Propagação de pontos característicos e suas incertezas utilizando a Transformada Unscented

Leyza Elmeri Baldo Dorini

Dissertação de Mestrado

Instituto de Computação
Universidade Estadual de Campinas

Propagação de pontos característicos e suas incertezas utilizando a Transformada Unscented

Leyza Elmeri Baldo Dorini[1]

17 de Janeiro de 2006

Banca Examinadora:

- Siome Klein Goldenstein
  IC - Unicamp (Orientador)
- Luiz Carlos Pacheco Rodrigues Velho
  IMPA - Instituto de Matemática Pura e Aplicada
- Ricardo Machado Leite de Barros
  FEF - Unicamp
- Jorge Stolfi (Suplente)
  IC - Unicamp

[1]**Este trabalho foi financiado pelo CNPq (processo número 131816/2005-5) no período de Abril/2005 a Fevereiro/2006.**

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecário: Maria Júlia Milani Rodrigues – CRB8a / 2116

Dorini, Leyza Elmeri Baldo
D734p Propagação de pontos característicos e suas incertezas utilizando a transformada unscented / Leyza Elmeri Baldo Dorini -- Campinas, [S.P. :s.n.], 2006.

Orientador : Siome Klein Goldenstein
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Visão computacional. 2. Rastreamento de características. 3. Reconstrução de imagens. I. Goldenstein, Siome Klein. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Propagating feature points and its uncertainty using the unscented transform

Palavras-chave em inglês (Keywords): 1. Computer vision. 2. Feature tracking. 3. Image reconstruction.

Área de concentração: Visão computacional

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Siome Klein Goldenstein (IC-UNICAMP)
Prof. Luiz Carlos Pacheco Rodrigues Velho (IMPA-RJ)
Prof. Dr. Ricardo Machado Leite de Barros (FEF-UNICAMP)

Data da defesa: 20/02/2006

Propagação de pontos característicos e suas incertezas utilizando a Transformada Unscented

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Leyza Elmeri Baldo Dorini e aprovada pela Banca Examinadora.

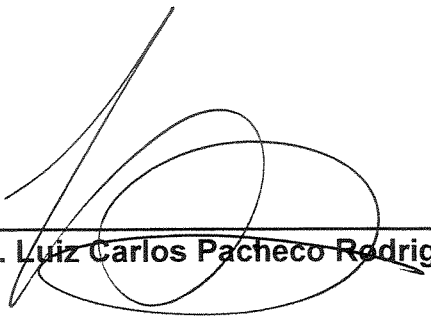
Campinas, 20 de fevereiro de 2006.

Siome Klein Goldenstein
IC - Unicamp (Orientador)

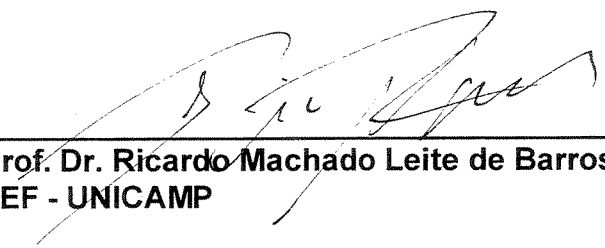
Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 20 de fevereiro de 2006, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Luiz Carlos Pacheco Rodrigues Velho
IMPA



Prof. Dr. Ricardo Machado Leite de Barros
FEF - UNICAMP

Prof. Dr. Siome Klein Goldenstein
IC - UNICAMP

Agradecimentos

Ao meu marido Fábio, por estar sempre ao meu lado, compartilhando momentos alegres e difíceis e, acima de tudo, me incentivando a sempre acreditar em mim mesma. Esta conquista também é tua.

Ao meu orientador, Siome, pelo incentivo e apoio prestados.

A todos que de alguma forma contribuíram para este trabalho: família, professores, colegas e funcionários do IC.

Ao CNPq pelo apoio financeiro.

Resumo

O correto estabelecimento de correspondências entre imagens tomadas de diferentes pontos de vista é um problema fundamental na área de visão computacional, sendo base para diversas tarefas de alto nível, tais como reconstrução 3D e análise de movimento. A grande maioria dos algoritmos de rastreamento de características não possui uma incerteza associada à posição estimada das características sendo rastreadas, informação esta de extrema importância, considerando sua vasta aplicabilidade. É exatamente este o foco principal deste trabalho, onde introduzimos um *framework* genérico que expande algoritmos de rastreamento de tal forma que eles possam propagar também informações de incerteza. Neste trabalho, por questão de simplicidade, utilizamos o algoritmo de rastreamento de características Kanade-Lucas-Tomasi (KLT) para demonstrar as vantagens do nosso método, denominado *Unscented Feature Tracking* (UFT). A abordagem consiste na introdução de Variáveis Aleatórias Gaussianas (GRVs) para a representação da localização dos pontos característicos, e utiliza a Transformada *Unscented* com Escala (SUT) para propagar e combinar GRVs. Mostramos uma aplicação do UFT em um procedimento de *bundle adjustment*, onde a função custo leva em conta a informação das GRVs, fornecendo melhores estimativas. O método é robusto, considerando que identifica e descarta anomalias, que podem comprometer de maneira expressiva o resultado de tarefas que utilizam as correspondências. Experimentos com sequências de imagens reais e sintéticas comprovam os benefícios do método proposto.

Abstract

To determine reliable correspondences between a pair of images is a fundamental problem in the computer vision community. It is the the foundation of several high level tasks, such as 3D reconstruction and motion analysis. Although there are many feature tracking algorithms, most of them do not maintain information about the uncertainty of the feature locations' estimates. This information is very useful, since large errors can disturb the results of the correspondence-based tasks. This is the goal of this work, a new generic framework that augments feature tracking algorithms so that they also propagate uncertainty infomation. In this work, we use the well-known Kanade-Lucas-Tomasi (KLT) feature tracker to demonstrate the benefits of our method, called Unscented Feature Tracking (UFT). The approach consists on the introduction of Gaussian Random Variables (GRVs) for the representation of the features' locations, and on the use of the Scaled Unscented Transform (SUT) to propagate and combine GRVs. We also describe an improved bundle adjustment procedure as an application, where the cost function takes into account the information of the GRVs, and provides better estimates. Experiments with real and synthetic images confirm that UFT improves the quality of the feature tracking process and is a robust method for detect and reject outliers.

Conteúdo

Agradecimentos **vii**

Resumo **viii**

Abstract **ix**

1 Introdução **1**

2 Rastreamento de Características **3**

2.1 Trabalhos Relacionados 5

2.2 Kanade-Lucas-Tomasi Feature Tracker (KLT) 8

2.2.1 Os Modelos de Movimento Translacional e Afim 8

2.2.2 Determinando o Movimento da Imagem 9

2.2.3 Seleção de Características 12

2.3 Erros no Rastreamento de Características 13

3 Rastreamento e filtros preditivos **14**

3.1 Conceitos Básicos 14

3.2 Trabalhos Relacionados - Filtros Preditivos 17

3.3 A Transformada Unscented com Escala 18

4 Nossa Proposta: Unscented Feature Tracking **21**

4.1 Descrição do Método 22

4.2 Rejeição de Anomalias 24

5 Reconstrução 3D **26**

5.1 Trabalhos Relacionados 27

5.2 Ambiguidade na Reconstrução 27

5.3 Recuperando Matrizes de Projeção e Estrutura Projetiva (Duas Vistas) . . 30

5.4 Reconstrução a Partir de Múltiplas Vistas 31

5.5 Atualização da Reconstrução Projetiva Para a Euclidiana 33

5.6 Bundle Adjustment 34

6 Resultados **39**

6.1 Cálculo do Erro 41

6.2 Análise do Rastreamento de Características 42

6.2.1 Sequências Reais 42

6.2.2 Sequências Sintéticas 44

6.3 Aplicação em reconstrução 3D 47

6.4 Comparação com Resultados do RANSAC 49

6.5 Considerações Sobre a Implementação 51

7 Conclusões e trabalhos futuros **52**

Bibliografia **54**

Lista de Tabelas

6.1 Tempos de execução. 51

Lista de Figuras

2.1 Ambiguidade no estabelecimento de correspondências. 4
2.2 O problema da abertura: somente o movimento ortogonal à aresta pode ser determinado 5
2.3 Estrutura geral dos algoritmos de rastreamento de correspondências. 5

3.1 A transformada *Unscented* 18

4.1 Nosso algoritmo. 24

5.1 Reconstrução projetiva. 28
5.2 Reconstrução Euclidiana. 28
5.3 Exemplo de Reconstrução 3D. (a) imagem original, (b) reconstrução projetiva e (c) reconstrução Euclidiana. 29

6.1 Cinco quadros da sequência *Artichoke* real. 39
6.2 Cinco quadros da sequência *Hotel* real. 40
6.3 Cinco quadros da sequência *Artichoke* sintética. 40
6.4 Cinco quadros da sequência *Hotel* sintética. 40
6.5 Cinco quadros da sequência *Hotel* sintética. 41
6.6 Média da distância dos pontos rastreados à linha epipolar (menor é melhor). 42
6.7 Número de características rastreadas. 43
6.8 Características rastreadas na sequência *Hotel* real. 44
6.9 Características rejeitadas pelo UFT. 44
6.10 Resultados para a sequência *Artichoke* sintética (menor é melhor). 45
6.11 Resultados para a sequência *Hotel* sintética (menor é melhor). 45

6.12 Resultados para a sequência *Cow* sintética (menor é melhor). 46
6.13 Resultados para a sequência *Hotel* sintética ao desconsiderarmos as características descartadas (menor é melhor). 47
6.14 Erro de reprojeção (sequências reais - menor é melhor). 48
6.15 Erro de reprojeção (sequências sintéticas - menor é melhor). 48
6.16 Erro de reprojeção na sequência *Cow* (menor é melhor). 49
6.17 Resultados do rastreamento para o último quadro da sequência *Cow*. $\oplus$ indica a posição estimada e $\square$ a real. 50
6.18 Resultados do rastreamento para o último quadro da sequência *Artichoke*. $\oplus$ indica a posição estimada e $\square$ a real. 51

Capítulo 1

Introdução

Devido fatores tais como oclusão, condições de iluminação, e até mesmo problemas numéricos, o rastreamento de correspondências é um procedimento sujeito a erros, os quais podem vir a comprometer a precisão das tarefas de mais alto nível que utilizam as correspondências.

Como a grande maioria dos algoritmos de rastreamento de pontos característicos não considera o erro intrínseco das estimativas, procuramos abordar este aspecto neste trabalho. Associamos a cada ponto característico uma medida de incerteza, representada por uma distribuição de probabilidade. Para limitar a demanda computacional do algoritmo, utilizamos apenas os dois primeiros momentos da distribuição (média e covariância), representando a localização de cada característica por uma variável aleatória Gaussiana (GRV - *Gaussian Random Variable*). Desta forma, modelamos o problema de rastreamento como uma transformação de variáveis aleatórias.

Para estimar a variável aleatória durante o processo de rastreamento, utilizamos a transformada *Unscented* com escala (SUT - *Scaled Unscented Transform*), a qual calcula as estatísticas de uma variável aleatória que sofre uma transformação não-linear.

Neste trabalho, a transformação não-linear é representada por qualquer algoritmo de rastreamento de características. Utilizamos o Kanade-Lucas-Tomasi (KLT) *feature tracker* [29, 39, 37], amplamente utilizado na área de visão computacional, para mostrar os benefícios e bons resultados de nosso método. O seu critério de similaridade baseia-se

na minimização da Soma das Diferenças ao Quadrado (SSD - *Sum of Squared Differences*) das intensidades em uma região de suporte em torno da característica de interesse. Denominamos nossa abordagem *Unscented Feature Tracking* (UFT).

Além de associar uma medida de incerteza, o método proposto pode ser considerado robusto, sendo que também detecta e descarta automaticamente anomalias (*outliers*), pontos que são rastreados para posições distantes da real.

Neste trabalho, também visamos melhorar a precisão das correspondências estimadas. Para tal, utilizamos conceitos de filtros preditivos, uma família de técnicas de estimação de parâmetros que buscam estimar o estado ótimo de um sistema. A idéia básica consiste em combinar à uma estimativa inicial outras medidas do sistema de forma a obter uma solução mais precisa. Neste trabalho, buscamos utilizar tanto informações locais da imagem quanto informações do próprio algoritmo de rastreamento.

Este trabalho está organizado como segue:

Capítulo 2: falamos sobre o problema das correspondências, abrangendo os conceitos teóricos básicos, e também sobre alguns algoritmos de rastreamento de características, com enfoque especial no KLT, utilizado neste trabalho.

Capítulo 3: buscamos modelar o problema de rastreamento, dentro do contexto de filtros preditivos. Apresentamos também a SUT.

Capítulo 4: apresenta a formalização do nosso método, o UFT.

Capítulo 5: trata do tópico reconstrução 3D, descrevendo também como a aplicação do UFT pode trazer uma melhoria na qualidade do processo. Além disso, descrevemos o algoritmo de *Structure from Motion* (SfM) utilizado neste trabalho e a técnica de *bundle adjustment*.

Capítulo 6: resultados ao comparar os algoritmos KLT e UFT (utilizando o KLT como algoritmo de rastreamento) em três aspectos: melhoria na precisão das correspondências estimadas, detecção/rejeição de anomalias e reconstrução 3D.

Conclusões e trabalhos futuros: conclusões do trabalho e também algumas de suas possíveis extensões.

Capítulo 2

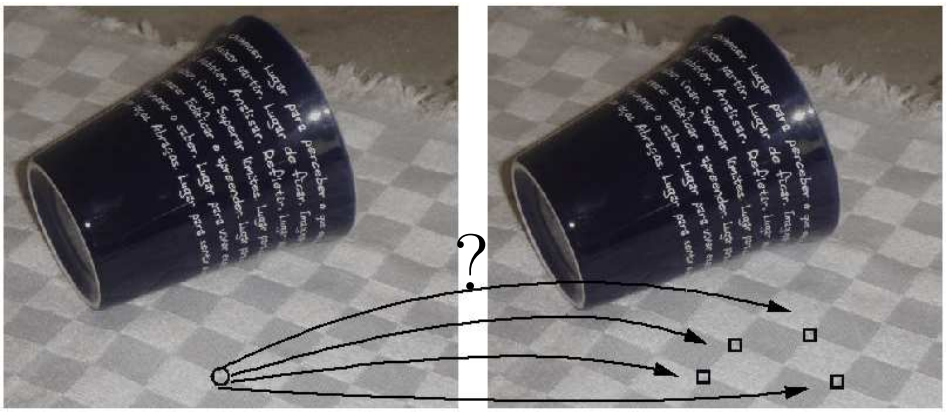
Rastreamento de Características

O rastreamento de características, também chamado de problema das correspondências, consiste em estabelecer qual ponto em uma imagem corresponde a um determinado ponto em outra, no sentido de eles corresponderem ao mesmo ponto físico [30].

O problema de rastrear características através de uma sequência de imagens é uma tarefa essencial em visão computacional, sendo base para diversos problemas de mais alto nível, tais como reconstrução 3D [3, 34], reconhecimento de objetos [6, 44] e análise de movimento. Ele é uma instância do problema mais geral de calcular o fluxo ótico, que representa a estimativa do movimento de diferentes posições de um quadro da sequência, descrevendo como a imagem muda com o tempo [30, 38]. Como exemplo, considere que devemos estabelecer correspondências entre as duas imagens da Figura 2.1.

A estratégia mais simples seria comparar o valor das intensidades dos pontos nas duas imagens, ou seja, se o valor do ponto $\mathbf{x}_1$ na primeira imagem é $I_1(\mathbf{x}_1)$, a correspondência na segunda imagem seria o ponto $\mathbf{x}_2$ tal que $I_1(\mathbf{x}_1) = I_2(\mathbf{x}_2)$.

No entanto, seria difícil ou até mesmo impossível determinar a localização de um ponto característico nos demais quadros da sequência baseando-se apenas em seu valor de intensidade, considerando que um mesmo valor pode se repetir diversas vezes em uma imagem, ou até mesmo mudar de uma imagem para outra, devido à ruídos e mudança nas condições de iluminação, por exemplo. Associamos, então, a cada *pixel* de interesse $\mathbf{x}$ um vetor contendo informações de intensidade de cada *pixel* em sua volta, $l(\mathbf{x}) = \{I(\tilde{\mathbf{x}}) \mid \tilde{\mathbf{x}} \in$



(a) (b)

Figura 2.1: Ambiguidade no estabelecimento de correspondências.

$W(\mathbf{x})\}$, onde $W(\mathbf{x})$ é uma janela de suporte em torno de $\mathbf{x}$ [30].

Como devido a mudanças no ponto de vista os *pixels* sofrem transformações, tais como translação e adição de ruído, procuramos pela transformação sofrida por uma janela que minimiza alguma medida de discrepância [30]. Note que o problema de rastreamento pode ser interpretado como um problema de minimização.

Se considerarmos características cuja janela de suporte possui *pixels* com valor de intensidade (aproximadamente) constante, várias regiões irão satisfazer o critério de minimização sendo utilizado. Para a característica destacada com um círculo na Figura 2.1(a), por exemplo, são várias as regiões que podem ser consideradas correspondentes na Figura 2.1(b), algumas das quais as representadas por quadrados. Isso se deve ao conhecido problema da abertura, ilustrado na Figura 2.2, onde somente o movimento ortogonal à aresta pode ser determinado [30, 42]

Desta forma, é preciso restringir a nossa atenção à pontos que contenham uma janela de suporte suficientemente rica em textura, tornando o rastreamento mais robusto à variações de aparência causadas por mudanças de iluminação e pontos de vista. Chamamos tais pontos de **características** ou **pontos característicos** [29, 37].

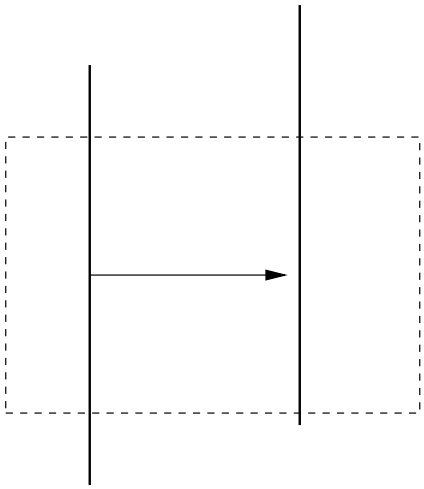


Figura 2.2: O problema da abertura: somente o movimento ortogonal à aresta pode ser determinado

2.1 Trabalhos Relacionados

De forma geral, os algoritmos de rastreamento de características possuem quatro passos principais: predição, detecção, casamento (*matching*) e atualização (Figura 2.3) [5], diferindo basicamente no método de seleção de características, no critério de similaridade e na forma como a predição é feita.

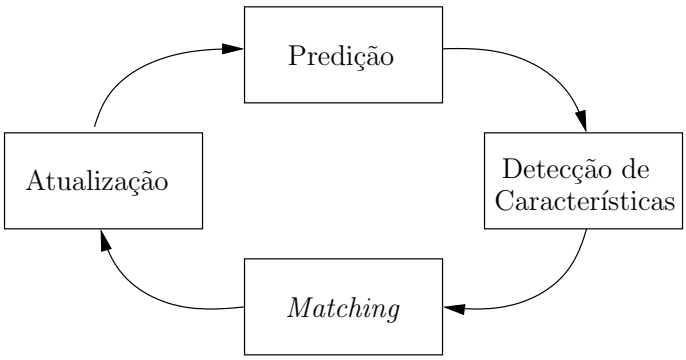


Figura 2.3: Estrutura geral dos algoritmos de rastreamento de correspondências.

Na etapa de predição, nos baseamos em modelos de movimento e na posição em quadros anteriores para estimar a localização de cada característica através dos quadros.

Tais modelos variam de simples modelos com velocidade constante a movimentos com distribuições de probabilidade determinadas.

A etapa de casamento visa estabelecer uma correspondência a cada característica do quadro atual, utilizando para tal algum critério de similaridade, tal como a mudança na aparência da característica através dos quadros ou a minimização da distância entre a característica original e as candidatas.

As características candidatas são escolhidas na etapa de detecção utilizando um algoritmo apropriado. Algumas estratégias utilizadas são selecionar janelas com desvios padrões altos [32], *zero crossings* do Laplaciano da intensidade da imagem [31] e usar informações de primeira e segunda derivada [25]. A etapa de atualização, por sua vez, consiste apenas na reiteração do algoritmo. Ressaltamos que estas quatro etapas muitas vezes estão relacionadas entre si. Podemos utilizar o mesmo modelo nas etapas de predição e casamento, por exemplo [5].

Neste trabalho, utilizamos um critério de seleção de características proposto por Tomasi, Lucas e Kanade [39, 29], onde as características são selecionadas visando o bom funcionamento do algoritmo de rastreamento proposto, o KLT (*Kanade-Lucas-Tomasi*) *Feature Tracker*, largamente utilizado na literatura e abordado com mais detalhes na Seção 2.2. Jin *et al.* [21] apresentaram uma extensão do algoritmo KLT que leva em conta mudanças na iluminação e reflexão.

O detector de Harris [18] define um limiar para escolha de características, dado por

$$C(G) = det(G) + k \times \text{trace}^2(G) \tag{2.1}$$

com

$$G = \begin{bmatrix} \nabla^2 x & \nabla x \nabla y \\ \nabla x \nabla y & \nabla^2 y \end{bmatrix}, \tag{2.2}$$

em que ∇x e ∇y representam o gradiente nas direções x e y, respectivamente. A escolha do escalar $k \in \mathbb{R}$ influi na exigência de uma variação do gradiente maior em uma ou outra direção, ou em ambas.

As abordagens de correspondências robustas, por sua vez, detectam automaticamente características que devem ser descartadas por consistirem de correspondências erradas.

Exemplos incluem Torr et al. [41] que adota uma abordagem RANSAC [12] para eliminar tais pontos e Fusiello et. al. [15], que propôs uma extensão ao KLT, introduzindo um esquema automático de descarte de características baseado em uma regra de rejeição chamada X84. A identificação destas características, denominadas anomalias (*outliers*), é de extrema importância, considerando que podem comprometer severamente o resultado de tarefas de mais alto nível que dependem do correto estabelecimento de correspondências.

A *Scale-Invariant Feature Transform* (SIFT) [28] extrai características distintivas de uma imagem, as quais são invariantes à escala e rotação, e parcialmente invariantes à mudanças de iluminação. Elas fornecem um casamento confiável mesmo quando há uma grande distorção afim e mudanças no ponto de vista 3D. A técnica usa uma abordagem em cascata, onde operações com um custo computacional alto são aplicadas somente em características que satisfazem a requisitos iniciais.

Bretzner [5] propôs um método robusto à variações de tamanho para rastreamento multi-escala incorporando *multi-cue matching*. Além disso, o autor também propôs uma representação multi-escala de objetos baseada em características.

Como podemos verificar, existem muitos algoritmos de rastreamento de características, cada um possuindo diferentes suposições e objetivos. No entanto, nenhum destes algoritmos considera a incerteza dos dados sendo rastreados, ou seja, a informação sobre a confiabilidade das estimativas. Enquanto há na literatura trabalhos que estimam a covariância do erro no quadro final [10, 8, 9], ou então utilizando uma métrica baseada unicamente em características locais da imagem [33], aqui nós introduzimos um meio de propagá-lo, utilizando uma técnica de filtragem não-linear em conjunto com o algoritmo KLT, detalhado a seguir.

Ressaltamos que outros algoritmos de rastreamento de características poderiam ser utilizados. Aqui, optamos por utilizar o algoritmo KLT padrão porque além de demonstrar os benefícios do nosso método, seu uso também possibilita a comparação com outros trabalhos que também utilizam o KLT como base.

2.2 Kanade-Lucas-Tomasi Feature Tracker (KLT)

O algoritmo de rastreamento de correspondências KLT foi proposto por Lucas e Kanade [29], e desenvolvido por Tomasi e Kanade [39]. O critério de similaridade utilizado é baseado na minimização da soma das diferenças quadradas (SSD - *Sum of Squared Differences*) das intensidades sob as janelas, assumindo um modelo de movimento puramente translacional.

Posteriormente, Shi e Tomasi [37] extenderam o algoritmo considerando também deslocamentos mais complexos, através da utilização do modelo de movimento afim. A qualidade das características rastreadas é monitorada, visando identificar anomalias (o erro acumulado do rastreamento se tornou muito grande). Enquanto o modelo translacional é mais adequado para rastreamento entre quadros adjacentes, o modelo afim deve ser utilizado para rastreamento entre quadros distantes. A seguir veremos com mais detalhes estes dois métodos.

2.2.1 Os Modelos de Movimento Translacional e Afim

Seja $h(\mathbf{x})$ a função que descreve o movimento da imagem (transformação do domínio). De uma maneira intuitiva, podemos escrever h como [30]:

$$h(\mathbf{x}) = \mathbf{x} + \Delta\mathbf{x}(\mathbf{X}), \tag{2.3}$$

em que $\mathbf{X} \in \mathbb{R}^3$ é o ponto de interesse e $\Delta\mathbf{x}(\mathbf{X})$ é o deslocamento de $\mathbf{x} \in \mathbb{R}^2$ (a imagem do ponto $\mathbf{X}$) de uma vista para outra.

O modelo de movimento mais simples é o translacional, $\Delta\mathbf{x}(\mathbf{X})$ constante, onde cada ponto na janela em questão (denotada por W) sofre a mesma transformação [30, 37]. Em resumo, assume-se que as intensidades dos *pixels* na janela transladada são aquelas na imagem original mais um resíduo que depende quase linearmente do vetor de translação. Assim, a função h passa a não depender mais de $\mathbf{X}$ [30]:

$$h(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}} + \Delta\mathbf{x} \qquad \forall\, \tilde{\mathbf{x}} \in W(\mathbf{x}), \tag{2.4}$$

em que $\Delta\mathbf{x} \in \mathbb{R}^2$.

Tal modelo é somente uma aproximação, válido apenas localmente no espaço e no tempo, quando utilizado com janelas de pequenas dimensões e onde há pouco movimento da câmera. Além disso, o modelo é válido apenas para regiões da imagem que são planas e paralelas ao plano da imagem, e que se movem paralelamente a ele [30, 37]. Entretanto, vale ressaltar que é um modelo bastante utilizado devido à simplicidade e eficiência da implementação.

Considerando que os *pixels* pertencentes a uma janela podem sofrer deslocamentos diferentes, é preciso trabalhar com modelos mais complexos. Isso ocorre principalmente quando não há um movimento contínuo da câmera, e temos apenas imagens de pontos de vista bastante distintos. Nestes casos, é adequado usar o modelo afim, onde o movimento dos pontos irá depender linearmente de sua posição e de um deslocamento constante. A função h é definida como [30]:

$$h(\tilde{\mathbf{x}}) = A\tilde{\mathbf{x}} + \mathbf{d} \qquad \forall\, \tilde{\mathbf{x}} \in W(\mathbf{x}), \tag{2.5}$$

em que $A \in \mathbb{R}^{2\times 2}$ e $\mathbf{d} \in \mathbb{R}^2$.

Este modelo é uma boa aproximação para regiões planares paralelas ao plano da imagem que sofrem translação e rotação arbitrárias em torno do eixo ótico e rotação (modesta) em torno de um eixo ortogonal a este [30, 37].

O modelo afim também é adequado quando as características são rastreadas por um grande número de quadros, onde o erro estimado se acumula no decorrer do tempo, levando à eventual perda de algumas das características. Neste caso, pode ser necessário realizar o casamento não entre quadros adjacentes, mas sim entre o primeiro e o atual.

2.2.2 Determinando o Movimento da Imagem

Seja $I(x, y, t)$ a função que representa uma seqüência de imagens, em que x e y são as variáveis espaciais e t o tempo. Tal função satisfaz a seguinte propriedade [37, 39]:

$$I(x, y, t + \tau) = I(x - \xi, y - \eta, t), \tag{2.6}$$

que nos diz que a imagem no tempo $t + \tau$ pode ser obtida deslocando-se a imagem no tempo t por uma determinada quantidade, denominada deslocamento no ponto $\mathbf{x} = (x, y)$,

e denotada aqui por $\delta = (\xi, \eta)$. Mesmo em ambientes controlados, a propriedade descrita pela Equação 2.6 é violada em muitas situações. Próximo a bordas, por exemplo, os pontos podem desaparecer e voltar a aparecer. Além disso, dependendo do ponto de vista, as condições de iluminação podem variar, influenciando os valores das intensidades dos *pixels*.

Para o modelo afim, o deslocamento é dado por

$$\delta = D\mathbf{x} + \mathbf{d}, \tag{2.7}$$

em que

$$D = \begin{pmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{pmatrix} \tag{2.8}$$

é denominada matriz de deformação e $\mathbf{d}$ é a translação do centro da janela [37, 39]. No caso do modelo translacional, que estamos considerando neste trabalho, D é zero. Temos, então,

$$\delta = \mathbf{d}. \tag{2.9}$$

Formalmente, se definirmos $J(x) = I(x, y, t + \tau)$ e $I(\mathbf{x} - \delta) = I(x - \xi, y - \eta, t)$ temos que nosso modelo é [39]:

$$J(\mathbf{x}) = I(\mathbf{x} - \delta) + n(\mathbf{x}) \tag{2.10}$$

em que $n(\mathbf{x})$ é um ruído. Como não trabalharemos com algo exato, escolhemos o deslocamento δ que minimize o erro residual definido pela seguinte integral dupla sobre a janela W.

$$\int\int_W [I(\mathbf{x} - \delta) - J(\mathbf{x})]^2 w(\mathbf{x}) d\mathbf{x}, \tag{2.11}$$

em que W é a janela em questão e $w(\mathbf{x})$ é uma função peso (usualmente uma Gaussiana ou 1 no caso mais simples). A minimização de 2.11 resultará em (ver [37]):

$$T\mathbf{z} = \mathbf{a}, \tag{2.12}$$

em que [37, 39]:

$$\mathbf{z}^T = [d_{xx}, d_{yx}, d_{xy}, d_{yy}, d_x, d_y] \tag{2.13}$$

possui informações da matriz de deformação D e do deslocamento $\mathbf{d}$,

$$\mathbf{a} = \int\int_W [I(\mathbf{x}) - J(\mathbf{x})] \begin{bmatrix} xg_x \\ xg_y \\ yg_x \\ yg_y \\ g_x \\ g_y \end{bmatrix} w \, d\mathbf{x}, \tag{2.14}$$

$$T = \int\int_W \begin{bmatrix} U & V \\ V^T & Z \end{bmatrix} w \, d\mathbf{x}, \tag{2.15}$$

com

$$U = \begin{bmatrix} x^2g_x^2 & x^2g_xg_y & xyg_x^2 & xyg_xg_y \\ x^2g_xg_y & x^2g_y^2 & xyg_xg_y & xyg_y^2 \\ xyg_x^2 & xyg_xg_y & y^2g_x^2 & y^2g_xg_y \\ xyg_xg_y & xyg_y^2 & y^2g_xg_y & y^2g_y^2 \end{bmatrix}, \tag{2.16}$$

$$V^T = \begin{bmatrix} xg_x^2 & xg_xg_y & yg_x^2 & yg_xg_y \\ xg_xg_y & xg_y^2 & yg_xg_y & yg_y^2 \end{bmatrix}, \tag{2.17}$$

$$Z = \begin{bmatrix} g_x^2 & g_xg_y \\ g_xg_y & g_y^2 \end{bmatrix}. \tag{2.18}$$

Conforme já exposto, quanto trabalhamos com o modelo translacional, a matriz de deformação D é zero. Neste caso, para estimar o deslocamento $\mathbf{d}$ basta resolver o sistema

$$Z\mathbf{d} = \mathbf{e}, \tag{2.19}$$

em que $\mathbf{e} = [g_x \; g_y]$ (duas últimas entradas da matriz $\mathbf{a}$).

Em [39], é apresentada uma abordagem multi-escala. O método aqui apresentado é aplicado a cada nível de uma pirâmide construída a partir da imagem original, através da sua suavização e amostragem, produzindo novas imagens em diferentes escalas. O deslocamento total é dado pela soma dos deslocamentos estimados em cada nível. Esta abordagem é adequada quando o deslocamento entre correspondências excede 3 *pixels*.Também

podemos realizar este procedimento novamente na imagem original (correspondente à escala mais fina). Tipicamente, 5 ou 6 iterações deste tipo são suficientes para produzir um erro de localização de um décimo de *pixel* em uma janela 7×7 [30].

Quando o deslocamento entre as imagens é muito grande, a técnica geralmente usada consiste em primeiro estabelecer correspondências entre um pequeno número de características e então tentar extender o *matching* à outras características utilizando técnicas estatísticas robustas, tais como o RANSAC [12].

2.2.3 Seleção de Características

Como vimos, é preciso selecionar pontos que contenham uma região de suporte com textura suficiente para evitar o problema da abertura. O critério utilizado pelo algoritmo KLT visa maximizar o seu desempenho, ou seja, ele seleciona características que possam ser rastreadas com confiança [37, 39].

Na seção anterior, vimos que é possível rastrear uma janela de um quadro para outro se o sistema 2.19 (para o caso do modelo translacional) pode ser resolvido de maneira confiável. Isso implica que a matriz Z deve ser bem condicionada e ter seus dois autovalores maiores que um limiar λ.

Dois autovalores pequenos na matriz Z indicam que a janela possui intensidades aproximadamente constantes. Ter um autovalor grande e outro pequeno sinaliza que a intensidade varia em apenas uma direção. Por outro lado, dois autovalores grandes indicam cantos, texturas ricas e outros padrões que podem ser rastreados de maneira confiável [37, 39].

Desta forma, se os dois autovalores de Z são λ_1 and λ_2, uma janela é "aceita" se

$$\min(\lambda_1, \lambda_2) > \lambda, \tag{2.20}$$

em que λ é o limiar escolhido como um meio termo entre um limitante inferior (obtido a partir dos autovalores de regiões com brilho aproximadamente uniforme) e um limite superior (obtido a partir de um conjunto de características tais como cantos).

Embora para o caso do sistema de movimento afim (Equação 2.12) as considerações sejam similares, é preciso destacar uma diferença essencial: a meta não é determinar a deformação em si, não importando se algum se seus componentes não pode ser estimado de

forma confiável. Isso porque as deformações são usadas para determinar se o casamento entre o primeiro quadro e o corrente está adequado, não afetando significativamente a janela. Na prática, a Equação 2.12 pode ser resolvida através do cálculo da pseudo-inversa de T [37].

Entretanto, mesmo selecionando características confiáveis, o algoritmo pode cometer erros, pois o processo está exposto a ruídos e erros (inclusive de máquina).

2.3 Erros no Rastreamento de Características

Os erros cometidos pelos algoritmos de rastreamento são classificados em duas categorias: erros na localização ("*bad locations*") e correspondências falsas ("*false matches*") [47].

Na primeira categoria, assume-se que o erro de localização de um ponto característico assume um comportamento Gaussiano, ou seja, enquanto grande parte dos pontos possui um erro pequeno (dentro de dois *pixels*), apenas alguns estão mais distantes da posição real. Nesta categoria, é possível modelar o erro obtido.

Na segunda categoria temos uma grande quantidade de anomalias, características mapeadas para uma localização completamente diferente da correta. As anomalias comprometem seriamente o resultado das demais tarefas que dependem do estabelecimento de correspondências. Algumas vezes, mesmo quando o conjunto de dados contém somente uma anomalia, toda a estimativa pode ser completamente perturbada [19].

Devido a tais problemas, e considerando que a ocorrência de anomalias é frequente, muitas técnicas robustas tem sido propostas. Dois dos métodos mais populares são os M-estimators [40] e a Mediana Mínima dos Quadrados (LMedS - *least-median-of-squares*) [48].

O método proposto neste trabalho visa aumentar a precisão com que as correspondências são estimadas, fornecendo também uma região de confiança associada à cada característica. Além disso, conforme veremos com mais detalhes no Capítulo 4, o método detecta e descarta anomalias. Como exemplo de aplicação, utilizamos as correspondências estimadas por nosso método em um procedimento de reconstrução 3D.

Capítulo 3

Rastreamento e filtros preditivos

Quando temos um problema, é preciso modelá-lo de uma maneira apropriada, para tornar possível o seu estudo e também para encontrar e formalizar a solução correta.

Se nosso objetivo fosse somente o rastreamento de características, usaríamos a formalização apresentada no capítulo anterior. No entanto, temos dois objetivos principais: associar uma noção de incerteza à cada correspondência e melhorar as estimativas realizadas. Assim, é preciso modelar o problema das correspondências de tal forma que estes dois itens sejam considerados. Neste capítulo apresentaremos dois conceitos que serão úteis: filtros preditivos e transformada *Unscented*.

3.1 Conceitos Básicos

O conjunto de parâmetros que descreve a configuração de um objeto é denominado vetor de estados o qual, juntamente com o modelo de sua dinâmica, forma o que chamamos de sistema. Temos também as observações, medidas indiretas do sistema usadas para inferir informações sobre parâmetros que por algum motivo não podem ser medidos diretamente. Em cada amostra de tempo k fazemos uma medida/observação do sistema, que será representada por $\vec{y}_k$. $D_k = \{\vec{y}_1 \dots \vec{y}_k\}$ representa toda a informação disponível até o instante de tempo k [16].

Além disso, como há um erro intrínseco tanto no modelo quanto nas medidas realizadas, é preciso levar em conta o conceito de incerteza.

De forma geral, em problemas de rastreamento temos um modelo para o movimento do objeto em questão (no Capítulo 2, vimos os modelos translacional e afim) e um conjunto de medidas tomadas da seqüência de imagens, tais como a posição e os momentos de uma região. Se assumirmos que o modelo de movimento representa os parâmetros iniciais e que as medidas serão as observações, podemos utilizar conceitos de filtros preditivos, uma família de técnicas de estimação de parâmetros que visam estimar o estado ótimo de um estado.

Inicialmente, os valores dos estados e suas incertezas são propagados através da dinâmica do sistema. As informações obtidas com as observações são combinadas com esta estimativa preliminar. Os filtros preditivos possuem três passos principais [16, 13]:

- predição: busca descobrir a predição do estado k a partir das observações D_{k-1};
- associação de dados (*weighting*): consiste basicamente em obter uma representação $p(\vec{x}_k|D_{k-1})$, a qual busca resolver a questão anterior, baseando-se em medidas obtidas no instante de tempo k;
- correção: considerando agora também a observação $\vec{y}_k$, queremos uma representação de $p(\vec{x}_k|D_k)$.

Seria como se a regra de Bayes, dada por

$$P(B|A) = \frac{P(A|B)\ P(B)}{P(A)} \tag{3.1}$$

fosse interpretada no seguinte sentido:

$$P(\text{parâmetros}|\text{medidas}) = \frac{P(\text{medidas}|\text{parâmetros})\ P(\text{parâmetros})}{P(\text{medidas})}. \tag{3.2}$$

Os termos $P(\text{parâmetros})$ e $P(\text{parâmetros}|\text{medidas})$ são denominados *prior* (descreve o que sabemos do sistema antes de as medidas serem efetuadas) e *posterior* (descreve a probabilidade de diversos modelos após medidas serem feitas), respectivamente. Uma das propriedades da regra de Bayes é que ela nos indica a escolha de parâmetros mais provável, dados o modelo e o conhecimento de informações anteriores sobre o sistema [13].

Quando estamos trabalhando com rastreamento, fazemos duas suposições, com o intuito de facilitar a solução do problema. A primeira delas é que somente o último estado

nos interessa, ou seja, $p(\vec{x}_k|\vec{x}_1, \ldots, \vec{x}_{k-1}) = p(\vec{x}_k|\vec{x}_{k-1})$. Isso irá simplificar muito o projeto dos algoritmos (é preciso guardar informação do último estado apenas). Na segunda, assume-se que as observações dependem apenas do estado corrente, o que significa dizer que $\vec{y}_k$ é condicionalmente independente de todas as outras observações dado $\vec{x}_k$. Formalmente: $p(\vec{y}_k, \vec{y}_j, \ldots, \vec{y}_i|\vec{x}_k) = p(\vec{y}_k|\vec{x}_k)\ p(\vec{y}_j, \ldots, \vec{y}_i|\vec{x}_k)$. Vamos então formalizar a interpretação do problema de rastreamento no contexto de filtros preditivos.

Inicialmente, temos apenas $p(\vec{x}_0)$, que é a predição do estado do nosso sistema na ausência de observações. No rastreamento de características, consiste das coordenadas de cada uma das características selecionadas. A correção, feita a partir de $D_0 = \vec{y}_0$ é [13]:

$$p(\vec{x}_0|D_0) = \frac{p(D_0|\vec{x}_0)\ p(\vec{x}_0)}{p(D_0)} \tag{3.3}$$

que é resultado da aplicação da Regra de Bayes (Eq. 3.1). Vamos assumir agora que temos uma representação de $p(\vec{x}_{k-1}|D_{k-1})$. A predição busca representar $p(\vec{x}_k|D_{k-1})$. A partir da suposição de independência que fizemos, podemos escrever [13]:

$$\begin{aligned} p(\vec{x}_k|D_{k-1}) &= \int p(\vec{x}_k, \vec{x}_{k-1}|D_{k-1})\ d\vec{x}_{k-1}, \\ &= \int p(\vec{x}_k|\vec{x}_{k-1}, D_{k-1})\ p(\vec{x}_{k-1}|D_{k-1})\ d\vec{x}_{k-1}, \\ &= \int p(\vec{x}_k|\vec{x}_{k-1})\ p(\vec{x}_{k-1}|D_{k-1})\ d\vec{x}_{k-1}. \end{aligned} \tag{3.4}$$

Ainda a partir da suposições de independência, podemos fazer a correção, a qual consiste em obter uma representação de $p(\vec{x}_k|D_k)$ [13]:

$$\begin{aligned} p(\vec{x}_k|D_k) &= p(\vec{x}_k|\vec{y}_k), \\ &= \frac{p(\vec{x_k}|\vec{y_k})\ p(D_{k-1}|\vec{y_k})\ p(\vec{y_k})}{p(D_k)}, \\ &= \frac{p(\vec{x_k}, D_{k-1}|\vec{y_k})\ p(\vec{y_k})}{p(D_k)}, \\ &= \frac{p(\vec{y}_k|\vec{x}_k, D_{k-1})\ p(\vec{x}_k|D_{k-1})\ p(D_{k-1})}{p(D_k)}, \\ &= p(\vec{y}_k|\vec{x}_k)\ p(\vec{x}_k|D_{k-1})\frac{p(D_{k-1})}{p(D_k)}, \\ &= \frac{p(\vec{y}_k|\vec{x}_k)\ p(\vec{x}_k|D_{k-1})}{\int p(\vec{y}_k|\vec{x}_k)\ p(\vec{x}_k|D_{k-1})\ d\vec{x}_k}. \end{aligned} \tag{3.5}$$

No caso do rastreamento, a correção consistiria em buscar uma estimativa melhor da localização das características do que a obtida na etapa de predição, utilizando a informação adicional fornecida pelas observações realizadas. Os filtros preditivos diferem na forma de solução das equações 3.4 e 3.5.

3.2 Trabalhos Relacionados - Filtros Preditivos

Quando a dinâmica do sistema e os modelos de observação são lineares, os modelos de probabilidade condicional são distribuições normais, e o filtro de Kalman (KF) pode ser utilizado. O KF, largamente usado devido a sua simplicidade, tratabilidade e robustez, é um filtro recursivo que estima o estado de um sistema a partir de medidas incertas e incompletas [46, 16].

Entretanto, em muitas aplicações de interesse, as condições de linearidade não são satisfeitas, e extensões do KF devem ser utilizadas. O Filtro de Kalman Extendido (EKF) explora a suposição de que todas as transformações são quase-lineares e lineariza todas as transformações não-lineares, de tal forma que as equações tradicionais do KF possam ser utilizadas. Entretanto, o EKF tem algumas desvantagens, levando à representações pobres das funções não-lineares e distribuições de probabilidade de interesse, resultando em estimativas incorretas [46, 43, 24].

O Filtro de Kalman *Unscented* (UKF) [22, 23, 24] busca superar as limitações do EKF através da utilização do modelo não-linear real, utilizando para tal uma abordagem de amostragem determinística.

A distribuição do estado é aproximada por uma GRV, representada por um conjunto de pontos amostrais escolhidos deterministicamente de forma a capturar a sua média e covariância reais. Ao propagar esses pontos pelo sistema não-linear, a média e covariância *a posterior* são capturados até a terceira ordem para qualquer não-linearidade [43, 24, 45].

O UKF é baseado na transformada *Unscented* (UT), que fornece um mecanismo para transformar as informações de média e covariância, suprindo as deficiências impostas pela linearização realizada pelo EKF [23, 22, 43].

3.3 A Transformada Unscented com Escala

A UT é um método para calcular as estatísticas de uma variável aleatória que sofre uma transformação não-linear. A idéia chave da Transformada *Unscented* é que é mais fácil aproximar uma distribuição Gaussiana do que uma transformação/função não-linear arbitrária. Para garantir que matriz de covariância predita seja positiva definida, usamos a Transformada *Unscented* com Escala (SUT), uma extensão da UT que garante esta condição [43].

A SUT é ilustrada na Figura 3.1. Um conjunto de pontos (denominados pontos sigma) é escolhido deterministicamente de tal forma que os pontos configurem uma média e uma covariância específicas. A função não-linear é aplicada a cada ponto, e então as estatísticas dos pontos transformados são calculadas, obtendo assim a média e a covariância transformadas.

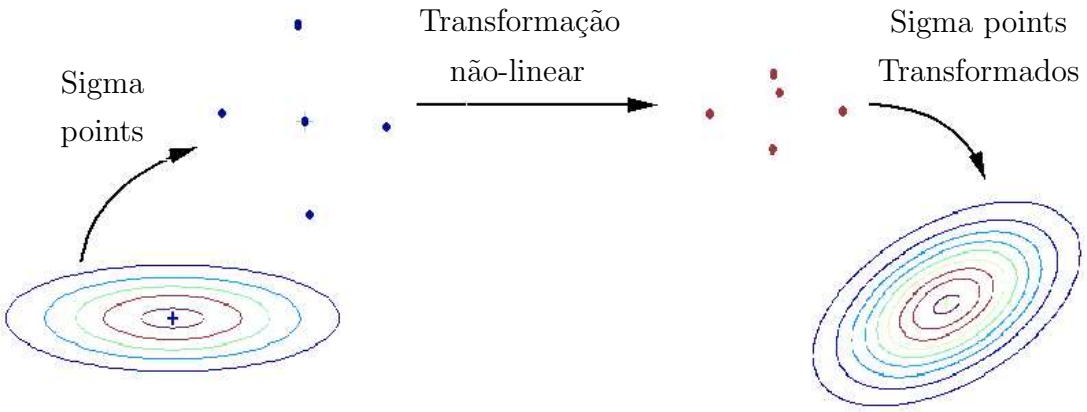


Figura 3.1: A transformada *Unscented*

Embora este método lembre os filtros de partículas, existem várias diferenças fundamentais [1, 36]. Primeiro, os pontos sigma não são escolhidos de forma randômica, mas sim deterministicamente, exibindo certas propriedades específicas - como ter uma dada média e covariância. Com isso, informações de mais alta ordem sobre a distribuição podem ser capturadas com um número pequeno e fixo de pontos. A segunda diferença está na atribuição de pesos aos pontos sigma (aqui os pesos podem ser negativos).

Considere uma variável aleatória $\vec{x}$ (dimensão n) que sofre uma transformação não-linear $\vec{y} = g(\vec{x})$. Sejam $\bar{\vec{x}}$ e Σ_x a média e a matriz de covariância de $\vec{x}$, respectivamente. Para calcular a média e a covariância de $\vec{y}$, nós geramos um conjunto $\mathcal{X}$ de $2n+1$ pontos sigma $\mathcal{X}_i$ como segue [43]:

$$\begin{aligned} \mathcal{X}_0 &= \bar{\vec{x}}, \\ \mathcal{X}_i &= \bar{\vec{x}} + (\sqrt{(n+\lambda)\Sigma_x})_i, \qquad i = 1, \ldots, n, \\ \mathcal{X}_i &= \bar{\vec{x}} - (\sqrt{(n+\lambda)\Sigma_x})_{i-n}, \qquad i = n+1, \ldots, 2n. \end{aligned} \tag{3.6}$$

em que [43]:

- $\lambda = \alpha^2(n+\kappa) - n$ é um parâmetro de escala;
- α determina o espalhamento dos pontos sigma em torno da média. Usualmente $10^{-1} \leq \alpha \leq 1$;
- κ é um parâmetro secundário de escala. Geralmente, $\kappa = 3 - n$;
- β busca incorporar conhecimento sobre os momentos de mais alta ordem da distribuição. Para distribuições Gaussianas, 2 é o valor ótimo para β;
- $(\sqrt{(n+\lambda)\Sigma_x})_i$ é a i-ésima coluna da raiz quadrada da matriz $(n+\lambda)\Sigma_x$.

  Para determinar a raiz quadrada desta matriz, foi utilizada a decomposição de Cholesky: seja $A \in \mathbb{R}^{n \times n}$ uma matriz simétrica positiva semi-definida. Seja também $A = GG^T$ a sua decomposição de Cholesky. Se $G = U\Sigma V^T$ é a decomposição SVD de G, e $X = U\Sigma U^T$, então X é simétrica positiva definida e

$$A = GG^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma^2 U^T = (U\Sigma U^T)(U\Sigma U^T) = X^2 \tag{3.7}$$

  temos, portanto, que X é a raiz quadrada de A [17].

Cada ponto sigma tem um peso associado W_i, sujeito a $\sum_{i=0}^{2n} W_i = 1$ [43]:

$$\begin{aligned} W_0^m &= \frac{\lambda}{n+\lambda}, \\ W_0^c &= \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta, \\ W_i^m &= W_i^c = \frac{1}{2(n+\lambda)} \qquad i = 1, \ldots, 2n. \end{aligned} \tag{3.8}$$

com W_i^m sendo o peso para cálculo da média e W_i^c da covariância. Os pontos sigma são propagados através da transformação não-linear

$$\mathcal{Y}_i = f(\mathcal{X}_i). \tag{3.9}$$

A média e a matriz de covariância são aproximadas como

$$\bar{\vec{y}} = \sum_{i=0}^{2n} W_i^m \mathcal{Y}_i, \tag{3.10}$$

e

$$\Sigma_y = \sum_{i=0}^{2n} W_i^m \ (\mathcal{Y}_i - \bar{\vec{y}})(\mathcal{Y}_i - \bar{\vec{y}})^T. \tag{3.11}$$

Dentre as propriedades da SUT temos que [24, 23]:

- Os pontos sigma capturam a mesma média e a mesma covariância independentemente do método usado para calcular a raiz quadrada da matriz.
- A SUT pode ser utilizada em qualquer tipo de modelo de processo, sendo que a média e a covariância são calculadas utilizando operações padrões sob vetores e matrizes.
- Para entradas Gaussianas, as aproximações resultantes da transformada *Unscented* são corretas até a terceira ordem para todas as não-linearidades. Para entradas não-gaussianas a corretude é garantida até a segunda ordem, e a de ordens mais altas depende da escolha dos parâmetros α e β (prova em [46]). Tal desempenho é superior ao do EKF, que calcula a média e covariância de forma precisa apenas até a primeira ordem.
- O custo computacional do algoritmo é da mesma ordem de magnitude do EKF. As operações de mais alto custo são calcular a matriz raiz quadrada e os produtos externos exigidos chegar à covariância dos pontos sigma projetados. Entretanto, ambas as operações são $O(N_x^3)$, que é o mesmo custo das multiplicações de matrizes $N_x \times N_x$ necessárias para calcular a covariância predita no EKF.

Como exposto em [24], a SUT pode ser usada para propagar qualquer informação de alta ordem sobre os momentos, anexando-se informações adicionais aos pontos sigma.

Capítulo 4

Nossa Proposta: Unscented Feature Tracking

Quando estimamos o estado de um sistema, raramente obtemos um resultado exato, considerando que a precisão dos instrumentos de medida e do processo é limitada. Sendo assim, é extremamente importante que consigamos representar a incerteza associada à estimativa. Uma forma de representação é através de uma distribuição de probabilidade.

Como uma parametrização completa da distribuição de probabilidade pode não ser viável computacionalmente, uma aproximação do estado pode ser gerada, mantendo-se um número menor de momentos da distribuição, de forma a limitar a demanda computacional do algoritmo.

Neste trabalho, onde representamos a localização de cada característica como uma GRV, precisamos manter apenas os dois primeiros momentos: média e covariância. Embora a utilização dos dois primeiros momentos seja uma representação relativamente simples do estado do sistema, possui diversos benefícios:

- Ao trabalharmos apenas com os dois primeiros momentos, apenas uma quantidade pequena e constante de informação precisa ser mantida. Como a informação é suficiente para os nossos objetivos, é um *trade off* entre flexibilidade de representação e complexidade computacional.

- Os dois primeiros momentos são linearmente transformáveis, isto é, suas estimativas

são preservadas quando submetidas à transformação lineares.

- Conjuntos de estimativas de média e covariância podem ser utilizados para representar características adicionais da distribuição. Métodos de rastreamento multimodal baseados em múltiplas estimativas incluem filtros Rao-Blackwellized e *sum-of-Gaussian* [24].

Utilizando esta forma de representação, modelamos o problema de rastreamento de características como uma transformação de variáveis aleatórias. Como as transformações ao longo da sequência de imagens podem ser não-lineares, é preciso lidar com filtragem bayesiana não-linear. Então, usamos a SUT para estimar a variável aleatória da localização da posição ao longo de uma transformação não-linear, representada neste trabalho pelo algoritmo KLT. Ressaltamos novamente que qualquer outro algoritmo de rastreamento de características poderia ser utilizado. Nós chamamos a nossa abordagem de *Unscented Feature Tracking* (UFT).

4.1 Descrição do Método

Seja $u(\mu_i, \Sigma_i)_k$, com $i = 1, \ldots, n$, em que n é o número de pontos, o vetor de estados de nosso sistema, onde para cada tempo k nós temos as GRVs que representam as localizações de cada característica.

Para lidar efetivamente com não-linearidades, utilizaremos uma técnica de filtragem baseada na SUT para propagar o estado do sistema u, conseguindo assim uma precisão de até pelo menos a segunda ordem, o que é suficiente para trabalhar com GRVs, descritas pelos dois primeiros momentos.

No capítulo anterior, formalizamos nosso método de tal maneira que conceitos de filtros preditivos podem ser utilizados. A idéia principal é buscar uma estimativa mais precisa da localização das características do que a obtida inicialmente (etapa de predição), utilizando a informação adicional fornecida por observações realizadas no sistema.

Na etapa de predição, a estimativa é baseada na SUT (Algoritmo 4), que é aplicada à cada ponto característico pertencente ao vetor de estados u: encontramos os $2n + 1$

pontos sigma, onde n é a dimensão do estado do sistema (Equação 3.6), os propagamos usando o algoritmo de rastreamento KLT, e finalmente calculamos a média e covariância correspondentes (Equações 3.10 e 3.11).

Algoritmo 1 SUT - Transformada *Unscented* com Escala

1: **função** TRANSFORMADA *Unscented* COM ESCALA
2: dados n características selecionadas pelo algoritmo KLT
3: **para** cada característica **faça**
4: gerar $2L + 1$ pontos sigma, em que L é a dimensão da GRV;
5: propague os pontos sigma usando o KLT;
6: calcule a média (Equação 3.10);
7: calcule a matriz de covariância (Equação 3.11);
8: **fim para**
9: **fim função**

Em cada tempo k, fazemos também uma observação do sistema, denotada por $v(\mu_i, \Sigma_i)_k$. A princípio, a covariância dos pontos característicos depende do algoritmo de rastreamento e das características locais da imagem [10]. Como a predição baseou-se no primeiro item, nossa observação é baseada na informação local da imagem, fornecendo informação extra para combinar com a GRV propagada. Cada característica irá ter uma covariância associada, dada pela inversa da matriz

$$C = \begin{bmatrix} \nabla^2 x & \nabla x \nabla y \\ \nabla x \nabla y & \nabla^2 y \end{bmatrix}, \tag{4.1}$$

em que ∇x e ∇y representam o gradiente nas direções x e y, respectivamente. A média é dada pela própria coordenada estimada pelo algoritmo de rastreamento de características, representado neste trabalho pelo KLT.

Finalmente, nós fazemos a fusão do vetor de estados e da observação, u e v, gerando o vetor de estados para o tempo $k + 1$ (Figura 4.1).

Para tal, é utilizado o método da Máxima Verossimilhança (MLE - *Maximum Likelihood Estimation*), uma estratégia de inferência que consiste em escolher os parâmetros do mundo que maximizam as probabilidades das medidas observadas. Para formalizar melhor o conceito de MLE, vamos definir a função verossimilhança [13]:

$$\mathcal{L}(\theta) = P(\mathbf{z}|\theta), \tag{4.2}$$

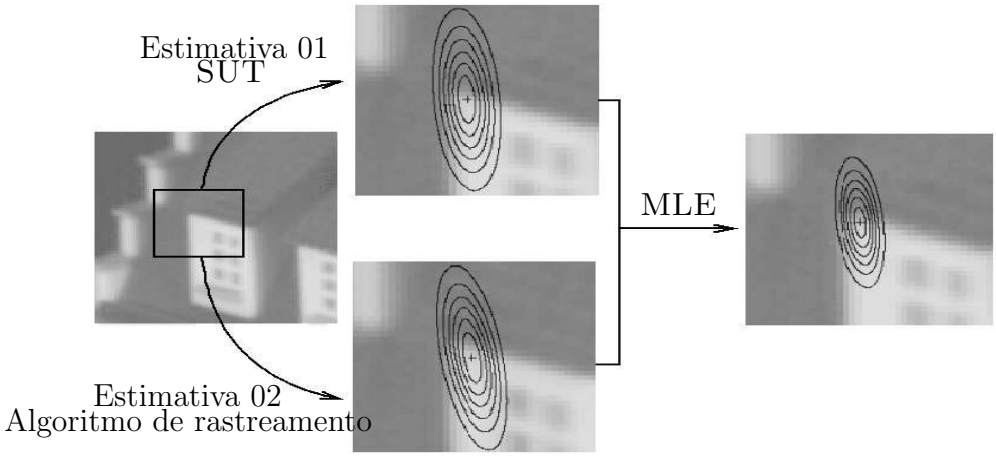


Figura 4.1: Nosso algoritmo.

que consiste na probabilidade condicional de obtenção de medidas $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ dado que o valor verdadeiro é θ. É importante notar que tal função não é uma distribuição de probabilidade sobre θ sendo, portanto, incorreto interpretá-la como a probabilidade dos dados.

Dados $\mathbf{z}$ e a função verossimilhança $\mathcal{L}(\theta)$, a máxima verossimilhança estimada de θ é o valor de θ que maximiza a função verossimilhança

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\theta). \tag{4.3}$$

Em outras palavras, o MLE seleciona o modelo para o qual a probabilidade dos dados observados é maior, ou seja, para o qual a probabilidade posterior para o modelo, dadas as observações, é maior [40].

Quando fazemos a fusão dos dados utilizando o MLE, estamos realizando as etapas de associação de dados e correção, presentes nos filtros preditivos. Após a fusão, teremos uma GRV para a nova localização da característica, e podemos reiterar o processo. O algoritmo UPF é sumarizado no Algoritmo 5.

4.2 Rejeição de Anomalias

A suposição que o erro das características estimadas é uma distribuição Gaussiana não é sempre válida (Seção 2.3), sendo que algumas vezes características são mapeadas para

Algoritmo 2 *Unscented Feature Tracking* - UFT

```
1: função Unscented Feature Tracking - UFT
2:     dada uma sequência de imagens;
3:     selecione pontos característicos e calcule a sua distribuição Gaussiana inicial
4:     para cada característica faça
5:         calcule a SUT utilizando o algoritmo 1;
6:         use um algoritmo de rastreamento para obter a observação e calcule a co-
               variância;
7:         faça a fusão destas duas estimativas utilizando a MLE;
8:     fim para
9: fim função
```

posições muito diferentes da correta. Isso pode causar distúrbios nos resultados de tarefas de mais alto nível que utilizam as correspondências estimadas. Então, é necessário identificar e descartar estas características, chamadas anomalias (*outliers*).

Existem inúmeros algoritmos disponíveis que visam lidar com este problema [15] [41]. De forma geral, a solução encontrada consiste um utilizar uma etapa adicional, responsável por decidir se a característica rastreada deve ou não ser mantida, baseando-se principalmente em regras pré-definidas. Podemos utilizar um algoritmo como o RANSAC, por exemplo, para detectar anomalias.

Nosso algoritmo, no entanto, realiza o descarte de anomalias de forma automática, podendo assim ser considerado um algoritmo robusto. Como vimos, o método UFT envolve a propagação de cinco pontos sigma e o cálculo de uma matriz de covariância para cada ponto característico em cada uma das três etapas: predição, observação e fusão.

Quando uma característica é mapeada para uma localização muito distante da correta, nosso método usualmente falha em um dos seguintes aspectos: (a) não consegue mapear os cinco pontos sigma e/ou (b) a matriz da etapa de fusão ou observação não é simétrica definida positiva (a SUT garante que a matriz de covariância da etapa de predição possui esta propriedade). Em ambos os casos a característica em questão é descartada. No próximo capítulo discutiremos novamente este tópico, apresentado exemplos.

Capítulo 5

Reconstrução 3D

A recuperação da estrutura tridimensional (informação de profundidade) de uma cena ou objeto a partir de imagens bidimensionais é uma tarefa importante em diversas aplicações. Vale lembrar que durante o processo de formação da imagem na câmera são perdidas informações tridimensionais sobre os objetos de interesse [13].

Em imagens médicas, a reconstrução 3D facilita vários procedimentos, levando possivelmente a diagnósticos mais precisos. A reconstrução de modelos de face, por sua vez, têm despertado o interesse de diversas áreas devido à ampla variedade de aplicações em que podem ser utilizados, tais como identificação para controle de acesso, reconhecimento de expressões e *surveillance* [3, 11, 49, 26].

São dois os subproblemas computacionais associados com a reconstrução 3D: o estabelecimento de correspondências entre características e a estimativa da estrutura. O problema das correspondências foi discutido no Capítulo 2, onde vimos que é uma tarefa extremamente difícil e sujeita a erros. A dificuldade do segundo subproblema, por sua vez, dependerá da quantidade (e qualidade) da informação disponível *a priori*.

A reconstrução 3D é uma aplicação do UFT em dois aspectos: primeiro, como obtemos estimativas mais precisas, o primeiro subproblema terá uma melhor aproximação. Além disso, podemos utilizar a informação da incerteza associada às correspondências em um procedimento de *bundle adjustment*, que consiste em um processo que minimiza o erro de reprojeção, visando a melhoria da precisão da estrutura 3D estimada.

5.1 Trabalhos Relacionados

Os métodos de reconstrução 3D diferem em fatores tais como número necessário de imagens e tipos de informações exploradas para inferir dados 3D. Uma abordagem comum para a reconstrução 3D utiliza múltiplas imagens e, com base no princípio que um ponto físico no espaço é projetado em diferentes localizações em imagens capturadas de diferentes pontos de vista, infere informação de profundidade a partir da diferença entre as localizações projetadas [7, 2].

Dois dos principais métodos que usam esta abordagem são o *structure-from-motion* (SfM) [4, 35, 20] e o *structure-from-stereo* [35, 20]. Enquanto o segundo usa imagens tomadas de diferentes pontos de vista para fazer a reconstrução, o SfM tipicamente envolve uma sequência monocular de imagens proximamente amostradas, onde dois tipos de movimento podem estar presentes: câmera e cena. Neste trabalho, assumimos que há somente um movimento rígido relativo entre a câmera e a cena, ou seja, os objetos da cena não possuem movimentos distintos.

Independentemente do método sendo utilizado, existem ambiguidades inerentes ao processo de reconstrução a partir de correspondências, como veremos a seguir [19].

5.2 Ambiguidade na Reconstrução

Os parâmetros obtidos no processo de reconstrução não são uma representação exata dos pontos e câmeras reais, diferindo da reconstrução real por uma transformação pertencente a uma dada classe, cujo tipo irá depender das informações que temos disponíveis.

Se não temos informações sobre a calibração ou a posição relativa das câmeras, a ambiguidade da reconstrução é representada por uma transformação projetiva (Figura 5.1). Este fato é um importante resultado, formalizado no Teorema 1 [19, 20].

Teorema 1 - Teorema da reconstrução projetiva [19]. *Suponha que* $\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}'_i$ *é um conjunto de correspondências entre pontos em duas imagens, e que a matriz fundamental* F *é unicamente determinada pela condição* $\boldsymbol{x}'_i F \boldsymbol{x}_i = 0$ *para todo* i*. Sejam* $(P_1, P'_1, \{\boldsymbol{X}_{1i}\})$ *e* $(P_2, P'_2, \{\boldsymbol{X}_{2i}\})$ *as reconstruções das correspondências* $\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}'_i$*. Então, existe uma matriz*

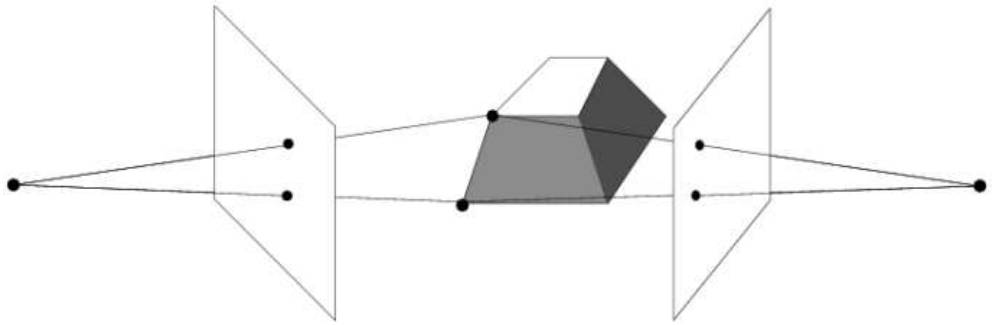


Figura 5.1: Reconstrução projetiva.

não-singular H *tal que* $P_2' = P_1' H^{-1}$, $P_2 = P_1 H^{-1}$ *e* $\boldsymbol{X}_{2i} = H\boldsymbol{X}_{1i}$ *para todo* i, *exceto para aqueles em que* $F\boldsymbol{x}_i = \boldsymbol{x}_i'^T F = 0$.

Em outras palavras, é possível obter a reconstrução projetiva de uma cena baseando-se apenas no conhecimento das correspondências entre as imagens, ou seja, independentemente do conhecimento dos parâmetros ou da pose (posição+orientação) das câmeras. Quaisquer duas reconstruções a partir destas correspondências são projetivamente equivalentes [19, 20].

Se informações externas estiverem disponíveis, é possível determinar a cena a menos de uma transformação Euclidiana (rotação, translação, escala). Na reconstrução Euclidiana, também chamada de reconstrução métrica ou de similaridade (Figura 5.2), as propriedades métricas, tais como ângulos, possuem os valores reais quando medidos na reconstrução [19, 20].

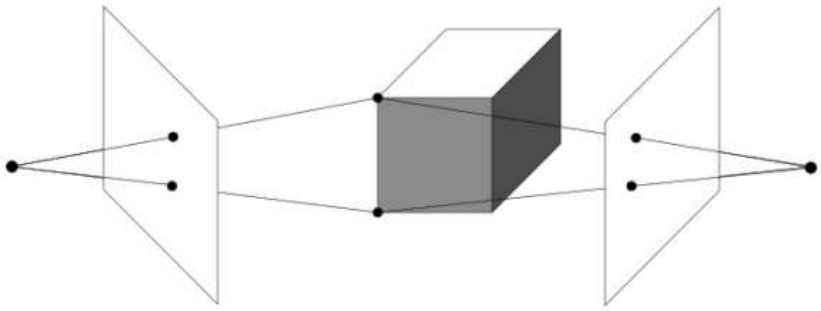


Figura 5.2: Reconstrução Euclidiana.

Em suma, sem a informação da localização da cena em relação a um sistema de

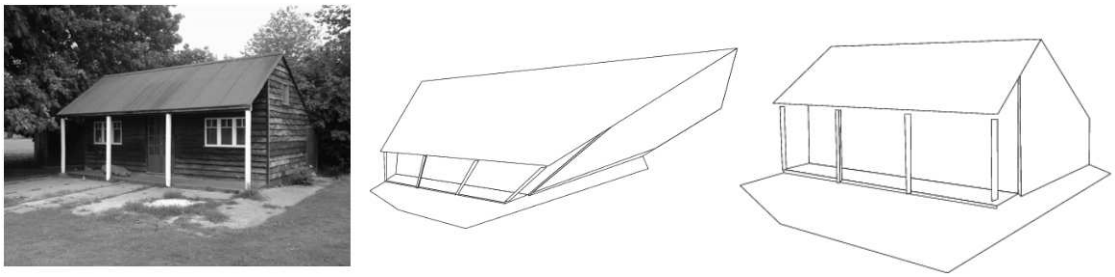
coordenadas 3D conhecido, geralmente não é possível reconstruir fielmente a orientação ou a posição, independentemente do número de vistas utilizado ou do conhecimento dos parâmetros intrínsecos ou extrínsecos da câmera. Além disso, a escala da cena também não pode ser determinada [19].

De uma maneira mais formal, considere o conjunto de pontos $\mathbf{X}_i$ e o par de câmeras P e P' que projetam $\mathbf{X}_i$ em pontos de imagem $\mathbf{x}_i$ e $\mathbf{x}'_i$. Seja

$$H_S = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & \lambda \end{bmatrix} \tag{5.1}$$

uma transformação Euclidiana qualquer, em que R é uma rotação, $\mathbf{t}$ uma translação e λ^{-1} a escala total.

Se substituirmos os pontos $\mathbf{X}_i$ por $H_S\mathbf{X}_i$ e as câmeras P e P' por PH_S^{-1} e $P'H_S^{-1}$, os pontos de imagem não mudarão, sendo que $P\mathbf{X}_i = PH_S^{-1}H_S\mathbf{X}_i$. Quando estamos trabalhando com câmeras calibradas, esta é a única ambiguidade existente, conforme provado em [27]. A Figura 5.3 mostra um exemplo de reconstrução a partir da imagem de uma casa.



(a) (b) (c)

Figura 5.3: Exemplo de Reconstrução 3D. (a) imagem original, (b) reconstrução projetiva e (c) reconstrução Euclidiana.

Outros tipos de ambiguidade na reconstrução resultam de certas suposições sobre os tipos de movimento ou conhecimento parcial dos parâmetros de câmera. Por exemplo, se as duas câmeras estão relacionadas por um movimento translacional, sem mudança na calibração, então a reconstrução é possível a menos de uma transformação afim [19].

5.3 Recuperando Matrizes de Projeção e Estrutura Projetiva (Duas Vistas)

A geometria epipolar relaciona duas vistas de uma cena estática, capturando a informação geométrica necessária para estabelecer correspondências entre um par de imagens [19, 30]. A geometria epipolar é representada por uma matriz 3×3 singular. Se os parâmetros internos da câmera são conhecidos, nós chamamos a matriz de matriz essencial, e trabalhamos com as coordenadas de imagem normalizadas. Caso contrário, apenas as coordenadas de *pixel* estão disponíveis, e trabalhamos com a matriz fundamental (este é o caso deste trabalho).

Existem inúmeras formas de decompor a matriz fundamental F para obter as matrizes de projeção e a estrutura tridimensional a partir de duas vistas[1]. É fato que [30]

$$F = \widehat{T}' K R K^{-1}, \tag{5.2}$$

onde $R \in \mathbb{R}^{3\times3}$ é a matriz de rotação, $T \in \mathbb{R}^3$ é o vetor de translação, $T' = KT$, $\widehat{T}$ é a matriz 3×3 tal que $T \times v = \widehat{T}v$, e $K \in \mathbb{R}^{3\times3}$ é a matriz de calibração dada por [30]

$$K = \begin{bmatrix} fs_x & fs_\theta & \theta_x \\ 0 & fs_y & \theta_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

em que f é a distância focal, s_θ, s_x e s_y são fatores de escala e θ_x e θ_y são as coordenadas (em *pixels*) do ponto principal em relação ao quadro de referência da imagem.

A partir da Equação 5.2, temos que todas as matrizes de projeção

$$\Pi_p = [KRK^{-1} + T'v^T, v^4T'] \tag{5.4}$$

produzem a mesma matriz fundamental para qualquer valor de $v = [v_1, v_2, v_3]^T$ e v_4 e, portanto, existe uma família de escolhas possíveis. Uma escolha comum, conhecida como decomposição canônica, é dada por [30]

$$\Pi_{1p} = [I, 0], \quad \Pi_{2p} = [(\widehat{T}')^T F, T'], \quad \lambda_1 \mathbf{x}'_1 = \mathbf{X}_p, \quad \lambda_2 \mathbf{x}'_2 = (\widehat{T}')^T F \mathbf{X}_p + T'. \tag{5.5}$$

[1]O código (em Matlab) deste processo está disponível em http://vision.ucla.edu/MASKS/. Nós o utilizamos como base para o desenvolvimento deste trabalho.

em que I é a matriz identidade, λ_i é um parâmetro de escala e $\mathbf{x}' = K\mathbf{x}$ (coordenadas de *pixel*).

Dependendo da escolha de v e v_4, serão obtidas diferentes matrizes de projeção Π_p, as quais resultarão em diferentes coordenadas projetivas $\mathbf{X}_p$. Teremos, assim, diferentes reconstruções, algumas das quais mais distorcidas que outras, no sentido de estarem mais longe da reconstrução Euclidiana verdadeira.

Na prática, é comum assumir que o centro ótico está no centro da imagem, que conhecemos uma aproximação da distância focal (a partir de calibrações anteriores da câmera, por exemplo) e que os *pixels* são quadrados sem inclinação. Desta forma, iniciamos com uma aproximação da matriz de parâmetros intrínsecos K, $\tilde{K}$.

Depois, podemos escolher v e v_4 exigindo que o primeiro bloco da matriz de projeção esteja o mais próximo possível da matriz de rotação entre duas vistas, sendo $R \approx v_4(\widehat{T}')^T F + T'v^T$. Se a rotação entre as duas vistas for suficientemente pequena, podemos escolher $\tilde{R} \approx I$, e resolver linearmente para v e v_4. Para o caso de rotação geral, a equação acima pode ser resolvida para v a partir de uma estimativa inicial de $\tilde{R}$. Se este não é o caso, há um método alternativo de decomposição canônica [30].

A partir da matriz de projeção obtida, podemos recuperar a estrutura 3D. Se a estimativa inicial de $\tilde{K}$ foi boa, os pontos 3D estimados devem ser visíveis, ou seja, sua escala deve ser positiva. Se isso não ocorrer, o valor da distância focal deve ser alterado de forma que a maioria dos pontos tenha escala positiva. O processo é resumido no Algoritmo 3.

5.4 Reconstrução a Partir de Múltiplas Vistas

Quando temos mais de duas imagens disponíveis, tomadas de diferentes pontos de vista, podemos utilizar os algoritmos de reconstrução a partir de múltiplas vistas. Dependendo do algoritmo sendo utilizado, as imagens podem ser adicionadas uma de cada vez ou simultaneamente [30]. Aqui, falaremos do caso não-calibrado.

Para a configuração de múltiplas vistas temos:

$$\lambda_i^j \mathbf{x}_i^j = \Pi_i \mathbf{X}^j, \quad i = 1, 2, \ldots, m, \;\; j = 1, 2, \ldots, n, \tag{5.6}$$

em que m é o número de vistas, n o número de pontos característicos, a matriz $\Pi_i = K_i\Pi_0 g_i \in \mathbb{R}^{3\times4}$ é a matriz de projeção, $K_i \in \mathbb{R}^{3\times3}$ a matriz de parâmetros intrínsecos da i-ésima câmera, $\Pi_0 = [I, 0] \in \mathbb{R}^{3\times4}$ é a matriz de projeção padrão, e $g_i \in SE(3)$[2] é o deslocamento rígido da câmera com relação ao quadro de referência do mundo.

O objetivo do algoritmo é recuperar todas as poses da câmera para as m vistas e a estrutura 3D de pontos que apareçam em pelo menos duas vistas. O princípio do algoritmo de múltiplas vistas consiste em explorar a seguinte equação [30]:

$$P_i = \begin{bmatrix} R_i^s \\ T_i \end{bmatrix} \doteq \begin{bmatrix} \mathbf{x}_1^{1T} \otimes \widehat{\mathbf{x}_i^1} & \alpha^1 \widehat{\mathbf{x}_i^1} \\ \mathbf{x}_1^{2T} \otimes \widehat{\mathbf{x}_i^2} & \alpha^2 \widehat{\mathbf{x}_i^2} \\ \dots & \dots \\ \mathbf{x}_1^{nT} \otimes \widehat{\mathbf{x}_i^n} & \alpha^n \widehat{\mathbf{x}_i^n} \end{bmatrix} \begin{bmatrix} R_i^s \\ T_i \end{bmatrix} = 0 \quad \in \mathbb{R}^{3n}, \tag{5.7}$$

em que $\otimes$ é o produto de Kronecker, $\alpha^j = 1/\lambda_1^j$ pode ser interpretado como a inversa da profundidade do ponto j com relação ao primeiro quadro, $R_i^s = [r_{11}, \dots, r_{33}] \in \mathbb{R}^9$, e $T_i \in \mathbb{R}^3$, com $i = 2, 3, \dots, m$, onde m é o número de vistas.

Um resultado importante é que se conhecemos α^j, a matriz P_i é de posto 11 se mais de $n \geq 6$ pontos são dados em posição geral. Neste caso, o espaço nulo de P_i é único a menos de um fator de escala, assim como também será a matriz de projeção.

O procedimento é detalhado no Algoritmo 4. As estimativas do movimento da câmera e da estrutura 3D são realizadas alternadamente, explorando as restrições de múltiplas vistas em todas as vistas. Após a convergência do algoritmo, o movimento da câmera é dado por $[R_i, T_i], i = 2, 3, \dots, m$ e a profundidade dos pontos (com relação à primeira câmera) são dados por $\lambda_1^j = 1/\alpha^j, j = 1, 2, \dots, n$. As matrizes de projeção e a estrutura 3D resultantes podem ser refinadas usando um algoritmo de otimização não-linear, conforme descrito em [30].

[2] $SE(3)$ é o conjunto de todos os movimentos ou transformações que preservam a norma e o produto cruzado entre quaisquer dois vetores [30].

5.5 Atualização da Reconstrução Projetiva Para a Euclidiana

A reconstrução projetiva obtida com o Algoritmo 4 está relacionada à estrutura Euclidiana por um transformação linear $H \in \mathbb{R}^{4\times4}$ [30].

$$\Pi_{ip} \approx \Pi_{ie} H^{-1}, \qquad \mathbf{X}_p \approx H\mathbf{X}_e, \qquad i = 1, 2, \ldots, m, \tag{5.8}$$

em que $\approx$ indica a igualdade a menos de um fator de escala, $\Pi_{1p} = [I, 0]$ e H tem a forma

$$H = \begin{bmatrix} K_1 & 0 \\ -v^T K_1 & 1 \end{bmatrix} \in \qquad \mathbb{R}^{4\times4}. \tag{5.9}$$

Na prática, assumimos como uma aproximação inicial que o eixo ótico é ortogonal ao plano da imagem, o intersectando em seu centro, e que os *pixels* são quadrados. A partir de tais suposições, estimativas razoáveis das parâmetros intrínsecos da câmera são obtidas, as quais podem ser refinadas utilizando esquemas de otimização não-lineares. A partir de tais suposições, a restrição quádrica absoluta (*absolute quadric constraint*) dada por [30]

$$\Pi_{ip} Q \Pi_{ip}^T \approx S_i^{-1}, \tag{5.10}$$

em que

$$Q \doteq \begin{bmatrix} K_1 K_1^T & -K_1 K_1^T v \\ -v K_1 K_1^T & v^T K_1 K_1^T v \end{bmatrix} \in \mathbb{R}^{4\times4} \quad \text{e } S^{-1} \doteq K_i K_i^T \in \mathbb{R}^{3\times3}, \tag{5.11}$$

assume uma forma particularmente simples:

$$\Pi_{ip} \doteq \begin{bmatrix} a_1 & 0 & 0 & a_2 \\ 0 & a_1 & 0 & a_3 \\ 0 & 0 & 1 & a_4 \\ a_2 & a_3 & a_4 & a_5 \end{bmatrix} \Pi_{ip}^T \approx \begin{bmatrix} f_i^2 & 0 & 0 \\ 0 & f_i^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.12}$$

As entradas de S_i^{-1} satisfazem os seguintes relacionamentos:

$$s_{11} = s_{22}, \quad s_{12} = s_{13} = s_{23} = 0, \quad s_{21} = s_{31} = s_{31} = 0, \tag{5.13}$$

os quais podem ser traduzidos nas restrições da matriz Q:

$$\left[\begin{array}{rcc} \pi_i^{1T} Q \pi_i^1 & = & \pi_i^{2T} Q \pi_i^2, \\ \pi_i^{1T} Q \pi_i^2 & = & 0, \\ \pi_i^{1T} Q \pi_i^3 & = & 0, \\ \pi_i^{2T} Q \pi_i^3 & = & 0, \end{array}\right], \tag{5.14}$$

em que $\Pi_i = [\pi_i^{1T}, \pi_i^{2T}, \pi_i^{3T}]$ é a matriz de projeção escrita em termos de suas linhas. Uma vez recuperada a matriz Q, extraímos K e v a partir da Equação 5.11 e H a partir da Equação 5.9. O procedimento é sumarizado no Algoritmo 3.

5.6 Bundle Adjustment

Conforme vimos anteriormente, no processo de reconstrução 3D, dados os pontos de imagem $\mathbf{x}_i^j$, desejamos encontrar o conjunto de matrizes de projeção Π_i e os pontos $\mathbf{X}^j$ tal que $\Pi_i \mathbf{X}^j = \mathbf{x}_i^j$, com $i = 1, \ldots, m$ e $j = 1, \ldots, n$, onde m é o número de vistas e n o número de características.

Como há um erro intrínseco nas medidas realizadas, esta equação não será satisfeita de maneira exata. Estimamos, então, as matrizes de projeção $\widehat{\Pi}_i$ e os pontos 3D $\widehat{\mathbf{X}}^j$, e minimizamos a distância entre o ponto reprojetado, $\widehat{\mathbf{x}}_i^j = \widehat{\Pi}_i \widehat{\mathbf{X}}_j$, e o ponto estimado (medido) $\mathbf{x}_i^j$, ou seja [40]:

$$min \sum d(\widehat{\Pi}_i \widehat{\mathbf{X}}^j, \mathbf{x}_i^j) \tag{5.15}$$

em que $d(x, y)$ é uma medida de distância entre os pontos x e y.

Este processo, envolvendo a minimização do erro de reprojeção para estimar os parâmetros desconhecidos e reconstruir a cena, é conhecido como *bundle adjustment*, e deveria ser utilizado como um passo final em qualquer algoritmo de reconstrução, de forma a melhorar a estimativa inicial realizada. Como o *bundle adjustment* envolve a minimização de uma função custo, a escolha de tal função é um importante passo [40, 14].

Neste trabalho, utilizamos a função de custo Soma dos Erros ao Quadrado com pesos (SSE - *Weight Sum of Squared Error*), dada por [40]:

$$f(x) \equiv \frac{1}{2} \sum_i \Delta z_i(x)^T W_i \Delta z_i(x), \tag{5.16}$$

em que $\Delta z_i(x) \equiv \tilde{z}_i - z_i(x)$ é o erro de predição das características, com $\tilde{z}_i$ representando vetores de observação preditos por um modelo $z_i = z_i(\mathbf{x})$, onde $\mathbf{x}$ é o vetor de parâmetros do modelo. W_i é uma matriz simétrica definida positiva arbitrária.

Note que ao utilizarmos esta função custo é possível atribuir covariâncias individuais à cada estimativa. É exatamente este aspecto que exploramos no nosso algoritmo. Como cada característica tem uma medida de incerteza associada, representada pelo segundo momento central (covariância), tomamos W como sendo a incerteza associada de cada ponto. Os resultados obtidos são apresentados no próximo capítulo (Seção 6.3).

Algoritmo 3 Reconstrução projetiva - duas vistas [30]

1: Dado um conjunto inicial de correspondências entre pontos expressadas em coordenadas de *pixel* $(\mathbf{x}_1'^j, \mathbf{x}_2'^j)$ para $j = 1, 2, \ldots, n$,

2: Aproxime a matriz de calibração $\tilde{K}$ escolhendo o centro ótico como o centro da imagem, assumindo que os *pixels* são quadrados e inicializando a distância focal como $\tilde{f}$. Por exemplo, para um plano de imagem de tamanho $D_x \times D_y$ *pixels*, uma inicialização comum é

$$\tilde{K} = \begin{bmatrix} \tilde{f} & 0 & D_x/2 \\ 0 & \tilde{f} & D_y/2 \\ 0 & 0 & 1 \end{bmatrix}$$

com $\tilde{f} = k \times D_x$, onde k é usualmente escolhido no intervalo $[0.5, 2]$.

3: Estime a matriz fundamental [30, 19]. As coordenadas normalizadas são denotadas por $\tilde{\mathbf{x}}_1 = \tilde{\mathbf{K}}^{-1}\mathbf{x}_1'$ e $\tilde{\mathbf{x}}_2 = \tilde{\mathbf{K}}^{-1}\mathbf{x}_2'$.

4: Calcule o epipolo T' como o espaço nulo (núcleo) de F^T: a partir da SVD de $F^T = USV^T$, T' é a última (terceira) coluna da matriz V.

5: Escolha $v \in R^3$ e $v_4 \in R$ tal que a parte rotacional da matriz fundamental $v_4(\widehat{T'})^T F + T'v^T$ seja o mais próximo de uma (pequena) rotação:

- Assuma $\tilde{R} \approx I$;
- Resolva a equação $v_4(\widehat{T'})^T F + T'v^T$, no sentido de mínimos quadrados, utilizando a SVD.

6: Considerando o primeiro quadro como referência, as matrizes de projeção são dadas por:

$$\Pi_{1p} = [I, 0], \qquad \Pi_{2p} = [v_4(\widehat{T'})^T F + T'v^T, T'] = [R, T']$$

.

7: A estrutura 3D projetiva $\mathbf{X}_p$ para cada $j = 1, 2, \ldots, n$ pode ser agora estimada como segue:

- Denote as matrizes de projeção por $\Pi_{1p} = [\pi_1^{1T}, \pi_1^{2T}, \pi_1^{3T}]$ e $\Pi_{1p} = [\pi_2^{1T}, \pi_2^{2T}, \pi_2^{3T}]$ escritos em termos de seus três vetores linha. Sejam $\tilde{\mathbf{x}}_1 = [\tilde{x}_1, \tilde{y}_1, 1]^T$ e $\tilde{\mathbf{x}}_2 = [\tilde{x}_2, \tilde{y}_2, 1]^T$ os pontos correspondentes nas duas vistas. A estrutura a ser estimada satisfaz as seguintes restrições:

$$(\tilde{x}_1\pi_1^{3T} - \pi_1^{1T})\mathbf{X}_p = 0, \quad (\tilde{y}_1\pi_1^{3T} - \pi_1^{2T})\mathbf{X}_p = 0,$$

$$(\tilde{x}_2\pi_2^{3T} - \pi_2^{1T})\mathbf{X}_p = 0, \quad (\tilde{y}_2\pi_2^{3T} - \pi_2^{2T})\mathbf{X}_p = 0,$$

- A estrutura projetiva pode então ser recuperada como a solução de mínimos quadrados de um sistema linear de equações $M\mathbf{X}_p = 0$. A solução para cada ponto é dado pelo autovetor de M^TM que corresponde ao seu menor autovalor, calculado novamente utilizando a SVD.
- As escalas desconhecidas λ_1^j são simplesmente a terceira coordenada da representação homogênea de $\mathbf{X}_p^j$ (com a quarta coordenada de $\mathbf{X}_p$ normalizada por 1), tal que $\mathbf{X}_p^j = \lambda_1^j\tilde{\mathbf{x}}_1^j$ para todos os pontos $j = 1, 2, \ldots, n$.

Algoritmo 4 Algoritmo de estimativa de estrutura e movimento a partir de múltiplas vistas [30]

1: Dadas m imagens $\mathbf{x}_1^j, \mathbf{x}_2^j, \ldots, \mathbf{x}_m^j$, $j = 1, 2, \ldots, n$, estime a matriz de projeção $\Pi_i = [R_i, T_i], i = 2, 3, \ldots, m$ como segue:

2: Inicialização: $k = 0$; Sejam $\alpha_0^j = 1/\lambda_1^j$ as escalas recuperadas a partir do algoritmo de inicialização de duas vistas (Algoritmo 1).

3: Determine a matriz P_i, usando as escalas α^j, assim como pela Equação 3.7 e calcule o vetor singular v_{12} associado com o menor valor singular de $P_i, i = 2, 3, \ldots, m$. Desempilhe as primeiras nove entradas de v_{12} para obter $\tilde{R}_i$: as últimas três entradas de v_{12} são $\tilde{T}_i$.

4: Como estamos trabalhando com câmeras não calibradas, as estimativas atuais de (R_i, T_i) são simplesmente $(\tilde{R}_i, \tilde{T}_i)$, para $i = 2, 3, \ldots, m$.

5: Seja $\Pi_{ik+1} = [R_i, T_i]$.

6: Dados todos os movimentos, recalcule as escalas α_{k+1}^j através da fórmula

$$\alpha_{k+1}^j = -\frac{\sum_{i=2}^m (\widehat{\mathbf{x}}_i^j T_i)^T \widehat{\mathbf{x}}_i^j R_i \mathbf{x}_i^j}{\sum_{i=2}^m \|\widehat{\mathbf{x}}_i^j T_i\|^2}$$

recalculando assim as coordenadas 3D de cada ponto $\mathbf{X}_{k+1}^j = \lambda_{1k+1}^j \mathbf{x}_1^j$.

7: Calcule o erro de reprojeção

$$e_r = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \|\mathbf{x}_i^j - \pi(\Pi_{ik+\mathbf{1}} \mathbf{X}_{k+1}^j)\|^2$$

se $e_r > \epsilon$ para um $\epsilon > 0$, então $k = k + 1$ e vá para o passo 2, senão pare.

Algoritmo 5 Recuperação da restrição quádrica absoluta e atualização Euclidiana

1: Dadas m matrizes de projeção $\Pi_i, i = 1, 2, \ldots, m$ recuperadas a partir do Algoritmo 2, para cada matriz de projeção estabeleça as restrições lineares em $Q \doteq \begin{bmatrix} a_1 & 0 & 0 & a_2 \\ 0 & a_1 & 0 & a_3 \\ 0 & 0 & 1 & a_4 \\ a_2 & a_3 & a_4 & a_5 \end{bmatrix}$.

Seja $Q^s \doteq [a_1, a_2, a_3, a_4, a_5]^T$ ser a versão vetorial de Q.

2: Forme uma matriz $\chi \in R^{4m \times 5}$ empilhando m dos seguintes blocos 4×5, um para cada $i = 1, 2, \ldots, m$, onde cada linha do bloco corresponde à uma das restrições da Equação 3.14

$$\begin{bmatrix} u_1^2 + u_2^2 - v_1^2 - v_2^2 & 2u_4u_1 - 2v_1v_4 & 2u_4u_2 - 2v_2v_4 & 2u_4u_3 - 2v_3v_4 & u_4^2 - v_4^2 \\ u_1v_1 + u_2v_2 & u_4v_1 + u_1v_4 & u_4v_2 + u_2v_4 & u_4v_3 + u_3v_4 & u_4v_4 \\ u_1w_1 + u_2w_2 & u_4w_1 + u_1w_4 & u_4w_2 + u_2w_4 & u_4w_3 + u_3w_4 & u_4w_4 \\ v_1w_1 + v_2w_2 & v_4w_1 + v_1w_4 & v_4w_2 + v_2w_4 & v_4w_3 + v_3w_4 & v_4w_4 \end{bmatrix}$$

onde$u = [u_1, u_2, u_3, u_4] \doteq \pi_i^1, v \doteq \pi_i^2, w \doteq \pi_i^3$ são as três colunas da matriz de projeção $\Pi_i = [\pi_i^1, \pi_i^2, \pi_i^3]$, respectivamente.

3: De maneira similar, forme um vetor $b \in R^{4m}$ empilhando m dos seguintes blocos 4D

$$[-u_3^2 + v_3^2 \qquad - u_3v_3 \qquad - u_3w_3 \qquad - v_3w_3]^T$$

4: Resolva para Q^s no sentido de mínimos quadrados: $\hat{Q}^s \doteq \chi^\dagger \mathbf{b}$, em que † denota a pseudo-inversa.

5: Organize Q^s em uma matriz $\tilde{Q}$ de acordo com a definição do passo 1.

6: Force a restrição de posto 3 em $\tilde{Q}$, calculando a sua SVD $\tilde{Q} = U_Q \text{diag}\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\} V_Q^T$. Obtenha Q setando o menor autovalor de $\tilde{Q}$ para zero:

$$Q = U_Q \text{diag}\{\sigma_1, \sigma_2, \sigma_3, 0\} V_Q$$

7: Uma vez que Q tenha sido recuperada usando o algoritmo acima, as distâncias focais f_i das câmeras individuais podem ser obtidas por substituição na Equação 5.12.

8: Faça a atualização Euclidiana usando H na Equação 5.9 com K_1 e v calculados a partir dos parâmetros da restrição quádrica absoluta Q através de

$$K_1 = \begin{bmatrix} \sqrt{a_1} & 0 & 0 \\ 0 & \sqrt{a_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad v = -[a_2/a_1, a_3/a_1, a_4]^T.$$

Capítulo 6

Resultados

Neste capítulo, validamos o método proposto em três diferentes aspectos: a precisão alcançada no rastreamento de características, detecção e rejeição de anomalias e qualidade obtida em um processo de reconstrução 3D (Capítulo 5) utilizando informações sobre a incerteza associada à cada correspondência. Note que a forma de representação utilizada possibilita que o algoritmo seja integrado mais facilmente à outras tarefas de mais alto nível.

Nós comparamos o desempenho do nosso algoritmo contra o KLT padrão em duas sequências reais e em três sequências sintéticas. As sequências reais são do banco de dados de imagem do CMU/VASC[1]. A primeira delas é a sequência *Artichoke* (Figura 6.1), uma cena em que o movimento é puramente translacional. A segunda é a sequência *Hotel* (Figura 6.2), uma cena estática observada por uma câmera fazendo movimentos de translação e rotação.

Figura 6.1: Cinco quadros da sequência *Artichoke* real.

[1]CMU/VASC Image Database. Disponível em http://vasc.ri.cmu.edu/idb/

Figura 6.2: Cinco quadros da sequência *Hotel* real.

As sequências sintéticas foram geradas artificialmente, para a criação uma base de comparação, um *ground truth* de correspondências. Para tal, nós aplicamos movimentos de rotação e translação arbitrários à um quadro de cada sequência real e também renderizamos a animação de um modelo texturizado de uma vaca.

A sequência *Artichoke* sintética possui somente movimentos translacionais, porém mais complexos que os sequência real. A sequência *Hotel* sintética, por sua vez, possui movimentos translacionais e rotacionais, mas somente em 2D. A sequência *Cow* consiste em uma séria de *warpings* controlados.

A mesma transformação aplicada às imagens foi aplicada a um conjunto de características previamente selecionadas pelo algoritmo KLT. Mostramos alguns quadros de cada sequência sintética nas Figuras 6.3, 6.4 e 6.5.

Figura 6.3: Cinco quadros da sequência *Artichoke* sintética.

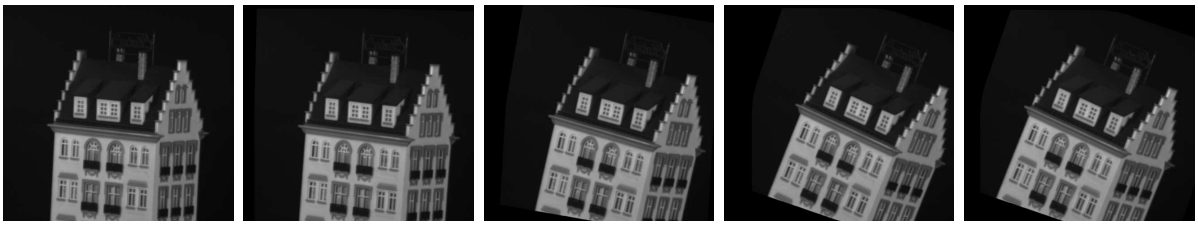


Figura 6.4: Cinco quadros da sequência *Hotel* sintética.

Figura 6.5: Cinco quadros da sequência *Hotel* sintética.

Conforme veremos, os testes experimentais comprovam que o UFT detecta e rejeita anomalias, resultando assim em uma estimativa melhor e mais robusta.

6.1 Cálculo do Erro

Inicialmente, comparamos a precisão obtida pelo KLT e pelo UFT na estimativa de correspondências, tendo como base as informações disponíveis no *ground truth*. Seja $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ um conjunto de n correspondências estabelecidas entre duas imagens e $\hat{\mathbf{x}}'_i$ a correspondência estimada por um dos métodos. A medida de distância RMS (*Root Mean Squared*), dada por [19]

$$E_{res} = \left(\frac{1}{2n} \sum_{i=1}^{n} d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \right)^{\frac{1}{2}}, \tag{6.1}$$

mede a diferença média entre os dados de entrada $\mathbf{x}'_i$ e os pontos estimados $\hat{\mathbf{x}}'_i$, e portanto é apropriado chamá-lo de erro residual. É uma medida de qualidade boa para mensurar o quanto a transformação calculada se aproxima dos dados de entrada.

Além disso, também calculamos a distância dos pontos rastreados à linha epipolar, considerando o Lema 1 a seguir:

Lema 1 - Matching epipolar [30]. *Dois pontos de imagem $\boldsymbol{x}_1$ e $\boldsymbol{x}_2$ correspondem a um único ponto no espaço se e somente se $\boldsymbol{x}_1$ está na linha epipolar $\mathcal{L}_1 = F^T \boldsymbol{x}_2$ ou, equivalentemente, se $\boldsymbol{x}_2$ está na linha epipolar $\mathcal{L}_2 = F\boldsymbol{x}_1$.*

em que F é a matriz fundamental. Em suma, o Lema nos diz que se a geometria epipolar é estimada de maneira exata, todos os pontos devem estar sobre a linha epipolar [47]. Esta

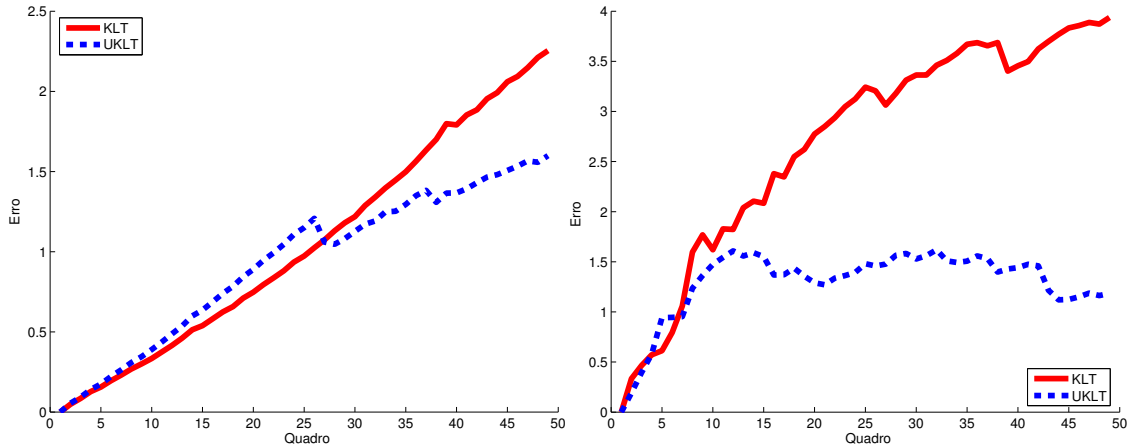
medida é válida porque estamos assumindo um movimento rígido relativo entre a cena e a câmera, ou seja, todos os objetos da cena sofrem o mesmo movimento. Nós utilizamos os pontos rastreados pelo UFT (utilizando o KLT como algoritmo de rastreamento) e pelo KLT para calcular a matriz fundamental entre o primeiro e o quadro corrente de cada sequência, e então calculamos a distância RMS dos pontos rastreados às linhas epipolares correspondentes. Nas duas métricas, quanto menor o erro obtido, melhor a estimativa.

6.2 Análise do Rastreamento de Características

Nesta primeira etapa, nosso objetivo é comparar nosso algoritmo ao KLT, analisando a qualidade das estimativas e as características descartadas por cada método, em sequências reais e sintéticas. Para as sequências sintéticas, utilizamos as duas medidas descritas na Seção anterior. Já para as sequências reais, medimos apenas a distância dos pontos rastreados à linha epipolar associada, pois não temos um *ground truth* disponível.

6.2.1 Sequências Reais

A Figura 6.6(a) mostra o gráfico da média da distância dos pontos rastreados à linha epipolar para a sequência *Artichoke* e a Figura 6.6(b) para a sequência *Hotel*.

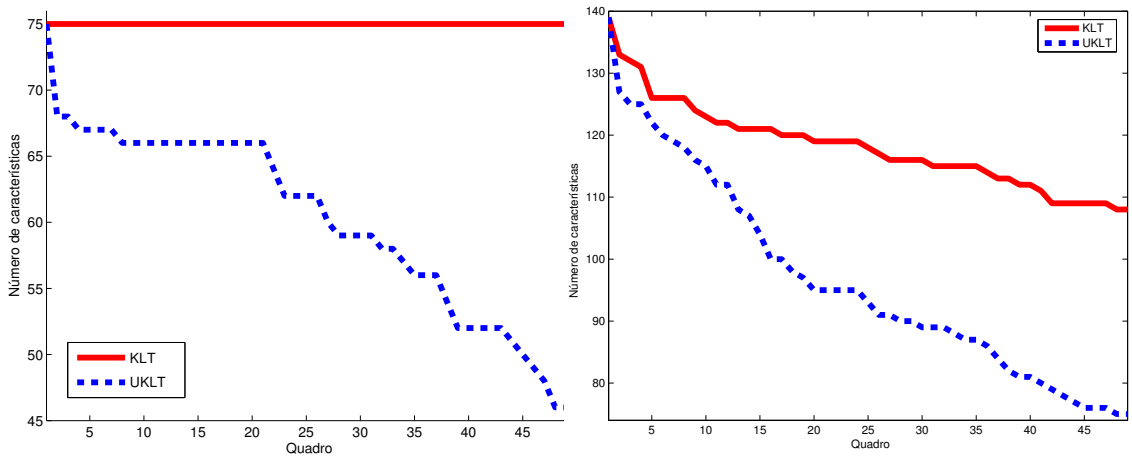


(a) Sequência *Artichoke* real. (b) Sequência *Hotel* real.

Figura 6.6: Média da distância dos pontos rastreados à linha epipolar (menor é melhor).

Para a sequência *Artichoke*, note que o nosso método apresentava um desempenho equivalente até o quadro 27. Depois disto, a qualidade das características rastreadas pelo nosso método é superior. A distância medida no último quadro é de 2.2543 para o KLT e 1.5989 para o UFT. Na sequência *Hotel*, o desempenho do nosso método é superior em praticamente toda a sequência. No último quadro, o erro é de 3.9366 para o KLT e 1.1854 para o UFT.

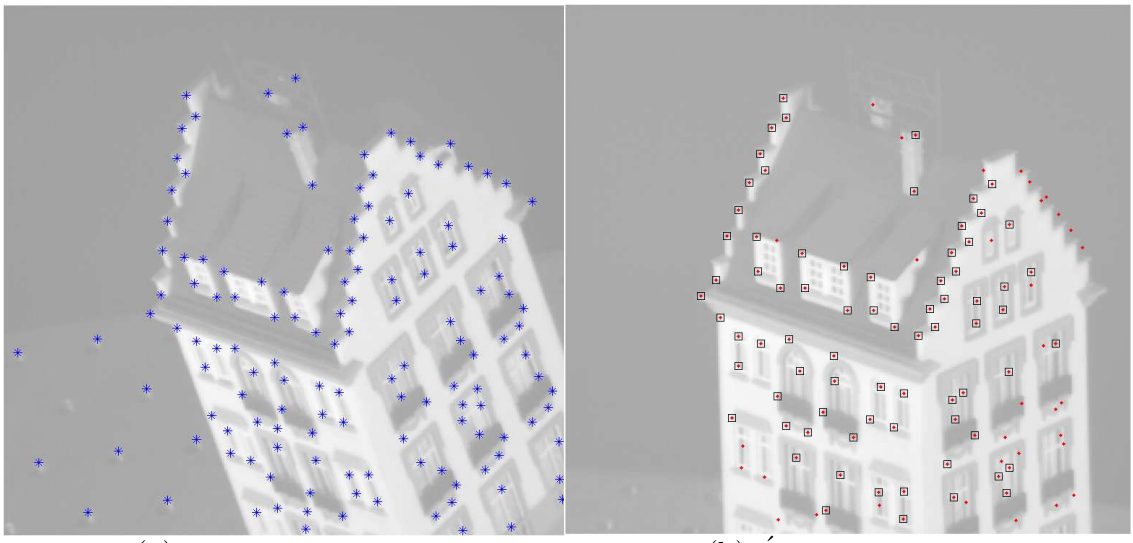
Em ambas as sequências o erro é menor não apenas por causa de melhores estimativas, mas também porque nosso método rejeita características que não são bem rastreadas, enquanto o algoritmo KLT ainda as preserva. A Figura 6.7 mostra o número de características rastreadas para cada sequência.



(a) Sequência *Artichoke* real. (b) Sequência *Hotel* real.

Figura 6.7: Número de características rastreadas.

Para a sequência *Artichoke*, enquanto o KLT rastreou um número constante de características, nosso algoritmo descartou 30 pontos. Para a sequência *Hotel*, o UFT descartou 32 características a mais que o KLT. Na Figura 6.8 mostramos o quadro inicial, com as características selecionadas para rastreamento, e o quadro final, com as características rastreadas pelo UFT (□) e pelo KLT (.), da sequência *Hotel*. Note que as características que foram rejeitadas pelo UFT de forma geral são características que consistem de correspondências erradas, ou seja, anomalias. Na Figura 6.9 mostramos uma região da Figura 6.8 onde isso acontece, para melhor visualização.



(a) Primeiro quadro. (b) Último quadro.

Figura 6.8: Características rastreadas na sequência *Hotel* real.

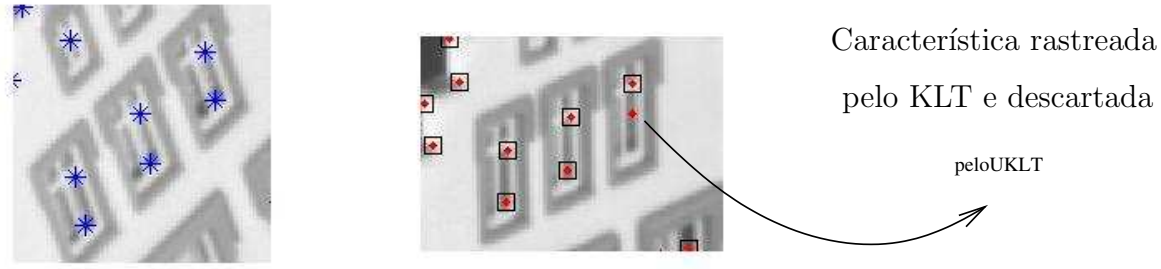


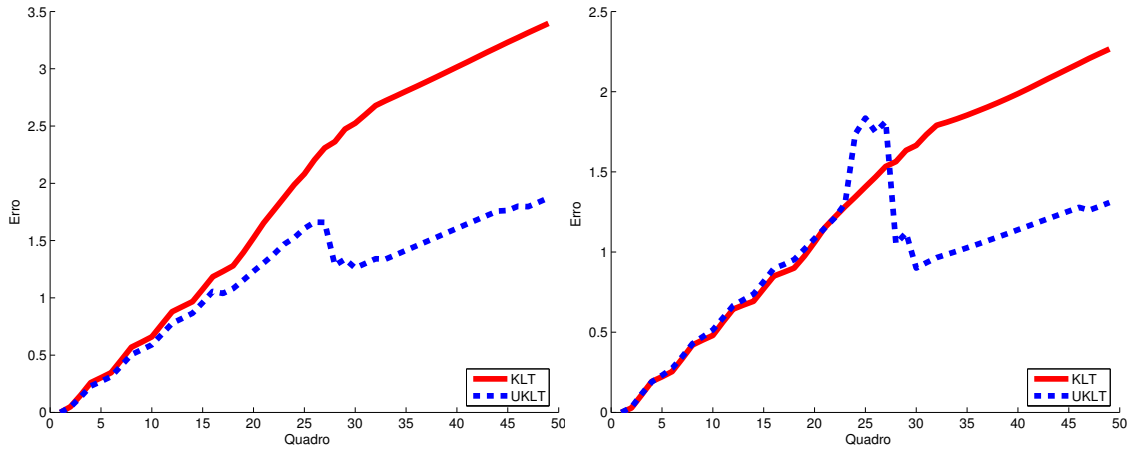
Figura 6.9: Características rejeitadas pelo UFT.

Tais características foram descartadas pelo nosso algoritmo por um dos seguintes motivos: (a) o algoritmo KLT conseguiu rastrear apenas um subconjunto dos cinco pontos sigma; (b) a matriz de covariância resultante em alguma das etapas do processo (observação ou fusão) não é definida positiva. Esses aspectos foram discutidos no Capítulo 4.

6.2.2 Sequências Sintéticas

Para a sequência *Artichoke*, nosso algoritmo possui um desempenho superior ao KLT. Quando medimos a distância ao *ground truth* (Figura 6.10(a)), nosso método apresentou valores de erro inferiores em todos os quadros. No último quadro, o KLT apresentou um erro de 3.3964 e o UFT de 1.8732. Quando medimos a distância à linha epipolar

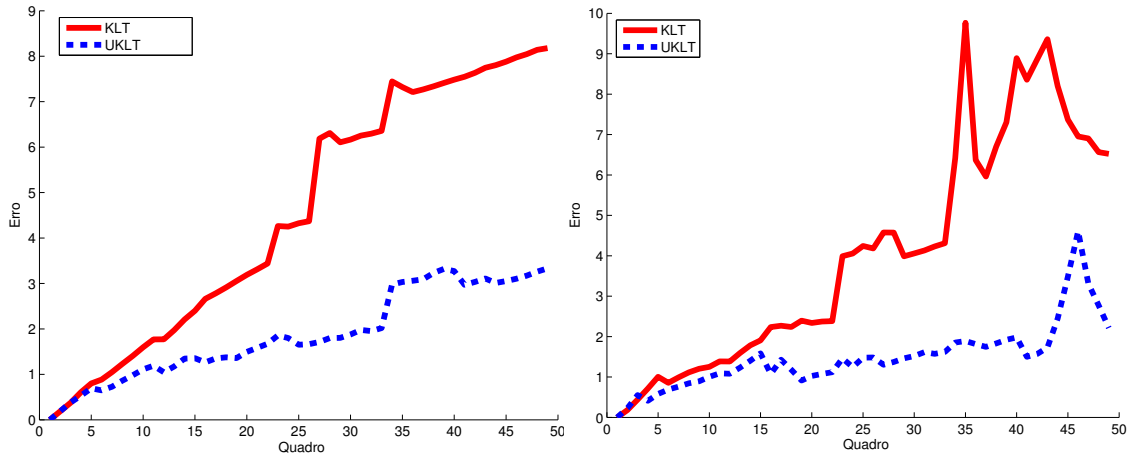
(Figura 6.10(b)), nosso método apresentou erro superior apenas do quadro 23 ao 26. Este salto se deve ao fato de um dos pontos sigma ter sido rastreado para um posição muito distante da real, comprometendo assim a estimativa da média. O erro medido no último quadro é de 2.2662 para o KLT e 1.3090 para o UFT.



(a) Erro em relação à base. (b) Distância da linha epipolar.

Figura 6.10: Resultados para a sequência *Artichoke* sintética (menor é melhor).

A Figura 6.11(a) mostra o gráfico da média da distância dos pontos rastreados à linha epipolar, e a Figura 6.11(b) mostra a distância dos pontos à base de comparação para a sequência *Hotel* sintética.

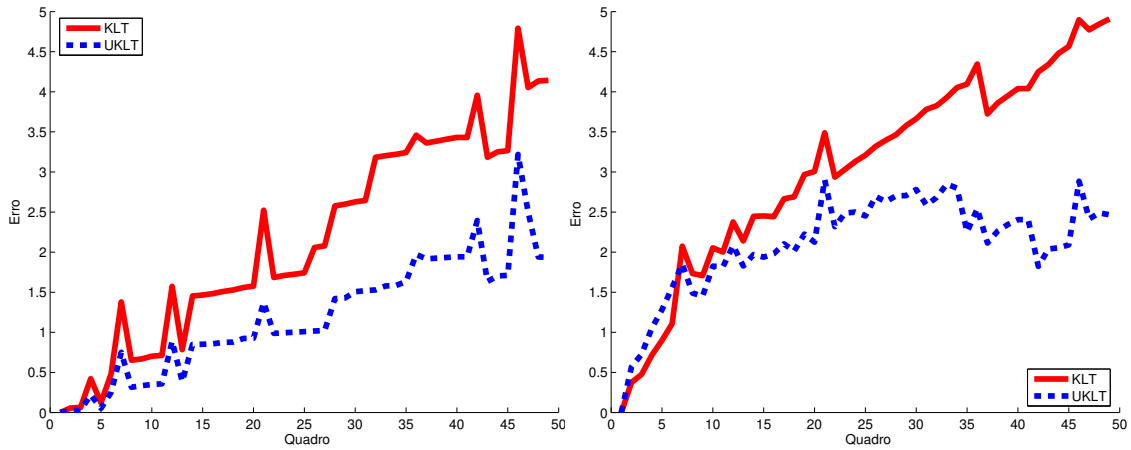


(a) Erro em relação à base. (b) Distância da linha epipolar.

Figura 6.11: Resultados para a sequência *Hotel* sintética (menor é melhor).

Note que para esta sequência o UFT teve um desempenho melhor em todos os quadros, nas duas métricas utilizadas. A distância em relação ao *ground truth*, medida no último quadro, é de 8.1791 para o KLT e 3.3322 para o UFT. A distância à linha epipolar, também no último quadro, é de 6.5228 utilizando o KLT e 2.2140 com o UFT.

Agora, apresentamos os resultados para a sequência *Cow*, que foi gerada a partir de um modelo 3D. Desta forma, foi possível gerar rotações e translações em todos os sentidos, validando melhor o algoritmo proposto. A Figura 6.12 ilustra os resultados obtidos.

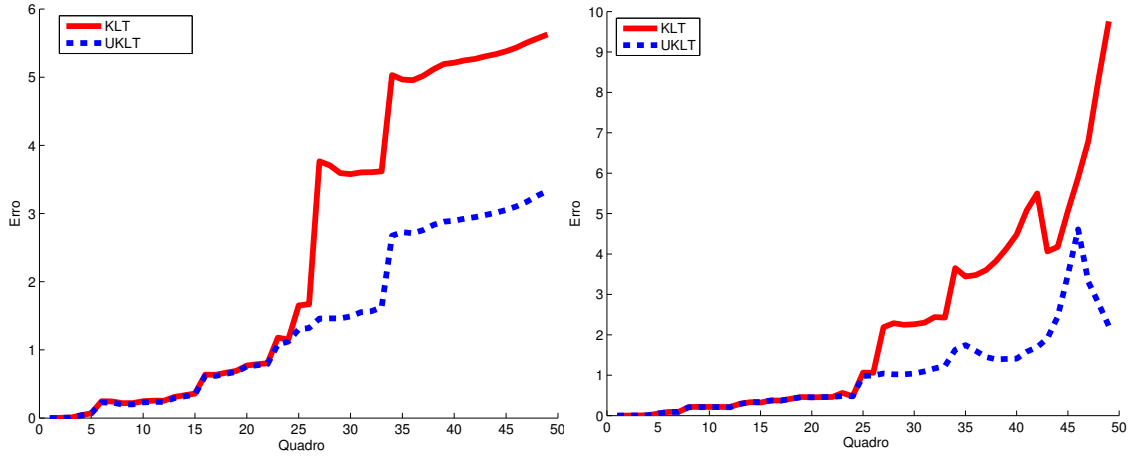


(a) Erro em relação à base. (b) Distância da linha epipolar.

Figura 6.12: Resultados para a sequência *Cow* sintética (menor é melhor).

Nosso método novamente possui melhores resultados que o KLT, apresentando erro inferior em todos os quadros da sequência. O erro em relação à base de comparação foi de 10.0029 para o KLT e 4.3993 para o UFT (Figura 6.12(a)). Quando medimos a distância à linha epipolar, o erro foi de 4.9080 para o KLT e 2.4664 para o UFT (Figura 6.12(b)).

Conforme comentamos, nosso algoritmo possui um desempenho melhor não apenas por detectar e descartar anomalias, mas também porque obtém melhores estimativas. Como teste experimental, selecionamos as características que não foram descartadas por nenhum dos métodos no teste anterior para a sequência *Hotel* real. Então, rastreamos estas características e medimos a diferença em relação ao *ground truth* (Figura 6.13(a)) e a distância à linha epipolar (Figura 6.13(b)). Desta forma, a melhoria dos resultados indicará a uma melhor precisão na estimativa das localizações das caraterísticas, sendo que o descarte de anomalias não está sendo considerado.



(a) Erro em relação à base. (b) Distância da linha epipolar.

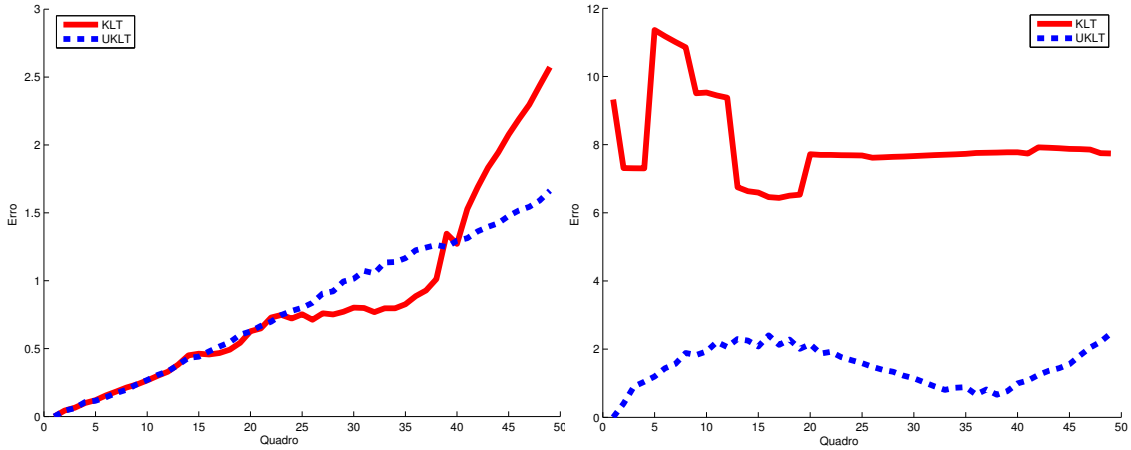
Figura 6.13: Resultados para a sequência *Hotel* sintética ao desconsiderarmos as características descartadas (menor é melhor).

Baseando-se nos resultados experimentais, podemos concluir que o UFT rastreia características com uma precisão maior que o KLT, alcançando melhores estimativas e rejeitando anomalias. Além disso, como a localização das características é representada com uma GRV, é possível obter um grau de confiança – ou medida de incerteza – relacionado à cada estimativa. Esta informação é útil em várias aplicações, como em um procedimento de *bundle adjustment*.

6.3 Aplicação em reconstrução 3D

Para avaliar a qualidade da reconstrução 3D, nós reprojetamos os pontos 3D estimados e medimos a magnitude do erro em relação ao *ground truth*, no caso das sequências sintéticas, e o RMS à linha epipolar para a sequência real.

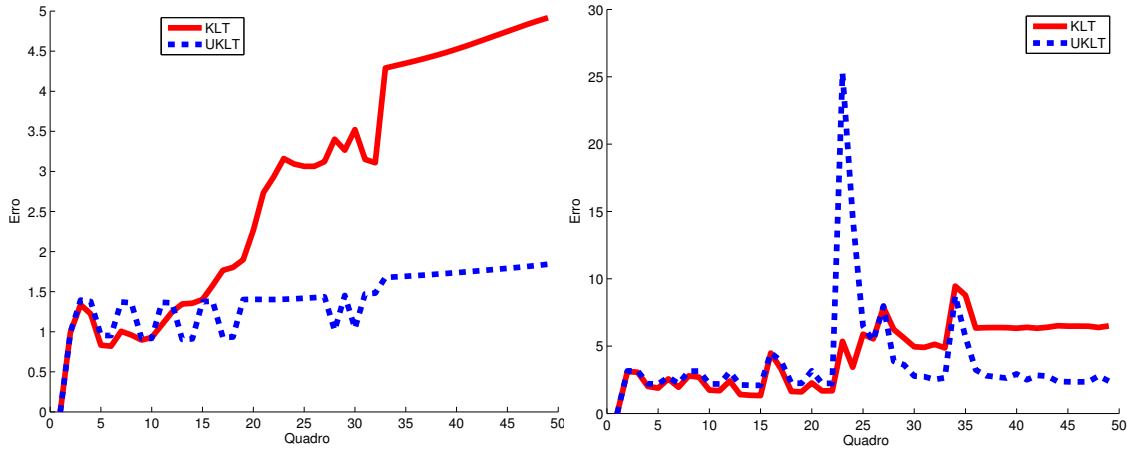
A Figura 6.14 ilustra os resultados para as sequências reais. Note que como nenhuma característica foi descartada pelo algoritmo KLT na sequência *Artichoke*, o erro aumenta de forma contínua no decorrer da sequência. Na sequência *Hotel*, confirmando os melhores resultados obtidos na etapa de estabelecimento de correspondências, o erro medido foi consideravelmente menor utilizando o algoritmo aqui proposto, o UFT.



(a) Sequência *Artichoke*. (b) Sequência *Hotel*.

Figura 6.14: Erro de reprojeção (sequências reais - menor é melhor).

A Figura 6.15 ilustra os resultados para as sequências sintéticas *Artichoke* e *Hotel*. Enquanto para a sequência *Artichoke* o erro de reprojeção é menor em praticamente todos os quadros da sequência, na sequência *Hotel* existem instabilidades, causadas pelas rotações "bruscas" realizadas na criação da sequência. O grande salto na Figura 6.15(b) se deve ao fato de um dos cinco pontos sigma ter sido rastreado para um posição muito distante da real, comprometendo assim a estimativa da média.



(a) Sequência *Artichoke*. (b) Sequência *Hotel*.

Figura 6.15: Erro de reprojeção (sequências sintéticas - menor é melhor).

Na sequência *Cow* estas instabilidades também estão evidentes nos quadros que sofreram transformações mais bruscas (Figura 6.16). Note quanto o KLT é sensível a tais transformações, estabelecendo correspondências erradas (*outliers*) e comprometendo assim o resultado final.

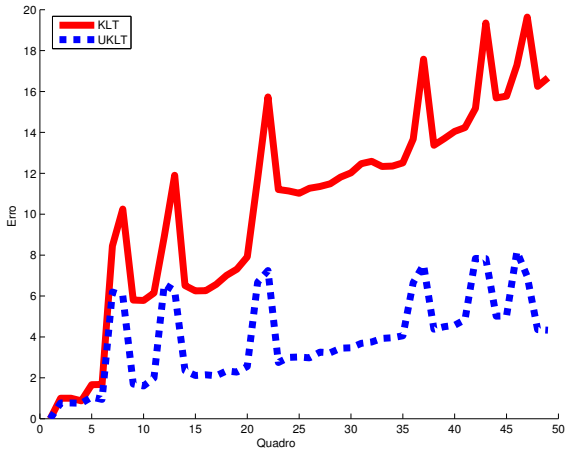


Figura 6.16: Erro de reprojeção na sequência *Cow* (menor é melhor).

6.4 Comparação com Resultados do RANSAC

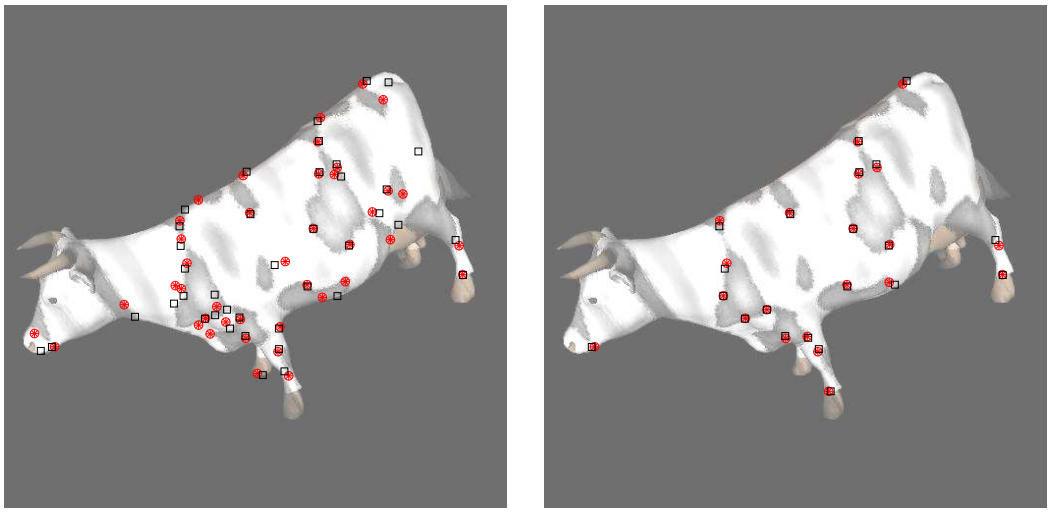
Fatores tais como oclusão e condições de iluminação tornam o rastreamento de características um processo bastante complicado, que mesmo os algoritmos mais elaborados encontram dificuldades para lidar. Desta forma, os algoritmos de rastreamento de correspondências muitas vezes apresentam falsas correspondências, chamadas anomalias (*outlier*). Assim, a detecção e rejeição destas anomalias é uma etapa fundamental para grande parte das tarefas de visão computacional.

O *Random Sample Consensus* (RANSAC) é um estimador robusto capaz de lidar com dados que possuem uma grande proporção de anomalias. O método consiste basicamente em partir de um conjunto inicial com um número mínimo de pontos e então agregar a este conjunto somente pontos que satisfazem a determinadas condições. Para que o processo termine, um número máximo de tentativas deve ser delimitado [12, 19].

Muitas vezes, o RANSAC é aplicado ao conjunto de correspondências definido por um algoritmo de rastreamento de correspondências com o intuito de identificar anomalias, ou seja, pontos que não estão se ajustando ao modelo. As anomalias identificadas são descartadas do conjunto de correspondências.

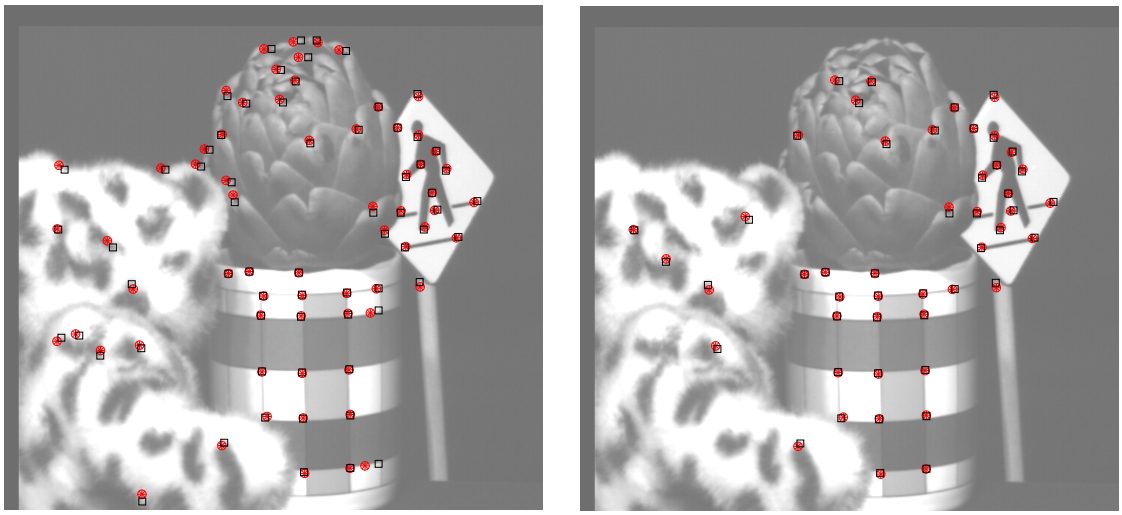
O que fazemos nesta seção é comparar as características selecionadas como anomalias pelo RANSAC e pelo nosso algoritmo. Ressaltamos, entretanto, que os dois métodos possuem objetivos completamente distintos, e a comparação sendo feita aqui diz respeito apenas às características descartadas.

A rejeição automática de anomalias feita pelo UFT apresentou um bom desempenho, descartando um número maior de anomalias que a abordagem KLT + RANSAC. Na Figura 6.17 nós mostramos os resultados do rastreamento no último frame da sequência *Cow* e a Figura 6.18 para a sequência *Artichoke*, para os dois algoritmos. Os símbolos + indicam as posições estimadas e □ as reais. Nós não mostramos as pontos característicos descartados.



(a) KLT tracking + RANSAC (b) UKF tracking

Figura 6.17: Resultados do rastreamento para o último quadro da sequência *Cow*. ⊕ indica a posição estimada e □ a real.



(a) KLT tracking + RANSAC (b) UKF tracking

Figura 6.18: Resultados do rastreamento para o último quadro da sequência *Artichoke*. $\oplus$ indica a posição estimada e $\square$ a real.

6.5 Considerações Sobre a Implementação

Nós implementamos o nosso algoritmo usando MATLAB. Os tempos de execução em um 3.2GHz Pentium 4 com 512Mb de memória estão na Tabela 6.1.

Sequence	UFT		KLT
	Total time	KLT part	
Artichoke real	111.67	73%	21.6570
Artichoke sintética	78.14	75%	20.15
Hotel real	133.29	77%	24.78
Hotel sintética	93.04	76%	22.64
Cow sintética	120.20	79%	19.95

Tabela 6.1: Tempos de execução.

O tempo médio de execução para o algoritmo UFT foi de 107.27 segundos, enquanto para o KLT este tempo foi de 21.83, cerca de cinco vezes menor.

Este tempo de execução já era esperado, considerando que o UFT precisa propagar cinco vezes mais pontos, devido à criação dos pontos sigma. Note o tempo gasto em rastrear estes pontos no UFT (utilizando o KLT).

Capítulo 7

Conclusões e trabalhos futuros

Neste trabalho, modificamos o algoritmo de rastreamento de características KLT através da introdução da incerteza associada à cada correspondência. Para tanto, representamos a localização de cada característica como uma GRV, cuja incerteza é representada pelo segundo momento central. Essa abordagem é um *trade-off* entre custo computacional e flexibilidade de rpresentação, pois é preciso propagar apenas os dois primeiros momentos através da dinâmica do sistema.

Além disso, modelamos o problema das correspondências utilizando conceitos de filtros preditivos, com o intuito de melhorar a precisão das estimativas realizadas, bem como levar em conta tanto informações locais da imagem quanto informações do algoritmo de rastreamento.

Os testes experimentais comprovaram o bom desempenho do nosso algoritmo, o UFT, que se mostrou mais robusto às transformações bruscas sofridas pela imagem, obtendo estimativas mais precisas e detectando e descartando anomalias. Nosso método possui um desempenho equivalente ao KLT padrão quando as sequências são relativamente simples, ou seja, quando possuem apenas movimentos contínuos, sem nenhuma transformação brusca. Obviamente, nestes casos nosso método apresenta a desvantagem de ter uma demanda computacional maior que a do KLT.

Através da forma de representação utilizada, é possível determinar não apenas o erro absoluto, mas também o erro em cada direção, possibilitando o uso de uma medida de

minimização de erro mais adequada. Esta informação é importante em uma grande quantidade de aplicações, tais como o *bundle adjustment*, conforme comprovado neste trabalho.

As principais fontes de erro do método são equivalentes à do algoritmo KLT padrão: baixa qualidade da sequência de imagens (imagens tomadas de pontos de vista muito diferentes, os quadros possuem uma textura muito pobre, entre outros) e problemas numéricos (embora neste trabalho não encontramos problemas sérios de estabilidade).

Algumas das possíveis extensões deste trabalho são apresentadas a seguir:

- Neste trabalho, a transformação não-linear sofrida pelas características na SUT foi representada pelo algoritmo KLT. No entanto, qualquer outro algoritmo poderia ser utilizado, como por exemplo o SIFT [28].

- Aqui, trabalhamos apenas com rastreamento de características em 2D. Um possível foco de pesquisa é extender os resultados do método para rastreamento 3D.

Bibliografia

[1] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):175–188, 2002.

[2] A. Azarbayejani and A.P. Pentland. Recursive estimation of motion, structure and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, 1995.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[4] J. Bohg. Real-time structure from motion using kalman filtering. Master's thesis, Technische Universitat Dresden, 2005.

[5] L. Bretzner. Multi-scale feature tracking and motion estimation. Master's thesis, Stockholms Universitet, 1999.

[6] M. Brown and D.G. Lowe. Unsupervised 3d object recognition and reconstruction in unordered datasets. In *Proceedings of the 5th International Conference on 3D Imaging and Modelling*, pages 56–63, 2005.

[7] S. Carlsson and D. Weinshall. Dual computation of projective shape and camera positions from multiple images. *International Journal of Computer Vision*, 27(3):1–16, 1998.

[8] A.K.R. Chowdhury and R. Chellappa. Face reconstruction from video using uncertainty analysis and a generic model. *Computer Vision and Image Understanding*, 91(1-2):188–213, 2003.

[9] A.K.R. Chowdhury and R. Chellappa. Stochastic approximation and rate-distortion analysis for robust structure and motion estimation. *International Journal of Computer Vision*, 55(1):27–53, 2003.

[10] A.K.R. Chowdhury and R. Chellappa. An information theoretic criterion for evaluating the quality of 3d reconstructions from video. *IEEE Transactions on Image Processing*, 13(7):960–973, 2004.

[11] M. Dimitrijevic, S. Ilic, and P. Fua. Accurate face models from uncalibrated and ill-lit video sequences. In *Proceedings of Computer Vision and Pattern Recognition*, pages 1034–1041, 2004.

[12] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[13] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[14] P. Fua. Regularized bundle-adjustment to model heads from image sequences without calibration data. *International Journal of Computer Vision*, 38(2):153–171, 2000.

[15] A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto. Improving feature tracking with robust statistics. *Pattern Analysis and Applications*, 2:312–320, 1999.

[16] S. Goldenstein. A gentle introduction to predictive filters. *Revista de Informatica Teórica e Aplicada (RITA)*, 11(1):61–89, 2004.

[17] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins, 1996.

[18] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference*, pages 147–151, 1988.

[19] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[20] Oliensis J. A critique on structure-from-motion algorithms. *Computer Vision and Image Understanding*, 80(2):172–214, 2000.

[21] H. Jin, P. Favaro, and S. Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *Proceedings of the International Conference on Computer Vision*, pages 684–689, 2001.

[22] S. Julier and J. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, Dept. of Engineering Science, University of Oxford, November 1996.

[23] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proceedings of the International Symposium on Optical Science, Engineering and Instrumentation*, 1997.

[24] S. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of IEEE*, 92(3):401–422, 2004.

[25] L. Kitchen and A. Rosenfeld. Gray-level corner detection. *Pattern Recognition Letters*, pages 95–102, 1982.

[26] A. Lanitis, C.J. Taylor, and T.F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, 1997.

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

[28] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[29] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[30] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3D Vision - From Images to Geometric Models*. Springer, 2004.

[31] D. Marr, S. Ullman, and T. Poggio. Bandpass channels, zero-crossings, and early visual information processing. *Journal of the Optical Society of America*, 69:914–916, 1979.

[32] H. P. Moravec. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. PhD thesis, 1980.

[33] K. Nickles and Seth Hutchinson. Measurement error estimation for feature tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3230–3235, 1999.

[34] F. Pighin, R. Szeliski, and D.H. Salesin. Modeling and animating realistic faces from images. *International Journal of Computer Vision*, 50(2):137–154, 2004.

[35] P.A. Rautenbach. Facial feature reconstruction using structure from motion. Master's thesis, Universidade de Stellenbosch, 2005.

[36] Y. Rui and Y. Chen. Better proposal distributions: Object tracking using unscented particle filter. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 786–793, 2001.

[37] J. Shi and C. Tomasi. Good features to track. In *Proceedings of Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[38] S. Srinivasan. Extracting structure from optical flow using the fast error search technique. *International Journal of Computer Vision*, 37(3):203–230, 2000.

[39] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[40] P.H.S Torr, A. Zisserman, and S. Maybank. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pages 298–372, 1999.

[41] P.H.S Torr, A. Zisserman, and Maybank S. Robust detection of degeneracy. In *Proceedings of the International Conference on Computer Vision*, pages 1037–1044, 1995.

[42] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[43] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. Technical Report CUED/F-INFENG/TR380, Cambridge University, August 2000.

[44] P. Viola and M.J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[45] E.A. Wan and R. van der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control*, 2000.

[46] E.A. Wan and R. van der Merwe. *Kalman Filtering and Neural Networks*. Wiley Publishing, 2001.

[47] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–198, 1998.

[48] Z. Zhang, R. Deriche, O. D. Faugeras, and T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995.

[49] Z. Zhang, Z. Liu, D. Adler, M.F. Cohen, E. Hanson, and Y. Shan. Robust and rapid generation of animated faces from video images: A model-based modeling approach. *International Journal of Computer Vision*, 58(2):93–119, 2004.