a coluna de melhor qualidade.

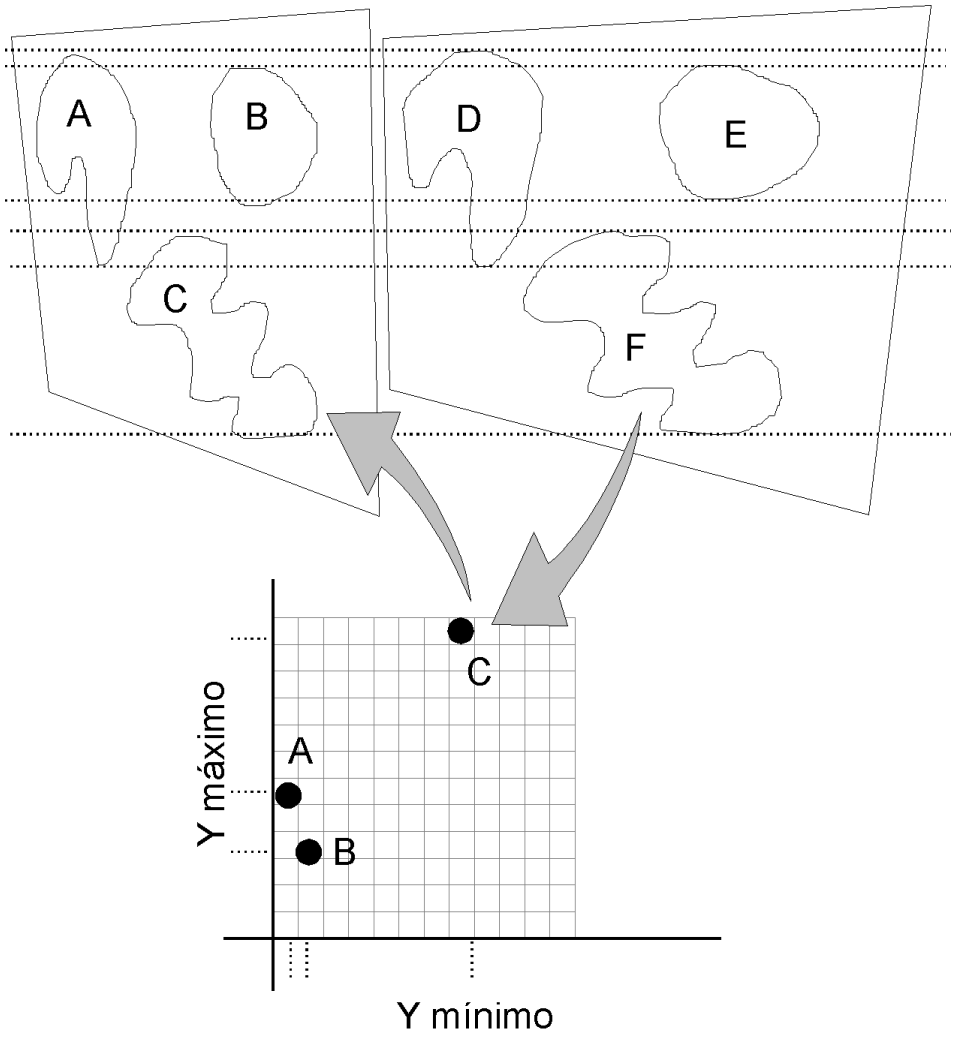


Figura 4-3 – *Matching* de feições extensas

As regiões correspondentes das duas imagens devem ter a coordenada y mínima e também a máxima iguais (a não ser pela imprecisão) para um par de imagens retificadas. Assim, a correspondência pode ser determinada por uma técnica de hash. Constrói-se uma tabela hash indexando as feições A, B e C da imagem pelos seus valores mínimo e máximo de y. Identificam-se as feições correspondentes a cada feição da imagem direita consultando a tabela. No exemplo da figura, as coordenadas y máxima e mínima de F indexam a mesma posição da tabela que a feição C, assim se conclui que F e C são feições homólogas.

Uma forma mais eficiente de se implementar a técnica de *matching* é através de uma tabela *hash*. Cria-se uma matriz indexada pelos valores de $y_{\min}$ e $y_{\max}$, mas amostrada em intervalos largos para considerar a imprecisão. A matriz é preenchida esparsamente com o valor de cada um dos rótulos da primeira imagem na posição correspondente aos valores de $y_{\min}$ e $y_{\max}$ da região associada ao rótulo. As correspondências são obtidas procurando o rótulo da feição homóloga na posição da matriz correspondente aos valores de $y_{\min}$ e $y_{\max}$ de cada região da segunda imagem. Pode-se ainda colocar um *timestamp* nas entradas da matriz para evitar ter que apagar a matriz a cada novo quadro. Pode-se também considerar uma vizinhança na tabela de *hash* e também colocar uma medida de qualidade da correspondência junto a cada entrada da tabela.

No caso de haver mais de uma feição na mesma imagem com o mesmo valor para o par $y_{\min}$ e $y_{\max}$, deve-se adotar uma forma de resolução de ambigüidade. É necessário utilizar informação adicional, que pode ser, por exemplo, a cor da região. Na ausência de informação adicional, é possível realizar todo o processo de alinhamento para cada possibilidade que cubra as ambigüidades selecionando o alinhamento mais adequado. Como essa alternativa é custosa, outras políticas como ignorar as feições com ambigüidade ou selecionar uma das possíveis correspondências arbitrariamente são mais apropriadas para tempo restrito.

4.3.2 Reconstrução 3D das feições

Uma vez determinada a correspondência entre feições, é necessário para a reconstrução determinar a correspondência entre pontos do contorno. Pela restrição epipolar, os pontos correspondentes estão sobre a mesma coordenada y. E, pela restrição de que não há inversão de ordem, sabe-se que o ponto mais à direita do contorno em uma imagem corresponde ao ponto mais à direita na segunda. Assim, os pontos são associados na ordem que se encontram considerando a coordenada x.

A alternativa à análise dos mínimos e máximos locais que consideramos é utilizar, ao invés do contorno da região, o contorno de seu fecho convexo dado que o fecho convexo é invariante à transformação projetiva. Para isso, é necessário supor que a feição reconstruída em 3D seja plana de forma que seu fecho convexo em 3D seja definido sobre um plano. A

borda do fecho convexo apresenta, em relação à ordenada *y* definida no plano-imagem, apenas um máximo e um mínimo globais, não havendo outros extremantes locais nem pontos de inflexão. Entretanto, o cálculo do fecho convexo pode implicar um custo adicional de ordenação dos vértices do polígono que representa o contorno da região. Para vértices ordenados pelo valor de um eixo de coordenadas, o fecho convexo é determinado em tempo linear. Adicionalmente, o fecho convexo simplifica a descrição da forma, por considerar um número menor de vértices.

Quando se considera a reconstrução do contorno do fecho convexo, sabe-se que a linha epipolar cruza o contorno no máximo 2 vezes. Dessa forma, a determinação dos pontos homólogos do contorno do fecho convexo é imediata. O contorno do fecho convexo pode ser dividido em uma curva à esquerda com y crescente e uma curva à direita com y crescente, quebrando o contorno nos extremantes globais.

O algoritmo de reconstrução (veja figura 4-4) consiste em percorrer os pontos de dois segmentos de curva homólogos, obtendo a partir de suas coordenadas $[x_1 \quad y]$ e $[x_2 \quad y]$, as coordenadas do ponto objeto no espaço tridimensional, utilizando a equação (3-79). A curva tridimensional é resultado do encadeamento dos pontos reconstruídos e da concatenação da reconstrução dos segmentos de curva que compõem o contorno de uma região.

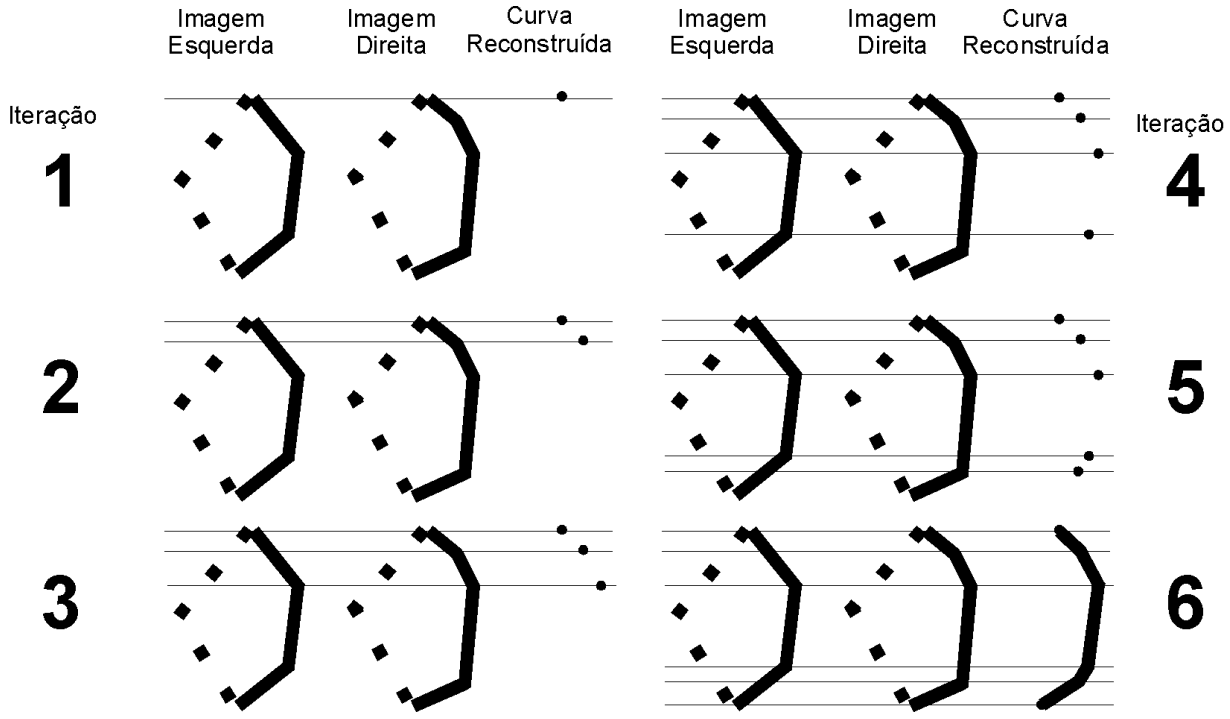


Figura 4-4 – Exemplo de reconstrução de linha poligonal com *y* crescente.

O algoritmo 4-1 é um possível algoritmo co-seqüencial de *scanlines* para realizar a reconstrução tridimensional de um segmento de curva com y crescente. Para um segmento decrescente, o algoritmo é análogo. Nesse algoritmo, as curvas são representadas por linhas poligonais, iniciando-se no vértice com valor mínimo de y. As duas curvas são percorridas simultaneamente e são criados pontos intermediários conforme se avança o cursor y. O cursor sempre avança assumindo o menor valor y considerando o próximo vértice a ser visitado de cada curva.

$y \leftarrow y_{\min}$

$y_1 \leftarrow y_{\min}$

$y_2 \leftarrow y_{\min}$

$y_1^+ \leftarrow$ *próximo y do segmento* 1

$y_2^+ \leftarrow$ *próximo y do segmento* 2

Enquanto $y < y_{\max}$

- *Se* $y_1^+ < y_2^+$, *então* $y \leftarrow y_1^+$; $y_1 \leftarrow y_1^+$; *reconstrói para* $x_1, y_1, \bar{x}_2$
- *Se* $y_1^+ > y_2^+$, *então* $y \leftarrow y_2^+$; $y_2 \leftarrow y_2^+$; *reconstrói para* $\bar{x}_1, y_2, x_2$
- *Se* $y_1^+ = y_2^+$, *então* $y \leftarrow y_1^+$; $y_1 \leftarrow y_1^+$; $y_2 \leftarrow y_2^+$; *reconstrói para* x_1, y_1, x_2
- *Atualizar* y_1^+, y_2^+

onde $\bar{x}_1 = \dfrac{(y_1^+ - y)x_1^+ + (y - y_1)x_1}{(y_1^+ - y_1)}$ *e* $\bar{x}_2 = \dfrac{(y_2^+ - y)x_2^+ + (y - y_2)x_2}{(y_2^+ - y_2)}$

Algoritmo 4-1 – Reconstrução 3D de linha poligonal.

4.3.3 Descrição das feições

A partir dos contornos reconstruídos em 3D, descrevemos as feições planares independentemente de suas formas. Como mostramos no item 3.5, utilizamos o método dos momentos para descrever as feições dados seus contornos em coordenadas tridimensionais. Consideramos que uma feição é descrita pela posição do centróide, pela direção normal ao plano de mínima dispersão e por outros atributos provenientes da imagem, como a cor, o que é ilustrado na figura 4-6. Veja na figura 4-5 como é representado um modelo geométrico pelas descrições das poses das feições regionais.

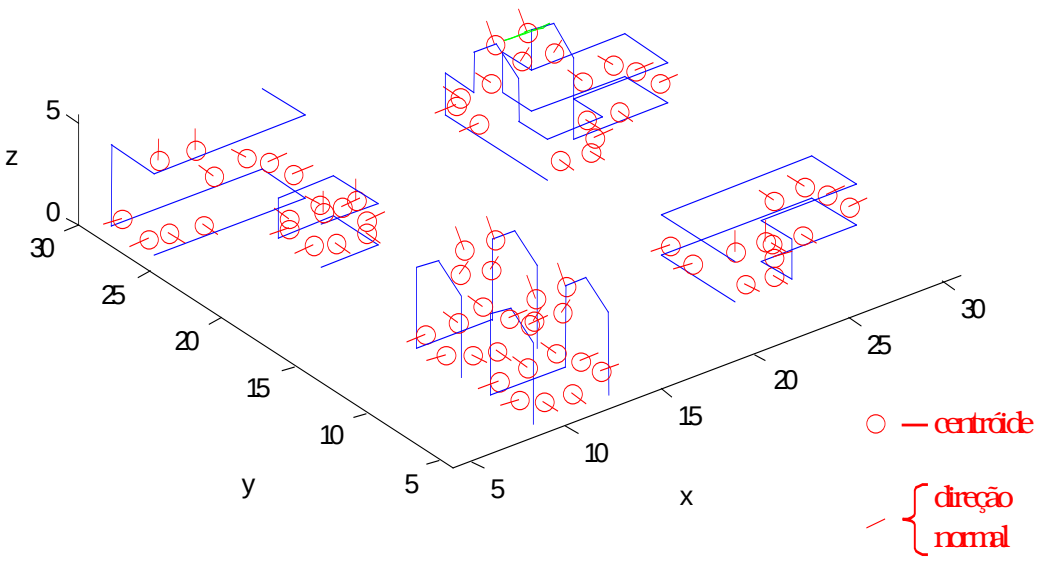


Figura 4-5 – Representação do modelo baseado nas feições descritas.

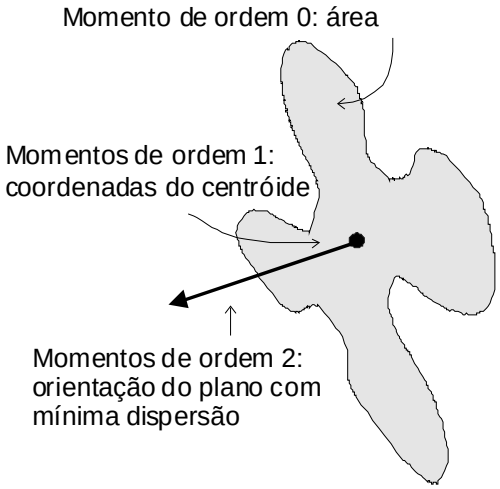


Figura 4-6 – Descrição de uma feição.

O grau de liberdade dado pela rotação de uma feição em seu plano não é levada em conta pelos seguintes motivos: se a forma da feição for simétrica, há mais de uma solução possível. Se não for simétrica, mas se aproximar de uma forma simétrica, pequenas variações na imagem podem provocar alterações súbitas no resultado estimado para essa rotação, tornando a solução instável.

4.4 Determinação automática das correspondências imagem-modelo

Há pelo menos três tipos de atributos que podem ser obtidos da descrição da geometria observada que enumeramos a seguir. (1) Atributos invariantes individuais da feição, como a cor da feição na imagem ou a área da feição reconstruída, simplificam o reconhecimento por serem invariantes à transformação. Quando é necessário considerar variações de iluminação, a cor pode ser decomposta em matiz e intensidade, de forma que a intensidade varia com a pose do objeto e o matiz pode ser considerado invariante no caso de reflexão difusa se a cor do iluminante não se alterar. (2) Atributos individuais da feição dependentes da transformação, como a posição dos centróides e a direção das normais das feições, são úteis na estimação da pose, uma vez identificadas as feições. (3) Atributos invariantes de pares (ou grupos maiores) de feições, como a distância entre centróides e os ângulos entre normais, e restrições topológicas, como o relacionamento entre arestas conectadas por uma junta, são úteis no reconhecimento das feições considerando a compatibilidade das hipóteses de correspondência entre imagem e modelo frente às relações entre feições do modelo.

Apresentamos um modelo para o processo de reconhecimento ou correspondência entre imagem e modelo baseado em atributos invariantes de feições individuais e de pares de feições e derivamos um algoritmo a partir desse modelo. Antes disso, porém, apresentamos um modelo probabilístico que nos permite explicar como tratar imprecisões e empregar uma forma de raciocínio abdutivo, que é a base dos processos de reconhecimento.

4.4.1 Modelagem probabilística para o reconhecimento

Começamos descrevendo os elementos do modelo e suas regras. O modelo do processo de reconhecimento pode ser visto na figura 4-7. Nessa figura, I_i são feições-imagem ou seja, em nosso caso, tratam-se de elementos segmentados das imagens e já reconstruídos em três dimensões. As feições M_m são feições-modelo. A existência de uma correspondência entre a feição I_i e a feição M_m é representada por C_{im}. O vetor de atributos de cada feição I_i é denominado A_i e o vetor de atributos de cada feição M_m é denominado B_m. A_i e B_m representam atributos individuais de cada feição.

Pode haver atributos também de pares de feições. Assim, os atributos que relacionem a feição I_i com a feição I_j são resumidos na variável R_{ij}. O par de feições é ordenado e não se considera repetições de forma que não se define R_{ii}. De forma análoga, os atributos de um par (M_m, M_n) são descritos por S_{mn}.

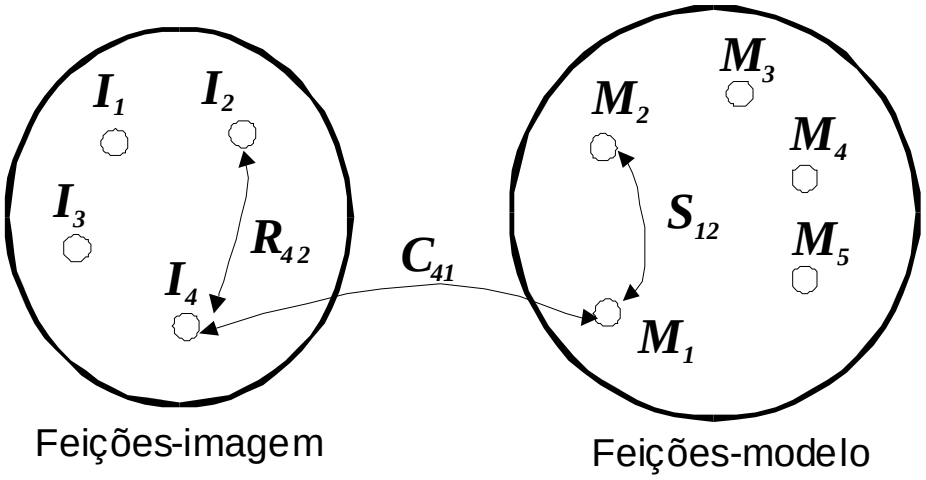


Figura 4-7 – Modelo do reconhecimento.

A correspondência é mutuamente exclusiva de forma que se uma feição-imagem se corresponde a uma feição-modelo, então não há correspondência entre essa feição-imagem

com qualquer outra feição-modelo nem há correspondência entre essa feição-modelo com outra feição-imagem. Isso pode ser escrito da forma (4-2)

$$\begin{aligned} C_{im} &\rightarrow \neg C_{in}, \quad \forall n \neq m \\ C_{im} &\rightarrow \neg C_{jm}, \quad \forall j \neq i \end{aligned} . \tag{4-2}$$

Entretanto, pode haver casos em que não se pareia uma feição-imagem, por não haver uma feição correspondente no modelo. Da mesma forma, uma feição do modelo pode estar escondida de forma que não haja uma feição-imagem que a represente. Assim, os índices i e m podem assumir um valor $\varnothing$ que representa a ausência de correspondência

$$\begin{aligned} m &= 1, \ldots, |M|, \varnothing \\ i &= 1, \ldots, |I|, \varnothing \end{aligned} . \tag{4-3}$$

Em termos de probabilidade, a exclusividade mútua e cobertura exaustiva das possibilidades podem ser escritas como (4-4)

$$\begin{aligned} P(C_{im}) &= 1 - \sum_{j \neq i} P(C_{jm}) \\ P(C_{im}) &= 1 - \sum_{n \neq m} P(C_{in}) \end{aligned} . \tag{4-4}$$

Os casos de não-correspondência são tratados separadamente. $P(C_{\varnothing m})$ é a probabilidade da feição M_m não estar representada na imagem. $P(C_{i\varnothing})$ é a probabilidade de encontrar uma feição espúria, ou em outras palavras, é uma medida do nível de *clutter* da imagem. Esses valores podem ser obtidos por treinamento, mas, na prática, funcionam como um valor de limiar (*threshold*) para o discernimento da identificação da feição I_i, de forma que podem ser arbitrados.

Pela lógica dedutiva, o modelo que relaciona a correspondência entre feições e os atributos das feições pode ser colocado como a proposição (4-5)

$$C_{im} \rightarrow A_i = B_m . \tag{4-5}$$

Que também pode ser entendido como a proposição (4-6)

$$\neg C_{im} \vee (A_i = B_m). \qquad \textbf{(4-6)}$$

A proposição (4-6) pode ser expandida na forma de uma tabela verdade, expressa na tabela 4-2.

C_{im}	$A_i = B_m$	$\neg C_{im} \vee (A_i = B_m)$
falso	falso	verdadeiro
falso	verdadeiro	verdadeiro
verdadeiro	falso	falso
verdadeiro	verdadeiro	verdadeiro

Tabela 4-2 – Tabela verdade da implicação.

É possível perceber que o fato de $A_i = B_m$ nada revela sobre C_{im} perante uma lógica dedutiva, pois se $A_i = B_m$, a proposição (4-6) já é verdadeira independente de C_{im}. A lógica tradicional ainda pode ser utilizada para refutar hipóteses, de forma que a hipótese C_{im} pode ser refutada uma vez que se saiba que $A_i \neq B_m$. Entretanto, mesmo que sejam refutadas todas hipóteses concorrentes a C_{im} (do ponto de vista da exclusividade mútua e da cobertura exaustiva), ainda assim não é suficiente para a solução do problema pois não há meios de refutar $C_{i\varnothing}$.

Para se obter informação útil a partir dos atributos conhecidos é necessária uma formulação que modele o raciocínio abdutivo. Esse tipo de raciocínio pode aparecer da seguinte forma, dada pela sentença (4-7)

$C_{im} \rightarrow A_i = B_m$ **(4-7)**

Se $A_i = B_m$, *então* C_{im} *tem mais credibilidade.*

Além disso, é difícil lidar com incerteza e imprecisão na lógica tradicional sendo mais conveniente uma abordagem flexível em que se possa fazer inferência com expressões lógicas considerando imprecisão. A modelagem probabilística bayesiana é bastante conveniente no nosso caso pois modela incerteza e imprecisão, é possível modelar raciocínio abdutivo e é fácil de se atualizar o conhecimento de forma incremental incluindo uma informação por vez.

Modelamos a relação entre correspondência e valores dos atributos através de probabilidades condicionais $P(C_{im} \mid A_i = B_m)$. A regra de Bayes aplicada a essa expressão é dada por (4-8)

$$P(C_{im} \mid A_i = B_m) = \frac{P(A_i = B_m \mid C_{im}) \cdot P(C_{im})}{\sum_n P(A_i = B_m \mid C_{in}) \cdot P(C_{in})}. \quad \textbf{(4-8)}$$

Nessa equação, $P(C_{im} \mid A_i = B_m)$ é a nova probabilidade de C_{im} frente ao novo fato de que $A_i = B_m$. A probabilidade *a priori* $P(C_{im})$ corresponde ao conhecimento anterior de C_{im} que é atualizado pela equação (4-8). O termo $P(A_i = B_m \mid C_{im})$ corresponde à verossimilhança, isto é, o quão plausível é a ocorrência de $A_i = B_m$ quando C_{im} for verdadeira, isto é, dada a imprecisão da medida. O somatório do denominador da expressão funciona como um fator de normalização e compreende todas as possibilidades de correspondência de I_i.

Para modelar o erro de medida dos atributos, pode ser utilizado (4-9)

$$P(C_{im} \mid A_i = a) = \frac{P(B_m = a \mid M_m)}{\sum_n P(B_m = a \mid M_n) \cdot P(C_{in})} P(C_{im}). \quad \textbf{(4-9)}$$

Neste caso, $P(B_m = a \mid M_m)$ é uma função de densidade de probabilidade de a que deve ser estimada na fase de autoria. Essa densidade de probabilidade modela a incerteza da medida do atributo, podendo ser estimada ou arbitrada. Em [Pope e Lowe, 1995], por exemplo, é

discutido o aprendizado probabilístico pela aparência a partir de vistas de um objeto para seu reconhecimento a partir de imagens. Quanto à determinação de $P(C_{i\varnothing})$, preferimos supor independente de $A_i = a$ e, portanto, $P(C_{i\varnothing} \mid A_i = a) = P(C_{i\varnothing})$ é sempre uma constante.

O reconhecimento apenas por atributos de feições individuais necessita de um vetor de funções $f : \mathbf{A} \to [0,1]$ que representam $P(B_m = a \mid M_m)$ para cada a, m, conforme a equação (4-9). $\mathbf{A}$ é o domínio dos atributos de feições individuais.

Estendendo o modelo para atributos de pares de feições, a regra que associa as correspondências e os valores dos atributos são escritas na proposição (4-10)

$$C_{im} \wedge C_{jn} \to R_{ij} = S_{mn}. \qquad \textbf{(4-10)}$$

A proposição (4-10) é equivalente a (4-11)

$$\neg C_{im} \vee \neg C_{jn} \vee \left(R_{ij} = S_{mn}\right). \qquad \textbf{(4-11)}$$

O fato $R_{ij} = S_{mn}$ pode ser pouco explorado pela lógica tradicional, de forma que utilizamos um tratamento probabilístico representado na equação (4-12)

$$P(C_{im}, C_{jn} \mid R_{ij} = r) = \frac{P\left(S_{mn} = r \mid \left(M_m, M_n\right)\right) \cdot P(C_{im}, C_{jn})}{\sum_{p,q} P\left(S_{pq} = r \mid \left(M_p, M_q\right)\right) \cdot P(C_{ip}, C_{jq})}. \qquad \textbf{(4-12)}$$

Veja que (4-12) permite apenas determinar a atualização da probabilidade conjunta $P(C_{im}, C_{jn})$ de duas correspondências, sendo que queremos na verdade a probabilidade de cada correspondência $P(C_{im})$ e $P(C_{jn})$ isoladamente, pois, embora as evidências sejam sobre pares de feições, queremos a identificação para cada feição. Além disso, o número de hipóteses de correspondências pode ser grande e trabalharmos com a equação (4-12) implica um custo proporcional ao quadrado desse número de correspondências. Calculamos, então, a probabilidade marginal de cada uma das correspondências. Na equação (4-13), calculamos a probabilidade condicional de uma correspondência dada a outra

$$P(C_{im} \mid R_{ij} = r, C_{jn}) = \frac{P(S_{mn} = r \mid (M_m, M_n)) \cdot P(C_{im} \mid C_{jn})}{\sum_p P(S_{pn} = r \mid (M_p, M_n)) \cdot P(C_{ip}, C_{jn})}. \quad \textbf{(4-13)}$$

Na equação (4-14) mostramos como calcular a probabilidade marginal de uma correspondência frente a todas as hipóteses de correspondência de uma feição I_i

$$P(C_{im} \mid R_{ij} = r) = \sum_q P(C_{im} \mid R_{ij} = r, C_{jq}) \cdot P(C_{jq}). \quad \textbf{(4-14)}$$

Assim, em (4-15), calculamos a probabilidade marginal de C_{im} frente à nova informação $R_{ij} = r$

$$P(C_{im} \mid R_{ij} = r) = \sum_q \frac{P(S_{mq} = r \mid (M_m, M_q)) \cdot P(C_{im} \mid C_{jq}) \cdot P(C_{jq})}{\sum_p P(S_{pq} = r \mid (M_p, M_q)) \cdot P(C_{ip} \mid C_{jq})}. \quad \textbf{(4-15)}$$

Entretanto, para utilizar (4-15), ainda é necessário manter $P(C_{im} \mid C_{jn})$ em memória a cada instante para todos as dimensões dadas pelos índices i, j, m, n. Assim, assumimos uma hipótese de independência entre as correspondências de feições-imagem diferentes, dada pela equação (4-16)

$$P(C_{im} \mid C_{jq}) = P(C_{im}), \quad (i \neq j, m \neq q). \quad \textbf{(4-16)}$$

Aplicando essa simplificação em (4-15) obtemos (4-17)

$$P(C_{im} \mid R_{ij} = r) = P(C_{im}) \sum_q \frac{P(S_{mq} = r \mid (M_m, M_q)) \cdot P(C_{jq})}{\sum_p P(S_{pq} = r \mid (M_p, M_q)) \cdot P(C_{ip})}. \quad \textbf{(4-17)}$$

Para utilizar (4-17) é necessário manter em memória apenas as probabilidades marginais das correspondências e uma tabela de funções $f : \mathbf{R} \rightarrow [0,1]$ que representam $P(R_{ij} = r \mid C_{im}, C_{jn})$ em função de r, m, n. $\mathbf{R}$ é o domínio dos atributos de pares de feições.

Introduzimos uma simplificação adicional dada na equação (4-18)

$$P(C_{im} \mid R_{ij} = r, C_{jn}) = P(C_{im} \mid R_{ij} = r)\,. \qquad \textbf{(4-18)}$$

Essa expressão representa a independência condicional entre C_{im} e C_{jn}, frente à evidência $R_{ij} = r$. A simplificação resultante é que não se utiliza o conhecimento sobre a identificação de outras feições para identificar a feição I_i. Assim, cada evidência da forma $R_{ij} = r$ pode ser tratada independentemente das demais levando a um algoritmo mais simples. A expressão resultante para o conhecimento *a posteriori* é descrita na equação (4-19)

$$P(C_{im} \mid R_{ij} = r) = P(C_{im}) \sum_q \frac{P(S_{mq} = r \mid (M_m, M_q))}{\sum_p P(S_{pq} = r \mid (M_p, M_q))}. \qquad \textbf{(4-19)}$$

A equação (4-19) é um esquema para iterativamente refinar a informação que temos sobre C_{im}, isto é, a identificação da feição I_i da imagem, a partir do conhecimento de um novo fato da forma $R_{ij} = r$, isto é, quando medimos os atributos do par de feições da imagem (I_i, I_j). Quando a informação está disponível para toda outra feição I_j, o conhecimento final (*a posteriori*, dada toda informação disponível) é expresso como um produtório, como na equação (4-20)

$$P\left(C_{im} \,\middle|\, \bigcup_j \{R_{ij} = r_j\} \right) = P(C_{im}) \cdot \prod_j \left(\sum_q \frac{P(S_{mq} = r_j \mid (M_m, M_q))}{\sum_p P(S_{pq} = r_j \mid (M_p, M_q)) \cdot} \right). \qquad \textbf{(4-20)}$$

Utilizando log-probabilidades, é possível expressar o conhecimento dada toda informação extraída dos atributos como um somatório que é representado de forma abreviada na equação (4-21)

$$\log P\left(C_{im} \,\middle|\, \bigcup_j \{R_{ij} = r_j\} \right) = \log P(C_{im}) + \sum_j \log \left(\sum_q \frac{P(S_{mq} = r_j \mid (M_m, M_q))}{\sum_p P(S_{pq} = r_j \mid (M_p, M_q)) \cdot} \right). \qquad \textbf{(4-21)}$$

Computando esse valor para uma dada feição I_i da imagem e para toda possível feição M_m do modelo, a melhor identificação para a feição I_i é aquela cujo valor de m o maximize. Este método é uma estimação por máxima *a posteriori* (MAP). Se não houver o conhecimento inicial *a priori* $P(C_{im})$, este é assumido como uma distribuição uniforme em m e se diz que a estimação é por máxima verossimilhança (ML, *maximum likelihood*). Assim, a identificação da feição I_i é dada por (4-22)

$$m = \arg\max_m P\left(C_{im} \left| \bigcup_j \{R_{ij} = r_j\} \right. \right). \qquad \textbf{(4-22)}$$

Note que (4-21) e (4-22) representam um esquema de votação. Cada nova informação da forma $R_{ij} = r_j$, contribui com votos ponderados para os vários C_{im}. Para cada feição I_i, aquela feição M_m para qual C_{im} contém o maior número de votos é a correspondência mais provável para I_i. O valor de cada voto é uma função de r e de m dada em (4-23)

$$f(r,m) = \log\left(\sum_q \frac{P(S_{mq} = r \mid (M_m, M_q))}{\sum_p P(S_{pq} = r \mid (M_p, M_q))} \right). \qquad \textbf{(4-23)}$$

Essa função f pode ser pré-calculada *off-line*, dado o modelo. Uma representação por uma estrutura de dados eficiente para essa função, ou para uma aproximação desta, permite criar algoritmos rápidos de reconhecimento. Esse é o papel das tabelas *hash* para o reconhecimento.

O esquema de votação, assim como a estimação por máxima verossimilhança, é um modelo para implementar o raciocínio abdutivo, necessário para inferir as correspondências a partir das observações das imagens.

4.4.2 Complexidade do método probabilístico

Fazemos agora uma análise da complexidade do custo computacional do reconhecimento. No caso de atributos individuais de feições, considerando que é necessário para cada nova informação atualizar as probabilidades, a complexidade do custo computacional para cada

nova informação da forma $A_i = a$ na equação (4-9) é $O(M)$, pois afeta apenas o índice i correspondente, mas todos índices m, para I feições-imagem e M feições-modelo. O somatório do denominador é independente de m e portanto não é contabilizado. Como o número de amostras consiste do atributo de cada feição, é no máximo I. Assim, a complexidade total para esse caso é $O(IM)$.

Para reconhecimento baseado em atributos de pares de feições conforme a equação (4-17), tem-se um grande número de entradas que corresponde a todos os pares de feições-imagens e é no máximo $O(I^2)$. Não sendo necessário utilizar todas essas entradas, amostramos aleatoriamente um total de R pares da forma $R_{ij} = r$. É necessário atualizar $P(C_{im})$ para cada nova entrada, assim são necessárias $O(M)$ atualizações, cada atualização afeta determinados índices i, j e todos os possíveis índices m. Cada atualização computa um somatório de $O(M)$ parcelas (índice q), não contabilizando o somatório do denominador que é independente do índice m (e portanto, pré-calculado). Assim, o custo do algoritmo correspondente é $O(RM^2)$ e no máximo $O(I^2M^2)$.

Se estruturarmos a função f através de uma representação esparsa eficiente, o somatório de votos pode ser calculado com mais eficiência. O método de *hashing* geométrico pode ser visto como uma representação esparsa eficiente do método probabilístico, onde se assume uma forma simplificada para as distribuições de probabilidades $P(S_{mn} = r \mid (M_m, M_n))$. A partir do valor de r, a tabela *hash* é indexada, obtendo-se uma lista de pares de índices (m,n) dentro da margem de erro da medida de r. As probabilidades $P(C_{im})$ e $P(C_{jn})$ são, então, atualizadas para cada par (m,n) encontrado. Se para cada uma das R evidências, ocorrerem em média S entradas (m,n) tabeladas, o custo total do algoritmo de reconhecimento é $O(RS)$.

4.4.3 Algoritmo proposto para o reconhecimento

O algoritmo proposto se baseia no raciocínio abdutivo. Na sentença (4-7), esse raciocínio é expresso para atributos individuais de feições. Expressamos na sentença (4-24) esse raciocínio para atributos de pares de feições

$$\frac{\begin{array}{l} C_{im} \wedge C_{jn} \rightarrow R_{ij} = S_{mn} \\ R_{ij} = S_{mn} \end{array}}{\left(C_{im} \wedge C_{jn}\right) \textit{tem mais credibilidade}}. \qquad \textbf{(4-24)}$$

Como é custoso manter uma tabela de 4 dimensões para representar o conhecimento da expressão $C_{im} \wedge C_{jn}$, consideramos os membros da conjunção separadamente. Assim, a sentença (4-25) expressa melhor o raciocínio implementado no algoritmo proposto

$$\frac{\begin{array}{l} R_{ij} = S_{mn} \\ C_{im} \wedge C_{jn} \rightarrow R_{ij} = S_{mn} \end{array}}{\begin{array}{l} C_{im} \textit{ tem mais credibilidade} \\ C_{jn} \textit{ tem mais credibilidade} \end{array}}. \qquad \textbf{(4-25)}$$

A imprecisão é tratada apenas na granularidade da tabela *hash*. Atributos são considerados iguais se indexam a mesma célula da tabela *hash*. Consideramos também, por simplicidade, que todos votos do processo de votação têm a mesma importância.

O algoritmo é, então, descrito da seguinte forma. Para cada par de feições da imagem $\left(I_i, I_j\right)$, medimos o atributo r e consultamos na tabela *hash* quais pares de índices (m,n) de feições modelos obtêm um atributo igual a r. Para cada par (m,n) encontrado, acumula-se um voto para C_{im} e um voto para C_{jn}.

Para identificar a feição I_i a partir da tabela de votos resultante, procura-se qual valor de m corresponde a mais votos para C_{im}. Em outras palavras, se C for uma tabela de votos com I linhas correspondendo às feições imagem indexadas por i e M colunas correspondendo às feições-modelo indexadas por m, procura-se na linha i a célula da tabela que contém o maior número de votos. O índice m da coluna que corresponde a essa célula é também o índice da feição-modelo mais confiável como identificação para a feição-imagem.

A construção da tabela *hash* é feita a partir do modelo. Para cada par de feições (M_m, M_n), com atributo S_{mn}, determina-se o índice da tabela correspondente ao valor do atributo e se acrescenta à célula indexada uma referência para o par (m,n).

O algoritmo de construção da tabela *hash* é resumido no algoritmo 4-2.

Construção da tabela *hash*, *off-line*
para cada feição *m* do modelo
  para cada outra feição *n* do modelo
    compute os atributos *S* do par *(m,n)*
    determine o índice da tabela hash dado *S(m,n)*
    insira na célula indexada da tabela uma referência para *(m,n)*
Fim

Algoritmo 4-2 Construção da tabela *hash*.

O reconhecimento das feições da imagem é descrito no algoritmo 4-3.

Reconhecimento das feições da imagem
para cada feição *i* da imagem
  para cada outra feição *j* da imagem
    compute os atributos *R* do par *(i,j)*
    determine o índice da tabela *hash* dado *R*
    para cada item na célula indexada da tabela
      obtenha a identificação do par *(m,n)* descrito no item
      verifique compatibilidade dos demais atributos de *(i,j)* e *(m,n)*
      caso positivo, coloque um voto para *C(i,m)* e *C(j,n)*
para cada feição *i* da imagem
  determine *X*, feição-modelo com mais votos $C(i,X) \geq C(i,m), \ \forall m$
  identifique *i* como correspondente a *X*.
Fim

Algoritmo 4-3 Reconhecimento por votação.

Alternativamente, pode-se fazer alguma análise mais complexa sobre a tabela de votos para se ter uma melhor idéia da qualidade do reconhecimento.

Considerando a escolha dos atributos das feições e o grupo das transformações rígidas, a fim de obter atributos invariantes, a base que utilizamos é formada por pares de feições. Cada

feição possui uma descrição dependente da pose, consistindo em seu centróide e a direção normal ao seu plano. Uma feição individual não constitui uma base porque há um grau de liberdade, desprezado a partir da hipótese de feição de forma livre, que consiste na rotação em torno do eixo normal. Os pares de feições podem ser caracterizados de forma invariante à transformação rígida. Se cada feição contribui com a determinação da transformação em 5 graus de liberdade e o espaço de transformações possui 6 graus de liberdade, um par de feições possui 4 dimensões redundantes que podem ser utilizadas como atributo do par.

Um sistema de coordenadas pode ser definido fixando-se a origem no centróide da primeira feição e um dos eixos (o eixo x por exemplo) na sua direção normal. O sistema de coordenadas é rotacionado em torno desse eixo até que o centro da outra feição se encontre no plano xy. As coordenadas x e y do centro da segunda feição e a direção normal da segunda feição nesse novo sistema de coordenadas compõem um vetor de 4 valores invariantes à transformação rígida.

Cada par de feições é descrito e tabelado numa tabela *hash* em função de seus atributos. Para manter uma tabela bidimensional, nossa implementação considera apenas as duas coordenadas transformadas do centróide da segunda feição para indexar a tabela hash. A direção normal da segunda feição, uma vez transformada, é utilizada para descartar hipóteses que apresentam uma medida de orientação muito discrepante. Assim, a tabela é mais fácil de ser tratada por ser bidimensional e ocorrem colisões com maior freqüência, tendo-se um número variável de entradas por índice da tabela.

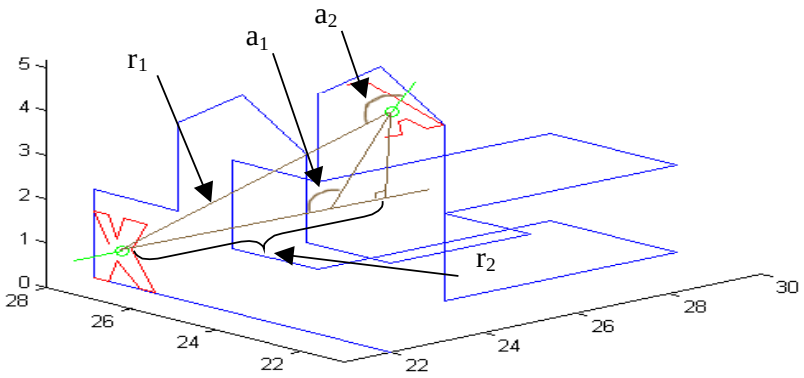


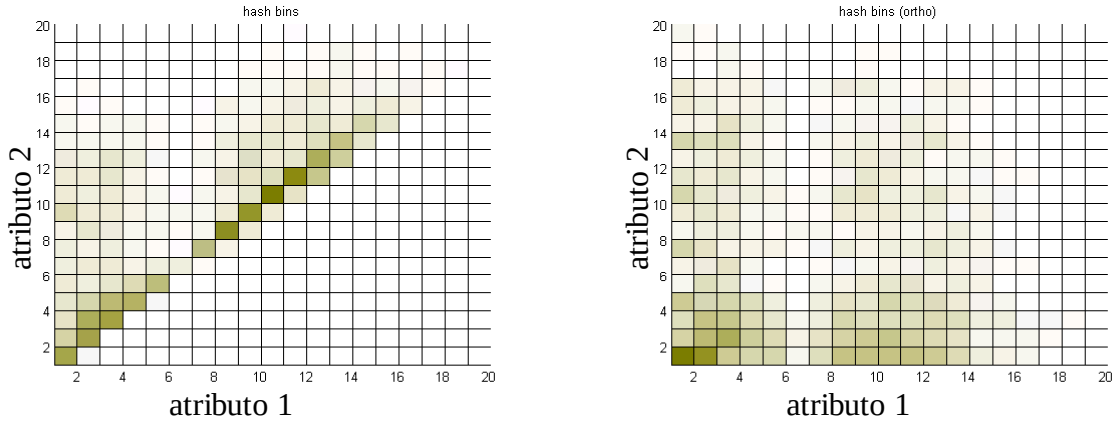
Figura 4-8 – Descrição de um par de feições.

Possíveis atributos para descrever um par de feições: r_1 distância entre os centróides, r_2 distância do centróide de uma feição à reta que passa pelo centróide da outra na direção de sua normal, a_1 ângulo entre as normais, a_2 ângulos entre a reta que une os centróides e uma normal, entre outros.

O conjunto de atributos influencia a forma da tabela *hash*. Um conjunto de atributos que podem ser utilizados é o seguinte (ver figura 4-8):

- distância entre os centróides (r_1);
- distância entre a reta normal à primeira feição e o centróide da segunda feição (r_2);
- ângulo entre as direções normais (a_1);
- ângulo entre a reta que une os centróides e a direção normal da segunda feição (a_2).

Este conjunto é interessante por conter a distância entre os centróides que é uma medida intuitiva para o operador da fase de autoria, entretanto pode-se notar que o segundo atributo é sempre menor que o primeiro, formando uma tabela *hash* triangular. Veja figura 4-9.



(a) (b)

Figura 4-9 – Densidade das tabelas hash utilizadas.

A tabela *hash* da figura 4-9a foi feita considerando a distância entre centróides como atributo do par de feições. Apesar de ser uma medida intuitiva e útil para uma fase de autoria, deve-se atentar para o formato da tabela *hash*. Utilizando a outra configuração descrita no item 4.4.3, a tabela *hash* da figura 4-9b é obtida aproveitando melhor o espaço de índices. Ambos os casos tiveram desempenho semelhante quanto ao reconhecimento. Lembramos que as medidas de ângulo não foram utilizadas para indexar, mas apenas para podar hipóteses.

Tendo-se reconhecido pelo menos duas feições, já é possível se estimar a pose. No entanto, um número maior de feições reconhecidas permite realizar uma aproximação determinando uma valor médio e mais preciso para os componentes da pose.

4.4.4 Complexidade do algoritmo de reconhecimento proposto

Para M feições do modelo, a complexidade tanto em tempo quanto em espaço da fase *off-line* de construção da tabela é $O(M^2)$, pois as feições do modelo são observadas aos pares. Na fase de reconhecimento, há duas etapas a se considerar: a coleta de votos e a análise dos votos. A coleta de votos considera cada par de feições-imagens, totalizando $O(I^2)$ pares, onde I é o número de feições da imagem. Para cada par se obtém $O(H)$ votos na tabela *hash*.

Assim, o custo total da coleta de votos é $O(I^2H)$. A análise dos votos consiste em procurar em cada linha da tabela de votos, a coluna cuja célula contém o maior número de votos para aquela linha, totalizando $O(IM)$, onde M é o número de feições-modelo. Como o modelo que consideramos é limitado em tamanho, é razoável percorrer essa tabela. Expressamos o custo total do método de reconhecimento como $O(I^2H + IM)$.

Para comparar o custo com o algoritmo de *hashing* geométrico de Wolfson e Rigoutsos [1997], consideramos que o atributo do grupo de feições representado por R_{ij}, é a posição de uma feição dada uma base. Segundo Wolfson e Rigoutsos [1997] para o tamanho b da base, no pior caso, em que todas as bases são testadas (muitas vezes a primeira base testada já fornece a solução), o custo do algoritmo de *hashing* geométrico é de ordem $O(I^{b+1}H)$, onde $O(H)$ é o custo de manipulação da tabela *hash*. Considerando a base de tamanho 2, o custo do algoritmo de *hashing* geométrico é $O(I^3H)$. O custo foi computado considerando que é utilizada uma terceira feição além da base, que é como se descreve o algoritmo de *hashing* geométrico. O algoritmo que propomos utiliza informação redundante da própria base (4 valores de um vetor com 10 valores, dos quais 6 determinam a transformação). Se considerarmos apenas a informação redundante da base, a complexidade passa a ser $O(I^2H)$.

4.4.5 Consolidação da pose

Tendo-se reconhecido K feições, tem-se K correspondências entre feições da imagem e feições do modelo. A informação de pose pode ser consolidada, equacionando-se sistemas lineares tendo como incógnitas os coeficientes r_{ij} e t_i na equação (4-26) e na (4-27)

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \\ Z_k \\ 1 \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix}. \qquad \textbf{(4-26)}$$

Nesta equação, k é o índice da correspondência entre feições, X, Y, Z são coordenadas no espaço das feições-modelo e x, y, z são coordenadas no espaço das feições-imagem

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} N_{X_k} \\ N_{Y_k} \\ N_{Z_k} \end{bmatrix} = \begin{bmatrix} N_{x_k} \\ N_{y_k} \\ N_{z_k} \end{bmatrix}. \quad \textbf{(4-27)}$$

Aqui, $N_{X_k}, N_{Y_k}, N_{Z_k}$ é a direção normal à feição modelo da correspondência k e $N_{x_k}, N_{y_k}, N_{z_k}$ é a direção normal à feição imagem da correspondência k. É desejável uma matriz de rotação ortonormal. Algum método para garantir a restrição (B-4, ver apêndice B) pode ser utilizado.

Dado que o reconhecimento de algumas feições pode falhar, um algoritmo robusto, como o RANSAC, pode ser utilizado por precaução. Assim, as correspondências mais provavelmente corretas serão utilizadas.

4.5 Avaliação dos algoritmos frente aos requisitos

Avaliamos os algoritmos quanto à sua complexidade, robustez e possibilidade de extensão.

4.5.1 Estudo da complexidade dos algoritmos

Se considerarmos N o número de pixels das imagens, I o número de feições nas imagens, P a soma do perímetro dos contornos de todas as feições visíveis nas imagens e M o número de feições no modelo, podemos estimar as complexidades das etapas do método proposto.

As etapas a analisar que são específicas da nossa implementação – segmentação, rotulação, filtragem por área, traçado das bordas e determinação do fecho convexo – são descritas no capítulo 5. Um algoritmo simples de segmentação baseado em cor tem complexidade $O(N)$ por ser necessário varrer toda a imagem. A rotulação tem complexidade $O(N)$. A filtragem por área tem complexidade $O(I)$. O traçado das bordas ou do fecho convexo tem complexidade $O(P)$.

O algoritmo de correspondência de feições entre imagens estéreo pode ser implementado para ter $O(I)$ utilizando as feições ordenadas por coordenada y mínima ou por uma tabela

hash, caso contrário. A tabela *hash* pode ser zerada apenas esporadicamente se não se avaliar as ambigüidades e acrescentar um *timestamp* a cada entrada.

O algoritmo de reconstrução toma tempo $O(P)$, assim como o processo de descrição e a retificação. O algoritmo de reconhecimento, como já analisamos, tem complexidade $O(I^2H + IM)$, onde o acesso à tabela *hash* tem complexidade $O(H)$.

Assim, para cada quadro analisado, a complexidade total do processo é dada por $O(N + P + I + I^2H + IM)$. Como $O(N)$ é dominante em relação a $O(P)$ e $O(I)$, visto que as feições apresentam uma área mínima, a complexidade total é dada por $O(N + I^2H + IM)$.

Podemos concluir dessa análise que é muito importante se ter um algoritmo de reconhecimento eficiente como o proposto. Considerando que as feições são selecionadas tendo uma área mínima pré-estabelecida e que o número de colisões na tabela *hash* pode ser controlado dado o conhecimento prévio do modelo, é possível garantir que $O(I^2H + IM)$ não supera $O(N)$. Os gargalos do processo são, portanto, a segmentação e a rotulação, dependentes do tamanho da imagem, e o reconhecimento, dependente do número de feições-modelo.

4.5.2 Análise da robustez do método

Existem três níveis de rejeição de feições espúrias. Num primeiro nível, uma feição espúria pode ser rejeitada por não se encontrar uma correspondência no par estéreo. Caso, ocorra a coincidência de algum elemento da outra imagem do par estéreo possuir atributos compatíveis com a feição detectada erroneamente, será construída uma representação tridimensional para essa feição e ela é descrita pelo método adotado como ocorre para feições normais. O outro nível em que a feição pode ser rejeitada é no reconhecimento. Como a posição, a orientação e demais atributos da feição espúria não são correlacionados com os atributos das demais, é muito improvável que sua presença cause grande influência na tabela de votos. Na consolidação da pose é possível ainda eliminar alguma feição espúria quando se procura o conjunto de feições para a qual melhor aproximação por quadrados mínimos é obtida. Para

uma aplicação de seqüência de imagens, como o método trata cada quadro independentemente, o erro em um quadro não se propaga aos demais quadros.

O método apresenta robustez em relação à oclusão parcial e campo de visão limitado. Desde que um número representativo de feições seja visível, o alinhamento pode ser resolvido para oclusão parcial. Feições parcialmente visíveis têm sua forma observada alterada, tratando-se de feição espúria.

O método também é robusto quanto à distração e ao *clutter* porque analisa globalmente a imagem e utiliza o consenso das múltiplas feições. O *clutter* contaminaria a tabela de votos, mas de forma desorganizada e não-correlacionada, dificilmente influenciando a identificação das feições.

Quanto a movimentos abruptos, o método é considerado robusto por tratar cada quadro independente e globalmente. Quanto a variações de iluminação, no item 5.2 descrevemos uma norma cromática que permite diminuir a sensibilidade a variações de intensidade, mas não variações cromáticas.

4.5.3 Possibilidades de extensão

A abordagem proposta permite fácil extensão no caso de seqüências de imagens, para que, combinada com uma abordagem incremental, seja possível realizar predição de estado. É também simples estender a proposta para outros tipos de feições com 5 graus de liberdade.

- **Combinação com método incremental**

A abordagem proposta pode ser utilizada de forma complementar com um algoritmo incremental para detectar situações de descontinuidade e reiniciar a abordagem incremental. A detecção pode também ser realizada por algum instrumento externo que avise da ocorrência de movimento abrupto, mas a abordagem incremental seria reiniciada através de nossa abordagem.

Pelo modelo probabilístico (veja propriedade A-10 no apêndice A, seção A.3), o uso de informação proveniente de um método de predição pode influenciar o algoritmo de reconhecimento através da tabela de votos. Uma modelagem da votação baseada em estimação

MAP (máxima *a posteriori*) deve substituir o modelo de máxima verossimilhança. Na forma de algoritmo, deve-se utilizar a pose estimada pelo algoritmo de predição para acrescentar votos para feições próximas das posições que se espera encontrá-las nas imagens.

Dessa forma, um método incremental pode influenciar a nossa abordagem através da preparação da tabela de votos, inicializando-a com uma distribuição inicial de votos a partir do conhecimento *a priori*. Assim é possível utilizar predição para se ter uma maior eficiência e a abordagem proposta para se ter maior robustez no processo de determinação da pose. Da mesma forma, a tabela de votos também pode ser influenciada num caso de fusão de dados sensoriais.

- **Extensão para feições curvilíneas, retilíneas e pontuais**

Outras formas de feição, além das feições regionais, ocorrem naturalmente nas cenas. É o caso, por exemplo, de feições curvilíneas e retilíneas, que devem ser tratadas por processos de segmentação diferentes. A representação da feição regional 3D com 5 graus de liberdade dados pela posição do centróide e pela direção normal é extensível para curvas, para as quais um plano pode ser ajustado e uma direção obtida, e para feições retilíneas, considerando a direção da própria reta. Assim, essas feições também podem ser incorporadas na representação para o reconhecimento.

O algoritmo de *matching* estéreo é facilmente adaptado para essas feições, por serem extensas. Já não é o caso de feições pontuais, cuja correspondência estéreo deve considerar um número grande de ambigüidades entre pontos que residem na mesma linha epipolar.

4.6 Discussão

A adição de mais uma vista facilitou em muito obter uma solução eficiente para o reconhecimento. A vista adicional permitiu coletar informação da posição 3D e orientação de cada feição que levou a uma descrição apropriada da feição para diminuir a complexidade do reconhecimento.

A representação por atributos de pares de feições indexada numa tabela *hash* foi um dos pontos mais importantes para permitir realizar *matching* com custo comparável ao processo de

segmentação e rotulação. Outro ponto positivo de destaque do método proposto consiste no pequeno espaço de memória.Utilizamos uma tabela *hash* com $O(M^2)$ itens, para M feições-modelo.

Os algoritmos desenvolvidos consistem em uma seqüência de etapas, que poderiam ser organizadas na forma de *pipeline*, por ser o resultado de uma, a entrada para a etapa seguinte. Lembramos que a implementação segundo uma estrutura de *pipeline* só é útil quando não há restrições quanto à latência.

Os mais importantes gargalos que permanecem na nossa abordagem são os algoritmos de segmentação e rotulação, que, apesar de muito simples, dependem de uma varredura pela imagem inteira. Técnicas mais sofisticadas de segmentação dominariam o custo total. É interessante procurar algoritmos que não necessitem dessa varredura. Porém, algoritmos dessa forma são usualmente incrementais, sendo sempre necessário conhecimento no primeiro quadro ou após uma mudança abrupta da vista.

A discussão do método probabilístico funcionou como guia para propor o algoritmo de reconhecimento, entendendo os compromissos entre todas as entidades modeladas e as simplificações adotadas. Uma das principais vantagens do método probabilístico por atributos invariantes de pares de feições é que é possível utilizar a propriedade da regra de Bayes para atualização incremental da informação. Assim, o método é facilmente combinado com a técnica de atributos de feições individuais e dados provenientes de preditores e outros sensores. Para fusão sensorial e uso de dados do rastreio contínuo, propomos utilizar esses dados externos para estimar a distribuição de probabilidade *a priori* que será utilizada no método. Entretanto, vale acrescentar que apesar do ferramental probabilístico baseado na regra de Bayes modelar adequadamente essas idéias, não é este o único meio possível de fazê-lo.

Ignorar a forma da região foi uma decisão importante. Para se trabalhar com a forma da região, é necessário que se tenha um método confiável e preciso para determinar o contorno das regiões. Por sua vez, um método desse tipo apresenta outras dependências como a resolução, o baixo nível de ruído, o emprego de um algoritmo robusto de segmentação e de reconstrução e precisão da orientação relativa das câmeras. Devido a essas dificuldades, é

preferível realizar o alinhamento independente da forma das feições, que são consideradas planares, e utilizar a multiplicidade das feições. Relaxando as restrições quanto à reconstrução 3D do contorno e evitando o problema da ambigüidade, obtém-se uma reconstrução aproximada dos contornos dos fechos convexos das regiões em 3D, que é suficiente para a estimação de 5 parâmetros de pose para cada região observada.

O momento de ordem zero é um atributo da região importante apesar de não o termos considerado na implementação porque no caso testado esse atributo variava pouco entre as feições. A área ou o perímetro da feição regional pode também ser utilizada para auxiliar a discriminação. Para o caso não calibrado, em que a relação entre as orientações das câmeras do par estéreo não é conhecida, a curva de contorno pode ser reconstruída dependente de um fator de escala. Neste caso, o momento de ordem zero não é utilizado para reconhecimento da curva, mas pode ser utilizado para inferir o fator de escala desconhecido. Esse fator de escala permite obter o parâmetro de linha de base do par estéreo. Se o par estéreo permanecer rígido, então a linha de base não será alterada e conforme se tenha medidas adicionais para várias feições ao longo do tempo, maior precisão pode ser obtida utilizando estimação.

Desmembramos o problema de alinhamento em problemas menores e descrevemos detalhadamente uma solução para cada um deles. Essas soluções têm compromisso com as restrições de tempo, robustez e precisão quando formam o conjunto que é a solução para o problema de alinhamento. Uma análise da complexidade e da robustez completa o capítulo justificando a adequação dos algoritmos aos problemas diante das hipóteses assumidas. Além disso, foram identificados os gargalos mais importantes. No capítulo 5, avaliamos uma implementação desses algoritmos quanto à robustez e a precisão para defendermos a viabilidade do método.

Capítulo 5 - IMPLEMENTAÇÃO E RESULTADOS

No capítulo 4, desenvolvemos algoritmos para descrever e reconhecer objetos e estimar suas poses a partir de um par estéreo de imagens. Descrevemos os requisitos e as hipóteses assumidas e avaliamos a complexidade dos algoritmos propostos.

Neste capítulo, uma implementação desses algoritmos é avaliada quanto à precisão e à robustez. Inicialmente, apresentamos a metodologia de testes adotada e os detalhes mais importantes da implementação. Em seguida, apresentamos e avaliamos os resultados dos diversos algoritmos e de seu conjunto. Mostramos o desempenho dos algoritmos de *matching* de feições em vistas estéreo, de reconstrução 3D das feições, de reconhecimento e de estimação de pose implementados aplicados a imagens de uma cena real. Com esses resultados, completamos nossa argumentação a favor da viabilidade do método.

5.1 Metodologia de testes

O objetivo principal desses experimentos é avaliar uma implementação dos algoritmos propostos com o intuito de mostrar a viabilidade da abordagem proposta. Para isso, concentramos os testes em imagens de uma cena real. Apesar da cena ter sido fabricada especificamente para o experimento, o processo de formação da imagem e seus problemas como oclusão, variações de iluminação e ruído estão presentes nas imagens adquiridas.

5.1.1 Construção

O conjunto de experimentos foi preparado da seguinte maneira. Um modelo virtual de uma cena contendo feições regionais foi criado. Esse modelo foi planificado e impresso em cartão de forma a se construir uma maquete física correspondente ao modelo virtual, mantendo suas proporções. O modelo físico é mostrado na sua forma planificada na figura 5-1. Trata-se de uma cidade, cujos prédios contêm faces com feições coloridas planas.

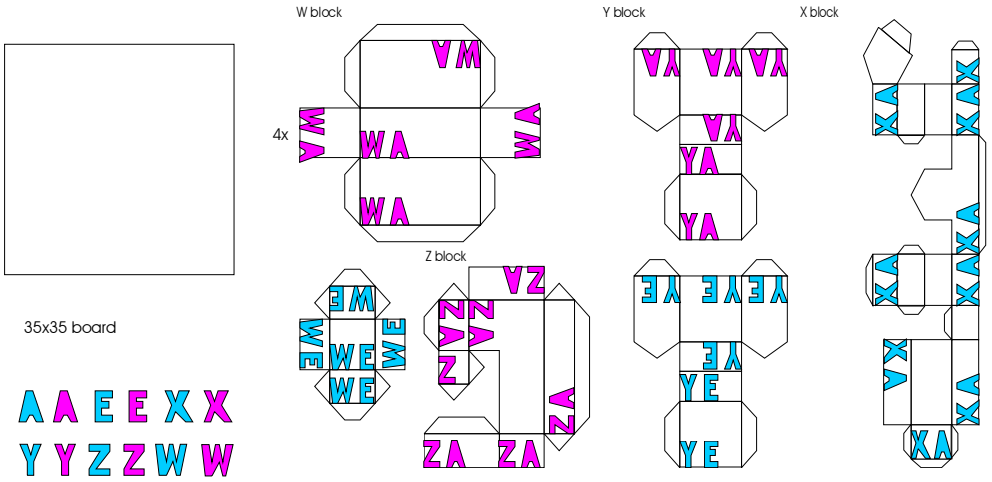


Figura 5-1 – Modelo da maquete.

Desta cena são obtidas imagens e seqüências de imagens com a câmera em movimento. As feições regionais foram planejadas para que se possa segmentá-las facilmente por cor. O modelo geométrico da cena já é previamente conhecido podendo ser facilmente utilizado para avaliar o alinhamento e construir o sistema de reconhecimento das feições.

Não houve necessidade de simulação de uma fase de autoria porque o modelo físico tinha suas dimensões conhecidas. Entretanto, há uma fase de preparação, onde as feições do modelo são representadas e é criada uma tabela *hash* a partir das descrições das feições do modelo para o processo de reconhecimento.

Na figura 5-2, mostramos uma vista da maquete. Essa vista foi obtida em alta resolução pela câmera utilizada no experimento. As imagens foram capturadas com uma câmera CCD do

modelo Sony Cybershot DSC-P30. As imagens podem ter uma resolução de até 1280x960 e seqüências de vídeo podem ter até 320x240 a 10 Hz. As imagens são armazenadas com compressão no formato JPEG, havendo perdas. No caso do vídeo, a qualidade é ainda mais baixa no formato MPEG.

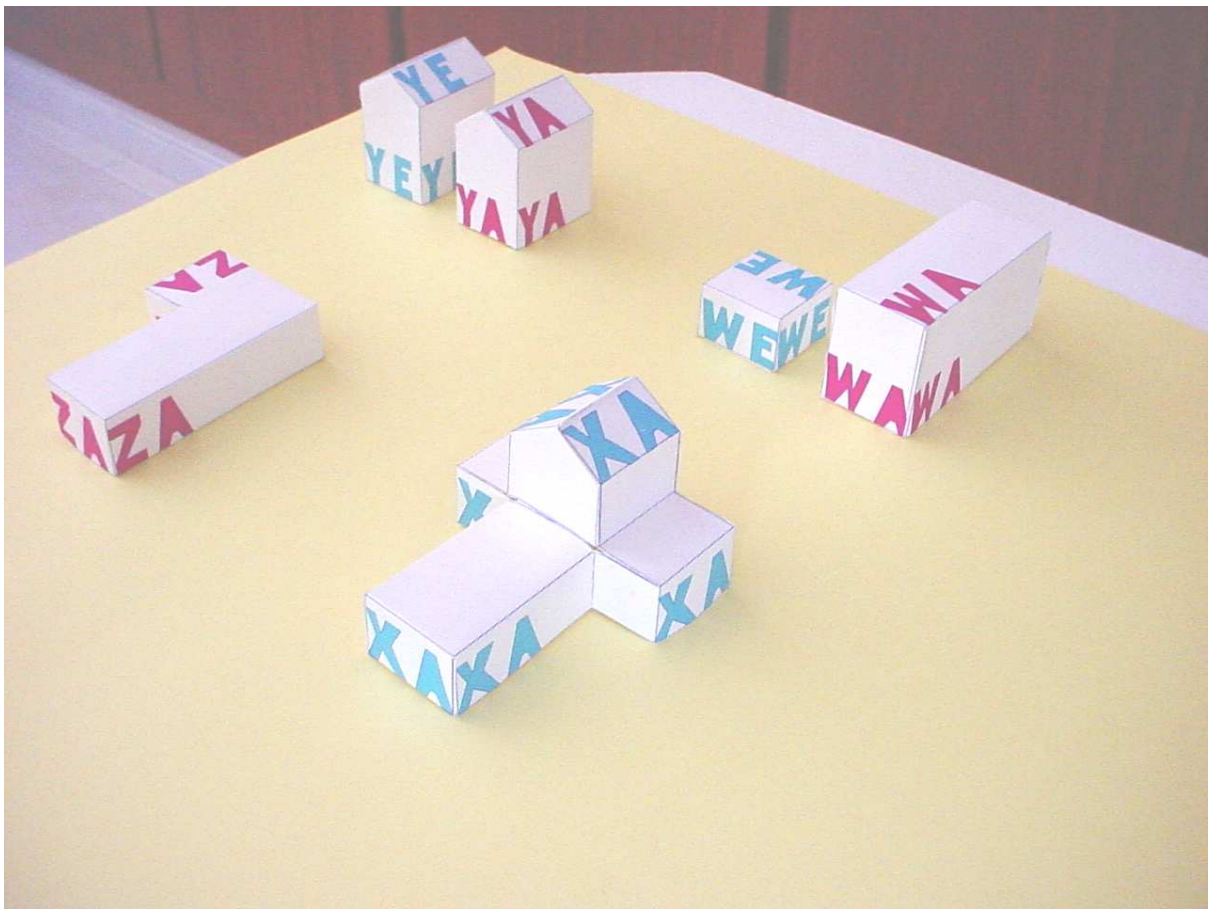


Figura 5-2 – Uma vista da maquete.

Apesar de se assumir o uso de um par de câmeras retificadas ou de orientação relativa conhecida, utilizou-se uma só câmera. Através da calibração manual para cada vista, foi simulada a obtenção de um par estéreo. Para efeito de experimento não há diferença, pois a cena permanece inalterada entre os quadros.

As coordenadas das quinas foram utilizadas para calibração de câmeras, servindo tanto para obter a orientação relativa entre as vistas como para produzir uma referência para avaliação da estimação de pose pelas feições regionais.

Para cada imagem do par estéreo calibramos a câmera utilizando as coordenadas de todos os cantos visíveis que foram marcados com um *mouse* através de uma interface que pode ser visualizada na figura 5-3. A seleção dos pontos da imagem foi feita utilizando o *mouse* para uma seqüência pré-estabelecida de pontos do modelo. Se um ponto da imagem correspondente ao ponto apresentado do modelo não for visível, esse ponto pode ser omitido pressionando o botão direito do *mouse*.

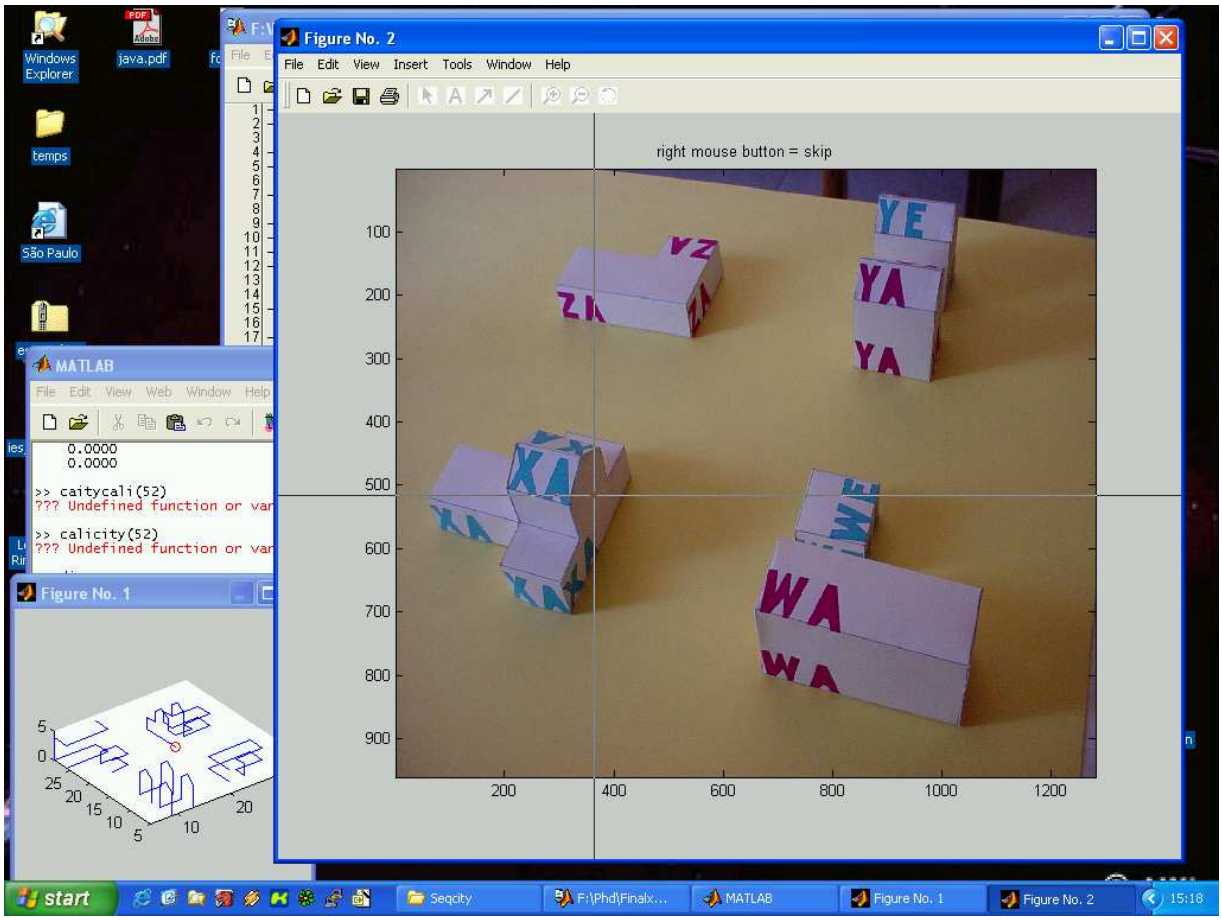


Figura 5-3 – Interface para calibração de câmeras.

A partir das matrizes de calibração, obtivemos a orientação relativa utilizada para efetuar a retificação.

A descrição das feições do modelo virtual é realizada pelo método dos momentos. Observamos pouca diferença entre os resultados dos momentos calculados para o contorno e dos calculados para o interior utilizando integral de linha no contorno, portanto consideramos esta última forma. Todas as feições regionais foram representadas no modelo 3D e as descrições baseadas nos momentos: centróide e direção normal ao plano são computadas. A representação para o modelo é ilustrada na figura 5-4. Cada feição apresenta seus atributos próprios e mais a posição do centróide e a direção da normal ao plano de mínima dispersão. A partir dessas descrições é construída uma tabela *hash* para o reconhecimento das feições.

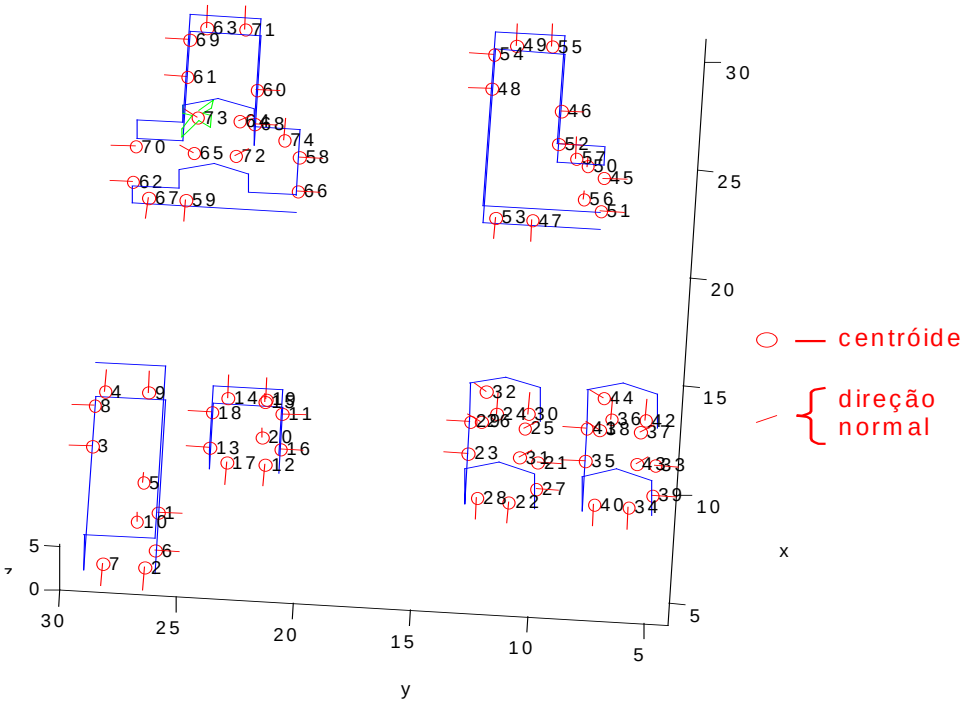


Figura 5-4 – Modelo virtual e descrição das feições.

5.1.2 Critérios de avaliação

Com o modelo construído, implementamos e avaliamos os algoritmos que fazem parte da abordagem proposta. Os itens seguintes descrevem o que foi implementado e a forma de avaliação adotada.

- **Rastreio das feições**

A fim de avaliar a dificuldade do rastreio preditivo em cenas complexas, implementamos um rastreio incremental para a cena construída utilizando um segurador (*holder*) de ordem 2. O rastreio estima a posição futura de cada feição a partir das posições da feição nos três quadros anteriores, considerando sua velocidade e sua aceleração em coordenadas de imagem. As posições dos quadros iniciais da seqüência foram marcadas manualmente. A partir daquele ponto, a imagem é varrida em espiral até encontrar a borda da região mais próxima ou até que se atinja um raio estabelecido para que se desista da busca.

A dificuldade esperada para esse rastreio serve como motivação para o emprego de métodos globais na determinação de correspondências entre feições do modelo e da imagem.

- **Correspondência estéreo**

Testamos o algoritmo de *matching* estéreo para imagens da maquete. Geramos a matriz de correspondências completa e, como critério de solução de ambigüidade, selecionamos aquelas correspondências melhor avaliadas de acordo com a precisão da correspondência dos máximos e mínimos globais. O processo de *matching* estéreo como explicado no capítulo 4, consiste na aplicação da restrição epipolar para cada hipótese de par homólogo. Mostramos o resultado para o *matching* através da visualização de uma matriz de correspondências.

- **Reconstrução 3D e descrição das curvas**

Reconstruímos o fecho convexo das feições em 3D e visualizamos a sobreposição ao modelo tridimensional da cidade para avaliação. A visualização interativa permite observar diversas projeções, alterando a posição da câmera.

Aplicamos o método dos momentos sobre as feições reconstruídas, obtendo suas posições e orientações. Avaliamos o erro das medidas de posição e orientação de cada feição

individualmente e da pose consolidada por mínimos quadrados no caso das imagens de alta resolução. Para o caso de baixa resolução, estimamos o erro da pose consolidada para as feições selecionadas e o erro da pose de uma feição individual ao longo de uma seqüência de imagens.

- **Alinhamento manual**

Estabelecemos manualmente a correspondência de 6 feições da imagem com suas feições do modelo para imagens de baixa resolução e avaliamos um primeiro resultado de alinhamento para essas imagens. Esse resultado é visualizado através da sobreposição de uma imagem de arestas do modelo alinhado com a imagem adquirida.

- **Reconhecimento e pose**

Finalmente, realizamos o alinhamento automático dado um par de imagens da cena. Determinamos automaticamente a correspondência entre feições do par estéreo e reconstruímos os fechos convexos das feições extraídas em 3D, de onde extraímos a descrição das feições observadas. As feições são reconhecidas automaticamente, estabelecendo-se a correspondência entre feições da imagem e do modelo. Dessa correspondência obtemos o alinhamento.

A qualidade do reconhecimento das feições pode ser avaliada observando os histogramas de votos para identificação das feições. O alinhamento é avaliado visualmente, através da reprojeção do modelo sobreposta à imagem, e também quantitativamente das seguintes formas: (1) pela diferença da posição e da orientação estimados e nominais da câmera, ou (2) pela distância em pixels dos contornos dos fechos convexos das feições.

A distância de Hausdorff é utilizada para auxiliar na avaliação do erro de alinhamento, provendo uma medida em pixels para a discrepância entre as feições regionais da imagem e as feições do modelo reprojetadas. Definimos essa distância a seguir.

A distância orientada $\vec{d}$ para os conjuntos de pontos A e B corresponde à maior distância entre um ponto de A e seu ponto correspondente mais próximo em B dada uma métrica d, que pode ser euclidiana

$$\vec{d}(A,B) = \sup_{a \in A} \inf_{b \in B} d(a,b). \qquad \textbf{(5-1)}$$

A distância de Hausdorff é definida por (5-2), sendo a maior distância $\vec{d}$ para as duas permutas do par de conjuntos

$$H(A,B) = \max\left\{\vec{d}(A,B), \vec{d}(B,A)\right\}. \qquad \textbf{(5-2)}$$

5.2 Detalhes do pré-processamento das imagens

Apresentamos os detalhes da implementação como sugestão para construção de um sistema de Visão Computacional bem como para a realização de experimentos adicionais. Como pré-processamento das imagens, consideramos o método de segmentação das feições, o processo de rotulação e o traçado da borda de cada região ou de seu fecho convexo, formando um polígono para representar cada região. Dada uma imagem, no final desse pré-processamento se obtém uma lista de polígonos rotulados e uma tabela de atributos (como cor) indexados por seus rótulos.

Por segmentação das regiões presumimos ser suficiente a detecção das feições regionais ao invés da segmentação completa da imagem. Os algoritmos de segmentação de imagens dividem naturalmente a imagem em regiões, em contraste aos detectores de feições. Como consideramos as feições sendo regiões de cor uniforme, o nosso processo de segmentação classifica os pixels de acordo com a proximidade a alguma das cores adotadas para as feições. No final do processo de segmentação, temos imagens binárias classificando os pixels de feições regionais conforme seus atributos. A estrutura de conectividade dos pixels é explorada pelo algoritmo de rotulação, para identificar de forma única cada região detectada na imagem.

A rotulação é a etapa que analisa a conectividade entre os pixels. Tendo classificado cada pixel conforme seus atributos na etapa de segmentação, queremos agora indexar cada região conexa da imagem por um rótulo. O processo de rotulação vai produzir, portanto, uma imagem que contém o rótulo da região a que cada pixel pertence. Da fase de rotulação

obtemos uma matriz que relaciona cada pixel a um rótulo e cada rótulo corresponde a uma região conexa da imagem na qual todos os pixels tem o mesmo atributo.

O traçado da borda de uma região consiste em determinar um polígono sobre a imagem cujos pixels interiores pertencem àquela região e os pixels exteriores não pertencem a ela. No final dessa operação, para cada rótulo considerado se tem um polígono representando a fronteira da região correspondente.

Por não estarmos interessados na forma exata do contorno, preferimos adotar o contorno do fecho convexo, que apresenta uma forma mais simples e permite a extração de medidas invariantes. A determinação do fecho convexo é muito conveniente para se trabalhar com regiões planares quando a forma não interessa. O fecho convexo de uma região é invariante à sua transformação rígida no espaço 3D e à transformação projetiva. Além disso, seu uso simplifica o processo de reconstrução do contorno da feição reduzindo a ocorrência de extremantes ao mínimo global e ao máximo global e elimina pontos de inflexão.

- **Segmentação das regiões**

Um método eficiente de segmentação é necessário para fim de cumprir as restrições de tempo do sistema. A segmentação mais simples necessita de pelo menos uma varredura completa da imagem. Técnicas de rastreio de regiões por *snakes* trabalham apenas em torno das bordas da região sendo promissoras para garantir a eficiência.

O método de segmentação que adotamos classifica cada pixel de acordo com sua cor. Para tornar o método robusto a variações de intensidade, utilizamos uma norma cromática apenas na intensidade do pixel em um espaço de cor diferente. A conversão de espaço de cor (5-3), utilizada por exemplo em [Duda e Hart, 1975], gera uma distribuição mais uniforme do histograma de cor e diminui a discrepância entre imagens obtidas sob circunstâncias variadas, além de permitir uma representação da cor no plano $R+G+B=1$

$$R=\frac{r}{r+g+b};\quad G=\frac{g}{r+g+b};\quad B=\frac{b}{r+g+b}. \qquad \textbf{(5-3)}$$

Os pixels são classificados utilizando a distância euclidiana no espaço RGB. A distância euclidiana com centro em uma determinada cor e um determinado raio discrimina pixels que pertencem ou não a uma classe de feições. Dividimos as feições em duas classes: feições magenta e feições ciano. Consideramos que esta classificação é robusta às variações de iluminação para as duas classes consideradas, porém dependente da aplicação.

Constatamos que este método sofre forte influência do ruído de compressão, especialmente do tipo "*ringing*" (veja [Gonzalez e Woods, 1992]). Assim, utilizamos um filtro passa-baixa nas imagens comprimidas, tendo como efeito uma perda de resolução. Uma máscara de convolução de tamanho 3×3 ou 5×5 mostrou-se suficiente.

- **Rotulação**

Duas regiões são vizinhas se, para uma determinada distância definida entre as coordenadas dos pixels, pelo menos um ponto de uma região tem distância a pelo menos um ponto da outra região menor ou igual a um valor fixo que chamaremos raio. Normalmente são utilizadas como distâncias as normas L_1 e L_∞ de um dos pixels com origem no outro. A primeira corresponde à soma dos valores absolutos da diferença de ordenadas e a diferença de abscissas. A segunda corresponde ao maior valor absoluto dentre a diferença de ordenadas e a diferença de abscissas. Para raio de um pixel, a norma L_1 permite 4 vizinhos e chamamos de vizinhança N_4. A norma L_∞ permite 8 vizinhos e chamamos de vizinhança N_8.

A estrutura de conectividade entre regiões pode ser explorada por algoritmos de reconhecimento. Entretanto, é necessário preservar a topologia para que isso possa ser feito. Por exemplo, devido à oclusão, essa restrição é quebrada quando se projeta uma cena 3D para uma imagem 2D.

A conectividade define as regiões segmentadas. Uma região é definida como um conjunto conexo de pixels com o mesmo atributo. Assim, para qualquer par de pixels de uma região é possível encontrar uma seqüência de pixels iniciando no primeiro pixel do par e terminando no último tal que a distância entre um pixel ao próximo seja menor ou igual ao raio.

Em outro problema a conectividade nos é útil. Um algoritmo de segmentação pode acabar retornando um número muito grande de regiões, que consiste do problema de supersegmentação, normalmente resultado de ruído. Podemos definir a área de uma região como o número de pixels e selecionar regiões com área maior que um limiar. Essa operação é conhecida como “abertura por área” em Morfologia Matemática.

O algoritmo de rotulação que utilizamos é equivalente àquele apresentado em [Gonzalez e Woods, 1992]. Numa rotulação, cada diferente região conexa é preenchida com um rótulo numérico diferente. Assim, cada região da imagem é identificada por um índice, o que é ilustrado na figura 5-5, onde, particularmente, o rótulo “1” é atribuído ao fundo da imagem.

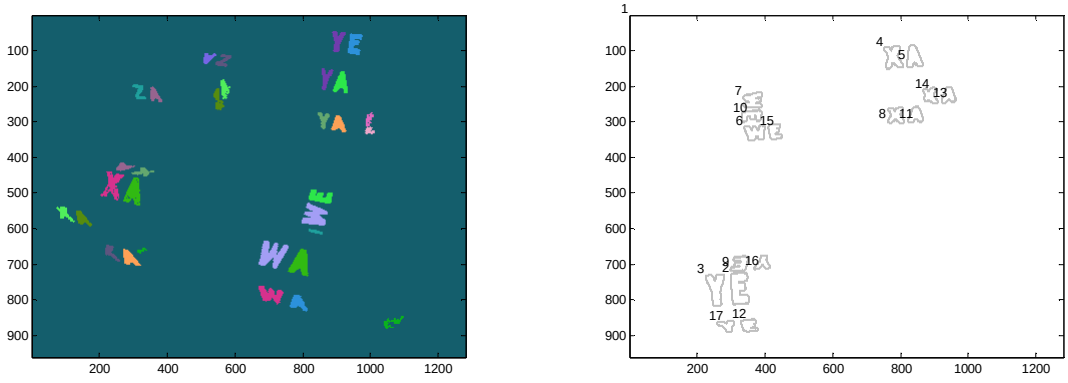


Figura 5-5 – Segmentação e rotulação de feições

No processo de rotulação, aproveita-se para calcular a área de cada região e se pode também associar ao índice de cada região outros atributos individuais próprios como a cor e o perímetro. Regiões com área inferior a um limiar são descartadas. É comum ordenar as demais regiões por sua área.

- **Traçado da borda**

É conveniente ter a borda da região na forma de uma lista de coordenadas dos pontos para reconstrução em 3D e descrição da feição. O encadeamento de bordas é feito por uma função, descrita pela figura 5-6, que baseada nos quatro pixels em torno da posição do cursor retorna a

direção em que este deve se mover. Um ponto inicial deve ser encontrado e o cursor posicionado entre quatro pixels de forma que algum deles esteja dentro da região e algum outro fora. Na figura 5-6, os quadrados cheios representam pixels do interior da região e quadrados vazados, pixels do exterior ou do fundo. Dado um padrão de quatro pixels em torno do cursor, a função mostrada na figura percorrerá a borda em sentido horário considerando vizinhança N_4 até voltar ao ponto inicial. Este método pode ser estendido para vizinhança N_8 para uma máscara 3×3. Note que essa função é dependente da posição anterior do cursor para os padrões 6 e 9.

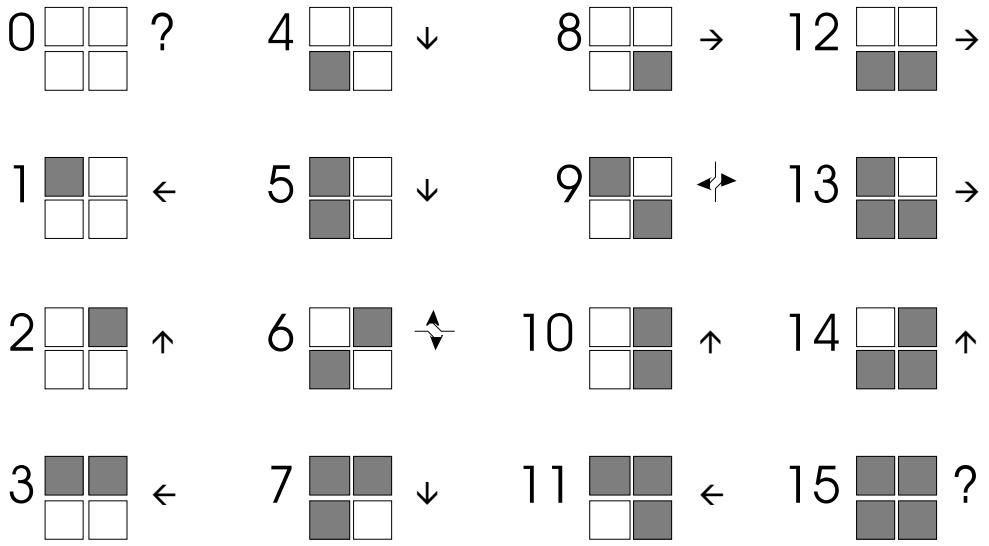


Figura 5-6 – Função de quatro pixels para traçado da borda.

As máscaras representam os quatro pixels a serem examinados, pixels do quadro branco devem ser zero e pixels do quadro escuro devem ser um. Nos casos 6 e 9, ainda é necessária a direção do último movimento para se poder determinar o movimento seguinte.

- **Fecho convexo**

Uma das melhores formas para computar o fecho convexo dos polígonos é feita ordenando primeiramente os vértices por y. O fecho convexo pode ser determinado a partir dos polígonos obtidos pelo traçado das bordas ou por *scanlines* sobre a imagem. Por *scanlines*, se armazena como lista de vértices as coordenadas do pixel de menor e de maior x para cada

linha y, de forma que os vértices são retornados ordenados. No caso do traçado da borda, é necessária uma etapa de ordenação. A partir da lista de vértices ordenados por y determinamos o fecho convexo conforme explicamos a seguir.

O algoritmo para determinar o fecho convexo consiste em ordenar os vértices em função de uma coordenada dada por um eixo qualquer, por exemplo, y. Partindo do ponto de mínimo y, criam-se duas listas: a lista da direita e a da esquerda. Cada novo vértice inserido é colocado no fim de cada lista após eliminar os vértices do fim da lista que estão em desacordo com a convexidade do polígono que está sendo construído.

O algoritmo que utilizamos para determinar o fecho convexo em tempo linear para vértices ordenados pelo valor da ordenada y consiste em manter duas listas de vértices. O vértice de mínimo y pertence ao contorno do fecho convexo. Assim, cada novo vértice acrescentado em ordem também é vértice do contorno do fecho convexo construído até esse instante considerando apenas os vértices anteriores porque seu valor de y é maior que os demais. O novo vértice é inserido na lista da esquerda e na lista da direita após remover os vértices que geram arestas cuja direção não se altere monotonamente. Os ângulos das arestas da lista da direita com a horizontal são sempre crescentes, enquanto que os da esquerda são decrescentes. O algoritmo 5-1 pode ser acompanhado visualmente na figura 5-7.

```
Construção do fecho convexo
  dados vértices ordenados em y
  insira o primeiro vértice e o segundo vértice nas listas esquerda e direita.
  para cada outro vértice V
    percorrer lista da esquerda inversamente removendo vértices até garantir que o ângulo
    das arestas em relação à horizontal é sempre decrescente.
    percorrer a lista da direita inversamente removendo vértices até garantir que o ângulo
    das arestas em relação à horizontal é sempre crescente.
    insira V nas duas listas.
Fim
```

Algoritmo 5-1 – Fecho convexo.

Como para N vértices o máximo número de vértices eliminados não ultrapassa N, então o método é $O(N)$. Mas veja que a ordenação inicial dos vértices pode tomar tempo $O(N \lg N)$.

A figura 5-8 ilustra uma região, o traçado de sua borda e o seu fecho convexo. A divisão em duas curvas de y crescente permite o uso fácil no algoritmo de reconstrução 3D.

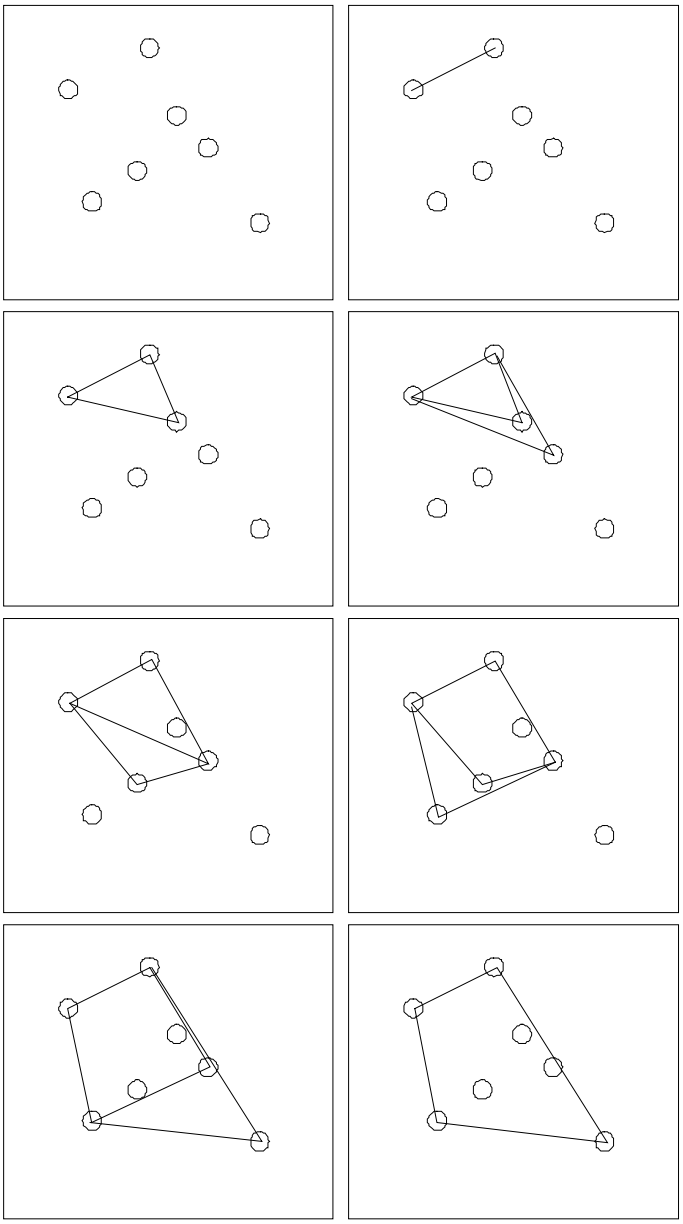


Figura 5-7 – Determinação do fecho convexo

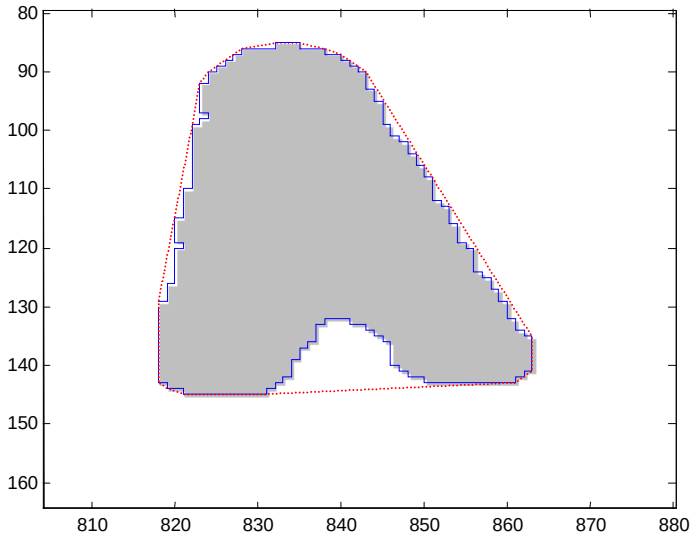


Figura 5-8 – Traçado da borda e fecho convexo.

5.3 Resultados

Apresentamos e analisamos a seguir os resultados dos experimentos realizados.

5.3.1 Rastreio de feições

Como esperado, o método incremental de rastreio de feições teve resultados adequados apenas para seqüências com variações muito pequenas. Um dos motivos é a taxa de amostragem da câmera de 10 quadros por segundo. Para uma resolução maior de tempo, espera-se ter imagens com variações menos abruptas. Outro motivo é o *clutter*. A grande quantidade de feições provoca freqüentemente a correspondência da feição incorreta, falhando o rastreio. Assim, apenas com um método que considere múltiplas hipóteses, como o *Condensation* [Blake e Isard, 1998], ou um método global como o que propomos, é que se pode perseguir satisfatoriamente essas feições. A figura 5-9 ilustra uma situação de sucesso.

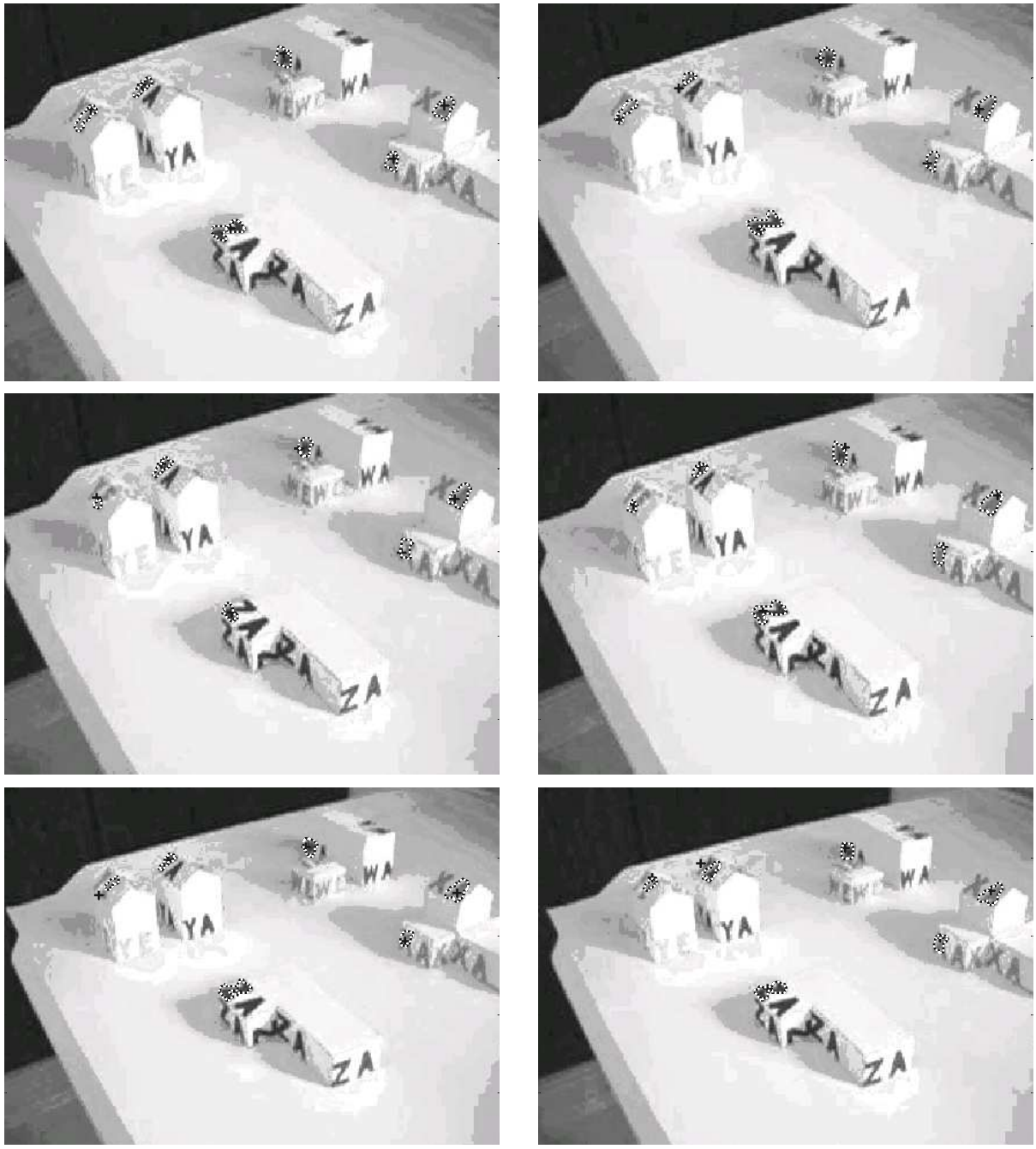


Figura 5-9 – Rastreio de feições *off-line* com predição por segurador de ordem 2.

O rastreio só foi possível em um curto intervalo da seqüência das imagens com variação muito pequena. O problema é a distração pelo grande número de feições, considerado um caso de rastreio com *clutter*.

5.3.2 *Matching* estéreo

A técnica proposta para o *matching* se mostrou bastante adequada, associando feições corretamente para casos de base larga e casos de base estreita. A equação (4-1) foi utilizada para ordenar as correspondências de acordo com a precisão das coordenadas *y* mínimas e máximas.

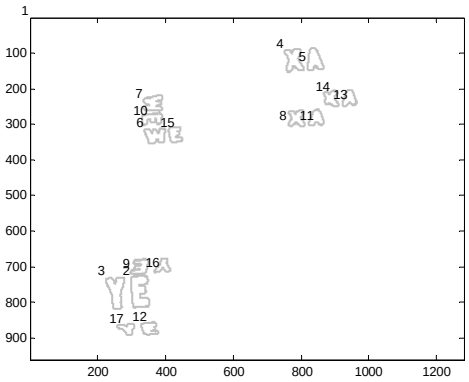


imagem 1

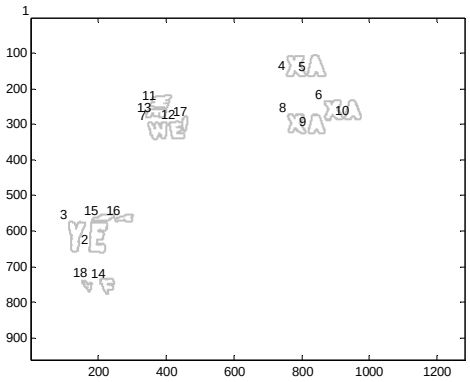
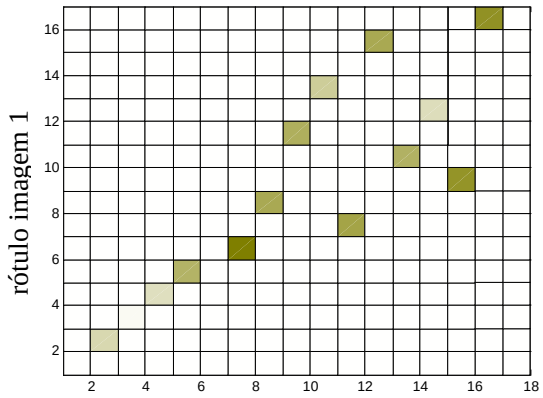


imagem 2



rótulo imagem 2

Figura 5-10 – Resultado do matching

Imagens acima contêm rótulos das feições observadas em vistas. O gráfico mostra a compatibilidade, segundo a equação (4-1), entre os rótulos de cada imagem, apenas para as associações determinadas.

O gráfico da figura 5-10 mostra as melhores compatibilidades dentre possíveis pares homólogos. Quanto menor a distância dada pela equação (4-1) mais escuro o retângulo no gráfico. Apenas os retângulos que determinaram as correspondências são mostrados.

5.3.3 Reconstrução das curvas

A reconstrução de uma feição em particular é mostrada na figura 5-11. É notável a baixa qualidade da reconstrução dos contornos pelo método proposto uma vez que a feição reconstruída destacada nessa figura não se encontra sobre o plano que se esperava quando observado de uma vista lateral. Entretanto, a informação de posição e orientação é obtida por um método estatístico para compensar o erro de reconstrução do contorno que acarreta imprecisão nas medidas de posição e orientação. Outra implicação é a inviabilidade de se utilizar a forma da feição para seu reconhecimento. O erro de posição e orientação é compensado no processo de consolidação da pose, pela multiplicidade de medidas dadas várias feições utilizando o método dos quadrados mínimos.

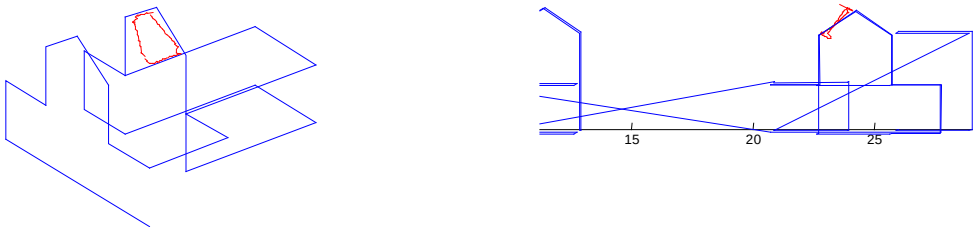


Figura 5-11 – Reconstrução de uma feição.

São observadas duas vistas do modelo geométrico da cena com o fecho convexo de uma feição reconstruída. A linha vermelha é o fecho convexo reconstruído de determinada feição. As linhas azuis são um esboço do modelo 3D da maquete.

5.3.4 Resultados da estimação de pose

Considerando que as feições são conhecidas, medimos o desempenho dos algoritmos de estimação de pose pelas feições regionais em algumas circunstâncias descritas a seguir. A tabela 5-1 apresenta os valores dos erros medidos para um par de vistas em alta resolução. O

erro é dado como a distância euclidiana entre a posição nominal e a posição estimada das feições. O erro da direção normal é o ângulo entre a normal estimada e a normal nominal. A média dos erros é dada na forma do erro absoluto médio. A posição nominal é dada pela calibração manual a partir dos cantos visíveis. A cena cabe num retângulo de 25 cm de lado.

Resolução: 1280x960 – 14 feições		
Erro (cm)		Erro direção normal (graus)
0.13		10.09
0.22		3.60
1.81		32.44
0.52		5.42
0.18		6.83
1.75		24.69
0.44		11.57
0.36		8.14
0.30		19.09
0.32		39.41
0.26		10.43
0.26		17.98
0.73		44.91
0.38		9.l6
soma		
7.31		243.78
média do erro absoluto		
0.50		17º 24’

Tabela 5-1 – Resultados da estimação de pose para as feições de um par de vistas.

O erro da direção normal é particularmente grande, mostrando a dificuldade que o método apresenta para estimar a orientação. Esse erro implica maior dificuldade no reconhecimento devido à maior incerteza na orientação. Por outro lado, o erro da posição das feições no espaço é razoável e a incerteza não prejudica o reconhecimento de forma significativa. Por esse motivo foi decidido utilizar os atributos mais dependentes da posição do que da orientação para indexar os pares de feições, no item 4.4.3.

Nas tabela 5-2 apresentamos o erro da consolidação de pose a partir das informações de várias feições, utilizando as mesmas condições da tabela 5-1. A pose foi estimada a partir das 14 feições. É notável a redução do erro de orientação em relação à análise de feições isoladas da tabela 5-1.

Imagem de alta resolução (1280x960) utilizando 14 feições. Erro da pose consolidada utilizando os centróides	
Erro da posição da câmera	0.99 cm
Erro do ângulo	1^{o} 36′
Erro utilizando as normais das feições também	
Erro da posição da câmera	0.95 cm
Erro do ângulo	1^{o} 41′

Tabela 5-2 – Resultados da pose consolidada.

No caso de imagens de baixa resolução, os erros das medidas individuais sobre as feições para posição e orientação crescem bastante. Entretanto, uma estimação grosseira da pose já pode ser obtida a partir de um pequeno número de feições. Além disso, a partir de certo número de feições (no caso do experimento foram 6 feições) não se espera melhoria da medida devido à imprecisão das medidas individuais aliada ao desconhecimento da distribuição do erro. A medida consolidada de posição não é afetada diretamente pela medida de orientação de uma feição dado que a medida de orientação influi apenas nos coeficientes da matriz de rotação. Essas conclusões são refletidas nas medidas de erro da tabela 5-3 feita para imagens de baixa resolução.

Imagem de baixa resolução (320x240) Erro da pose consolidada utilizando 4 feições	
Erro da posição da câmera	8.2 cm
Erro do ângulo	12^{o} 30′
Erro da pose consolidada utilizando 6 feições	
Erro da posição da câmera	2.0 cm
Erro do ângulo	5^{o} 23′

Tabela 5-3 – Resultados da pose consolidada para um número reduzido de feições.

A tabela 5-4 mostra os resultados da medida do centróide e da orientação de uma determinada feição ao longo de uma seqüência de imagens de baixa resolução. Nota-se que o erro de orientação de uma feição individual no caso de baixa resolução piora em relação ao caso de alta resolução da tabela 5-1.

Resolução: 320x240 – 1 feição – 9 quadros		
Erro (cm)		Erro direção normal (graus)
0.64		30.98
0.84		39.81
0.43		17.98
0.47		42.23
0.39		7.70
0.29		42.22
0.41		25.69
0.28		37.63
0.29		25.40
soma		
4.05		269.66
média do erro absoluto		
0.45		29º 57′

Tabela 5-4 – Resultados da estimação de pose de uma feição em uma seqüência.

Resumindo a análise desses dados, o que pode ser observado é que a multiplicidade de feições compensa os erros individuais da pose das feições. É também notável o efeito da resolução nos resultados, de forma que, como esperado, para resoluções maiores a medida de pose é melhor. Observa-se também que um número pequeno de feições selecionadas já é suficiente para obter uma medida razoável da pose. Verificando que a inclusão de feições além do necessário não contribui para melhoria da medida, concluímos que aplicar métodos estatísticos que utilizam poda pode ser uma melhor opção do que considerar todos os dados imprecisos.

5.3.5 Resultados do alinhamento manual

O alinhamento mostrado na figura 5-12 foi obtido a partir da consolidação de pose a partir de 6 feições regionais escolhidas manualmente em imagens de baixa resolução de uma seqüência de imagens. O alinhamento obtido utilizando as coordenadas de todas as quinas visíveis selecionadas manualmente é utilizado como termo de comparação. É possível ver que nessas diversas situações o alinhamento pelas feições regionais se aproximou do alinhamento nominal, considerando a baixa resolução.

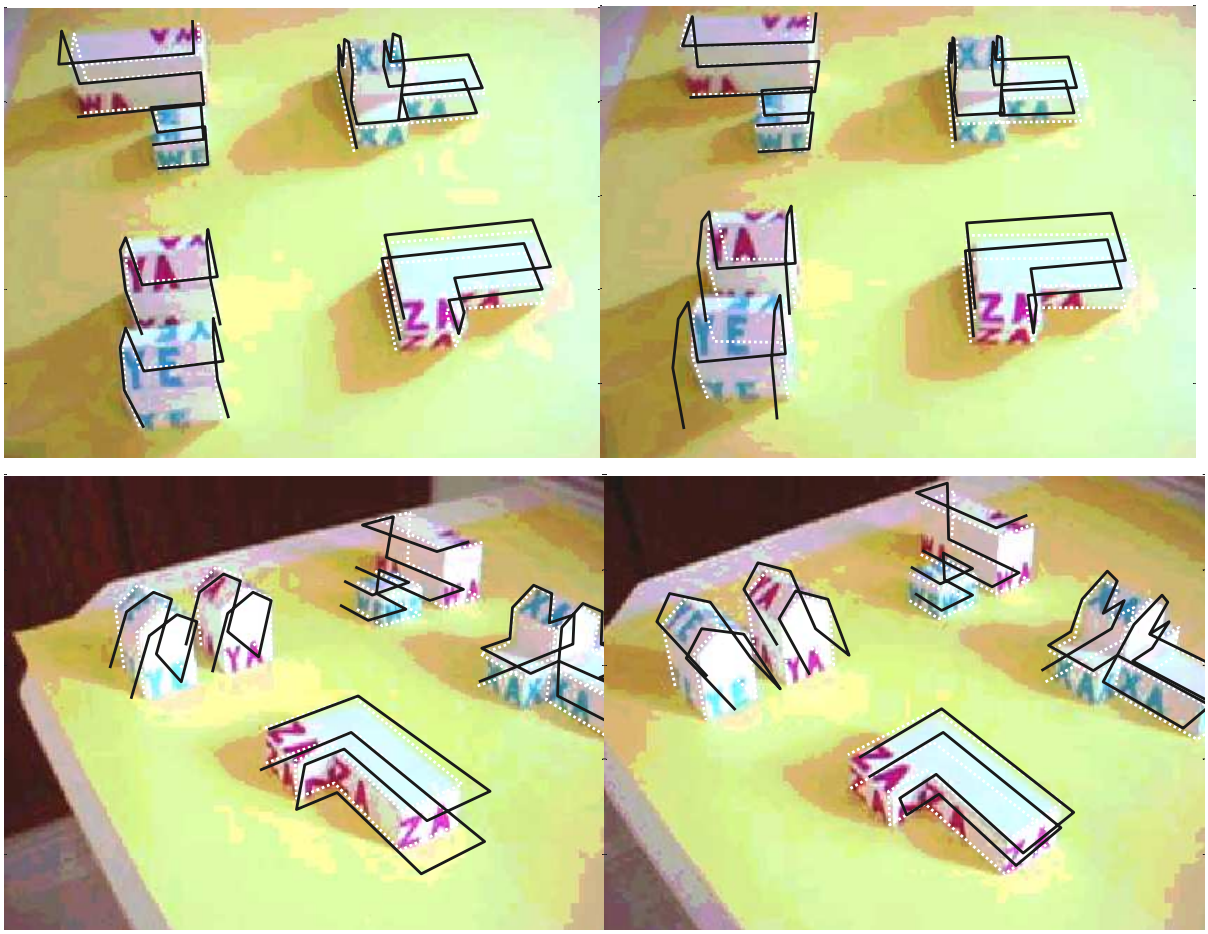


Figura 5-12 – Alguns resultados do alinhamento para baixa resolução.

A linha pontilhada branca mostra o modelo alinhado utilizando-se todos as quinas visíveis selecionadas manualmente. A linha preta contínua mostra o alinhamento obtido utilizando-se um conjunto de 6 feições regionais arbitrárias.

5.3.6 Reconhecimento

A figura 5-13 mostra os histogramas de votos obtidos pelo reconhecimento. Cada linha corresponde a um histograma de votos que avalia qual a feição-modelo que é mais provavelmente uma identificação correta para a feição-imagem da linha. Assim, para cada linha uma feição-imagem é identificada.

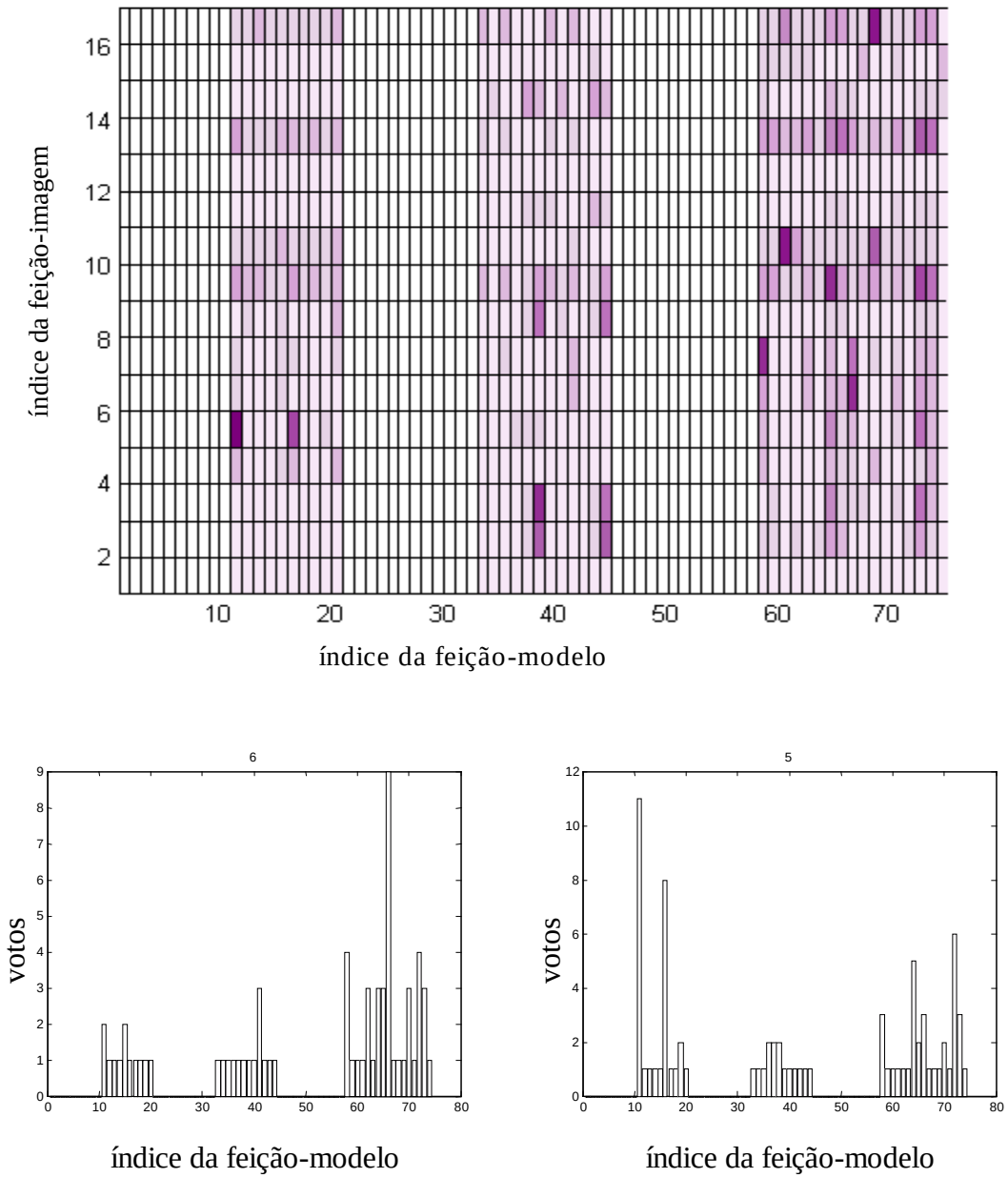


Figura 5-13 – Resultado do reconhecimento.

No gráfico acima, cada linha representa uma feição-imagem e cada coluna representa uma feição-modelo. A intensidade corresponde ao número de votos que a respectiva correspondência imagem-modelo obteve. Os histogramas abaixo correspondem às linhas 6 e 5, ilustrando a identificação das respectivas feições-imagem.

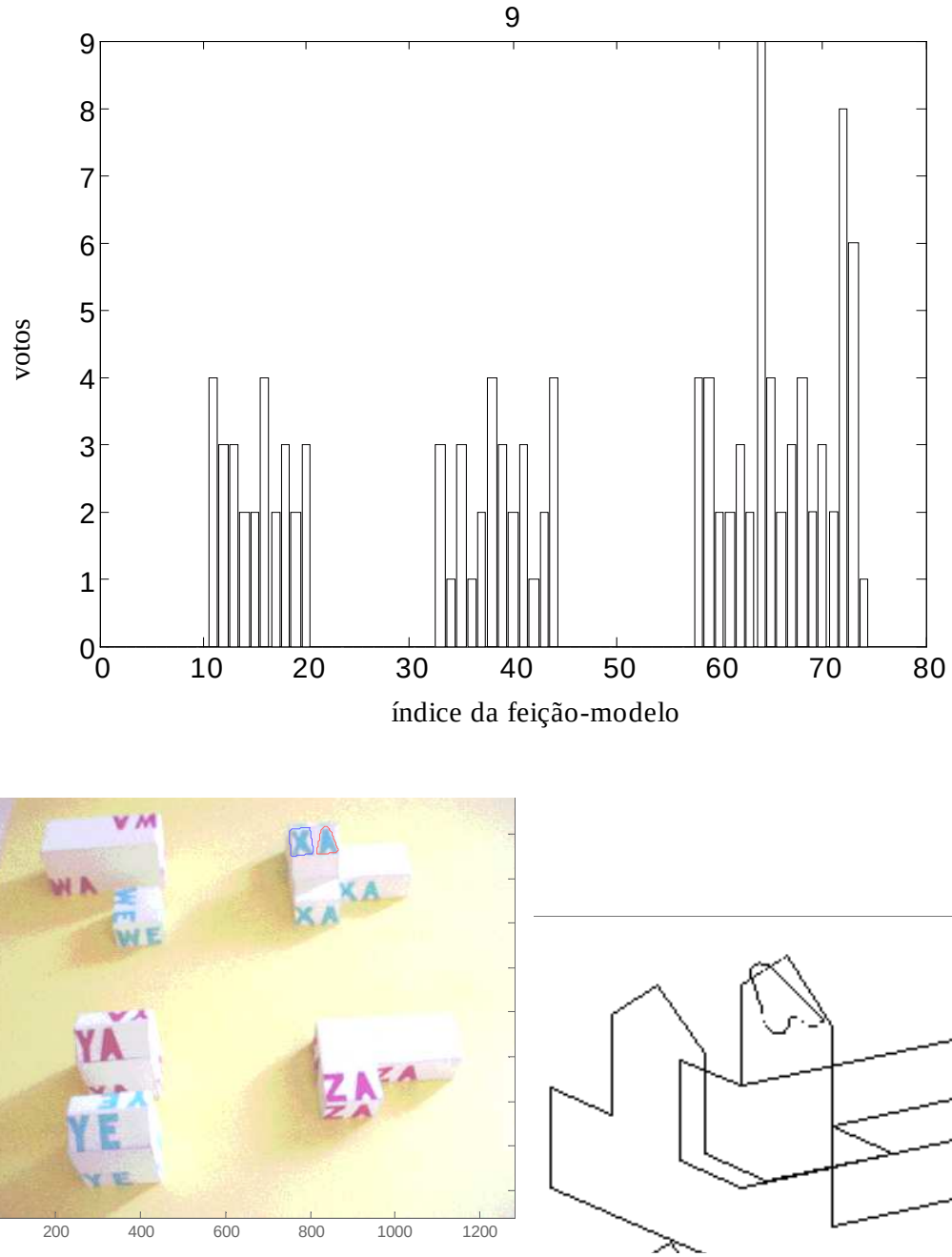


Figura 5-14 – Uma feição identificada.

Este é um histograma de votos discriminando a feição-imagem de número 9 como a feição-modelo de número 64 que por obter o maior número de votos é, portanto, a identificação mais plausível. Abaixo, uma das vistas utilizadas, com a feição escolhida (uma das assinaladas) para teste do reconhecimento. Na direita, o contorno reprojetado da feição é apresentado em detalhe.

Concluímos, pela discriminação realizada, que o método é capaz de discriminar feições pela estrutura de múltiplas feições regionais. A figura 5-14 mostra a identificação de uma determinada feição. A feição na forma de um “A” (índice 64) é mostrada na imagem da maquete com seu fecho-convexo marcado. O histograma mostra que nesse caso foi difícil discriminá-la da feição marcada com forma de um “X” (índice 72) por terem número de votos muito próximos devido a sua proximidade e paralelismo.

Entretanto, não é necessário reconhecer todas as feições. Nossos experimentos mostram que com 6 feições se obtém um bom resultado de alinhamento. Assim, casos de discriminação frágeis podem ser desconsiderados.

Variando-se a granularidade das tabelas *hash*, isto é, modificando a indexação e o número de linhas e colunas, concluímos que o resultado é muito dependente deste fator. Algumas vezes não foi possível decidir por uma feição porque mais de uma feição apresentou o número máximo de votos. Por outro lado, a granularidade das tabelas deve considerar a imprecisão das medidas dos atributos e o nível aceitável de colisões da tabela *hash*.

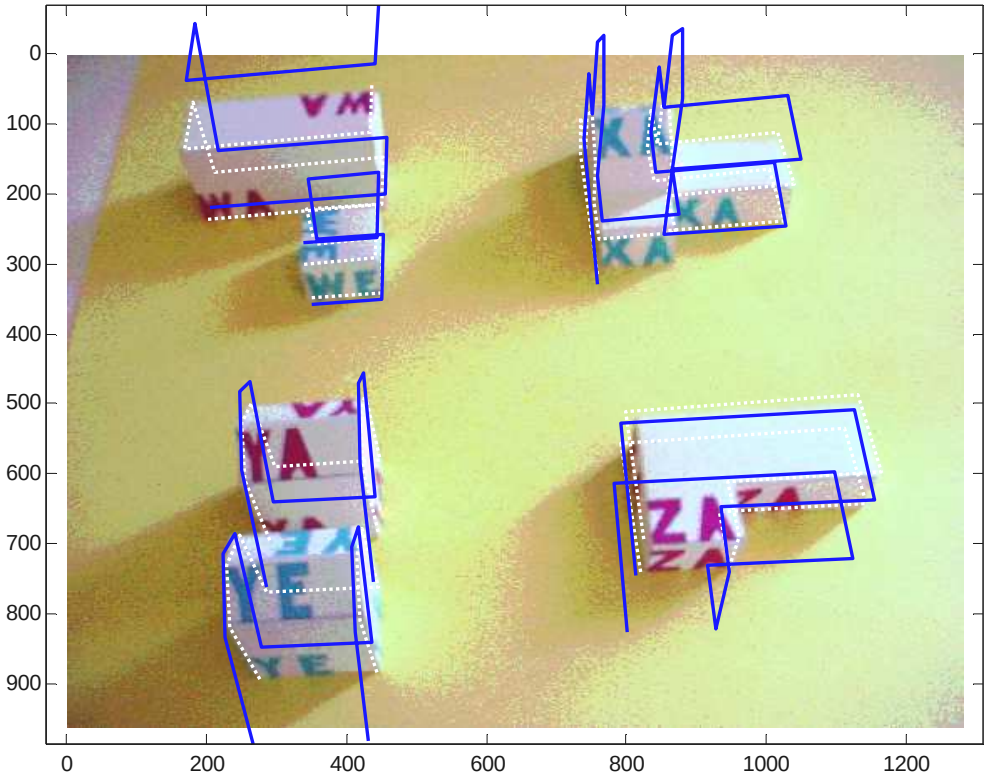


Figura 5-15 – Alinhamento de imagem de alta resolução (1280x960) utilizando apenas 4 feições.

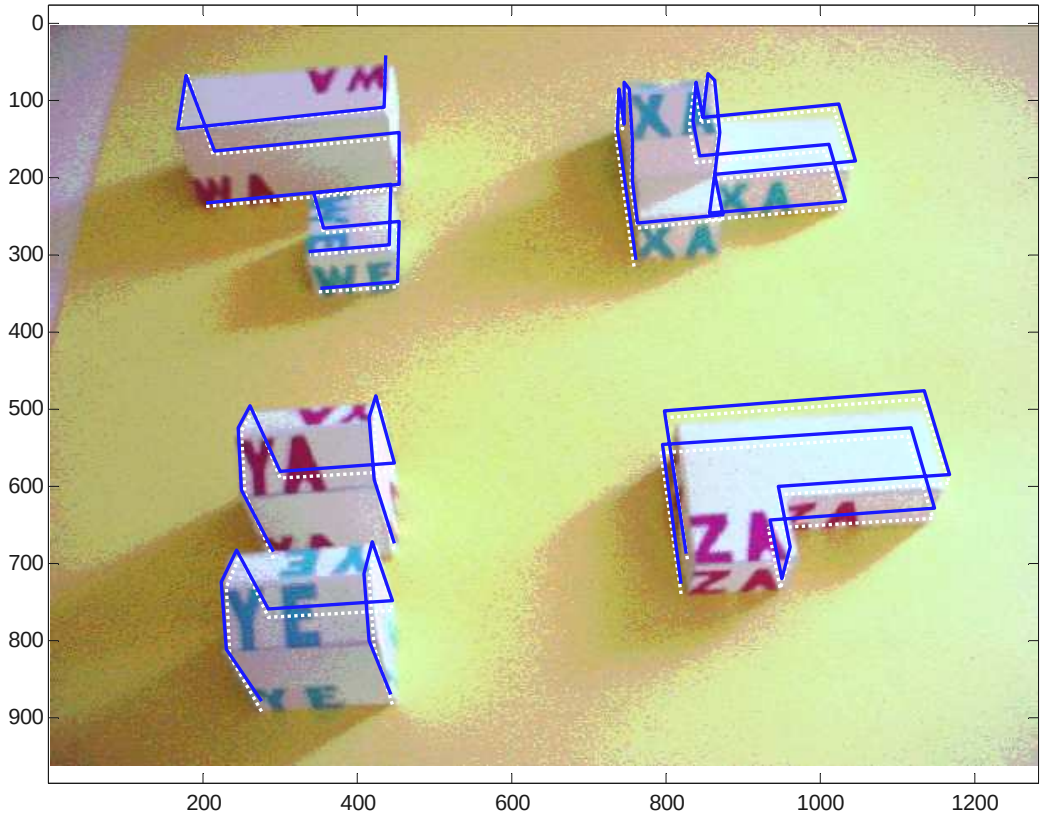


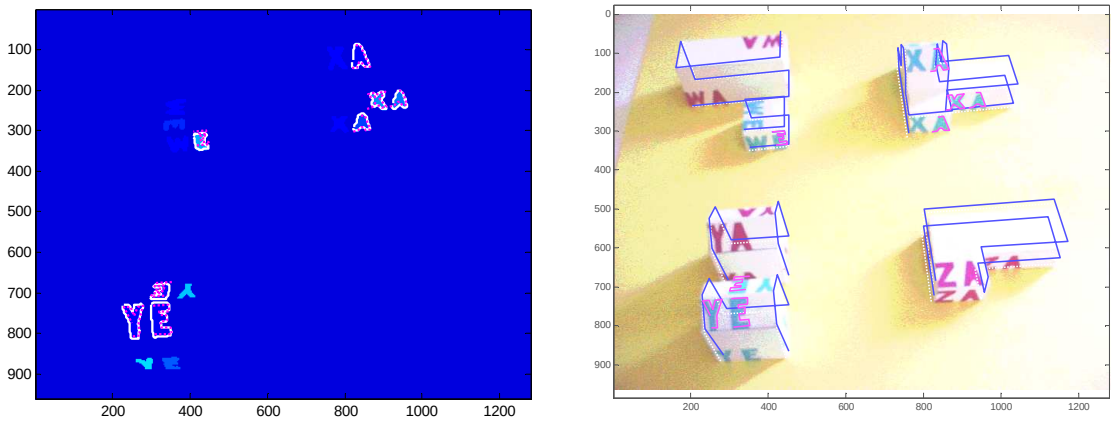
Figura 5-16 – Alinhamento de imagem de alta resolução (1280x960) utilizando apenas 6 feições.

Uma vez identificadas as correspondências entre feições da imagem e feições do modelo e ordenadas essas correspondências em função do número máximo de votos, passamos a avaliar a determinação da pose consolidada.

Para a imagem JPEG com resolução 1280x960 das figuras 5-15 e 5-16, foram reconhecidas adequadamente as 7 melhores feições, a partir da oitava melhor associação começaram a surgir associações incorretas. Utilizando 4 feições para o alinhamento, obtivemos erro de 8,7 cm de posição da câmera e 7,68 graus de orientação dado que a distância da câmera à cena era da ordem de 0,5 m. Utilizando 6 feições, obtivemos erro de 1,1 cm de posição da câmera e 0,98 graus de orientação.

Em outro experimento, de 16 feições, erram-se apenas 5 associações. As 8 melhores feições foram identificadas corretamente. Utilizando-se as 8 feições se obteve erro de 1,18 cm para posição e 0,99 graus para orientação da câmera.

O critério para avaliar o reconhecimento da feição é o número de votos conforme a estimação por máxima verossimilhança, porém, outros critérios podem ser definidos baseados no histograma como um todo. Utilizando a distância de Hausdorff para comparar as regiões originais de uma das imagens com as regiões do modelo reprojetadas utilizando a pose estimada, obtivemos medidas de erro em pixels de 3,4 a 8,6 para as 8 feições reconhecidas. Assim, o erro máximo estimado para o alinhamento obtido em pixels é de 8,6 pixels para elementos próximos às feições. Veja na figura 5-17 as feições reconhecidas utilizadas.



(a) (b)

Figura 5-17 – Feições reconhecidas utilizadas na estimação da pose

Em (a) os contornos em branco contínuo são contornos das regiões da imagem e aqueles em magenta tracejado são reprojeções do modelo. Em (b) é mostrado o alinhamento e o contorno das feições do modelo reprojetadas sobrepostas.

5.4 Discussão

Descrevemos a metodologia de testes e sua implementação. Apresentamos os resultados obtidos nos experimentos implementados. Analisamos esses resultados, mostrando a viabilidade de utilizar métodos eficientes de reconhecimento para determinar a correspondência entre feições da imagem e feições do modelo para realizar alinhamento geométrico de forma eficiente.

Conseguimos um bom resultado de alinhamento quanto à precisão e robustez. Mostramos os resultados para estimação de pose e alinhamento para as feições reconhecidas, com um resultado bastante satisfatório no fim do processo. O método se mostrou potencialmente robusto à oclusão por realizar o alinhamento detectando apenas parte do conjunto de feições. O método também é robusto considerando o *clutter*, pois a presença de feições espúrias não impediu o reconhecimento das feições relevantes. Aplicamos o algoritmo de reconhecimento para essas feições, conseguindo identificar um número grande de feições.

Efeitos prejudiciais externos podem vir também do *clutter* que pode produzir uma quantidade de votos espúrios, não correlacionados, mas que eventualmente podem influenciar o resultado se a discriminação se der por uma diferença pequena de votos. No caso analisado, pode-se considerar que há *clutter* produzido por feições muito deformadas pela perspectiva, onde não foi possível se determinar adequadamente a orientação, e pedaços de feições regionais produzidos por oclusão parcial das feições ou problemas de segmentação.

Uma possível causa para o erro está na baixa qualidade da determinação da orientação e posição do centróide de cada feição individual. Acreditamos que uma técnica baseada em RANSAC também pode ser utilizada para ajuste de plano nesses casos substituindo o método dos momentos (item 3.4.3) e com isso obter uma medida melhor. Outras fontes de erro que podem ser consideradas são a discrepância entre o modelo físico construído e o modelo virtual original, os erros no processo manual de calibração de câmera e a baixa qualidade das imagens, principalmente quanto ao uso de imagens com compressão.

Possíveis melhorias do método incluem àquelas já relatadas na literatura de espalhamento geométrico [Wolfson e Rigoutsos, 1997]. A função de *hash* pode ser adaptada conforme a

distribuição de entradas do modelo. Técnicas de histograma podem ser utilizadas para diminuir a ocorrência de colisões e ponderar o melhor tamanho de tabela. Outra alternativa é modelar através de probabilidades condicionais, para lidar com a imprecisão das medidas e inferir o resultado junto com a sua qualidade na forma de probabilidade.

Sugere-se, também, a fim de evitar que um erro de associação das feições melhor cotadas não implique um grande erro de estimação da pose, que se utilize uma técnica baseada em poda, como o RANSAC. Assim, podem-se escolher as 8 melhores feições e delas analisar cada grupo de 6 feições (total de 28 grupos) ou uma amostra desses grupos selecionando o que melhor se ajusta pelo método dos quadrados mínimos. Outra possibilidade, que apresenta custo maior, é selecionar aquele grupo cuja reprojeção gera a menor distância em pixels.

No próximo capítulo, resumimos as contribuições deste trabalho e sugerimos trabalhos futuros.

Capítulo 6 - CONCLUSÕES

Propusemos, nesta tese, uma solução para o problema de alinhamento geométrico automático considerando precisão, robustez e custo computacional. O problema mais difícil a ser resolvido é a determinação automática de correspondências entre feições da imagem e do modelo. Abordamos esse problema a partir das idéias para solução do problema de reconhecimento de objetos. Para uma configuração específica de sistema de Visão Computacional que considera feições regionais, visão estéreo e transformações rígidas em 3D, arquitetamos uma solução que se mostrou capaz de resolver o problema mesmo com requisitos severos.

Contribuímos para a solução do problema de alinhamento através da proposta de uma abordagem com as seguintes características:

- é automática em suas etapas *on-line*, dispensando a intervenção do usuário;
- é global, analisando cada nova imagem por inteiro, o que permite ter, em seqüências de imagens, robustez a movimentos abruptos e descontinuidades;
- utiliza exclusivamente câmeras como sensores;
- não considera detalhes da forma das regiões que podem ser de difícil detecção por algoritmos de segmentação e não assume um formato específico para as regiões;

- apresenta baixo custo, conforme a análise de complexidade realizada, sendo o processo de extração das feições o principal gargalo;
- resolve para a projeção perspectiva sem recorrer a aproximações;
- resolve para os 6 graus de liberdade de movimento rígido em 3D;
- é extensível para outros tipos de feições e para combinar com um método incremental preditivo;
- é robusta à oclusão parcial, dado que não é necessária a visibilidade de todas as feições, conforme discussão no item 4.5.2;
- é robusta ao *clutter*, conforme discussão no item 4.5.2;
- é adequadamente precisa no caso avaliado no experimento, dado que foi medido um erro em pixels na proximidade das feições inferior a 1% das dimensões da imagem.

As seguintes condições são as premissas assumidas para o desenvolvimento da abordagem proposta:

- É necessário conhecimento prévio do modelo da cena e só será necessária a modificação artificial do ambiente de trabalho quando feições naturais não estiverem presentes em quantidade suficiente.
- É necessário descrever e representar em memória esse modelo utilizando uma estrutura de dados própria.
- São consideradas feições regionais planas. É um tipo de feição mais genérico que arestas, cônicas e feições de forma pré-determinada. Esse tipo de feição simplifica o processo de autoria e o número de feições consideradas é facilmente controlado por uma filtragem da imagem baseada em área das regiões.
- São consideradas 2 vistas com orientação relativa conhecida.

- As feições não podem ser vistas pelo verso, ocorrer em quantidade insuficiente nem ter estrutura ambígua, como no caso de figuras simétricas.

Além disso, contribuímos com os seguintes algoritmos individuais para análise de imagens com feições regionais:

- extração do fecho convexo de feições coloridas uniformes;
- determinação da correspondência estéreo de feições não-pontuais;
- reconstrução de contorno de região convexa plana em 3D;
- retificação baseada apenas na matriz fundamental.

Outras contribuições do trabalho incluem:

- revisão bibliográfica não-exaustiva de métodos de reconhecimento e rastreio de objetos;
- revisão de métodos da Visão Computacional para análise de imagens estéreo, estimação de pose e descrição de feições;
- modelo e algoritmo para reconhecimento baseado em atributos invariantes de pares de objetos.

A abordagem proposta é uma ferramenta básica de Visão Computacional, podendo ter um grande número de aplicações e trabalhos derivados. Assim, nos itens seguintes, enumeramos possíveis aplicações e trabalhos futuros.

6.1 Aplicações

Apresentamos nos itens a seguir sugestões de aplicações para a abordagem proposta. No apêndice E, encontra-se uma análise mais profunda das aplicações.

- **Inicialização do rastreio óptico**

A nossa abordagem pode ser utilizada para reiniciar uma técnica de rastreio incremental em condições de variação abrupta das imagens e verificá-la em longas seqüências contínuas. O problema de rastreio que consideramos utiliza o modelo tridimensional de objetos sob restrições de objeto rígido, corrigindo ao longo do tempo a pose do objeto.

Técnicas incrementais não são suficientes para resolver o problema do alinhamento no caso geral. Primeiramente, porque os métodos incrementais de rastreio dificilmente respondem de forma adequada aos problemas de variação abrupta. Em segundo lugar, esses métodos precisam do conhecimento do estado inicial da pose da câmera ou dos objetos para poder computar estimações para os estados seguintes.

Tanto o requisito da robustez ao movimento abrupto de câmera como a necessidade do conhecimento de um estado inicial são solucionados com a adoção de uma técnica global. Assim, em oposição às técnicas incrementais, uma técnica global deve determinar a informação necessária a partir somente da observação do último quadro adquirido, não utilizando, portanto, variações. Uma técnica global de alinhamento pode ser empregada ao se iniciar o funcionamento do sistema, quando se detecta que o rastreio incremental falhou ou ainda como supervisora de uma técnica incremental para compensar os erros acumulados em longas seqüências.

A inicialização automática do rastreio é importante porque permite voltar a rastrear de forma incremental após um acidente de oclusão ou movimento brusco sem necessidade de intervenção do usuário. Uma técnica automática e global permite tratar descontinuidades devido a falhas de pré-processamento, oclusão e movimento abrupto e supervisionar o rastreio para corrigir problemas de *drift*, ou erro acumulado. Além disso, técnicas globais são mais aptas para lidar com *clutter*, isto é, distratores em grande quantidade.

- **Robótica, navegação autônoma e auto-localização**

O alinhamento baseado em imagens adquiridas por câmeras é um instrumento de medida da posição e da orientação entre as câmeras e os objetos da cena. Dessa forma, pode ser utilizado em muitas aplicações que dependem de medida geométrica do ambiente, das quais se destaca

a Robótica. Um robô dotado de um sistema de câmeras pode utilizar técnicas de alinhamento para determinar sua posição dadas imagens do ambiente adquiridas por suas câmeras. Além disso, esse robô poderia posicionar adequadamente uma garra de acordo com a posição e a orientação de um objeto a ser manipulado, o que se chama coordenação mão-olho.

- **Edição de vídeo**

Considerando uma imagem de uma cena real obtida por uma câmera, o alinhamento geométrico é um dos requisitos para que a imagem seja modificada de forma realista, por exemplo, acrescentando-se um objeto. O alinhamento permite posicionar adequadamente a imagem do objeto sobre a imagem da cena. Além disso, conhecendo a posição desse objeto virtual a ser acrescentado e a iluminação do ambiente real, o objeto pode ser sombreado de forma coerente com a porção real da imagem.

- **Interfaces modernas, particularmente Realidade Aumentada**

A localização de objetos no espaço é importante para interfaces modernas e pode ser realizada pela abordagem proposta. É uma aplicação que requer precisão, robustez, baixo custo computacional e baixa latência.

- **Fusão de dados, estabilização e super-resolução**

Fusão de dados, estabilização e super-resolução são também possíveis aplicações para o método proposto. A análise de dados de múltiplas fontes é uma questão de fusão de dados, que tem uma boa introdução, por exemplo, em [Hall e Llinas, 1997]. Um dos princípios da fusão de dados é a necessidade de se determinar como primeiro passo o alinhamento dos dados. Em [Irani e Peleg, 1993], imagens de um mesmo objeto sob poses variadas são alinhadas para se obter uma resolução melhor. Assim, é por exemplo possível se ler uma placa de um carro a partir de uma seqüência de imagens pouco nítidas.

6.2 Trabalhos futuros

A seguir, iniciamos a exploração de possíveis caminhos abertos por esse trabalho.

6.2.1 Melhoria dos algoritmos

Em primeiro lugar, é necessária uma avaliação experimental mais profunda da abordagem, realizando testes em conjuntos maiores de cenas. O tempo de processamento, a robustez e a precisão devem ser avaliados em experimentos específicos e considerando as diversas situações possíveis.

O algoritmo de reconhecimento pode ser melhorado através de técnicas mais refinadas para construção da tabela de *hashing* geométrico e esquema de votação. A partir do ferramental probabilístico apresentado, é possível criar um processo de votação ponderada que pode ser integrado com fusão de dados e predição. A integração com uma técnica de rastreio incremental é importante para a criação de um sistema causal de rastreio.

É também importante associar uma técnica para melhorar o alinhamento pixel a pixel. Uma medida de distância entre o alinhamento estimado e o alinhamento correto pela análise pixel a pixel pode ser utilizada para auxiliar na poda para a aproximação por mínimos quadrados. Em aplicações com restrições de tempo menos severas pode-se tentar reprojetar as feições e ajustar a pose até que o alinhamento pixel a pixel seja satisfatório. Isto pode ser feito, por exemplo, com poucos passos de um algoritmo de otimização que considere a soma das distâncias entre pixels ao quadrado.

O método de retificação apresentado é útil na abordagem desenvolvida por trabalhar diretamente sobre as coordenadas de vértices de polígonos. Entretanto, há vezes em que é necessário produzir um novo par de imagens retificadas por reamostragem das imagens originais do par estéreo. O método de retificação proposto não deve ser utilizado dessa forma porque o problema da distorção das imagens não foi considerado em seu desenvolvimento. Uma técnica para automatizar a determinação dos parâmetros livres das transformações de retificação, resultando em menor distorção, deve ser criada.

A reconstrução do contorno das feições não apresentou boa qualidade, resultando em erros de posição e orientação que acabam sendo compensados pelo consenso das múltiplas feições. É preciso investigar um método de melhor qualidade que possa resultar numa precisão melhor para o alinhamento final.

6.2.2 Extensões da abordagem

Uma extensão do método de reconhecimento para abranger modelos deformáveis ou parametrizados, como em [Koller, 1993] pode ser importante para algumas aplicações.

Outra extensão importante é a possibilidade de trabalhar com objetos desconhecidos previamente. Seria necessário um aprendizado de novas feições e relações entre feições durante a fase *on-line* de reconhecimento.

É importante também investigar que tipo de informação adicional pode ser utilizada por um sistema num nível superior para resolver ambigüidades. Essa informação pode se basear em experiências colaterais durante a fase de reconhecimento em uma seqüência de imagens.

Outro desafio é encontrar uma solução para o caso monocular em seqüências de imagens, resolvendo a auto-calibração a partir de feições regionais. Com a auto-calibração o problema estéreo considera uma única câmera em posições diferentes a cada instante. Uma idéia pode partir do método de Basri e Jacobs [1995], que estimam a pose dada a correspondência entre regiões. Outra possibilidade interessante é resolver o caso monocular considerando menos graus de liberdade para a câmera.

6.2.3 Integração com rastreio preditivo

A importância do rastreio preditivo está na construção de sistemas que estimem o estado presente antes da leitura atual dos sensores, como é o caso necessário para sistemas de Realidade Aumentada.

Mesmo que uma técnica global que resolva o alinhamento considerando os dados obtidos em dado instante seja muito eficiente ao ponto de poder substituir o uso de uma técnica incremental, existem situações em que não se pode desconsiderar as variações em uma seqüência de imagens. Este é o caso dos sistemas causais, como em Realidade Aumentada, pois deve-se predizer a informação do instante futuro antes que se possa medi-lo. Assim, a integração dos dois tipos de técnicas é importante.

Exemplos de técnicas de predição são a filtragem de Kalman e os filtros de partículas utilizados em [Fox *et al*., 1999] e [Isard e Blake, 1998].

6.2.4 Aplicação em Realidade Aumentada

Um trabalho interessante seria a aplicação em Realidade Aumentada, construindo um ambiente de trabalho aumentado com objetos conhecidos e um sistema VST, do qual o par de câmeras é utilizado para realizar o alinhamento. Uma arquitetura de sistema de Realidade Aumentada deve incluir uma fase de autoria, para construção dos modelos 3D dos objetos, e uma fase de exibição onde ocorre o rastreio dos objetos ou da câmera.

A fase de exibição é dividida em dois casos: o caso contínuo, em que pouca diferença existe entre cada par de imagens consecutivas de uma seqüência e o caso descontínuo provocado por mudança de contexto, movimento abrupto, oclusão ou falha na detecção de feições. O caso descontínuo pode ser invocado também em seqüências contínuas muito longas como supervisor.

O caso contínuo assume que já existe uma estimação para a pose dos objetos e, portanto, para a posição das feições na imagem. As operações do caso contínuo são: predição das coordenadas futuras, detecção apenas das feições rastreadas, verificação utilizando a multiplicidade de vistas, reconstrução tridimensional das feições e consolidação da pose.

O caso descontínuo ocorre no início da fase de exibição do sistema ou devido a descontinuidades detectadas no processamento do caso contínuo. No caso descontínuo, não se utiliza informação do passado e se estima a pose apenas a partir das feições detectadas nas imagens do quadro corrente. Este caso é, portanto, o principal alvo de nossa abordagem, já que abordagens incrementais são capazes de resolver o caso contínuo uma vez inicializadas corretamente. Algoritmos de reconhecimento permitem a inicialização e a reinicialização do sistema para rastreio sem a intervenção do usuário, permitindo o funcionamento para casos de descontinuidade e movimento abrupto da câmera ou de objetos e corrigir *drift* em seqüências longas.

Apêndice A - INTRODUÇÃO AO RACIOCÍNIO PROBABILÍSTICO

O ferramental probabilístico baseado na fórmula de Bayes é valioso por permitir modelar incerteza da informação, raciocínio baseado em plausibilidade e atualização incremental da informação. Assim, os conceitos apresentados aqui, retirados principalmente de [Pearl, 1998] e [Duda e Hart, 1973] são utilizados em nossa abordagem para o problema de reconhecimento, ao invés da lógica tradicional.

A.1 Probabilidades condicionais

A probabilidade conjunta de dois eventos A e B é escrita como $P(A,B)$ e representa a probabilidade de ocorrência simultânea dos dois eventos. Uma probabilidade condicional de A dado o fato de que o evento B ocorreu é escrita como $P(A \mid B)$. Essas duas formas se relacionam através da equação (A-1)

$$P(A \mid B) = \frac{P(A,B)}{P(B)}. \qquad \textbf{(A-1)}$$

Dois eventos são independentes se a ocorrência de um não altera a expectativa do outro ocorrer. Se A e B forem independentes, então (A-2) ocorre

$$P(A \mid B) = P(A). \qquad \textbf{(A-2)}$$

A probabilidade marginal de um evento corresponde à soma das probabilidades conjuntas desse evento sob todas as possibilidades das demais variáveis. Essa soma é representada em (A-3)

$$P(A) = P(A, B) + P(A, \neg B). \quad \textbf{(A-3)}$$

Se decompusermos B em todas as possibilidades B_i de forma exaustiva e mutuamente exclusiva, a probabilidade marginal de A é dada pela soma (A-4)

$$P(A) = \sum_i P(A, B_i). \quad \textbf{(A-4)}$$

Como a probabilidade conjunta pode ser escrita da forma (A-5),

$$P(A, B) = P(A \mid B) \cdot P(B) \quad \textbf{(A-5)}$$

a probabilidade marginal $P(A)$ pode ser escrita em termos de probabilidades condicionais, o que é dado na equação (A-6)

$$P(A) = \sum_i P(A \mid B_i) \cdot P(B_i). \quad \textbf{(A-6)}$$

A.2 Regra da "inversão" de Bayes

Utilizando a equação (A-1) e o fato que $P(A, B) = P(B, A)$, é fácil obter a fórmula da inversão dada por (A-7)

$$P(H \mid e) = \frac{P(e \mid H) \cdot P(H)}{P(e)}. \quad \textbf{(A-7)}$$

Aqui cabe uma interpretação. A variável H é uma hipótese e a variável e é um novo evento ou informação. $P(H)$ é a probabilidade de ocorrer a hipótese H sem esse novo conhecimento do evento e, chamada probabilidade *a priori*. A partir da ocorrência desse evento, a fórmula computa a nova distribuição de probabilidade da variável H escrita como

$P(H \mid e)$ e chamada probabilidade posterior. A verossimilhança é o termo $P(e \mid H)$. O termo chamado $L(e \mid H)$ em (A-8) é a razão de verossimilhança do evento e dada a hipótese H

$$L(e \mid H) = \frac{P(e \mid H)}{P(e \mid \neg H)}. \qquad \textbf{(A-8)}$$

O termo $P(e \mid H)$ é obtido através de experimentos ou treinamento ou ainda arbitrado por um especialista. O treinamento consiste da estimação dos parâmetros da distribuição de probabilidades a partir de amostras. O termo $P(e)$ é a probabilidade marginal do evento e. É computada através de um somatório como dado pela equação (A-6). Assim a fórmula de inversão pode ser escrita na forma da equação (A-9)

$$P(H \mid e) = \frac{P(e \mid H) \cdot P(H)}{\sum_i P(e \mid H_i) \cdot P(H_i)}. \qquad \textbf{(A-9)}$$

Para isso, a variável H é decomposta em eventos H_i de forma exaustiva e mutuamente exclusiva.

A.3 Inclusão recursiva de informação adicional

Uma das propriedades da formulação bayesiana para o raciocínio probabilístico é que a inclusão de informação adicional é facilmente modelada pela regra da inversão como mostramos em (A-10)

$$P(H \mid e_n, e) = P(H \mid e_n) \cdot \frac{P(e \mid e_n, H)}{P(e \mid e_n)}. \qquad \textbf{(A-10)}$$

Dado um novo fato e acrescido à informação decorrente de fatos antigos e_n, basta multiplicar a distribuição de probabilidades conhecida em um instante pela verossimilhança da nova informação. Essa propriedade permite que a modelagem pelos métodos bayesianos seja utilizada no rastreio incremental e na fusão de dados, uma vez que cada dado novo na forma de uma distribuição de probabilidades pode ser facilmente incorporado de forma incremental.

A.4 Estimação de parâmetros

A estimação de parâmetros da distribuição de probabilidades é o constituinte básico dos processos de treinamento (paramétricos) a partir de amostras ou aprendizado, seja supervisionado ou não. Vamos assumir o caso supervisionado em que amostras D devem ter o comportamento $P(\mathbf{x})$, cuja forma é controlada pelos parâmetros $\boldsymbol{\theta}$. Para isso deve-se otimizar um determinado critério.

O critério da máxima verossimilhança consiste da maximização de (A-11)

$$P(D \mid \boldsymbol{\theta}) = \prod_{k=1}^{n} P(x_k \mid \boldsymbol{\theta}). \tag{A-11}$$

Para adotar esse critério foi necessário assumir a independência das amostras x_k. Esse critério não requer o conhecimento de uma distribuição de probabilidades *a priori*.

É usual utilizar log-probabilidades a fim de simplificar as derivadas da função objetivo. Assim, a solução da estimação de parâmetros por máxima verossimilhança é dada por (A-12)

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{k=1}^{n} \log\big(P(x_k \mid \boldsymbol{\theta})\big). \tag{A-12}$$

Se houver o conhecimento de uma distribuição de probabilidade *a priori*, pode ser mais conveniente utilizar uma estimação MAP ou máxima *a posteriori*, que consiste de adotar como critério a probabilidade posterior dada por (A-13)

$$P(\boldsymbol{\theta} \mid D) = \frac{P(D \mid \boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})}{P(D)}. \tag{A-13}$$

A solução da estimação é dada em (A-14). Se a distribuição conhecida *a priori* for uniforme, o critério MAP é equivalente ao critério de máxima verossimilhança

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log\big(P(x_k \mid \boldsymbol{\theta})\big) + \log\big(P(\boldsymbol{\theta})\big). \tag{A-14}$$

Apêndice B - TRANSFORMAÇÕES GEOMÉTRICAS, RIGIDEZ E ESTIMAÇÃO DE POSE

Definimos, a seguir, os tipos de transformações geométricas mais importantes para a análise do processo de formação da imagem. Essas transformações são as transformações afins, as similaridades, as isometrias e as homografias. As transformações rígidas ou isometrias são subconjunto das similaridades, que são por sua vez subconjunto das transformações afins lineares.

Transformações afins tem a forma da equação (B-1), consistindo em combinações lineares das coordenadas do sistema de referência do domínio mais uma translação. São utilizadas devido à linearidade que simplifica muitos problemas

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \qquad \textbf{(B-1)}$$

Transformações de similaridade são transformações que multiplicam todas as distâncias pelo mesmo fator de escala. É uma generalização do conceito de semelhança de triângulos. Toda similaridade euclidiana pode ser escrita como a composição de escala uniforme, translação, rotação e eventual reflexo.

Transformações rígidas ou isometrias conservam as distâncias de forma que as distâncias entre os pontos transformados são idênticas àquelas entre os pontos originais. Se for considerada a distância euclidiana, as transformações rígidas são constituídas pela composição de transformações de rotação e de translação.

Para a formação da imagem através do modelo de câmera de orifício, é necessário considerar a transformação projetiva planar. Essa transformação pode ser definida em espaços diferentes, por exemplo, com domínio tridimensional e contra-domínio bidimensional. Se considerarmos que o objeto a ser projetado existe sobre um plano, podemos definir uma transformação de plano para plano, a transformação projetiva planar ou homografia, dada pela equação (B-2). Essa transformação é não-linear e difícil de se trabalhar. Seus parâmetros podem ser relacionados a uma transformação rígida seguida de uma projeção perspectiva, porém há, no espaço 8D de parâmetros, regiões que admitem a alteração do aspecto (*aspect ratio*) e da distância focal, que são considerados parâmetros intrínsecos da câmera

$$\begin{aligned} x &= \frac{k_1 X + k_2 Y + k_3}{k_7 X + k_8 Y + 1} \\ y &= \frac{k_4 X + k_5 Y + k_6}{k_7 X + k_8 Y + 1} \end{aligned} \quad . \qquad \textbf{(B-2)}$$

Determinar os parâmetros dos vários tipos de transformações pode ser visto como uma generalização da estimação de pose. A extração da transformação rígida de uma transformação perspectiva planar implica a consideração de restrições não lineares. Os parâmetros k_i da equação (B-2) acima, dados 4 conjuntos de valores para x, y, X, Y originários da correspondência entre 4 pontos de duas imagens, podem ser facilmente obtidos pela solução de um sistema linear. Todavia, não se pode garantir que estes parâmetros representem uma transformação rígida da câmera em 3D. Se a transformação rígida é dada pela equação (B-3), a restrição de rigidez consiste da garantia de ortonormalidade da matriz de rotação 3×3 definida por $\mathbf{R} = \left(r_{ij}\right)$,

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \textbf{(B-3)}$$

que pode ser escrita como as restrições da equação (B-4)

$$\mathbf{R}^T\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad \textbf{(B-4)}$$

Para câmeras de orifício, é importante entender que se não se alterar o centro de projeção da câmera, a transformação na imagem é equivalente a uma transformação de *warping*, isto é, um mapa bijetor entre os espaços de coordenadas das imagens. O fenômeno que corresponde à translação do centro de projeção é chamado paralaxe. A paralaxe é responsável pela percepção de profundidade em um sistema estéreo. A transformação na imagem correspondente à paralaxe não é bijetora no caso geral, sendo assim, são criadas dificuldades para alguns problemas como a construção de imagens panorâmicas a partir de múltiplas imagens de projeção planar. Imagens de câmeras de orifício com mesmo centro de projeção são equivalentes a não ser pela transformação de *warping*, pelo campo de visão e pela amostragem, de forma que imagens planares, cilíndricas e onidirecionais podem ser transformadas entre si através de uma função de *warping*. A distorção de uma câmera também é modelada como uma função de *warping*, usualmente radial.

A estimação da pose consiste na determinação de uma transformação rígida entre o sistema de referência do objeto e o sistema de referência da câmera que melhor represente as associações entre elementos correspondentes nos dois sistemas.

Para estimar a pose da câmera a partir da correspondência de pontos do modelo da cena com o modelo geométrico observado pelas imagens, é esperado que o conjunto de pontos obedeça à restrição de rigidez do objeto para que haja de fato compatibilidade das correspondências com a transformação.

Suponha que seja possível marcar um objeto físico com seis pontos de cores diferentes, medir as coordenadas tridimensionais desses pontos e capturar uma imagem desse objeto onde os seis pontos sejam visíveis e facilmente detectados pelo sistema. Os pontos seriam identificados por sua cor e a correspondência com os pontos do modelo dos quais as coordenadas tridimensionais são conhecidas é, então, imediata. Relacionando as coordenadas de imagem e as coordenadas tridimensionais do modelo através de equações, a pose da câmera em relação ao objeto pode ser estimada.

Usualmente se relaciona estimação de pose com calibração de câmera por serem problemas muito semelhantes. A calibração da câmera consiste da estimação de seus parâmetros intrínsecos e extrínsecos. A diferença é que a estimação de pose busca apenas os chamados parâmetros extrínsecos da câmera, que constituem os parâmetros da transformação rígida. Os parâmetros intrínsecos correspondem a transformações do plano imagem, que podem ser resumidas como uma função de *warping*, comumente não-linear. Entre os parâmetros intrínsecos citamos o centro óptico da imagem, a direção do eixo x sobre a imagem, o ângulo entre os eixos x e y, o aspecto e a distância focal.

- **Pose pelo método não-linear de três pontos**

O algoritmo de estimação de pose por 3 pontos é clássico de Fotogrametria e Visão Computacional e é revisitado por Quan e Lan [1999]. Apresentamos esse algoritmo aqui para ilustrar o problema da não-linearidade da projeção perspectiva e da ambigüidade da solução.

Considerando que se tenha a correspondência entre três pontos $\mathbf{u}_i$ de uma imagem e três pontos P_i do modelo 3D do objeto e o centro de projeção é o ponto C como mostrado na figura B-1, cada par de correspondências contribui com uma restrição quadrática dada por (B-5)

$$f_{ij}(x_i, x_j) = x_i^{\,2} + x_j^{\,2} - 2x_i x_j \cos\theta_{ij} - d_{ij}^{\,2} = 0 \qquad \textbf{(B-5)}$$

onde

$$\begin{aligned} x_i &= \left\| P_i - C \right\| \\ d_{ij} &= \left\| P_i - P_j \right\| \end{aligned} \quad \textbf{(B-6)}$$

É assim formado um sistema polinomial (B-7) onde cada equação é quadrática, sendo equivalente à solução de um polinômio de oitavo grau em x_1. Das 8 soluções possíveis, 4 soluções podem ser eliminadas por consistirem de objeto atrás da câmera, o polinômio pode ser resolvido como um polinômio de quarto grau em $x_1^{\,2}$

$$\begin{cases} f_{12}(x_1, x_2) = 0 \\ f_{13}(x_1, x_3) = 0 \\ f_{23}(x_2, x_3) = 0 \end{cases} . \quad \textbf{(B-7)}$$

A ambigüidade ainda presente pode ser resolvida testando com mais uma correspondência de pontos. Esse método é adequado para se determinar uma transformação rígida, resolvendo o sistema não-linear de equações.

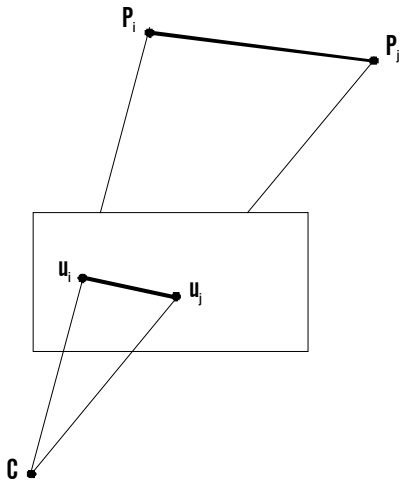


Figura B-1 – Construção geométrica básica para estimação de pose por pontos.

- **Método linear de quatro pontos**

O algoritmo de 4 pontos de Quan e Lan [1999] baseia-se na solução anterior para o caso de 3 pontos. Cada grupo de 3 correspondências, das 4 correspondências entre pontos possíveis,

produz um polinômio de quarto grau. Esse sistema polinomial pode ser escrito da forma (B-8)

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a_1' & a_2' & a_3' & a_4' & a_5' \\ a_1'' & a_2'' & a_3'' & a_4'' & a_5'' \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \end{bmatrix} = 0 . \qquad \textbf{(B-8)}$$

O espaço nulo da matriz (a_{ij}) é gerado por dois vetores $\mathbf{v}_4, \mathbf{v}_5$ e pode ser obtido por SVD. A solução procurada para o sistema subdeterminado é uma combinação linear desses vetores, representada em (B-9) com coeficientes λ e ρ

$$\begin{bmatrix} 1 & x & x^2 & x^3 & x^4 \end{bmatrix}^T = \lambda \mathbf{v}_4 + \rho \mathbf{v}_5 . \qquad \textbf{(B-9)}$$

Utilizando as relações não-lineares entre as variáveis $x^i, 0 \le i \le 4$ dadas em (B-10),

$$x^i x^j = x^k x^l \quad \text{para} \quad i + j = k + l, \; 0 \le i, j, k, l \le 4 \qquad \textbf{(B-10)}$$

um novo sistema linear homogêneo de 7 equações é gerado em função dos dois coeficientes λ, ρ sob relações quadráticas (B-11)

$$\mathbf{B}_{7\times 3} \begin{bmatrix} \lambda^2 \\ \lambda\rho \\ \rho^2 \end{bmatrix} = 0 . \qquad \textbf{(B-11)}$$

Esse sistema é superdeterminado e pode ser resolvido por SVD, encontrando um vetor nulo e a razão entre os coeficientes λ/ρ. Voltando aos componentes da equação (B-9), finalmente é possível se determinar cada coeficiente e a incógnita x.

- **Métodos com número superior de pontos**

Quan e Lan [1999] também apresentam um algoritmo para 5 ou mais pontos. O algoritmo é mais simples, sendo necessária apenas a solução de um sistema homogêneo da forma (B-8)

porém com 6 ou mais linhas (de fato, para n pontos, são obtidas $\frac{(n-1)(n-2)}{2}$ equações), bastando fazer a decomposição SVD e adotar a última linha da matriz $\mathbf{V}$ como solução.

O método de Quan e Lan [1999] tem algumas limitações, uma delas é que o número de equações cresce proporcionalmente ao quadrado do número de correspondências de pontos. Outra é a influência de correspondências incorretas, devendo-se aplicar um método robusto. A grande vantagem do método de Quan e Lan [1999] é que se obtém a solução para a pose considerando a transformação rígida e suas restrições não-lineares.

O método mais comum para a estimação da pose é a solução apresentada no item 3.1.4 dadas as correspondências de pontos estabelecendo valores para x, y, X, Y, Z. Cada nova correspondência entre pontos é um novo par de equações, formando um sistema superdeterminado para 6 ou mais pontos. Dessa forma, para um número grande de pontos e para uso prático é preferível utilizar este método. A inconveniência é que se obtém como resultado uma transformação mais geral que pode não ser rígida, sendo necessária uma técnica para encontrar a transformação rígida mais próxima à solução obtida. Uma forma de se fazer isso é através da fatoração da matriz de transformação em rotação, translação e deformação linear (escala e cisalhamento), como apresentamos no item 3.1.5.

Apêndice C - ESTIMAÇÃO ROBUSTA PELO RANSAC

A aproximação por quadrados mínimos é utilizada porque um sistema superdeterminado pode não ter solução. Sendo a correspondência e a detecção de feições tarefas difíceis, é comum a existência de erros e falhas. Por isso, não é bom se resolver a aproximação por mínimos quadrados para exatamente todos os dados de entrada. É melhor que se tenha um algoritmo capaz de tolerar alguns erros dos dados de entrada, através de poda. Dos métodos estatísticos robustos, o RANSAC (consenso de amostras aleatórias) [Fischler e Bolles, 1981] é o mais utilizado em Visão Computacional. A descrição simplificada do algoritmo a seguir é segundo [Fisher].

Tendo-se M itens no total e sabendo que os parâmetros podem ser estimados a partir de N itens, definem-se a probabilidade p_g de um item fazer parte de um bom modelo e a probabilidade p_f de que o algoritmo termine sem encontrar uma estimação adequada. O algoritmo RANSAC consiste nos seguintes passos:

1. Selecione N itens aleatoriamente e estime os parâmetros;
2. Determine quantos itens dos M se encaixam no modelo com os parâmetros atuais dada uma tolerância, chame essa quantia de K;
3. Se K for grande o suficiente, aceite a estimação e termine;

4. Repita de 1 a $4L$ vezes;

5. Se chegar nesse ponto, termine como uma falha.

O valor de L é dado por (C-1)

$$L = \frac{\log p_f}{\log\left(1 - p_g{}^N\right)}. \qquad \textbf{(C-1)}$$

Assim, esse algoritmo seleciona um conjunto de pontos que admite uma aproximação tolerável. Os demais pontos são considerados *clutter*, objetos de distração ou ruído.

Apêndice D - ESPALHAMENTO GEOMÉTRICO NO CASO AFIM 2D

Descrevemos, a seguir, o método de espalhamento geométrico para o caso bidimensional e transformação afim, de acordo com [Veltkamp, 2001].

- **Construção da tabela**

A construção da tabela *hash* ocorre numa fase *off-line* e todas possíveis combinações de feições contribuem com itens a adicionar na tabela. Escolhendo-se três pontos e_0, e_1, e_2 para formar uma base, todos pontos do plano podem ser representados da forma $e_0 + \kappa(e_1 - e_0) + \lambda(e_2 - e_0)$, o plano (κ, λ) é quantizado numa tabela bidimensional com par de índices inteiros (k, l).

Sejam N conjuntos de pontos alvos B_i. Para cada conjuntos de pontos alvos, faz-se o seguinte. Para cada três pontos não colineares e_0, e_1, e_2, do conjunto de pontos, expresse os demais pontos da forma $e_0 + \kappa(e_1 - e_0) + \lambda(e_2 - e_0)$ e adicione a tupla (i, e_0, e_1, e_2) a entrada (k, l) da tabela. Para $O(m)$ pontos em cada conjunto B_i, a construção da tabela *hash* tem complexidade $O(Nm^4)$.

- **Referência à tabela**

A tabela é indexada *on-line* de forma muito eficiente. Dado um conjunto de pontos A, que se deseja identificar e determinar a transformação afim, escolhem-se três pontos não colineares e'_0, e'_1, e'_2 do conjunto de pontos e se expressam os demais pontos como $e'_0 + \kappa(e'_1 - e'_0) + \lambda(e'_2 - e'_0)$ e se acumula um voto para cada tupla (i, e_0, e_1, e_2) na entrada (k,l) correspondente da tabela. A tupla (i, e_0, e_1, e_2) que receber mais votos indica o conjunto de pontos alvo T_i contendo o conjunto de pontos procurado. A transformação afim que mapeia e'_0, e'_1, e'_2 na base vencedora e_0, e_1, e_2 é assumidamente a transformação entre as duas formas. A complexidade de realizar o *matching* para um conjunto de n pontos é $O(n)$. A fim de obter a pose, deve-se combinar a transformação de normalização com a transformação tabelada.

Apêndice E - ÁREAS DE APLICAÇÃO

Concentramos o estudo de aplicações em três áreas. Em navegação autônoma, o problema de alinhamento ocorre na forma da auto-localização dos sensores. Em interfaces modernas, a localização e reconhecimento de objetos e ambientes são questões importantes aonde soluções para o problema do alinhamento são aplicadas. E a última área de aplicação que consideramos é a realidade aumentada, onde o alinhamento entre modelo e dados sensoriais deve ser estimado sob severas restrições de tempo, precisão e robustez.

E.1 Navegação autônoma

Em robótica, o problema de localização global para a navegação de veículos autônomos, pode ser abordado de forma equivalente ao problema considerado na tese. O problema de localização consiste em determinar a posição e a orientação de um robô a partir de seus dados sensoriais e do conhecimento do mapa do recinto. O robô precisa conhecer sua localização para planejar sua trajetória e atingir seu objetivo.

De forma similar ao problema geral de rastreio, há duas formas de localização, o caso local e o caso global. O caso local é equivalente ao problema incremental do rastreio, onde já se tem uma idéia inicial aproximada da posição que precisa ser corrigida a partir dos dados sensoriais. No caso global, o robô não sabe mais sua posição e deve ser capaz de estimá-la apenas pelos dados sensoriais e pelo mapa conhecido. O problema de localização global é conhecido como o problema do robô seqüestrado [Se *et al.*, 2001]. Essa segunda forma corresponde ao

problema abordado na tese, com a seguinte ressalva, é admitido utilizar uma seqüência de quadros até que seja possível determinar a localização do robô sem ambigüidades.

É costume a forma incremental e a forma global aparecerem em conjunto, pois, uma vez determinada a pose do robô pela forma global, o algoritmo incremental pode ser reiniciado para atualizar rapidamente a situação do robô a partir de suas observações sensoriais. Abordagens como a de Fox *et al*. [1999] integram totalmente as duas formas, através de uma etapa de predição de estado que analisa globalmente as hipóteses devidas aos dados sensoriais instantâneos. No item 4.5.3 propomos uma forma de integração da nossa abordagem com técnicas incrementais.

A abordagem que propomos pode ser utilizada para a localização global da câmera ou de objetos no caso de robôs que possuem câmeras por sensores desde que o ambiente possa ser descrito com os tipos de feições que admitimos. Enquanto que em nossa abordagem consideramos 6 graus de liberdade para movimento rígido em 3D, normalmente, as abordagens para o problema da robótica que são encontradas na literatura consideram 3 graus de liberdade para movimento sobre o plano: um ângulo de rotação e duas coordenadas do plano para translação.

E.2 Interfaces modernas

Interfaces modernas são definidas em contraposição ao modelo de *desktop* ou modelo WIMP (*windows, icons, menus and pointers*) e buscam eliminar descontinuidades (*seams*) entre mundo físico e mundo virtual. Modelos anteriores tem como limitação mais óbvia a não utilização de todos os meios sensoriais humanos na comunicação e dificuldades em representar a informação espacial tridimensional. Além disso, o trabalho auxiliado por computador, como atualmente se apresenta, é caracterizado pela comunicação em dois modos de atenção distintos, ora o trabalhador se comunica com o computador, ora com o restante do ambiente. Esse isolamento do ambiente é chamado imersão e nem sempre é conveniente.

Enumeramos os princípios de maior destaque que são utilizados no projeto de interfaces modernas. O princípio mais importante é a continuidade entre mundo real e mundo virtual. Isso é obtido através de dispositivos para mediar a percepção, através de grandes superfícies

funcionando como displays e sensores e através da adição de significado digital a objetos físicos. Outro princípio diretor do projeto de interfaces modernas é o uso de todo espectro sensorial, com a interpretação de movimentos, gestos e fala. A presença remota de um usuário, por exemplo, é possível através da transmissão desses sinais. Dentro do espectro sensorial, a periferia dos sentidos é utilizada para prover informação importante que, porém, não necessita de atenção imediata. Outro princípio importante é fazer as interfaces capazes de reconhecer o contexto, simplificando a seqüência de interações.

A evolução das interfaces recebe vários nomes na literatura de acordo com os princípios mais importantes para cada tipo de abordagem. Assim, interfaces tangíveis se referem a manipulação tridimensional de objetos no mundo físico e a objetos físicos com significado digital. A computação ubíqua considera a continuidade entre os mundo físico e virtual escondendo a computação sob objetos do dia a dia. Ambientes perceptivos e computação sensível ao contexto analisam a situação do mundo físico para contextualizar a interface. A realidade aumentada também é uma forma de interface moderna, discutida em destaque, que consiste de uma forma para mediar a percepção do usuário.

Esses modelos de interfaces se beneficiariam diretamente de métodos de rastreio e reconhecimento de objetos. Tomamos como exemplo as interfaces tangíveis. O objetivo das interfaces tangíveis [Ishii e Ullmer, 1997] [Fitzmaurice *et al.*, 1995] é aproveitar as habilidades desenvolvidas para manipulação de objetos com as duas mãos no mundo físico: agrupar, empurrar e até atirar objetos e a capacidade de lembrar de objetos pelo posicionamento no espaço físico. Para lidar com as posições e orientações dos diversos objetos físicos relacionados a dados no mundo cibernético são necessários sensores para determinar suas posições e orientações.

Outro exemplo é a computação ubíqua (*Ubiquitous computing*), Weiser [1991], que consiste no conceito da presença de interfaces com computadores por todo lugar a fim de que toda situação cotidiana do ser humano seja auxiliada por computador. É sugerido que estes dispositivos computacionais sejam pequenos e em grande quantidade para que operem de forma transparente e que as descontinuidades sejam abrandadas, ou seja, trata-se de um paradigma diferente de computação e interface com o usuário que coloca o computador em um

segundo plano, tentando deixá-lo imperceptível, e tira proveito da periferia dos sentidos para interface com o homem. A posição e a orientação de objetos, neste caso, correspondem à posição e à orientação de dispositivos que podem ser, por exemplo, computadores pessoais, superfícies de display ou mesmo usuários humanos. Algumas concepções incluem tornar todas as superfícies de um espaço arquitetônico dispositivos de comunicação com o mundo virtual, de forma que estas superfícies constituam displays e sensores.

Atualmente, é mais comum encontrar, nos protótipos de interfaces modernas, sensores magnéticos para rastreio com 6 graus de liberdade, o que implica a necessidade de um acoplamento físico por cabos. Nos casos em que sensores ópticos são adotados, geralmente se utilizam marcas ativas ou passivas introduzidas artificialmente para reconhecimento e estimação de pose. Outras técnicas de percepção visual são também utilizadas, por exemplo, detecção e interpretação de gestos.

Há aplicações para as quais o alinhamento geométrico é útil, porém é mais do que suficiente para resolver o problema. Pode ser que apenas uma noção aproximada da posição e da orientação sejam necessárias, ou até mesmo apenas a identificação do objeto ou a intensidade do movimento dos objetos da cena. Nestes caso, formas mais simples de solução são adotadas. O primeiro exemplo a mencionar são interfaces caligráficas ou gestuais [Rubine, 1991] [Baudel e Beaudouin-Lafon, 1993]. Para esse tipo de interface, o rastreio da trajetória traçada por objetos pontuais pode ser suficiente para reconhecer um comando na forma gestual. Em interfaces baseadas em visão, há casos em que a orientação da mão do usuário pode ser utilizada para servir como um controle tipo *joystick*. Neste caso, métodos estatísticos simples aplicados sobre os pixels das regiões da imagem já são suficientes para aproximar a orientação. Outro exemplo são as interfaces diagramáticas [Wellner, 1993] [Saund, 1999], onde a análise de imagens é responsável pelo reconhecimento de comandos grafados acompanhados de sua localização sobre o plano.

Outras situações em que o problema de alinhamento não necessita ser resolvido por completo ocorrem no caso dos ambientes perceptivos. Ambientes perceptivos assistem o usuário e interagem com ele observando suas ações e inferindo suas intenções. Um carro capaz de observar o motorista e a cena em que o carro está presente pode ajudar a prevenir

acidentes, alertando o motorista ou até mesmo acordá-lo se detectar que este caiu no sono na direção. Na arte dramática, o ambiente computadorizado pode se tornar um ator e contracenar com os demais atores no palco [Pinhanez e Bobick, 2003]. Em entretenimento, um ambiente virtual conduz uma estória que pode ter seu rumo modificado pela atividade dos expectadores [Bobick *et al.*, 2000]. Nesses casos, o conhecimento completo das poses dos objetos da cena permite resolver o problema, mas da mesma forma podemos resolvê-lo apenas com medidas simples de posição e área de regiões sobre a imagem e pela medida de energia das variações da imagem.

O problema do tratamento de *phicons* em interfaces tangíveis e computação ubíqua, muitas das vezes, pode ser resolvido por meios mais simples do que o alinhamento. *Phicons* ou *physical icons* são objetos físicos que representam uma determinada informação digital. Por exemplo, um crachá pode representar seu dono, um alfinete num mapa pode representar uma cidade, uma fita de vídeo pode representar um filme específico. Ícones físicos são utilizados por interfaces tangíveis para simular a manipulação direta da informação, organizada espacialmente. A informação associada a um ícone físico pode ser modificada, assim, constituindo um container virtual de dados. Algumas vezes, a posição e orientação de um *phicon* no espaço é relevante. Por exemplo, um ícone representando uma câmera pode ser utilizado para definir a pose de uma câmera virtual no espaço tridimensional para síntese de imagens.

Um conceito derivado do phicon que é muito importante para as interfaces modernas são os *passengers*. *Passengers* [Streitz *et al.*, 1999] são uma forma especial de ícone físico em que uma informação é associada ao objeto físico e depois pode ser reconhecida e recuperada em outra parte do sistema. Tratam-se de containeres virtuais de dados com a função especial de definir um processo de transferência de dados entre dispositivos ou contextos. Para serem caracterizados como *passengers* devem poder ser identificados por um dispositivo, chamado *bridge*, e devem ter identificação única. Não há necessidade de implementá-los como dispositivos ativos, bastando um mecanismo passivo para sua identificação.

Um exemplo de ambiente que utiliza marcação para rastreio de phicons é o *Illuminating Light* [Underkoffler e Ishii, 1998]. Nele, os *phicons* que se encontram sobre uma superfície

plana são rastreados por câmera utilizando marcas feitas de material retrorefletivo. Esses *phicons* são representações de componentes ópticos (lentes, fontes de luz etc) utilizados numa simulação onde a entrada de dados é a posição e a orientação desses objetos e a saída é uma representação gráfica dos raios de luz projetada sobre a superfície em que se encontram os *phicons*.

Além de rastrear *phicons* que são representações de dados, é também prática rastrear dispositivos. A posição geográfica de computadores pessoais sobre uma superfície pode ser utilizada para controlar mecanismos de troca de informação e para identificar o dono de cada computador pessoal. Há, na verdade, a necessidade de detectar as relações entre dispositivos que, por sua vez, podem ser conseqüência da posição geográfica. Por exemplo, enquanto que no i-LAND [Streitz *et al.*, 1999] o computador deve ser conectado a um dispositivo tipo *docking station* de posição geográfica fixa para ser identificado e para se poder negociar a transferência de informação, no *Augmented Surfaces* [Rekimoto e Saitoh, 1999] os computadores são identificados visualmente por câmeras através de marcas e suas posições sobre uma grande mesa-display são rastreadas. A interface da mesa pode ser utilizada, então, para especificar operações de transferência de documentos entre computadores através de *drag-and-drop* ou *hyperdragging*, denominação para essa operação de arraste de objetos entre dispositivos. Nesse ambiente, além de computadores pessoais, *phicons* também podem ser utilizados como fontes, containeres e receptores de documentos. Por exemplo, um phicon pode ser a representação das caixas de som do ambiente de forma que um arraste de um documento de som para a posição desse phicon ative uma operação de *play-back*. As marcas no *Augmented Surfaces* são marcas consistindo de quadrados pretos e brancos estruturados na forma de uma matriz. Os quatro cantos da matriz são sempre pretos de forma a se poder realizar estimação de pose. As posições vizinhas dos cantos são sempre brancas e as demais codificam a identificação do objeto físico marcado.

Marcas são também utilizadas por Rekimoto e Nagao [1995] para criar um protótipo de computação sensível ao contexto. O paradigma de computação sensível ao contexto permite simplificar a interface do usuário detectando o contexto do usuário automaticamente de forma que não seja necessário questioná-lo a esse respeito. O NaviCam [Rekimoto e Nagao, 1995] é um sistema portátil com câmera e display onde o contexto é detectado visualmente através do

reconhecimento de códigos de barras coloridas. Por exemplo, quando o usuário se aproxima da porta fecha do escritório de uma pessoa, o NaviCam pode reproduzir uma mensagem deixada por ela uma vez reconhecidas as marcas. A codificação das marcas no NaviCam consiste da leitura das cores de 4 barras.

Como exemplo do uso de *phicons* sem a necessidade de alinhamento preciso, citamos o exemplo do *Active Badge*. No sistema de *Active Badge* [Want *et al.*, 1992], pessoas são identificadas e têm suas posições geográficas rastreadas através de um crachá que emite sinal infravermelho que é detectado por sensores colocados nos diversos recintos de um ambiente de trabalho. A localização espacial tem uma granularidade muito larga (entre salas de um edifício) para que um alinhamento preciso seja necessário.

Nas interfaces modernas, um exemplo notável em que o alinhamento preciso em 3D é feito por rastreio magnético é o *Chameleon*. O *Chameleon* [Fitzmaurice, 1993] é uma implementação do paradigma de lentes, é um display que funciona como uma janela para o mundo virtual e o que é apresentado em sua tela depende de sua posição e orientação no espaço. O *Chameleon* consiste em um receptor de TV portátil do tamanho da mão que é rastreado. Um sistema computadorizado transmite o sinal de TV. Entretanto, o *Chameleon* não utiliza câmeras, mas sim um sensor magnético de posição e orientação de seis graus de liberdade. O conjunto da pequena tela do *Chameleon* com o rastreio consegue produzir uma experiência equivalente à visualização de cenas virtuais em *displays* de 21 polegadas estáticos.

E.3 Realidade aumentada

Em ambientes virtuais, o usuário é imerso numa realidade criada artificialmente através da obstrução de sua percepção do mundo real. Recentemente, cresceu o interesse em deixar de obstruir a percepção do mundo real e tentar unir as duas realidades. Assim, dispositivos para mediar a percepção do mundo real, editando-a conforme a intenção do autor do ambiente virtualmente aumentado, passam a ser utilizados a fim de criar uma realidade aumentada. Realidade aumentada é um caso particular de ambiente virtual criado por computador em que o usuário percebe o mundo físico em que habita e simultaneamente é capaz de perceber objetos virtuais alinhados no mesmo espaço tridimensional do usuário. O artigo de Azuma

[1995] é a principal introdução ao assunto (atualizada em [Azuma *et al.*, 2001]), mostrando os diversos problemas e objetivos da realidade aumentada. O autor tenta definir o que exatamente é realidade aumentada e conclui que os seguintes três elementos devem estar presentes: (1) composição do real e do virtual, (2) interação em tempo real e (3) alinhamento em 3D. Observa-se que o autor deseja excluir o caso *off-line* de edição de vídeo. Aparentemente, apenas o sentido da visão está compreendido nesta definição, entretanto, tato e audição também podem ser alinhados em 3D e sobrepostos à realidade física. Definições anteriores a esta limitam ainda a realidade aumentada para sistemas com HMD, os *head mounted displays*, capacetes com *displays* semi-transparentes.

A mediação da percepção visual é atualmente conseguida através de displays semi-transparentes. Tecnologias de *display* semi-transparentes permitem compor informação visual do mundo físico com informação gráfica sintética. A grande vantagem desse modelo de interação é que o usuário, imerso nesta realidade, pode continuar a se comunicar com o mundo físico da forma tradicional.

A percepção do mundo físico e conseqüentemente a interação do usuário neste mundo pode ser melhorada na realidade aumentada. Os objetos virtuais fornecem informação que o usuário não pode obter diretamente de seus sentidos. Essa informação auxilia o usuário a executar tarefas do mundo real. A realidade aumentada é um paradigma que permite facilitar o projeto da interface com o usuário. Através da percepção mediada, objetos do mundo físico podem ser virtualmente modificados, não havendo necessidade de alterações físicas e adição de componentes eletrônicos ativos a toda ferramenta que se for utilizar.

Essa forma de interface entre homem e computador permite melhor colaboração no ambiente de trabalho ao mesmo tempo que o usuário é inserido num contexto virtual. Um dos principais princípios de interfaces colaborativas é o princípio de *awareness* que consiste no conhecimento do que os demais usuários estão fazendo. A realidade aumentada propicia um ambiente colaborativo adequado por permitir a observação visual direta das ações dos usuários do mesmo ambiente. A existência desse canal social direto facilita entre outras coisas evitar *race conditions* entre usuários humanos através da observação dos movimentos dos outros

usuários e de comunicação face-a-face. Veja, por exemplo, [Fluckiger, 1995] para uma introdução a interfaces colaborativas multimídia.

Dentre os tipos de aplicações da técnica de realidade aumentada, destacamos anotação e visualização de informação espacial. Anotação permite, por exemplo, identificar objetos sem ter que consultar uma fonte externa, uma vez que a informação procurada já é apresentada sobreposta ao mundo físico. Assim, rótulos podem ser associados a objetos e seqüências de operações de manutenção podem ser exibidas virtualmente sobre o mundo físico. Anotação também pode ser utilizada para apresentar modelos tridimensionais de objetos em posições relacionadas a objetos do mundo físico. Como exemplo de aplicação, uma edificação a ser futuramente construída ou que já tenha sido demolida no passado pode ser visualizada no local correspondente no momento atual, com o uso dessa tecnologia. Um exemplo de visualização de informação espacial consiste em sistemas que permitem guiar os movimentos do usuário, por exemplo, auxiliando um cirurgião a fazer uma incisão. Há espaço para aplicações em muitos campos, como turismo, educação, entretenimento, propaganda além de manutenção, construção e medicina.

A arquitetura de sistemas de realidade aumentada pode ser bastante simples, contendo três processos essenciais: (1) extração da informação espacial, (2) *rendering* ou síntese gráfica e (3) apresentação visual através de uma tecnologia de *display*. Por extração da informação espacial se entende que é preciso executar medições sobre os objetos do mundo real, criando modelos geométricos. Síntese gráfica compreende criar uma representação gráfica realista alinhando os objetos do mundo virtual ao mundo físico. A tecnologia de *display* corresponde ao aparato em que se dá a composição dos mundos produzindo os sinais que são enviados aos sentidos do usuário.

Existem alguns esforços para classificar sistemas que misturam elementos físicos e virtuais. O mais notável é o de Milgram e Kishino [1994] que consideram seis classes de sistemas em que há mistura de real e virtual. Na figura E-1, representamos o contínuo de virtualidade e os outros três eixos que Milgram e Kishino adotam para auxiliar a classificação de sistemas de realidade mista.

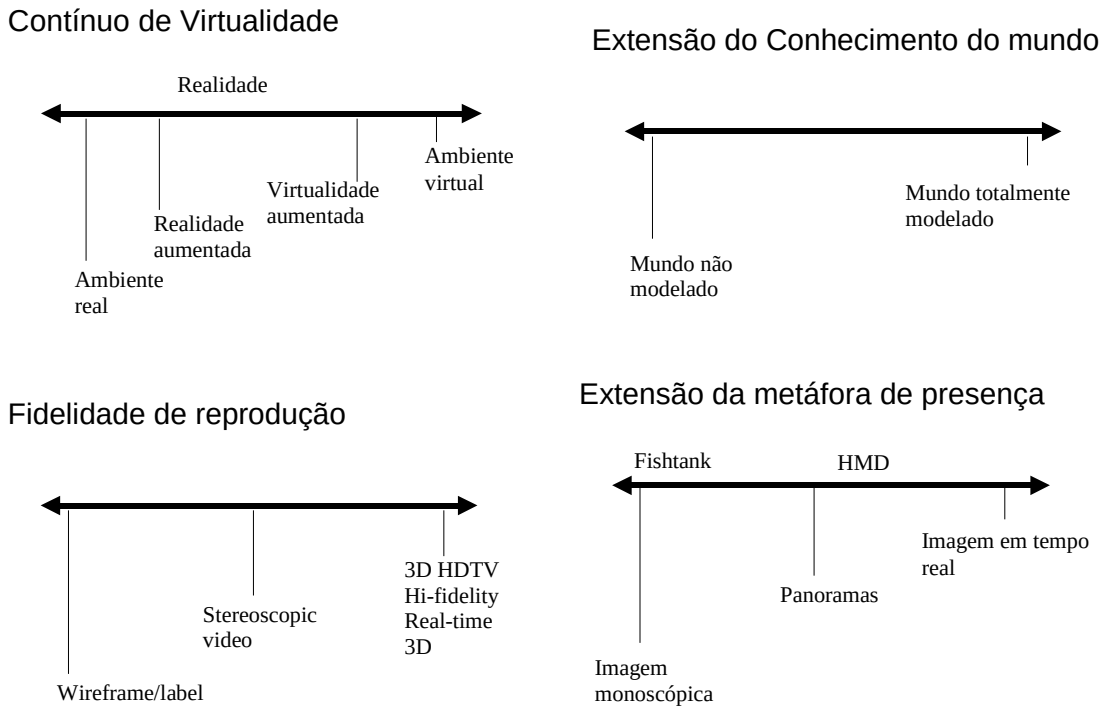


Figura E-1– Eixos de Milgram e Kishino

Atualmente há duas tecnologias principais de *display* para realidade aumentada: *Video See-Through* (VST) e *Optical See-Through* (OST). Um *display see-through* é aquele capaz de misturar a visão do ambiente real com gráficos sintéticos. No caso VST, o que é apresentado no display é a composição do vídeo adquirido com outro vídeo gerado pelo computador. Assim, a visão do usuário é bloqueada, de forma que ele vê o mundo através de câmeras. Esse vídeo pode, então, ser alterado pelo sistema para que o usuário veja o que a aplicação determinar. No caso OST, utiliza-se um espelho semitransparente e um *display* cuja imagem se reflete sobre o espelho. Assim a imagem do ambiente real e a imagem sintética são combinadas de forma óptica. O uso de projetores sobre o mundo real se assemelha ao caso OST. Veja a figura E-2.

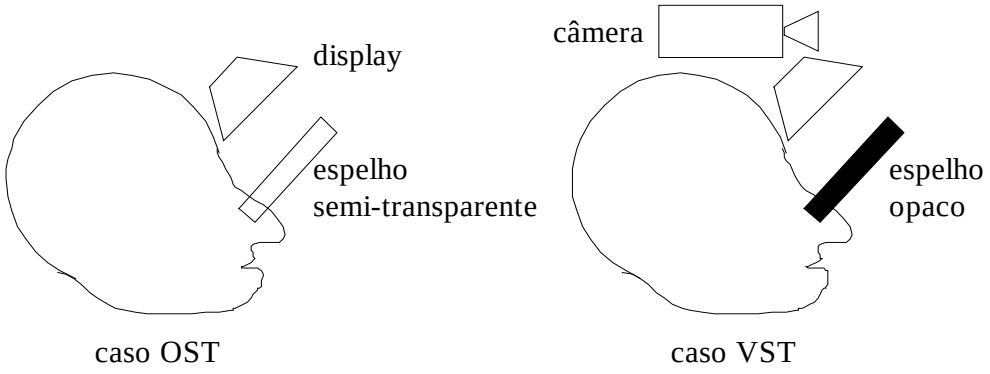


Figura E-2 – Composição de imagens em Realidade Aumentada

Caso OST ou composição por óptica é representado à esquerda com espelho semi-prateado. O caso VST ou composição por vídeo é representado à direita com espelho opaco.

Azuma [1995] discute vantagens e desvantagens dos dois modelos de display. Displays OST diminuem a luz recebida pelo ambiente de forma que a iluminação de um ambiente de trabalho deve ser intensificada. A composição é simples e baseada em c*hroma-key* e *depth buffer*. O problema de atraso é reduzido em OST uma vez que se vê diretamente o mundo físico, entretanto o problema de latência é mais sério. OST não limita a resolução, pois o mundo é visto diretamente, assim problemas devido a amostragem espacial (efeito Moiré) são reduzidos. OST é mais seguro contra queda de energia, uma vez que o mundo ainda pode ser visto com o equipamento desligado. Menor também é o efeito de estroboscopia, pois a visão do mundo real não é amostrada no tempo. Finalmente, em OST não há o problema de paralaxe dos olhos, como no caso de vídeo, onde as câmeras não são posicionadas de forma que o centro de projeção coincida com o do olho.

Por outro lado, sistemas VST são mais flexíveis quanto à composição, visto que objetos reais oclusos por objetos virtuais não são removíveis no caso OST. O projeto de um VST permite um campo de visão maior. É possível no caso de vídeo compensar o erro de alinhamento devido ao atraso entre a imagem do mundo físico e a imagem sintética, entretanto, não compensa o atraso entre o movimento de cabeça e a apresentação da imagem do mundo. Por causa disso, Bajura e Neumann [1995] propõem que se faça do caso VST um sistema em malha fechada, onde a composição é controlada, sendo possível estimar o erro de alinhamento e realimentar o sistema. Já no caso OST, considerado um caso em malha aberta, a

composição sempre terá erro devido ao atraso entre os dois sinais. As câmeras do caso VST podem ser aproveitadas para realizar rastreio ou outras operações de visão por computador. O problema de contraste é resolvido em VST, ajustando-se o brilho de objetos virtuais em relação ao brilho dos objetos físicos.

Há algumas concepções de interface baseadas em projetores sobre superfícies de um escritório ou laboratório. No *I/O Bulb* [Underkoffler e Ishii, 1998] e no projeto *Office of the Future* [Raskar *et al.*, 1998], os pesquisadores chegam ao extremo de querer substituir todas as luzes do escritório por câmeras e projetores controlados por computador a fim de criar ambientes aumentados. O *Office of the Future*, em particular, procura estender um escritório com imagens projetadas de outro escritório remotamente localizado e vice-versa.

Alguns problemas centrais na questão da realidade aumentada são o alinhamento da imagem gráfica com o mundo físico, as restrições críticas de tempo, a síntese de uma percepção ou ilusão convincente e a dificuldade em se especificar o ambiente aumentado. O destaque desta tese é para o alinhamento, mas todas essas questões enumeradas são relacionadas. O problema mais importante da realidade aumentada é o alinhamento em tempo real dos objetos virtuais aos objetos do mundo físico. Os objetos desenhados no *display* sobrepondo a visão do mundo físico devem ser corretamente alinhados em três dimensões para que a ilusão seja realista e para que a aplicação seja útil.

O uso de técnicas de Visão Computacional para criar ambientes aumentados é uma questão em discussão. Por um lado, é uma idéia promissora, pois, através dessas técnicas é possível determinar com precisão as posições e orientações de objetos físicos, reconhecer esses objetos, entender a cena e seu contexto, interpretar gestos e movimentos, eliminar o acoplamento mecânico entre as partes do sistema, coletar informações do ambiente para alterar a percepção com realismo e representar a informação visual. Por outro lado, os problemas de robustez e custo computacional da análise de imagens frente aos severos requisitos de tempo dos sistemas de tempo real diminuem a aceitação desse tipo de solução.

Na realidade virtual, o rastreio de objetos é utilizado principalmente para determinar como projetar a imagem do mundo virtual nos *displays* e coordenar a posição de objetos virtuais

utilizando objetos físicos como instrumento. O rastreio na realidade virtual é raramente acurado, pois os efeitos de erros de alinhamento são dificilmente percebidos. Segundo Azuma [1995], especula-se que a discrepância entre a impressão visual e os sentidos da cinestesia e da propriocepção seja responsável pela sensação de enjôo provocada por muitos sistemas de realidade virtual. Outro fenômeno, chamado captura visual, ajuda encobrir falhas no rastreio. A captura visual é a tendência do cérebro de acreditar mais no que é visto do que no que é captado por outros sentidos.

Ao contrário do que acontece na realidade virtual, em que erros de alinhamento são mascarados, na realidade aumentada, o alinhamento é crítico. Discrepâncias entre o mundo real e o mundo virtual são capturadas diretamente pelo olho e são aceitas apenas na periferia da visão. Na realidade aumentada além de ser necessário alinhar quadro a quadro uma seqüência de imagens ao sistema de referência do modelo numa taxa de quadros por segundo entre 10 e 100 Hz de forma precisa, é necessário eliminar a latência. Devido à sensibilidade do olho humano a variações, erros de alinhamento são detectados muito facilmente e tornam o trabalho impraticável. Erros de alinhamento provocam efeitos somáticos como náuseas. Além disso, os atrasos são responsáveis por supercompensações e oscilações induzidas por piloto.

Uma solução para o problema da latência consiste em considerar métodos preditivos de rastreio. O rastreio preditivo resolve problemas de robustez e permite o funcionamento sem a inserção de atrasos. Requer-se do sistema obtido que seja causal porque é necessário que dependa apenas de informação do passado para predizer o alinhamento no instante presente.

Mann [1994] estuda casos que não necessitam do alinhamento em tempo real para construir sistemas simples com alguma utilidade e no limiar da tolerância do organismo de usuário. Uma das possibilidades para criar um ambiente de colaboração é apresentar em um olho o que outro usuário está vendo e deixar o outro olho livre. O cérebro do usuário é capaz de coordenar o contexto e escolher qual olho se deve dar atenção. Entretanto, esse tipo de esquema vai limitar o usuário por retirar sua capacidade natural de visão estéreo.

Percebe-se, entretanto, que é limitado o número de aplicações sem um efetivo alinhamento tridimensional com o mundo real. Mesmo técnicas que criam uma realidade aumentada

alinhada através de um sistema de referência afim com a imagem [Kutulakos e Vallino, 1995] são de uso limitado por funcionarem nas coordenadas de *display* e não nas coordenadas tridimensionais. O rastreio das posições dos olhos e da cabeça para composição correta das imagens no *display* é importante, mas também a importância dos dados posicionais e de orientação para as aplicações deve ser considerada.

Apêndice F - VALIDAÇÃO DA CALIBRAÇÃO DE CÂMERAS E DA RETIFICAÇÃO

Dois cenários diferentes a respeito do posicionamento das câmeras foram testados. Um objeto em forma de cubo foi criado com centro na origem. Duas posições arbitrárias de câmeras são estabelecidas. Os gráficos descrevendo a cena podem ser vistos na figura F-1. Avalia-se a projeção dos pontos do objeto desenhando-a no espaço e verificando se os pontos estão sobre o plano imagem e se cada ponto da imagem é colinear ao ponto objeto correspondente e ao centro de projeção. Isto é feito modificando interativamente a visão do gráfico. Da mesma forma, os epipolos no espaço podem ser avaliados se são colineares aos dois centros de projeção.

Testamos a retificação em casos sintéticos. A partir das matrizes de projeção conhecidas para as duas vistas, obtêm-se as matrizes essencial e fundamental. A partir da matriz fundamental são calculadas as transformações de retificação pelo método mostrado no capítulo 3. O gráfico mostrando as imagens originais, retas epipolares e suas retificações é visto na figura F-1. A figura F-2 mostra a retificação aplicada em outro cenário, onde as câmera foram reposicionadas. A figura F-3 corresponde a uma retificação aplicada a um par de imagens da cena construída.

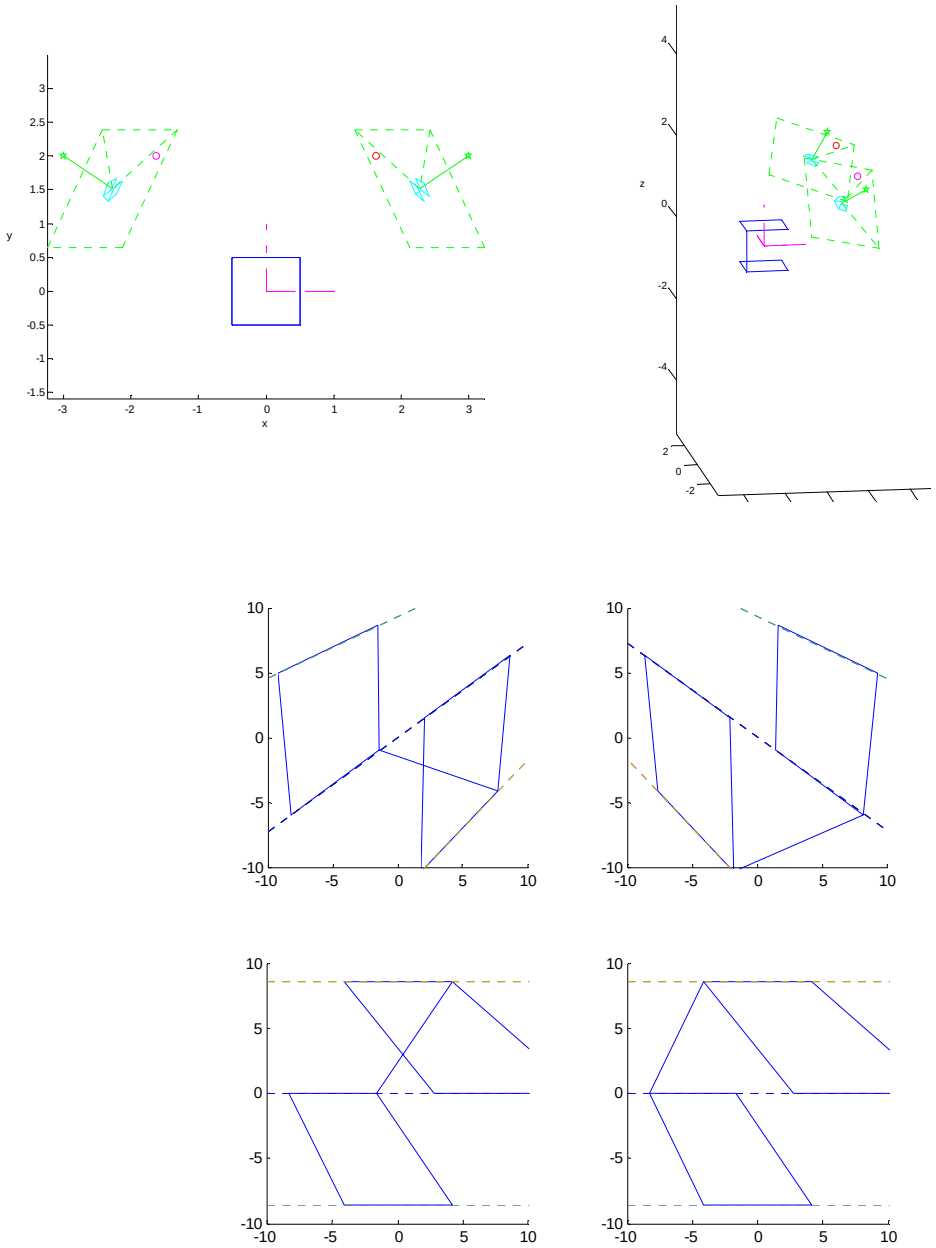


Figura F-1 – Cenário sintético com câmeras estéreo e retificação.

Imagens originais (acima) mostrando retas epipolares e imagens retificadas (abaixo). São mostradas as posições dos centros de projeção, eixos ópticos, imagens e epipolos das duas vistas e objeto para teste.

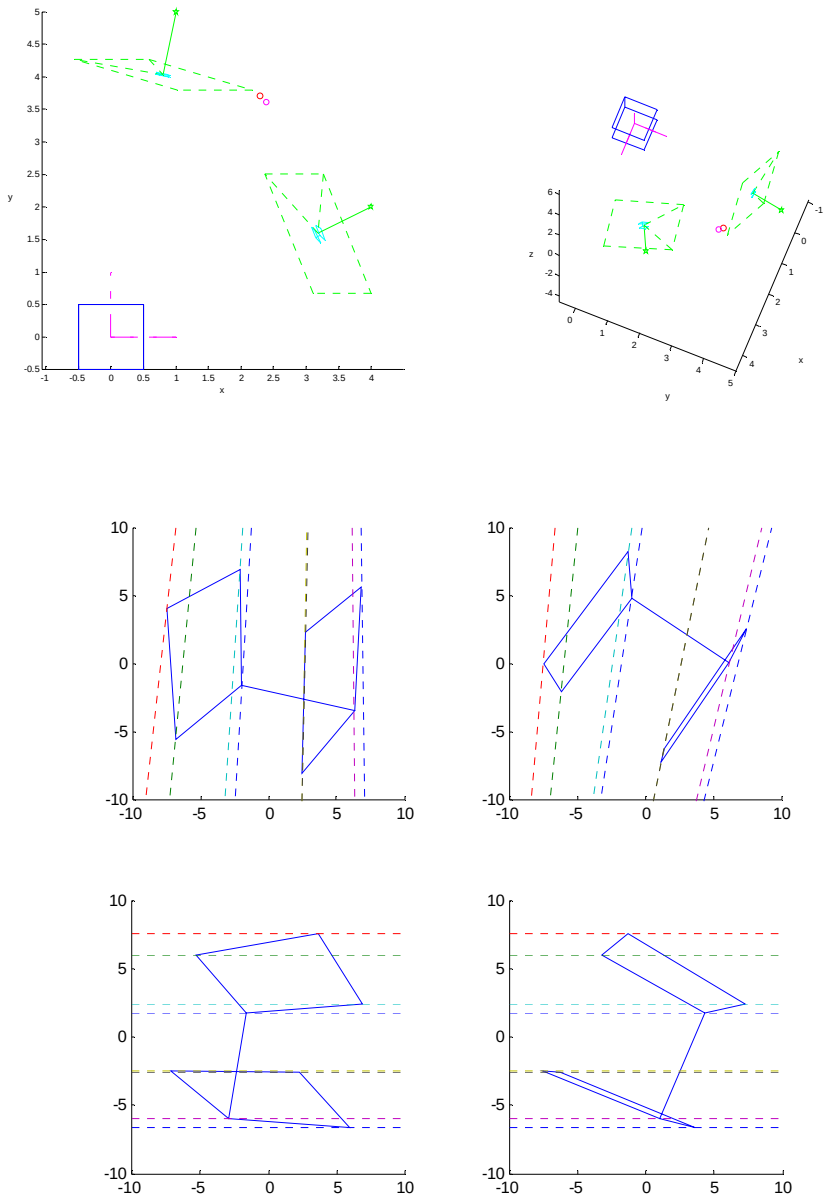


Figura F-2 – Outro cenário de teste para calibração de câmeras e retificação.

São mostradas a cena, imagens originais e imagens retificadas (de cima para baixo). As linhas pontilhadas são retas epipolares.

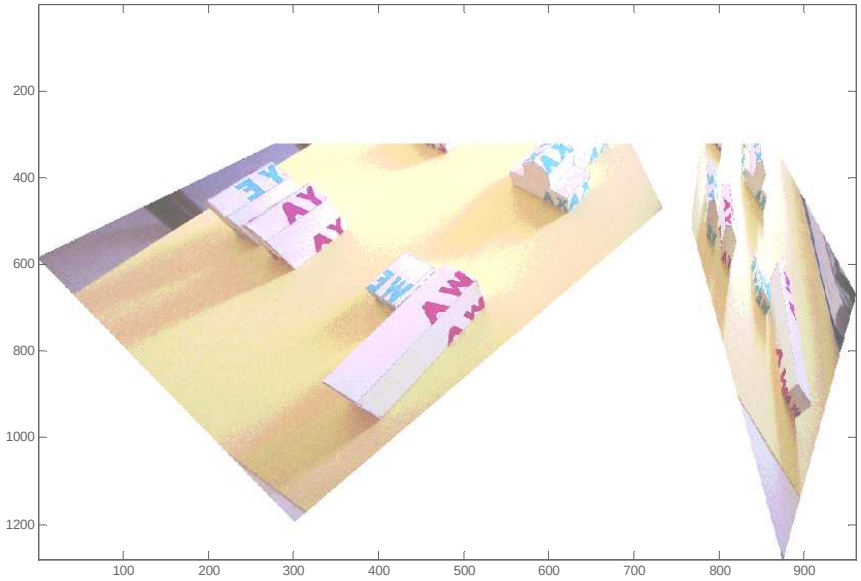


Figura F-3 – Retificação ilustrativa de imagens da cena.

REFERÊNCIAS

1. Azuma, R. T., 1995. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments* 6, 4 (August 1997), 355 - 385. Earlier version appeared in Course Notes #9: Developing Advanced Virtual Reality Applications, *ACM SIGGRAPH '95* (Los Angeles, CA, 6-11 August 1995), 20-1 to 20-38.

2. Azuma, R. T., 1999. The Challenge of Making Augmented Reality Work Outdoors. In *Mixed Reality: Merging Real and Virtual Worlds*, Yuichi Ohta and Hideyuki Tamura, editors. Springer-Verlag, 1999, ISBN 3-540-65623-5. Chapter 21, pp. 379-390. Associated with invited presentation at the First International Symposium on Mixed Reality (ISMR '99) (Yokohama, Japan, 9-11 March 1999).

3. Azuma, R., Y. Baillot, R. Behringer, S. Feiner, S. Julier e B. MacIntyre, 2001. Recent advances in augmented reality. *IEEE Computer Graphics and Application*, 21(6):34-47.

4. Bajura, M., Fuchs, H. e Ohbuchi, R., 1992 Merging Virtual Reality with the Real World: Seeing Ultrasound Imagery within the Patient. *Computer Graphics* (Proceedings of SIGGRAPH 1992), pp. 203-210.

5. Bajura, M. e Neumann, U., 1995. Dynamic Registration Correction in Augmented Reality Systems, *IEEE VRAIS 1995 proceedings*, ISBN 0-8186-7084-3, pp. 189-196.

6. Basri, R. e Jacobs, D., 1995. Recognition using region correspondences, *Proc. 5th Int. Conf. Comput. Vision*, pp. 8-13.

7. Baudel, T., e Beaudouin-Lafon, M., 1993. Charade: Remote Control of Objects Using Freehand Gestures. *Communications of the ACM*, Vol. 36, No. 7, pp. 28-35.

8. Beis, J. S. e Lowe, D. G., 1997. Shape Indexing Using Approximate Nearest-neighbour Search in high-Dimensional Spaces. *Conference on Computer Vision and Patter recognition*, Puerto Rico, pp. 1000-1006.

9. Blake, A. e Isard, M., 1998. *Active Contours*. Springer.

10. Bobick, A., S. Intille, J. Davis, F. Baird, C. Pinhanez, L. Campbell, Y. Ivanov, A. Schutte e A. Wilson, 2000. The KidsRoom. *Communications of the ACM*, 43(3), pp 60-61.

11. Cipolla, R. e Giblin, P. J., 1999. *Visual Motion of Curves and Surfaces*. Cambridge University Press, Cambridge.

12. Cornelis, M. V. K., Pollefeys, M. e Gool, L. V., 2000. Augmented reality from uncalibrated video sequences. In *3D Structure from Images - SMILE 2000*. LNCS 2018, pp 144-160, Springer-Verlag.

13. Duda, R. O. e Hart, P. E., 1973. *Pattern Classification and Scene Analysis*. Wiley-Interscience, New York.

14. Feiner, S., MacIntyre, B. e Seligmann, D., 1993. Knowledge-based Augmented Reality. *Communications of the ACM*, 36(7), pp 52-62.

15. Fischler, M. A. e Bolles, R. C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. ACM*, 24-6, 381-395.

16. Fisher, R. The RANSAC Algorithm. . http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/FISHER/RANSAC

17. Fitzmaurice, G. W., 1993. Situated Information Spaces and Spatially Aware Palmtop Computers. *Communications of the ACM*, Vol. 36, No. 7, pp 38--49.

18. Fitzmaurice, G., Ishii, H. e Buxton, W., 1995. Bricks: Laying the Foundations for Graspable User Interfaces. *Proceedings of the Conference on Human Factors in Computer Systems CHI'95*. pp 442-449.

19. Fluckiger, F., 1995. *Understanding Networked Multimedia: Applications and Technology*. Prentice Hall.

20. Forsyth, D. A., Mundy, J. L., Zisserman, A., Coelho, C., Heller, A. e Rothwell, C., 1991. Invariant Descriptors for 3-D Object Recognition and Pose, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, pp. 971-992.

21. Fox, D., Burgard, W., Dellaert, F. and Thrun, S., 1999. Monte Carlo localization: Efficient Position Estimation for Mobile Robots. *Proc. of the 16th National Conference on Artificial Intelligence*, Orlando, Florida, pp 343-349.

22. Fusiello, A., Trucco, E. e Verri, A., 2000. A Compact Algorithm for Rectification of Stereo Pairs. Machine Vision and Applications 12, pp 16-22.

23. Gdalyahu, Y. e Weinshall, D. Contour matching: problem and related work http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/GDALYAHU/gdalyahu.html

24. Gonzalez, R. C. e Woods, R. E., 1992. *Digital Image Processing*. Reading, MA:Addison-Wesley.

25. Hager, G. D. e Belhumeur, P. N., 1998. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025--1039.

26. Hall, D. e Llinas, J., 1997. An introduction to multisensor data fusion. *IEEE Proceedings*, 85(1), pp 6-23.

27. Haustler, G. e Ritter, D., 1999. Feature-Based Object Recognition and Localization in 3D-Space Using a Single Video Image. *CVIU* 73(1) pp 64-81.

28. Horn, B. K. P., 1989. *Robot Vision*. McGraw-Hill, seventh edition.

29. Irani, M. e Peleg, S., 1993. Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency, *J. of of Visual Comm. and Image Representation*, vol. 4, pp. 324-335, December 1993.

30. Isard, M. e Blake, A., 1998. Condensation - conditional density propagation for visual tracking. *IJCV*, 29:2-28.

31. Ishii, H., e B. Ullmer, 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. *CHI'97*, pp 234-241.

32. Jurie, F., 1998. Tracking objects with a recognition algorithm. *Pattern recognition Letters* (19)3-4, pp 331-340

33. Kanade, T., Narayanan, P. e Rander, P., 1995. Virtualized reality: Concepts and early results. In *Proc. IEEE Workshop on Representation of Visual Scenes*, pages 69-76.

34. Kanbara, M., Yokoya, N. e, Takemura, H., 2002. Registration for Stereo Vision-based Augmented Reality Based on Extendible Tracking markers and Natural Features. Proc. *16th IAPR Int. Conf. on Pattern Recognition (ICPR2002)*, Vol. II, pp. 1045-1048.

35. Koller, D. 1993. Moving Object Recognition and Classification based on Recursive Shape Parameter Estimation. In *Proc. of the 12th Israeli Conf. on Artificial Intelligence, Computer Vision, and Neural Networks*, pp. 359-368, Tel-Aviv, Israel.

36. Kutulakos, K. e Vallino, J., 1995. Calibration-free augmented reality. *IEEE Trans. on Visualization and Computer Graphics*, 4(1):1-20.

37. Lowe, D., 1999. Object Recognition from Local Scale-Invariant Features. *International Conference on Computer Vision*. Corfu, Greece, pp 1150-1157.

38. Lu, F. e Milios, E. E., 1994. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*. Seattle, WA, pp 935-938.

39. Mann, S. 1994. *Mediated Reality*. TR 260, M.I.T. Media Lab Perceptual Computing Section, Cambridge, MA.

40. Milgram, P. and Kishino, F., 1994. A Taxonomy of Mixed Reality Visual Displays, *IEICE Transactions on Information Systems*, vol. E77-D, no. 12. pp 1321-1329.

41. Mundy, J. L., 1994. Object Recognition: the Search for Representation. *ORCV* 95, pp 19-50.

42. Neumann, U. e Park, J., 1998. Tracking for Augmented Reality on Wearable Computers. *Virtual Reality Journal*, No.3, pp.167-175, Springer-Verlag, London.

43. Olson, C. F., 1994. Time and Space Efficient Pose Clustering, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 251-258.

44. Pearl, J., 1998. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgen Kaufmann Publishers, Inc.

45. Peters, G., 2000. Theories of Three-Dimensional Object Perception - A Survey. accepted for: *Recent Research Developments in Pattern Recognition*. Transworld Research Network.

46. Pinhanez, C. e Bobick, A., 2003. Interval Scripts: a Programming Paradigm for Interactive Environments and Agents. *Pervasive and Ubiquitous Computing*; vol. 7(1): 1-21.

47. Pope, A. R. e Lowe, D. G., 1995. Learning Object Recognition Models from Images. In *Early Visual Learning*, Poggio, T. e Nayar, S. (eds.). Oxford University Press, pp. 67-97.

48. Quan, L. e Lan, Z., 1999. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 21, no. 8, pp. 774-780.

49. Raskar R., G. Welch, M. Cutts, A. Lake, L. Stesin e H. Fuchs, 1998. The office of the future: A unified approach to image-based modeling and spatially immersive displays, Proc. SIGGRAPH'98, pp. 179-188.

50. Reiss, T. H., 1993. Recognizing Planar Objects using Invariant Image Features. Springer, *LNCS* 676, Berlin.

51. Rekimoto, J. e Nagao, K., 1995. The World through the Computer: Computer Augmented Interaction with Real World Environments. *Proceedings of UIST '95*, 29-38. ACM Symposium on User Interface Software and Technology.

52. Rekimoto, J. e Saitoh, M., 1999. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *Proceedings of ACM Conference on Human Factors in Computing Systems: CHI 1999*: ACM Press. pp. 378-385.

53. Robert, L., Zeller, C., Faugeras, O.D., e Hebert, M., 1993. *Applications of Non-Metric Vision to Some Visually-Guided Robotics Tasks*, TR2584, INRIA, Sophia-Antipolis.

54. Rothwell, C. A., Zisserman, A., Forsyth, D. A. e Mundy, J. L., 1995. Planar Object Recognition using Projective Shape Representation. *IJCV*(16), No. 1, pp. 57-99.

55. Rubine, D. 1991. Specifying Gestures by Example. *Computer Graphics*, 25(4), pp. 329-337.

56. Rucklidge, W., 1996. Efficient Visual Recognition using the Hausdorff Distance. *LNCS* 1173. Springer-Verlag, Berlin.

57. Saund, E., 1999. Bringing the Marks on a Whiteboard to Electronic Life, *Proceedings of CoBuild'99*. Second International Workshop on Cooperative Buildings. pp. 69-78.

58. Se, S., Lowe, D. e Little, J., 2001. Local and global localization for mobile robots using visual landmarks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Maui, Hawaii pp 414-420.

59. Se, S., Lowe, D. e Little, J., 2002. Global Localization using Distinctive Visual Features. *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, pp 226-231.

60. Simon, G. e Berger, M. O. 1997, *A two-stage robust statistical method for temporal registration from features of various type*. INRIA TR 3235.

61. Stolfi, J., 1991. *Oriented Projective Geometry*: *A Framework for Geometric Computations*, Academic Press, San Diego, CA.

62. Streitz, N.A., Geiler, J., Holmer, T., Konomi, S., Mller-Tomfelde, C., Reischl, W., Rexroth, P, Seitz, P. e Steinmetz, R., 1999. i-LAND: An interactive landscape for creativity and innovation. *Proc. CHI'99*, pp 120-127.

63. Trucco, E. e Verri, A., 1998. *Introductory Techniques for 3D Computer Vision*. Upper Saddle River, NJ. Prentice Hall.

64. Tsai, R., 1987. A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology using Off-the-shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, RA-3(4), pp. 323-344.

65. Uenohara, M e Kanade, T., 1995. Vision-based object registration for real-time image overlay. *LNCS* 905 (*Proc. First International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine*, Nice, France) pp 13-22.

66. Underkoffler, J., and Ishii, H. Illuminating Light: an optical design tool with a luminous-tangible interface, *Proceedings of CHI '98* (Los Angeles CA, April 1998), ACM Press. 542-549.

67. Veltkamp , R. C. e Hagedoorn, M., 1999. *State-of-the-art in shape matching*. Technical Report UU-CS-1999-27, Utrecht University, the Netherlands.

68. Want, R., Hopper, A., Falcao, V. e Gibbons, J., 1992. The active badge location system, *ACM Transactions on Information Systems*, vol. 10, pp. 91-102.

69. Weiser, M., 1991. The computer for the 21st Century. *Scientific American* 265(3): 66-75.

70. Wellner, P., 1993. Interacting with paper on the Digital Desk. *Communications of the ACM*, Vol. 36, No. 7, pp 87-96.

71. Wolfson, H. J. e Rigoutsos, I., 1997. Geometric hashing: An introduction. *IEEE Computational Science & Engineering*, vol. 4, no 4, pp 10-21.

72. Zisserman, A., Simon, G., Fitzgibbon, A., 2000. Markerless tracking using planar structures in the scene. In *IEEE and ACM International Symposium on Augmented Reality*, 2000. pp. 120-128.