UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Modelos 3D densos a partir de imagens com sobreposição parcial: factorização com dados desconhecidos

Rui Filipe Cardoso Guerreiro
(Licenciado)

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador: Doutor Pedro Manuel Quintas Aguiar

Júri

Presidente: Doutor João Paulo Salgado Arriscado Costeira

Vogais: Doutor Helder de Jesus Araújo
Doutor Pedro Manuel Quintas Aguiar

Setembro de 2004

RESUMO

Este trabalho aborda o problema de estimar estrutura tridimensional a partir de movimento (*structure from motion* (SFM), na literatura anglo-saxónica), com informação desconhecida. Neste problema, estima-se o modelo tridimensional de uma cena estática e o movimento da câmara de filmar a partir de uma sequência de video com oclusão e inclusão. A oclusão e inclusão ocorre quando partes de objectos deixam de aparecer ou passam a aparecer nas imagens, respectivamente.

Apesar do método de factorização de Tomasi e Kanade ser muito usado pela comunidade de visão por computador para resolver o problema *structure from motion*, este não consegue lidar com oclusão e inclusão, o que limita a sua aplicação prática e provoca um aumento significativo na complexidade computacional da solução. Nesta tese descreve-se um novo estimador de informação desconhecida em matrizes singulares, que estima as trajectórias completas dos pontos a partir das trajectórias incompletas observadas, eliminando o problema da oclusão e inclusão e reduzindo consideravelmente a complexidade computacional necessária.

O estimador de informação desconhecida em matrizes singulares duma dada característica é constituído por um algoritmo que calcula uma estimativa inicial da matriz completa e por dois algoritmos iterativos óptimos que refinam esta estimativa. Um destes é um algoritmo *Expectation-Maximization* e o outro implementa o *power method* com informação desconhecida. Estes algoritmos estimam a matriz completa de característica deficiente que melhor aproxima a matriz de observações com a mesma característica mas com entradas desconhecidas. Estes algoritmos permitem usar o método de factorização de Tomasi e Kanade em sequências de video reais em que existe muita oclusão e inclusão.

Nesta tese conclui-se que o estimador de informação desconhecida em matrizes singulares apresenta importantes vantagens face aos algoritmos existentes, tendo em conta a qualidade da estimação, custo computacional e generalidade. O uso deste método para eliminar o problema da oclusão permite a estimação do modelo tridimensional óptimo, tendo em conta os dados disponíveis.

De forma a facilitar a estimação de movimento nas imagens é vulgar considerar uma aproximação de primeira ordem em que se assume que o movimento na imagem pode ser aproximado por uma homografia. Nesta tese é também descrito um novo estimador de homografia. Implementou-se também um estimador de correspondências, de fluxo óptico baseado em correlação, com uma estratégia de multi-resolução.

Palavras-chave: *Structure from motion*, oclusão, inclusão, homografia, correspondências, observações incompletas.

ABSTRACT

This thesis addresses the problem of estimating 3D structure from motion with missing data. In this problem, I estimate the 3-D model of a static scene and the 3-D motion of the video camera from a video sequence with occlusion and inclusion. Occlusion and inclusion occur when parts of objects stop appearing or start appearing on the images, respectively.

Although Tomasi and Kanade's factorization method is used extensively by the computer vision community to estimate structure from motion, it cannot handle occlusion and inclusion, which limits its practical use e causes a significative increase in computational complexity. In this thesis, I describe a new estimator of missing data in rank deficient matrices, that estimates complete point trajectories from incomplete observed trajectories, therefore eliminating the problem of occlusion and inclusion and reducing required computational complexity.

The estimation of missing data in rank deficient matrices consists of an algorithm that computes an initial estimate of the complete matrix and two iterative optimum algorithms that refine this estimate. One implements an Expectation-Maximization algorithm, and the other implements the Power Method algorithm for missing data. These algorithms estimate the complete rank deficient matrix that best approximates the observation matrix with the same rank and missing data. These algorithms allow the use of Tomasi and Kanade's factorization method in real video sequences with a lot of occlusion and inclusion.

In this study, I conclude that the missing data estimator in rank deficient matrices has important advantages regarding existing algorithms, in such aspects as estimation quality, computational cost and generality. This method eliminates the occlusion problem in a way that it estimates the optimum 3-D model with observed data.

In order to facilitate motion estimation in the video sequence, it is common to consider a first-order approximation in which we assume that the image motion can be approximated by a homographic transformation. In this thesis, I also describe a new homography and correspondence estimator, based on correlation optical flow, with a course-to-fine strategy.

Key-Words: Structure from motion, occlusion, inclusion, homography, correspondences, missing observations.

Conteúdo

Lista de Figuras

Capítulo 1

INTRODUÇÃO

1.1 Motivação e objectivos

Há muito que o Homem sonha em construir uma máquina tão inteligente como o próprio Homem. Um dos seus requisitos seria a capacidade de processar dados visuais e conseguir extrair informação de diversos níveis conceptuais, desde o simples reconhecimento de contornos ao complexo reconhecimento de objectos, passando pelo reconhecimento de caracteres, a interpretação de imagens médicas, etc. O campo da ciência que estuda este assunto denomina-se *visão por computador*.

A percepção de tridimensionalidade é um tópico de grande interesse por parte da comunidade de visão por computador. Existem vários métodos que inferem a estrutura tridimensional a partir de uma só imagem, usando informação tal como sombras (*shape from shading*) [15], textura (*shape from texture*) [9], fronteiras (*shape from boundary*) [44] e desfocagem (*depth from focus*) [43]. Tornou-se evidente que estes métodos não conseguem produzir estimativas fiáveis da estrutura trimensional a partir de imagens de ambientes reais sem restrições [32].

Foram desenvolvidos outros métodos que usam duas ou mais imagens da mesma cena para reconstruir, de forma mais fiável, a respectiva estrutura estática tridimensional. Na secção 1.3 faz-se uma revisão destes métodos. Estes métodos baseiam-se na análise dos movimentos estimados nas imagens pelo que são conhecidos por *structure from motion* (SFM). Em SFM, pretende-se estimar a estrutura 3D da cena filmada e o movimento da câmara a partir duma sequência de duas ou mais imagens obtidas de posições e ângulos desconhecidos.

Em 1979, Ullman mostrou que é possível estimar a estrutura e o movimento tridimensional duma configuração rígida de quatro pontos a partir de três projecções ortográficas desses pontos usando uma câmara estacionária [39]. No entanto, desde então a maioria dos investigadores em SFM optaram por usar o modelo de projecção perspectivo, processando apenas um par de imagens. Apesar desta abordagem garantir a existência e a unicidade da solução, excepto em configurações singulares, o uso de apenas duas imagens mostrou ser altamente sensível ao ruído na imagem, especialmente quando o objecto está muito longe da câmara. A investi-

gação mais recente tem sido orientada para o uso de sequências de video mais longas. As abordagens adoptadas fazem uso de filtros de Kalman extendidos e métodos de optimização não lineares de elevado peso computacional.

No principio dos anos 90, Tomasi e Kanade recuperaram a formulação original de Ullman e introduziram um método computacionalmente simples que permite resolver o problema SFM com múltiplas imagens sem recorrer a optimização não linear, usando o modelo de projecção ortográfico [37]. Este método foi extendido para o modelo de projecção paraperspectivo [29], projecção perspectiva [35], análise de movimento de linhas [25], de superfícies [2] e análise de múltiplos movimentos [8].

O método de factorização de Tomasi e Kanade tem sido reconhecido como um método fundamental para estimar a estrutura a partir do movimento. Este método usa uma matriz de observação de característica 4 formada pelas coordenadas das projecções bidimensionais no plano da imagem dos pontos notórios. As matrizes de movimento e de forma obtém-se factorizando a matriz de observações e aplicando determinadas restrições à estrutura da matriz de movimento.

A oclusão é um fenómeno em que porções da estrutura tridimensional dos objectos deixam de aparecer nas imagens, e a inclusão é outro fenómeno em que porções da estrutura tridimensional que não apareciam nas imagens passam a aparecer -nesta tese os dois fenómenos são designados simplesmente como *oclusão*. Estes fenómenos surgem naturalmente com o movimento da câmara, ou porque as zonas passam a estar, ou estavam, ocultas por detrás de outras ou, simplesmente, porque saem ou entram no plano da imagem. Estes fenómenos aumentam consideravelmente a dificuldade na estimação de SFM, uma vez que as observações das trajectórias dos pontos resultam incompletas. Daí dar-se o nome SFM com informação desconhecida ao problema SFM discutido anteriormente, quando existe oclusão. No método de factorização de Tomasi e Kanade, as oclusões traduzem-se numa matriz de observação que contém valores desconhecidos, correspondentes às trajectórias que ficaram por seguir. Uma vez que não é possível factorizar uma matriz com valores desconhecidos, não é possível aplicar o método. Dada a frequência com que se dão os fenómenos de oclusão (agravados pelo facto dos algoritmos que efectuam a correspondência falharem frequentemente), é pouco eficiente usar um método que estima SFM sem lidar com oclusões. Os próprios Tomasi e Kanade apresentaram um algoritmo iterativo para lidar com oclusões [37], mas este é de tal forma pesado computacionalmente e propenso a erros que impossibilita o uso de sequências de video longas [18].

Nesta tese são propostos dois algoritmos iterativos óptimos e complementares no seu modo de funcionamento que, em condições bem determinadas, convergem para a matriz de observação completa que melhor aproxima todos os dados conhecidos. É também apresentado um algoritmo sub-óptimo que estima de forma eficiente a matriz de observação completa a partir de uma matriz de observação incompleta. Apesar dos algoritmos apresentados terem sido desenvolvidos no contexto do método de factorização de Tomasi e Kanade, que lida com modelos de projecção lineares, tais

como o modelo ortográfico e o modelo de paraperspectiva, estes algoritmos podem anteceder qualquer método linear de SFM que não consiga lidar com oclusões.

A generalidade dos algoritmos de factorização propostos nesta tese tornam-nos também adequados para o caso geral em que se pretende estimar informação desconhecida em matrizes singulares de qualquer característica, de qualquer problema de engenharia. De facto, abordagens recentes a diversos problemas de visão por computador tais como o reconhecimento de objectos [4, 40], aplicações em fotometria [33, 27, 5] e o alinhamento de imagem [45, 46] estimam sub-espaços de baixa dimensão a partir de observações ruidosas, usando decomposição em valores singulares [11]. Na prática, a matriz de observação destes problemas pode estar incompleta por não terem sido observadas todas as suas entradas, o que limita a utilização das aplicações. Os novos algoritmos aqui apresentados permitem estimar os valores das entradas desconhecidas, completando as matrizes de observação e permitindo uma utilização prática sem restrições das aplicações em questão.

Estes novos algoritmos aqui apresentados são integrados com um novo método de SFM baseado em fluxo óptico que suporta oclusão. Um dos problemas cruciais dos algoritmos de SFM é a estimação das correspondências entre as projecções bidimensionais de cada ponto tridimensional. Uma forma habitual de facilitar esta estimação consiste em efectuar uma primeira aproximação à estimação, considerando que a cena tridimensional pode ser aproximada por um plano e estimando a transformação homográfica entre cada par de imagens. A homografia é uma transformação projectiva bidimensional que pretende alinhar duas imagens duma região planar, obtidas de diferentes posições. A aproximação de que a cena tridimensional pode ser aproximada por um plano faz sentido porque se assume que a distância entre a câmara e a cena é muito maior do que o relevo tridimensional dos objectos filmados. Além disso, a aproximação é também justificada pelo facto de, se o centro óptico da câmara estiver sempre no mesmo ponto do espaço, a relação entre as imagens é dada exactamente por uma transformação homográfica, qualquer que seja a estrutura 3D da cena. Nesta tese é apresentado um novo e robusto estimador de homografia, iterativo, baseado em fluxo óptico de correlação, de resolução *sub-pixels* e com uma estratégia de processamento multi-resolução, que permite alinhar duas ou mais imagens de superficies planares, ou estimar uma aproximação das correspondências entre as projecções bidimensionais de pontos tridimensionais. O estimador de homografias tem importantes vantagens face aos métodos existentes por não usar pontos notórios e ser computacionalmente mais simples do que os restantes métodos baseados em fluxo óptico.

Uma vez que se pretendem modelos densos da estrutura tridimensional, difíceis de obter com uma estratégia baseada em pontos notórios, é apresentado um novo estimador de correspondências de fluxo óptico baseado em correlação, também com uma estratégia de multi-resolução. O uso de fluxo óptico permite calcular um mapa 3D denso porque que estima as correspondências de todos os *pixels* das imagens.

1.2 Abordagem

O método de SFM proposto nesta tese está dividido, como habitual, em dois passos distintos: o passo em que se estima a correspondência entre as projecções dos mesmos pontos tridimensionais ao longo das imagens; e o passo em que se calcula a estrutura tridimensional da cena e o movimento da câmara.

A estimação das correspondências baseia-se em fluxo óptico de correlação, i.e., estima-se a correspondência entre todos os *pixels* de um par de imagens através da minimização duma medida de distância entre as intensidades dos *pixels* correspondentes. De forma a melhorar e facilitar o processo de correspondência entre duas imagens consecutivas, este processo usa dois algoritmos em sequência: primeiro, estima-se a homografia que melhor descreve o movimento do padrão de intensidade entre as duas imagens; depois, estima-se o deslocamento dos padrões de intensidade nas imagens. Ambos os algoritmos usam um processamento iterativo multi-resolução denominado *coarse-to-fine*, em que as imagens são analisadas em várias resoluções: nas resoluções mais baixas (*coarse*) é feita uma correspondência mais grosseira, que é melhorada com o detalhe dos níveis de resolução mais altos (*fine*).

No segundo passo do algoritmo de SFM são propostos novos algoritmos que estimam as trajectórias completas de cada ponto a partir de trajectórias incompletas, o que permite reduzir o problema SFM com informação desconhecida, de grande complexidade, ao problema bastante mais simples de SFM com informação completa. Os algoritmos apresentados estimam a matriz de observação completa de característica 4 que melhor aproxima a matriz conhecida, mas incompleta, minimizando o funcional que mede a distância entre as matrizes nas posições conhecidas. Mostrar-se-à que a matriz que minimiza este funcional é a matriz de observação desejada. Estes algoritmos tomam partido da redundância nas trajectórias e combinam uma estimativa sub-óptima da matriz de observação completa, com um de dois algoritmos iterativos que refinam essa estimativa inicial de forma óptima. A estimativa inicial combina o espaço das linhas com o espaço das colunas da porção conhecida da matriz de observação para estimar uma matriz completa.

O primeiro algoritmo iterativo é baseado no método *Expectation-Maximization* (EM), que tem sido usado com muito sucesso na área do processamento de sinal [21]. Este algoritmo funciona em dois passos: *i*) no passo *expectation* estimam-se os dados desconhecidos a partir da estimativa anterior dos parâmetros; *ii*) no passo de *maximization* estimam-se os parâmetros a partir dos dados completos.

O segundo algoritmo iterativo, denominado *Row-Column* (RC), consiste numa generalização do *power method* [11] e inclui a generalização da pseudo-inversa de Moore-Penrose [11] para o caso em que existe informação desconhecida. O *power method* tem sido muito usado como alternativa à SVD e para calcular aproximações singulares de matrizes, quando toda a informação é conhecida. Este algoritmo funciona também em dois passos e estima alternadamente o espaço das linhas e o espaço das colunas da matriz a determinar.

Uma vez determinadas as trajectórias completas de cada ponto, usa-se o método de factorização de Tomasi e Kanade, com um modelo de projecção ortográfico, para obter a estrutura tridimensional da cena e o movimento da câmara.

1.3 Estrutura 3D a partir de imagens 2D

Nesta secção revêem-se as estratégias conhecidas para estimar SFM.

Os métodos que recuperam a estrutura tridimensional a partir do movimento detectado em duas ou mais imagens podem ser divididos em várias categorias, consoante sejam ou não conhecidas as características e o posicionamento das câmaras, e consoante o número de imagens por cena. Assim, existem métodos de estereo binocular: para duas imagens obtidas de posições e ângulos conhecidos, com câmaras calibradas [10, 20]; estereo multi-vista: para várias imagens obtidas de posições e ângulos conhecidos, com câmaras calibradas [19]; e *structure from motion* (SFM): para duas ou mais imagens obtidas de posições e ângulos desconhecidos, com câmaras não calibradas, onde se obtém também o movimento de translação e rotação da câmara. A designação *structure from motion* surge do facto de se inferir a estrutura tridimensional da cena e o movimento da câmara somente a partir do movimento estimado nas imagens, uma vez que a única informação conhecida são as intensidades das imagens duma sequência de video. Nesta tese será apenas considerado SFM.

Os métodos de SFM podem usar vários mecanismos para inferir a estrutura tridimensional a partir do movimento estimado entre as imagens. Estes mecanismos são: a stereopsis, i.e., o deslocamento lateral de pontos dos objectos nas imagens; o parallax de movimento, i.e., o diferente movimento dos pontos em relação a um ponto de referência consoante a sua profundidade [6]; e fenómenos de expansão (*looming*) e contracção devidos a efeitos de perspectiva [32]. A maioria dos métodos de SFM, em que se incluem os métodos estudados nesta tese, faz uso do mecanismo de stereopsis descrito acima, uma vez que a geometria deste mecanismo é bem conhecida e permite obter estimativas de alta qualidade da estrutura tridimensional da cena e do movimento da câmara.

Os métodos de SFM funcionam essencialmente em dois passos distintos: a estimação do movimento na imagem e a obtenção da estrutura tridimensional e do movimento da câmara. Para estimar o movimento na imagem é necessário estimar onde é que os mesmos pontos tridimensionais reais, que pertencem a objectos tridimensionais, estão projectados bidimensionalmente nas várias imagens obtidas. Este é o problema da estimação da correspondência entre pontos duma imagem e os mesmos pontos das imagens seguintes. Conhecendo o movimento de pontos das imagens, i.e., as suas trajectórias no plano da imagem, o segundo passo consiste em determinar a estrutura tridimensional e o movimento tridimensional da câmara que melhor descreve a maioria dos movimentos estimados.

Em geral, os métodos de SFM podem ser divididos em dois grandes conjuntos de métodos tendo em conta a forma como é resolvido o problema da correspondência: os primeiros baseiam-se em pontos notórios (*features*) e os segundos baseiam-se em fluxo óptico. Os métodos baseados em pontos notórios apenas procuram efectuar correspondência entre pontos *de interesse* ou *cantos*, em princípio fáceis de determinar em imagens consecutivas, obtendo-se uma estimativa de alta qualidade da estrutura tridimensional, apesar de esparça, e principalmente do movimento da câmara [38]. Sabendo com rigor o movimento da câmara, usam-se algoritmos de estereo multi-vista para obter uma estimativa densa, i.e. completa, da estrutura tridimensional. Os métodos baseados em fluxo óptico estimam directamente a estrutura tridimensional e o movimento da câmara, usando quantidades mensuráveis em todos os *pixels* para procurar estimar a correspondência entre todos os *pixels* das imagens [3, 16]. Os métodos baseados em fluxo óptico podem-se dividir em métodos baseados em gradiente (*differential* ou *gradient-based*), baseados em correlação (*region based matching* ou *correlation*), baseados em modelos (*direct model-based* ou *area-based regression*), baseados em energia (*energy-based methods*) e baseados na fase (*phase-based methods*) (ver [3], para mais detalhes sobre estes métodos).

Os métodos baseados em pontos notórios têm várias vantagens face aos métodos de fluxo óptico. Os pontos notórios exibem uma grande invariância fotométrica, i.e., mesmo com grandes mudanças na iluminação entre duas imagens, os pontos notórios detectados ainda irão corresponder ao mesmo ponto 3D. Outras características como linhas notórias (linhas fáceis de corresponder) exibem também uma grande invariância geométrica, umas vez que as linhas projectam-se em linhas, sendo portanto invariantes em transformações projectivas. Nos métodos baseados em fluxo óptico, caso seja usado o pressuposto habitual de que o brilho da projecção dos mesmos pontos 3D é constante, mudanças na iluminação entre duas podem dar origem a correspondências erradas. Nos métodos de fluxo óptico, a formulação duma função de verosimilhança não é directa, sendo difícil modular o ruído na correspondência e, consequentemente, efectuar uma minimização global do erro. Do ponto de vista do custo computacional, os métodos baseados em pontos notórios permitem calcular estimativas de alta qualidade do movimento da câmara com um custo computacional baixo, por efectuar a correspondência de poucos pontos, mas detectados com grande precisão [38]. O baixo custo computacional permite o processamento de sequências de video com muitas imagens. Se todos os *pixels* da imagem fossem usados, o custo computacional aumentaria bastante e a solução seria menos rigorosa, uma vez que muitos dos *pixels* usados teriam erro na correspondência, por corresponderem a zonas pouco texturadas onde a correspondência é ambígua.

Por outro lado, os métodos baseados em fluxo óptico têm várias vantagens face aos métodos baseados em pontos notórios. Enquanto é necessário encontrar pontos distintos e facilmente identificáveis para usar os métodos baseados em pontos notórios, é apenas necessário que haja alguma textura para que os métodos de fluxo óptico funcionem correctamente. Os métodos baseados em fluxo óptico cal-

culam directamente a estimativa da estrutura tridimensional e do movimento da câmara, enquanto os métodos baseados em pontos notórios necessitam de um pós-processamento de complexidade e ambiguidade semelhantes aos métodos de fluxo óptico. A diferença fundamental prende-se com o uso do movimento conhecido da câmara para deduzir e restringir a área de procura das correspondências às linhas epipolares obtidas. Por outro lado, encontrar as correspondências entre as projecções no plano da imagem de pontos notórios tridimensionais, apesar de em principio ser uma tarefa relativamente simples, é um problema em aberto de grande complexidade [16].

1.4 Trabalho relacionado

Nesta secção analisam-se as diferenças entre os métodos propostos nesta tese e as soluções existentes na literatura.

O algoritmo iterativo apresentado por Tomasi e Kanade para resolver o problema das oclusões [37] é semelhante ao algoritmo que calcula a estimativa inicial neste método. No entanto, a solução de [37] é computacionalmente muito mais pesada do que a que se propõe, por requerer um grande número de cálculos de SVD [18]. Foi também proposto um algoritmo iterativo bidireccional em [23], semelhante ao algoritmo EM usado no método aqui proposto. Por tratar a translação da câmara à parte e necessitar de efectuar processamento intermédio, não usufrui da generalidade, simplicidade e rapidez do algoritmo EM aqui apresentado. O algoritmo RC proposto, implementa a generalização do *power method* e da pseudo-inversa de Moore-Penrose para o caso em que existe informação desconhecida. Este algoritmo é semelhante ao algoritmo de Wiberg [41] e está relacionado com o algoritmo usado em [34] para modular poliedros. A integração duma boa estimativa inicial com algoritmos iterativos de refinamento, como aqui se propõe, evita problemas como a propagação de ruído (como acontece em [37]) ou a divergência de métodos iterativos (como acontece em [23]).

Todos os métodos referidos acima funcionam em lote, i.e., só depois de ter sido recolhida toda a informação da sequência de vídeo se pode resolver o problema *structure from motion*. A referência [26] propõe um método sequencial em que são calculadas estimativas para a estrutura 3D dos objectos filmados, que vão sendo refinadas sempre que se recebe uma imagem nova, o que permite processamento em tempo-real. A referência [7] propõe uma extensão do algoritmo em [26] que permite lidar com oclusão e inclusão. Os algoritmos de [26, 7] estimam directamente SFM, não resolvendo o problema de estimar a matriz de observações completa a partir de observações incompletas, como acontece nos algoritmos propostos nesta tese. Por esta razão, dão origem a piores estimativas da estrutura 3D do que o método proposto pois não têm em conta a rigidez da cena ao longo de todo o video.

1.5 Contribuições

Nesta tese são propostos algoritmos gerais de estimação de informação desconhecida em matrizes singulares. Estes transformam o problema SFM com informação desconhecida no problema SFM, o que permite usar o método de factorização de Tomasi e Kanade em sequências de video longas. Estes algoritmos combinam uma estimativa inicial sub-óptima da matriz de observação completa com um de dois algoritmos iterativos que refinam essa estimativa inicial. Um dos algoritmos iterativos é um algoritmo *Expectation-Maximization* (EM). O outro, denominado por *Row-Column* (RC), é uma generalização do *power method*.

É introduzido um método novo para estimar homografias baseado em fluxo óptico, com resolução *sub-pixels*. Este método baseia-se em correlação, usando um procedimento multi-resolução. É introduzido também um método novo para estimar a correspondência densa entre as projecções bidimensionais de pontos tridimensionais baseado em fluxo óptico de correlação com um procedimento multi-resolução.

O código destes algoritmos (desenvolvidos em $MatLab^{©}$) está disponível em [49]. Partes deste trabalho foram publicadas em [12, 13, 14].

1.6 Organização da tese

Esta tese está dividida em duas partes. Na primeira parte, capítulos 2 a 6, faz-se a descrição dos métodos desenvolvidos. Na segunda parte, capítulos 7 a 9, faz-se uma análise experimental dos mesmos e descrevem-se aplicações práticas que demonstram a relevância das contribuições desta tese.

No capitulo 2 descreve-se o novo algoritmo de correspondência denso proposto, baseado em fluxo óptico de correlação. No capitulo 3 é apresentado o novo algoritmo de estimação de homografia, também baseado em fluxo óptico de correlação. No capítulo 4 descreve-se o método de factorização de Tomasi e Kanade. No capítulo 5 introduz-se o problema *structure from motion* com informação desconhecida e propõe-se o novo algoritmo para calcular uma estimativa sub-óptima da matriz de observação completa. No capítulo 6 derivam-se os dois novos algoritmos iterativos que calculam, de forma óptima, estimativas da matriz de observação completa, a partir de trajectórias incompletas dos pontos.

Na segunda parte, no capitulo 7 é feita uma análise experimental do novo estimador de correspondência e do novo estimador de homografias. No capítulo 8 é feita uma análise experimental do método de factorização de Tomasi e Kanade e dos novos algoritmos que estimam informação desconhecida em matrizes singulares. No capítulo 9 são usados os novos algoritmos em duas aplicações práticas: a estimação de modelos 3D para realidade virtual e a compressão de video, em que se conseguiu uma taxa de compressão de cerca de 1000.

O capítulo 10 conclui a tese.

Parte I

Metodologia

Capítulo 2

MAPA DENSO DE CORRESPONDÊNCIAS

2.1 Formulação do problema

Neste capítulo é descrito um novo método para efectuar o passo de correspondência de um algoritmo de SFM, que usa fluxo óptico baseado em correlação.

Quando se fotografa ou filma um cenário tridimensional real, o resultado é uma ou várias projecções bidimensionais dessa realidade no plano da imagem. Ao efectuar a projecção bidimensional referida, em que se representa informação tridimensional com apenas duas dimensões, introduz-se uma ambiguidade que impossibilita a determinação das coordenadas tridimensionais exactas dos pontos filmados. De facto, tal como está ilustrado na figura 2.1, o ponto tridimensional p, de coordenadas $(x_{3Dp}, y_{3Dp}, z_{3Dp})$ no sistema de coordenadas da câmara e representado na imagem f nas coordenadas (x_{2Dp}, y_{2Dp}), podia estar localizado em qualquer ponto da recta que une a origem do centro focal ao ponto real, uma vez que a sua projecção bidimensional seria a mesma.

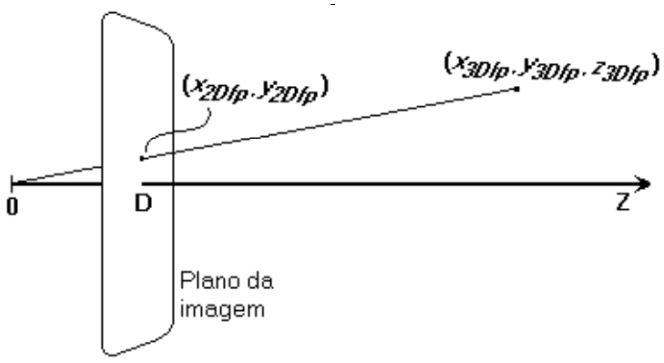


Figura 2.1: Ambiguidade quanto à localização de um ponto tridimensional real, dada a sua projecção bidimensional.

Conhecendo a projecção do mesmo ponto tridimensional em pelo menos duas imagens obtidas com câmaras em posições conhecidas e de distância focal, D, con-

hecida, é possível eliminar a ambiguidade anterior e determinar a localização exacta do ponto tridimensional.

Assim, o primeiro passo para resolver o problema de SFM consiste em determinar as coordenadas das projecções dos pontos tridimensionais, tendo como única pista as intensidades das imagens de uma sequência de video. Pretende-se, portanto, encontrar a transformação $\boldsymbol{C}$ que faz corresponder todas as coordenadas bidimensionais de uma imagem com a imagem seguinte, tal que ambas as coordenadas sejam projecções no plano da imagem do mesmo ponto tridimensional. Temos, matematicamente,

$$\begin{gathered}\boldsymbol{C} : [0,\ \#\boldsymbol{X} - 1] \times [0,\ \#\boldsymbol{Y} - 1] \longrightarrow [0,\ \#\boldsymbol{X} - 1] \times [0,\ \#\boldsymbol{Y} - 1] \\ (x_{f+1}, y_{f+1}) = \boldsymbol{C}(x_f, y_f)\end{gathered} , \qquad (2.1)$$

em que $\#\boldsymbol{X}$ e $\#\boldsymbol{Y}$ representam o número de *pixels* das imagens na horizontal e na vertical, respectivamente. Na figura 2.2, os pontos (x_{f+1}, y_{f+1}) e (x_f, y_f) representam pontos da imagem $f + 1$ e f da sequência de video, respectivamente.

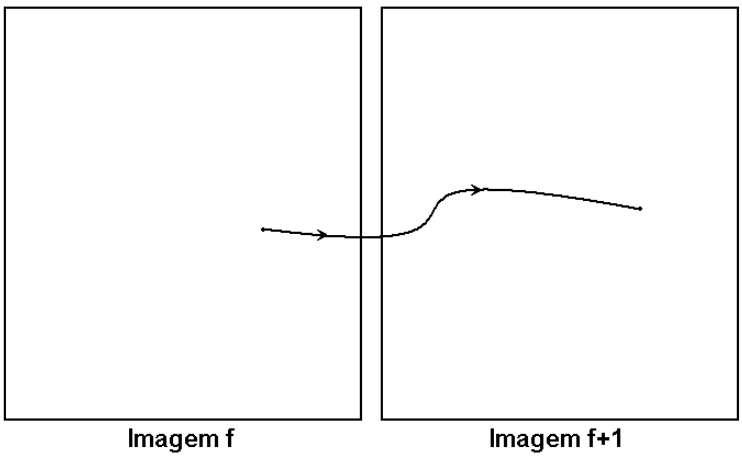


Figura 2.2: Correspondência entre os pontos (x_f, y_f) e (x_{f+1}, y_{f+1}), através da transformação $\boldsymbol{C}$.

2.2 Definições

Para determinar a correspondência entre os pontos de uma imagem com os pontos da imagem seguinte, são assumidos alguns pressupostos, entre os quais o pressuposto de brilho constante (*brightness constancy constraint*, na literatura anglo-saxónica). Este pressuposto, que é o ponto de partida para a maioria dos métodos baseados em fluxo óptico [3], consiste em considerar que a projecção bidimensional do mesmo ponto tridimensional tem a mesma intensidade em todas as imagens, o que permite definir à partida a transformação $\boldsymbol{C}$ como a transformação de que resulta a maior

semelhança entre as intensidades dos pontos. Este princípio torna-se grosseiro quando a região contém transparências, reflexões, sombras ou oclusão. Em casos em que estes fenómenos ocorrem com maior frequência ou um que as imagens são obtidas a partir de sensores com respostas cromáticas muito diferentes, justificam-se medidas de distância menos dependentes deste pressuposto [16].

Assume-se ainda outro pressuposto comum: a restrição de continuidade (*continuity constraint*), que afirma que os objectos tendem a ser suaves e contínuos e que, portanto, os cenários em que estes estão inseridos também variam suavemente. Este pressuposto não contempla descontinuidades no fluxo óptico resultantes de descontinuidades na cena, mas permite afirmar que as projecções dos pontos tridimensionais na segunda imagem estão distribuídas localmente de forma muito semelhante à da imagem original.

Assim, tendo em conta os pressupostos referidos acima, que são pressupostos clássicos nos métodos de fluxo óptico, é possível considerar que, caso existam duas regiões de *pequena* dimensão muito semelhantes nas duas imagens, estas são projecções das mesmas regiões tridimensionais. Como é habitual nos métodos de fluxo óptico por correlação, impõe-se que as *pequenas* regiões das imagens a comparar sejam quadradas e de dimensão $T \times T$, a que se denomina por *blocos*. Das duas imagens que se pretende fazer corresponder, a primeira é divida em blocos, pretendendo-se encontrar o deslocamento $d_b = (d_x, d_y)_b$ de cada bloco b que minimiza a medida de distância dada por

$$\boldsymbol{D}_b(d_x, d_y) = \sum_{i=0}^{T-1} \sum_{j=0}^{T-1} |\boldsymbol{I}_f(x_b + i, y_b + j) - \boldsymbol{I}_{f+1}(x_b + i + d_x, y_b + j + d_y)|. \quad (2.2)$$

Esta medida soma os erros absolutos e mede a distância entre o bloco b da imagem f e o bloco b da imagem $f + 1$, deslocado de d_b; x_b e y_b é o canto superior esquerdo do bloco a correlacionar na imagem f.

A imposição de que as regiões têm um tamanho pré-definido dá origem a um problema comum nos métodos de fluxo óptico por correlação: o problema generalizado da abertura (*generalized aperture problem*) [3], i.e., a escolha do tamanho dos blocos. Por um lado, é desejável que os blocos sejam *grandes* para que contenham suficiente textura para se conseguir uma boa estimativa da correspondência. Por outro lado, quanto maior for o bloco a corresponder, menos provável será o pressuposto de movimento único (*single motion assumption*) dentro do bloco, que advém implicitamente dos dois pressupostos referidos acima. Este pressuposto diz que numa região de análise existe apenas um movimento, que, no método aqui apresentado, se supõe que pode ser aproximado por translação. Okutomi e Kanade [28] propuseram um método que ajusta o tamanho dos blocos de forma a minimizar a incerteza nas estimativas.

A escolha do tamanho dos blocos nos métodos de fluxo óptico levanta um problema de difícil resolução e ainda em aberto relacionado com a ausência de textura. De facto, para que a medida de distância apresentada, a soma do módulo das diferenças

(ver equação (2.2)), ou outras medidas semelhantes tais como a soma do quadrado dos erros, funcionem correctamente, é necessário que os blocos a comparar tenham textura suficiente para que a superfície de erro destas medidas de distância tenham um mínimo vincado no valor do deslocamento em que as texturas coincidem. Na realidade, a ausência de textura nos blocos pode dar origem a estimativas erradas do deslocamento, uma vez que o pressuposto de brilho constante não se verifica em rigor devido às razões referidas anteriormente tais como alterações na iluminação de uma imagem para a seguinte, e também devido a distorções devidas à captação e digitalização das imagens por parte das câmaras fotográficas ou máquinas de filmar, tais como distorções geométricas, *scattering*, *blooming*, variações entre as diferentes células da câmara, *clipping* ou *wrap-around*, distorção cromática e efeitos de quantização [32]. A estratégia mais comum para lidar com a ausência de textura consiste em interpolar o deslocamento de regiões sem textura a partir do deslocamento de regiões com textura em seu redor, fazendo uso do pressuposto de continuidade referida acima.

2.3 Multi-resolução

No método de correspondência aqui proposto é usada também uma estratégia muito comum em métodos de correspondência baseados em fluxo óptico, denominada multi-resolução. Estes métodos processam as imagens a corresponder em vários níveis de resolução, da mais pequena para a maior, propagando as estimativas obtidas em resoluções inferiores para as resolução superiores. Este processamento permite algumas vantagens importantes face a um simples processamento das imagens no seu nível de resolução mais detalhado.

Os níveis de resolução a que se reduzem as imagens são, por ordem, 64×64 (*coarse*), 128×128, 256×256 e 512×512 (*fine*). (No método aqui proposto, as imagens da sequência de video são deformadas para a resolução mais alta de 512×512, de modo a permitir sempre resoluções inferiores múltiplas de 2, para facilitar o processamento). Cada *pixels* das imagens de menor resolução é dado pela média de 8×8, 4×4, 2×2 e 1×1 *pixels* da imagem original, respectivamente. Além disso, para cada nível de resolução, é necessário definir como parâmetro de entrada o tamanho do bloco (variável T introduzida equação 2.2), que é uma potência de 2, de forma a que toda a imagem seja contemplada.

O procedimento de multi-resolução tem várias vantagens. Uma vez que o nível de detalhe das imagens resultantes é muito baixo, as diferenças entre duas imagens, resultantes de rotações ou até pequenas violações dos pressupostos assumidos e descritos acima, são muito pequenas, o que permite determinar correctamente uma estimativa grosseira das correspondências entre as imagens, que servem de ponto de partida para as estimativas em níveis de resolução com maior detalhe. O facto das imagens terem uma resolução muito pequena faz com que os blocos sejam al-

tamente texturizados, o que facilita a obtenção de estimativas iniciais fiáveis para a correspondência. Nos níveis de resolução mais altos, os blocos, que são sempre pequenos, vão-se particularizando em zonas cada vez mais especificas da imagem, onde pode não haver suficiente textura para haver um mínimo pronunciado na superfície de distância. Além disso, o facto de cada *pixels* representar 8×8 *pixels* das imagens originais reduz grandemente ($8 \times 8 = 64$ vezes, no caso do algoritmo aqui proposto) a área de procura do mínimo da função de distância para cada bloco, e, consequentemente, reduz consideravelmente o esforço computacional.

Nos níveis de maior detalhe, e em particular no nível de maior detalhe, são usadas as estimativas de correlação grosseiras determinadas no nível de resolução anterior para restringir a área de procura na obtenção de uma estimativa mais fina, tendo em conta o maior nível de detalhe na imagem. No entanto, esta estratégia não elimina alguns problemas próprios dos altos níveis de resolução. Se nas imagens de menor detalhe existe uma grande tolerância face aos efeitos de perspectiva, rotações ou a pequenas violações dos pressupostos admitidos, nos níveis de maior detalhe esses efeitos (tal como os efeitos devidos à captação e digitalização, mencionados anteriormente) já não são desprezados e podem dar origem a estimativas ruidosas de correspondência. Além disso, para imagens com mais detalhe, a quantidade de blocos com textura suficiente para efectuar uma boa correspondência decresce consideravelmente, sendo necessário estimar o deslocamento desses blocos com base em algum pressuposto (através de interpolação, por exemplo, como foi referido acima, que assume o principio de continuidade), o que aumenta o ruído nas estimativas da correspondência. As estimativas grosseiras mas fiáveis obtidas nos níveis anteriores ajudam, no entanto, a restringir o erro.

2.4 Estimação de correspondências

Para cada nível de resolução e cada deslocamento, $d_b = (d_x, d_y)_b$, é calculada a distância entre cada bloco b da imagem f e os blocos respectivos na imagem $f + 1$ deslocados de d_b *pixels*, usando a medida de distância da equação 2.2. Para cada bloco em que a distância é menor do que um limiar definido para o erro, Th, verifica-se se esta correspondência, a que denominarei *ajuste*, é boa, i.e., se é um mínimo local prenunciado da distância entre os blocos. Para verificar se é um mínimo prenunciado da distância entre os blocos, é necessário calcular a distância entre o bloco b em questão da imagem f e os blocos em torno do deslocamento d_b, obtendo-se uma matriz de distâncias 3×3 tal como a que está indicada na figura (2.3).

Para que haja um mínimo pronunciado da distância entre blocos, o que passarei a designar como um *bom ajuste*, do bloco b no deslocamento $d_b = (d_x, d_y)_b$ é necessário que a distância para o deslocamento $(d_x, d_y)_b$ seja menor do que qualquer distância circundante. Para cada nível de resolução, o parâmetro D_{min} refere a diferença minima entre $\boldsymbol{D}_b(d_x, d_y)$ e as outras distâncias de forma a que se considere que este

(dx-1, dy-1)	(dx, dy-1)	(dx+1, dy-1)
(dx-1, dy)	(dx, dy)	(dx+1, dy)
(dx-1, dy+1)	(dx, dy+1)	(dx+1, dy+1)

Figura 2.3: Matriz de distâncias 3×3, em torno do deslocamento $d_b = (d_x, d_y)_b$.

deslocamento dá origem a um *bom ajuste*, para o bloco b referido. Uma vez que, para o mesmo bloco podem haver vários deslocamentos de que resultam *bons ajustes*, este método assume o deslocamento de que resulta o *ajuste* com menor distância, para cada bloco.

No nível de resolução mais baixo, o método desenvolvido testa os deslocamentos numa área de tamanho $(2 \times varre_y + 1) \times (2 \times varre_x + 1)$, centrada em $(0, 0)$. O método calcula os deslocamentos onde há *bons ajustes* e faz uma lista com estes deslocamentos. No nível de resolução acima, este método irá testar os deslocamentos numa área de tamanho $(2 \times varre_y + 1) \times (2 \times varre_x + 1)$ centrados nos deslocamentos estimados no nível de resolução anterior (multiplicados por 2, de forma a contabilizar o aumento da resolução para o dobro) e construir uma nova lista de deslocamentos onde houve *bons ajustes*. Este processo é repetido até ao último nível de resolução. Note-se ainda que os valores de $varre_x$ e $varre_y$ são diferentes consoante o nível de resolução, tendo-se definido que são grandes nos níveis de resolução mais baixos e muito pequenos nos níveis de resolução acima, de modo a serem apenas refinamentos da estimativa obtida no nível de resolução mais baixo. É desejável que haja apenas um refinamento da estimativa inicial porque as estimativas obtidas nos níveis de resolução mais baixos são de grande fiabilidade, uma vez que estas imagens têm muita textura, são menos sensíveis a efeitos perspectivos, efeitos de rotação e a pequenas violações dos pressupostos assumidos. Além disso, permitem uma eficácia muito maior do ponto de vista computacional.

Como foi referido, só os *bons ajustes* são considerados na lista dos deslocamentos e, em particular, só nos blocos em que houve um *bom ajuste* é que se considera que foi encontrada uma correlação. São excluídos, portanto, os casos em que os blocos não têm suficiente textura para se encontrar um mínimo vincado. Para que o resultado final seja uma estimativa de correlação densa e completa, procura-se estimar o deslocamento dos blocos com fraca textura no nível de resolução mais alto. Para determinar o deslocamento de um bloco com textura fraca, diz-se que o seu deslocamento é a média dos deslocamentos dos *bons ajustes* situados na sua vizinhança. Esta estratégia é, portanto, uma aplicação directa do principio da continuidade e assume que este bloco tem um deslocamento semelhante ao deslocamento dos *bons ajustes* na sua vizinhança.

É apenas no último nível de resolução que se decide qual é o deslocamento de

cada bloco de uma imagem para a outra e que se detectam oclusões. Diz-se que um bloco da imagem original foi ocultado quando a medida de distância entre esse bloco e os blocos na imagem final é grande, maior do que Th, considerando-se que não é possível determinar a sua correspondência para a imagem seguinte.

Capítulo 3

ESTIMAÇÃO DE HOMOGRAFIA

3.1 Formulação do problema

Quando a distância entre a câmara de filmar e a cena é muito maior do que o relevo tridimensional dos objectos filmados, pode-se considerar como primeira aproximação que a cena é aproximada por um plano, donde a relação entre as imagens é dada por uma transformação homográfica. Além disso, se ao longo duma sequência de video com objectos tridimensionais o centro óptico da câmara estiver sempre no mesmo ponto no espaço, a relação entre as imagens é dada também por uma transformação homográfica. Assim, é possível determinar uma estimativa de primeira ordem para a correspondência entre as duas imagens obtidas de pontos de vista diferentes estimando a transformação homográfica entre elas.

A transformação homográfica consiste no mapeamento de uma superfície planar em outra, em que cada coordenada do segundo plano relaciona-se com as coordenadas do primeiro através da expressão

$$\begin{cases} x_{f+1}(\phi, x_f, y_f) = \frac{ax_f+by_f+c}{gx_f+hy_f+1} \\ \\ y_{f+1}(\phi, x_f, y_f) = \frac{dx_f+ey_f+f}{gx_f+hy_f+1} \end{cases}, \tag{3.1}$$

em que os 8 parâmetros a a h não têm restrições (excepto $gx_f + hy_f + 1 \neq 0$), e definem o vector de parâmetros ϕ. Assim, a estimação da transformação homográfica entre duas imagens consecutivas consiste em determinar o vector ϕ que minimiza a distância

$$\min_{\phi} \sum_{x_f=1}^{\#\boldsymbol{X}} \sum_{y_f=1}^{\#\boldsymbol{Y}} |\boldsymbol{I}_f(x_f, y_f) - \boldsymbol{I}_{f+1}(x_{f+1}(\phi, x_f, y_f), x_{f+1}(\phi, x_f, y_f))|, \tag{3.2}$$

em que I_f e I_{f+1} representam as imagens f e $f+1$, respectivamente, duma sequência de video; e $\#\boldsymbol{X}$ e $\#\boldsymbol{Y}$ representam o número de *pixels* da imagem na coordenada x e y, respectivamente.

3.2 Alinhamento de duas imagens

O método aqui apresentado para estimar o vector de parâmetros ϕ da transformação homográfica usa uma estratégia baseada em fluxo óptico, assumindo também o pressuposto de brilho constante, explicado anteriormente. Usa também uma estratégia de multi-resolução semelhante à do método de correspondência discutido anteriormente, o que faz com que este método permita lidar com grande facilidade com imagens com grandes transformações perspectivas, como será discutido mais à frente. Este estimador começa também por re-amostrar as imagens originais para que estas tenham resolução 512×512, para permitir resoluções inferiores múltiplas de 2. As resolução usadas são crescentes e são, por ordem, 64×64, 128×128, 256×256 e 512×512. Uma vez que o vector ϕ a estimar é constituído por 8 parâmetros e a correspondência de cada ponto contribui com duas equações, tal como se vê a partir da equação (3.1), basta estimar a correspondência de um número mínimo de 4 pontos representativos. Os métodos baseados em fluxo óptico minimizam directamente a equação (3.2), o que é computacionalmente muito pesado [22]. O método aqui proposto divide as imagens em quatro blocos e estima o seu movimento para a imagem seguinte, melhorando as estimativas em cada iteração.

Assim, no nível de resolução mais baixo, a primeira imagem a corresponder, f, é dividida em quatro blocos com 1/4 do tamanho da imagem total, tendo metade do tamanho em cada direcção (portanto, com dimensão 32×32 no nível de resolução mais baixo), localizados nos cantos da imagem. Para cada um dos blocos b ($b\in[1..4]$), determina-se o deslocamento d_b de que resulta o mínimo duma função de distância tal como a soma do erro absoluto (definida de forma semelhante à da equação (2.2)) ou a soma do quadrado das diferenças. A partir dos deslocamentos estimados para os 4 blocos, é possível calcular uma estimativa para o vector de parâmetros ϕ que permita transformar a imagem $f+1$ de forma a aproximá-la à imagem f, considerando o ponto central de cada bloco como a sua coordenada representativa. De facto, a partir da equação (3.1), obtém-se, na forma matricial,

$$\begin{bmatrix} x_f & y_f & 1 & 0 & 0 & 0 & -x_{f+1}x_f & -x_{f+1}y_f \\ 0 & 0 & 0 & x_f & y_f & 1 & -y_{f+1}x_f & -y_{f+1}y_f \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} x_{f+1} \\ y_{f+1} \end{bmatrix} \Longleftrightarrow \tag{3.3}$$

$$\Longleftrightarrow \Psi\phi = p_b,$$

para cada bloco, b, para $b\in[1..4]$. As equações obtidas para os 4 blocos permitem

obter o vector de parâmetros ϕ, através de

$$\phi = \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \\ \Psi_4 \end{bmatrix}^{-1} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \tag{3.4}$$

em que a matriz a inverter é sempre não singular, uma vez que as posições centrais dos blocos não estão dispostas sobre uma linha, condição em que não é possível extrair qualquer informação acerca da homografia. Essa situação não ocorre no caso em questão porque os blocos estão localizados nos cantos da imagem.

Uma vez que se pretende aproximar a imagem $f+1$ à imagem f, as coordenadas x_{f+1} e y_{f+1} da equação (3.4) têm sempre o valor central de cada bloco b na sua posição original (por exemplo, para a resolução 512×512, os valores de (x_{f+1}, y_{f+1}) seriam $\{(128, 128), (128, 384), (384, 128), (384, 384)\}$) e as coordenadas x_f e y_f são as respectivas coordenadas x_{f+1} e y_{f+1} somadas dos deslocamentos estimados no passo de correspondência deste método. Desta forma, transformam-se as coordenadas x_f e y_f nas coordenadas x_{f+1} e y_{f+1} respectivas.

Uma vez determinada uma estimativa para o primeiro vector de parâmetros globais, ϕ_0, que é igual ao primeiro valor de ϕ estimado, é feita uma homografia na segunda imagem a correlacionar, $f+1$, com a estimativa mais recente do vector de parâmetros, de modo a aproximá-la à primeira imagem, f, e possibilitar uma melhor estimativa do deslocamento dos 4 blocos do método em iterações seguintes. De facto, se a estimativa de ϕ_0 fosse a ideal, a imagem $f+1$ após a homografia seria igual à imagem f nos pontos conhecidos. Após a transformação da segunda imagem a correlacionar, $f+1$, com o vector de parâmetros $\phi_{(t-1)}$ é repetida a divisão da imagem f em 4 blocos; repetida a estimação dos deslocamentos dos blocos da imagem f para a imagem $f+1$; e repetida a estimação de um novo vector de parâmetros ϕ entre a homografia da imagem $f+1$ transformada com o vector de parâmetros $\phi_{(t-1)}$, e a imagem f. Para estimar o vector de parâmetros total na iteração t deste método, $\phi_{(t)}$, a partir do vector total na iteração anterior, $\phi_{(t-1)}$, e a nova estimativa de ϕ, basta usar a equação da transformação homográfica em forma matricial, usando coordenadas homogéneas [42],

$$\begin{bmatrix} a_t & b_t & c_t \\ d_t & f_t & f_t \\ g_t & h_t & 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & f & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} a_{t-1} & b_{t-1} & c_{t-1} \\ d_{t-1} & f_{t-1} & f_{t-1} \\ g_{t-1} & h_{t-1} & 1 \end{bmatrix}. \tag{3.5}$$

Aplicando novamente uma transformação homográfica na imagem $f+1$ original com a nova estimativa do vector de parâmetros da homografia, $\phi_{(t)}$, obtém-se uma imagem ainda mais próxima da imagem f, repetindo-se o processo referido acima até que o deslocamento dos blocos entre a imagem $f+1$ transformada e a imagem f seja aproximadamente nulo. Nesse momento passa-se para o nível de resolução acima, onde se repete o mesmo processo, tendo como ponto de partida, $\phi_{(0)}$, a melhor

estimativa deste vector obtida no nível de resolução anterior, até chegar ao último nível de resolução, onde o detalhe da imagem permite obter um vector $\phi_{(k)}$ final com grande precisão.

Tal como no capítulo anterior, o uso duma estratégia multi-resolução tem várias vantagens face ao uso de uma resolução única. A correlação efectuada em baixa resolução é bastante insensível às diferenças resultantes de transformações perspectivas, rotações ou até pequenas violações dos pressupostos assumidos, o que permite determinar uma boa estimativa das correspondências entre os quatro blocos da imagens mesmo quando estas são bastante diferentes. Este facto é especialmente importante na primeira correspondência do primeiro nível de resolução, em que não existe nenhuma estimativa da homografia entre as imagens e estas são, em geral, bastante diferentes. No método de correspondência descrito no capítulo anterior, o facto das imagens terem uma resolução muito pequena também é importante porque os blocos são altamente texturizadas, o que evita a existência de superficies de erro sem um mínimo vincado. Neste estimador de homografia, uma vez que os blocos são sempre 1/4 da imagem, qualquer que seja o nível de resolução, a textura que estes contêm é sempre a mesma e em grandes quantidades, não havendo, portanto e em geral, o problema de não se encontrar um mínimo vincado na superfície de distância entre o bloco original da imagem $f+1$ transformada e os blocos em torno da sua posição original, na imagem f. Além disso, tal como no estimador de correspondências descrito no capitulo anterior, o uso de uma estratégia multi-resolução aumenta consideravelmente a eficiência computacional, uma vez que o deslocamento dos blocos de que resulta a melhor correlação só é grande, em geral, nas primeiras correlações do nível de resolução mais baixo, quando ainda não há uma estimativa de ϕ muito próxima do resultado ideal. Para os níveis de resolução mais altos, a área de procura do mínimo da distância é muito pequena pois está bastante próxima do resultado ideal.

3.3 Precisão *sub-pixels*

Uma vez que as imagens estão discretizadas no espaço, tendo uma resolução finita, os *pixels* de uma imagem f não se deslocam exactamente para as posições correspondentes aos *pixels* da imagem seguinte, $f+1$, mas deslocam-se, em rigor, para posições *entre* os *pixels*. Isto faz com que a intensidade de cada *pixels* da imagem f esteja distribuída pelos *pixels* em torno da posição para onde cada *pixels* se desloca, na imagem a correlacionar $f+1$. Os métodos de fluxo óptico com precisão *sub-pixels* estimam o deslocamento real de cada *pixels* da imagem f para a imagem $f+1$.

O método de estimação de homografia aqui apresentado permite implementar com grande naturalidade uma estratégia de resolução *sub-pixels*. Como é descrito acima, este método estima o vector de parâmetros da transformação homográfica determinando a melhor correspondência entre 4 blocos da imagem original f e a

imagem $f+1$. Após ter sido estimado o deslocamento de cada bloco da imagem f para a imagem $f+1$ transformada, consideram-se também os valores da distância em torno da posição de distância minima, obtendo-se uma matriz 3×3 para cada bloco. A partir da matriz 3×3 estimada é possível interpolar o deslocamento *aproximado* de cada bloco e, consequentemente, o vector de parâmetros *aproximado* para a transformação homográfica.

3.4 Comparação com outras abordagens

Nesta secção é feita uma comparação entre a estratégia adoptada no estimador de homografias aqui apresentado e os estimadores de homografia existentes. Em geral, para obter os oito parâmetros que compõem a transformação projectiva é necessário obter pelo menos quatro pontos representativos da transformação. Caso sejam usados mais do que quatro pontos procura-se encontrar a transformação que melhor adapta a maioria dos dados estimados. Para o efeito é necessário combinar um método de eliminação de *outliers*, i.e., de dados que não respeitam a transformação imposta pela maioria dos dados, tal como o algoritmo RANSAC, com métodos de minimização do erro, como o método dos mínimos quadrados.

Também nos estimadores de homografia existem dois grandes conjuntos de métodos de estimação dos pontos referidos: os que se baseiam no uso de pontos notórios e os de fluxo óptico [16]. As vantagens e desvantagens de uma abordagem face à outra são semelhante às do caso em que se considera que a cena filmada é tridimensional. Os métodos baseados em pontos notórios têm de encontrar pontos distintos e fáceis de determinar a correspondência, não sendo usada informação acerca do contexto de cada um desses pontos, que tem grande utilidade no aumento da eficácia do processo de correspondência. Os métodos baseados em fluxo óptico apenas necessitam de alguma textura para efectuar uma boa correspondência. No entanto, o que pode ser a principal desvantagem do uso de pontos notórios, que é a dificuldade de encontrar *bons* pontos notórios, pode tornar-se numa grande vantagem quando a textura da imagem desrespeita as aproximações iniciais de um algoritmo que procura estimar a homografia. Por exemplo, para efectuar a correspondência entre duas imagens consecutivas duma sequência de video em que se supõe que os objectos filmados são rígidos e estão dispostos sobre um plano, a presença de árvores a ondular ao vento, pessoas, etc, pode ser facilmente ignorado num sistema de pontos notórios (sendo considerados como *outliers*). Num sistema de fluxo óptico, este tipo de acontecimentos pode introduzir erro em toda a estimação ou pode dificultar o processo de eliminação de *outliers*. No entanto, a grande quantidade de textura que está presente nas imagens, que é ignorada pelos métodos baseados em pontos notórios, é a grande vantagem dos métodos de fluxo óptico, especialmente no caso da homografia, em que a imagem é modulada com poucos parâmetros, podendo-se obtê-los com grande rigor, se as imagens respeitarem o modelo assumido.

A maioria dos métodos de fluxo óptico que estimam a transformação homográfica dividem as imagens em blocos e usam o deslocamento de todos os blocos na equação (3.4) acima para estimar os oito parâmetros necessários. Esta abordagem dá origem a resultados bastante inferiores aos do algoritmo aqui apresentado. Como foi visto no capítulo anterior, a divisão da imagem em blocos de pequena dimensão faz com que muitos deles não tenham textura suficiente para se estimar correctamente o deslocamento, o que introduz ruído na estimação dos parâmetros. Além disso, o uso de medidas de distância que assumem que o movimento dos blocos pode ser aproximado por translação pode dar origem a estimativas incorrectas dos seus deslocamentos, o que também introduz ruído. Estes problemas não surgem no algoritmo aqui proposto porque, além dos blocos terem sempre 1/4 do tamanho da imagem, havendo sempre textura suficiente para se determinar uma boa estimativa do deslocamento, a transformação que é aplicada na segunda imagem a corresponder, $f+1$, faz com que os blocos a corresponder estejam muito próximos de uma imagem para a imagem seguinte. Além disso, este algoritmo tem também a vantagem de ser muito simples e de não necessitar de qualquer algoritmo de eliminação de *outliers*.

3.5 Alinhamento de várias imagens

Na secção 3.2 descreve-se um método que permite estimar o vector de parâmetros ϕ da homografia que mapeia a imagem $f+1$ na imagem f. Embora o alinhamento entre duas imagens seja suficiente para o problema particular de SFM, em que se pretende efectuar uma estimação de primeira ordem da correspondência entre duas imagens, existem outras aplicações em que é necessário estimar o alinhamento entre um conjunto de várias imagens duma sequência de video. Assim, para estimar o vector $\phi_{(f)}$ que efectua a transformação homográfica da imagem f para o plano da imagem 1, basta compor $\phi_{(f-1)}$ com a transformação ϕ entre as imagens $f-1$ e f, usando a equação (3.5).

No entanto, a experiência mostrou que, após algumas imagens da sequência de video, as transformações homográficas acumuladas dão origem a vectores $\phi_{(f)}$ com valores bastante elevados, que criam problemas de precisão, como será visto. Uma vez que os vectores ϕ estimados entre duas imagens contém algum erro (apesar de usarem a estratégia de precisão *sub-pixels* referida na secção 3.3), a composição deste vector com um vector $\phi_{(f)}$, com valores elevados, faz com que os pequenos erros do vector ϕ se transformem em erros muito grandes de que resultam transformações homográficas defeituosas. Esta situação existe em todos os métodos que compõem as estimações entre pares de imagens, criando uma estimação cada vez mais deficiente das transformações homográficas ao longo duma sequência de video, o que impossibilita o uso de sequências de video longas.

De forma a ultrapassar este problema, é feita uma alteração ao estimador de homografias aqui proposto. Como foi visto na secção 3.2, o vector de parâmetros ϕ da

homografia entre a imagem f e $f+1$ é determinado mantendo a imagem f constante e transformando continuamente a imagem $f+1$ com estimativas de ϕ, de forma a aproximá-la de f, nas posições conhecidas. A alteração efectuada a este algoritmo consiste em afectar a imagem f da estimativa global $\phi_{(f)}$ e procurar directamente a transformação homográfica $\phi_{(f+1)}$ que aproxima a imagem $f+1$, transformada com uma estimativa de $\phi_{(f+1)}$, da imagem f, transformada com o vector $\phi_{(f)}$. Esta alteração no algoritmo introduz um mecanismo de *feedback* em que se estima *directamente* o vector $\phi_{(f+1)}$ que melhor aproxima a imagem f transformada com com o vector $\phi_{(f)}$, no resultado global final.

No entanto, a alteração efectuada introduz outro problema. Quando a transformação aplicada à imagem f a deforma muito, a estimação do vector $\phi_{(f+1)}$ é dificultada face ao caso em que as imagens estão pouco modificadas. Para garantir a convergência do estimador e acelerar o processo de estimação é calculada uma estimativa inicial do vector $\phi_{(f+1)}$. Assim, começa-se por determinar o vector ϕ que define a homografia entre a imagem f e a imagem $f+1$ (da forma definida na secção 3.2) e diz-se que a primeira estimativa para $\phi_{(f+1)}$ é dada pela composição do vector $\phi_{(f)}$ com o vector ϕ, através da equação (3.5). Uma vez que esta estimativa inicial está, em teoria, bastante próxima do resultado final, espera-se uma convergência fácil e rápida do vector $\phi_{(f+1)}$.

Capítulo 4

ESTRUTURA 3D A PARTIR DE VIDEO USANDO FACTORIZAÇÃO MATRICIAL

4.1 Formulação do problema

Neste tese será usado o método de factorização matricial introduzido por Tomasi e Kanade [37]. Este método, descrito em detalhe neste capítulo, não consegue lidar com oclusão e inclusão, o que compromete a sua utilização na prática dada a frequência com que se dão estes fenómenos. Nos capítulos 5 e 6 é proposto um conjunto de novos algoritmos que permitem estimar as trajectórias completas a partir da informação conhecida, i.e., a partir das trajectórias incompletas, o que possibilita o uso prático e praticamente livre de restrições do método de factorização de Tomasi e Kanade.

No método de factorização de Tomasi e Kanade [37], a localização de P pontos seleccionados e seguidos ao longo de F imagens da sequência de vídeo, $\{(xfp, yfp)|f = 1, ..., F, p = 1, ..., P\}$, é armazenada numa matriz denominada por matriz de observação não-registada, $\boldsymbol{W}$. A designação *não-registada* indica que a matriz de observação não está normalizada de forma a que o centro de coordenadas dos pontos armazenados seja o ponto $(x, y) = (0, 0)$.

Assim, define-se a matriz de observação não-registada $\boldsymbol{W}$, de dimensão $2F \times P$, através de

$$\boldsymbol{W} = \left[\frac{\boldsymbol{X}}{\boldsymbol{Y}}\right], \tag{4.1}$$

em que $\boldsymbol{X}$ é uma matriz $F \times P$ que contém as coordenadas horizontais de cada ponto em cada imagem, x_{fp}, com uma linha por imagem e uma coluna por ponto. A matriz $\boldsymbol{Y}$ é definida de forma semelhante a $\boldsymbol{X}$, mas contendo as coordenadas verticais de cada ponto e em cada imagem, y_{fp}.

A restrição de que a estrutura tridimensional dos objectos que são filmados é estática e que, portanto, o movimento na imagem é apenas devido ao movimento da

câmara, faz com que a projecção no plano da imagem dos pontos do espaço apenas possua movimento de rotação, translação e perspectiva na imagem. Assim, fixando o centro de coordenadas do objecto no centróide dos pontos, é possível relacioná-lo com o centro de coordenadas da câmara de filmar em qualquer instante f, S_f, através de

$$\boldsymbol{S}_f^T = \boldsymbol{R}_{3Df}\boldsymbol{S}^T + \boldsymbol{T}_{3Df}.e_p, \tag{4.2}$$

em que $\boldsymbol{R}_{3Df}$ é uma matriz de rotação 3×3, $\boldsymbol{S}$ é a matriz de forma [37] $P \times 3$ que indica as coordenadas dos pontos em relação ao centro de coordenadas do objecto; $\boldsymbol{T}_{3Df} = [t_{xf}t_{yf}t_{zf}]^T$ representa a diferença, distribuída por cada ponto, entre a localização do centro de coordenadas dos pontos e o centro de coordenadas da câmara -a translação-; e $e_P = [1\ 1...1]$ é um vector com P uns. Assim, a relação entre as projecções bidimensionais no plano da imagem dos pontos, $\boldsymbol{W}_f$, e as respectivas coordenadas tridimensionais em relação ao centro de coordenadas do objecto é, para cada imagem f, dada por

$$\boldsymbol{W}_f = F(\boldsymbol{R}_{3Df}\boldsymbol{S}^T + \boldsymbol{T}_{3Df}.e_p), \tag{4.3}$$

em que $F(.)$ representa o modelo de projecção.

A figura 4.1 representa o modelo de projecção em perspectiva, semelhante à realidade, considerando que a câmara é descrita pelo modelo *pin-hole*.

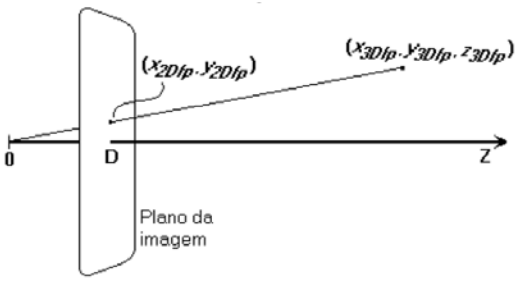


Figura 4.1: Modelo de projecção perspectivo.

Analisando a figura 4.1, verifica-se que a equação que rege este modelo é dada, para cada ponto, por

$$\boldsymbol{w}_{fp} = (x_{2Dfp}, y_{2Dfp}) = \left(\frac{D}{z_{3Dfp}}\right).(x_{3Dfp}, y_{3Dfp}) = \frac{1}{\lambda_{fp}(z_{3Dfp})}.(x_{3Dfp}, y_{3Dfp}), \tag{4.4}$$

em que D representa a distância focal; x_{3Dfp}, y_{3Dfp} e z_{3Dfp} representam as coordenadas do ponto no espaço tridimensional no sistema de coordenadas da câmara; e λ_{fp} representa a profundidade de projecção, tal como definida em [36]. O modelo de projecção ortogonal assumido em [37] e neste trabalho obtém-se de (4.4) considerando que a distância ao objecto é muito maior do que a distância focal, D, o que torna λ_{fp} aproximadamente constante, o que faz com que $w_{fp} \propto (x_{3Df}, y_{3Df})$,

aproximadamente. Assim, à excepção de uma constante multiplicativa, a equação (4.4) reduz-se a

$$\boldsymbol{W}_f = \boldsymbol{R}_f \boldsymbol{S}^T + \boldsymbol{T}_f . e_p, \tag{4.5}$$

em que $\boldsymbol{R}_f = \begin{bmatrix} i_f \\ j_f \end{bmatrix}$ e $\boldsymbol{T}_f$ são as duas primeiras linhas de $\boldsymbol{R}_{3Df}$ e $\boldsymbol{T}_{3Df}$, respectivamente. O modelo de projecção ortogonal permite simplificar bastante o problema SFM ao considerar-se que λ_{fp} é aproximadamente constante, o que elimina a não-linearidade neste problema, facilitando grandemente a obtenção de uma solução. Como consequência, não é possível determinar a translação da câmara ao longo do eixo dos zz e os movimentos na imagem que advém de efeitos de perspectiva tais como *zoom ins*, *zoom outs* e distorções dos objectos, introduzem erro na estimação das matrizes de rotação e de forma.

4.2 Estimação da translação: registo das observações

Definindo as matrizes $\boldsymbol{R}$ e $\boldsymbol{T}$ de acordo com

$$\boldsymbol{R} = \begin{bmatrix} i_1 \\ \vdots \\ i_F \\ j_1 \\ \vdots \\ j_F \end{bmatrix}, \boldsymbol{T} = \begin{bmatrix} t_{x1} \\ \vdots \\ t_{xF} \\ t_{y1} \\ \vdots \\ t_{yF} \end{bmatrix}, \tag{4.6}$$

em que $\boldsymbol{R}$ é denominada a matriz de rotação [37], é possível relacionar a matriz de observação, $\boldsymbol{W}$, com a matriz de rotação, $\boldsymbol{R}$, a matriz de forma, $\boldsymbol{S}$, e a matriz de translação, $\boldsymbol{T}$, da câmara, obtendo-se a equação (4.7), fundamental no método de factorização de Tomasi e Kanade:

$$\boldsymbol{W} = \boldsymbol{R}\boldsymbol{S}^T + \boldsymbol{T}.e_p = [\boldsymbol{R}|\boldsymbol{T}] \begin{bmatrix} \boldsymbol{S}^T \\ \hline e_p \end{bmatrix}. \tag{4.7}$$

Como o centro de coordenadas do objecto filmado está no seu centróide, a soma dos valores das linhas de $\boldsymbol{R}\boldsymbol{S}^T$ é zero. Este facto faz com que a diferença entre o centro de coordenadas do objecto e o centro de coordenadas da câmara, $P\boldsymbol{T}$, seja dado pela soma das linhas de $\boldsymbol{W}$, donde se deduz que $\boldsymbol{T}$ se obtém de

$$\boldsymbol{T}_i = \frac{1}{P} \sum_{j=1}^{P} w_{ij}, i \in [1, 2F]. \tag{4.8}$$

Subtraindo este vector coluna a cada coluna de $\boldsymbol{W}$, tal como está descrito na equação (4.9), obtém-se a matriz de observação registada, i.e., uma matriz de observação

em que o centro das coordenadas da projecção dos pontos armazenados é o ponto $(x, y) = (0, 0)$,

$$\widetilde{\boldsymbol{W}} = \boldsymbol{W} - \boldsymbol{T}.e_p = \boldsymbol{R}\boldsymbol{S}^T, \tag{4.9}$$

que, por ser o produto de uma matriz $2F \times 3$ por uma matriz $3 \times F$ tem, no máximo, característica 3 no caso em que não existe ruído [37]. Assim, da equação (4.7) deduz-se que, caso não haja ruído, a matriz de observação não-registada, $\boldsymbol{W}$, tem no máximo característica 4.

4.3 Decomposição

A solução de máxima verosimilhança (ML) da equação 4.9 dá origem ao problema de minimização

$$\min_{\boldsymbol{R},\boldsymbol{S}} \|\widetilde{\boldsymbol{W}} - \boldsymbol{R}\boldsymbol{S}^T\|_F, \tag{4.10}$$

em que se pretende encontrar as matrizes de rotação e forma, $\boldsymbol{R}$ e $\boldsymbol{S}$ respectivamente, cujo produto melhor aproxima a matriz $\widetilde{\boldsymbol{W}}$. $\|.\|_F$ representa a norma de Frobenius.

Usando a decomposição em valores singulares (SVD) para característica 3 na matriz $\widetilde{\boldsymbol{W}}$, obtém-se três matrizes, $\boldsymbol{U}$, $\boldsymbol{\Sigma}$ e $\boldsymbol{V}$, que apenas contabilizam os três primeiros valores singulares de $\widetilde{\boldsymbol{W}}$. As matrizes $\boldsymbol{U}$ e $\boldsymbol{V}$ são matrizes ortogonais, $2F \times 3$ e $P \times 3$ respectivamente, e $\boldsymbol{\Sigma}$ é uma matriz diagonal não negativa 3×3, tal que $\widetilde{\boldsymbol{W}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. É possível obter as matrizes $\boldsymbol{R}$ e $\boldsymbol{S}$ a partir de

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}\boldsymbol{N} = \boldsymbol{A}\boldsymbol{N} = \left[\begin{array}{c|c} a_{i1} & t_{x1} \\ a_{j1} & t_{y1} \\ a_{i2} & t_{x2} \\ a_{j2} & t_{y2} \\ \vdots & \vdots \\ a_{iF} & t_{xF} \\ a_{jF} & t_{yF} \end{array}\right] \boldsymbol{N}, \tag{4.11}$$

$$\boldsymbol{S} = \boldsymbol{V}\boldsymbol{\Sigma}^{1/2}(\boldsymbol{N}^{-1})^T,$$

em que a_{if} e a_{jf} definem vectores de rotação não normalizados e $\boldsymbol{N}$ é uma matriz de normalização a determinar que faz uso das restrições da matriz $\boldsymbol{R}$ de modo a recuperar o espaço de movimento. Essas restrições são a norma unitária de cada vector de rotação de $\boldsymbol{R}$ e a ortogonalidade entre os vectores de rotação de cada imagem.

Considerando que os valores singulares que não os três maiores apenas se devem ao ruído, o que é verdade se a correspondência dos pontos foi razoavelmente estimada, o facto de serem desprezados faz com que as matrizes $\boldsymbol{R}$ e $\boldsymbol{S}$ obtidas das matrizes resultantes do cálculo da SVD minimizem (4.10) mesmo na presença de ruído [37].

Aquilo que tem sido o grande problema do método de factorização de Tomasi e Kanade, que é o facto de este não suportar oclusão, surge essencialmente no cálculo da SVD. Quando existe oclusão, não são conhecidas as trajectórias completas de cada ponto, havendo posições na matriz de observação $\boldsymbol{W}$ que não são conhecidos. Este facto faz com que não seja possível calcular a matriz de observação registada, $\widehat{\boldsymbol{W}}$, (este problema pode ser ultrapassado formulando o problema SFM de forma a não ser necessário calcular a matriz de observação registada para o resolver, como será visto no capítulo 5) e não seja possível calcular a SVD, pois esta requer que a matriz seja completa. Nos capítulos 5 e 6 propõe-se um conjunto de novos métodos para estimar a matriz de observação completa a partir das trajectórias conhecidas, o que permite calcular a SVD e, portanto, usar o método de factorização de Tomasi e Kanade.

4.4 Normalização

Para estimar a matriz de normalização, $\boldsymbol{N}$, é necessário ter em conta as restrições impostas pela matriz de rotação, descritas por

$$\begin{cases} i_f.i_f^T = 1 \\ j_f.j_f^T = 1 \\ i_f.j_f^T = 0 \end{cases} \iff \begin{cases} a_{if}\boldsymbol{N}\boldsymbol{N}^T a_{if}^T = 1 \\ a_{jf}\boldsymbol{N}\boldsymbol{N}^T a_{jf}^T = 1 \\ a_{if}\boldsymbol{N}\boldsymbol{N}^T a_{jf}^T = 0 \end{cases} . \tag{4.12}$$

Definindo a matriz simétrica 3×3 [26] de acordo com

$$\boldsymbol{L} = \boldsymbol{N}\boldsymbol{N}^T = \begin{bmatrix} l_1 & l_2 & l_3 \\ l_2 & l_4 & l_5 \\ l_3 & l_5 & l_6 \end{bmatrix}, \tag{4.13}$$

o sistema (4.12) pode ser reescrito como

$$\boldsymbol{G}.l = c, \tag{4.14}$$

em que as matrizes $\boldsymbol{G}$, l e c são

$$\boldsymbol{G} = \begin{bmatrix} g(a_{i1}, a_{i1}) \\ \vdots \\ g(a_{iF}, a_{iF}) \\ g(a_{j1}, a_{j1}) \\ \vdots \\ g(a_{jF}, a_{jF}) \\ g(a_{i1}, a_{j1}) \\ \vdots \\ g(a_{iF}, a_{jF}) \end{bmatrix}, l = \begin{bmatrix} l_1 \\ \vdots \\ l_6 \end{bmatrix}, c = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, g(a,b) = \begin{matrix} [a_1b_1 & a_1b_2 + a_2b_1 \\ a_1b_3 + a_3b_1 & a_2b_2 \\ a_2b_3 + a_3b_2 & a_3b_3] \end{matrix}, \tag{4.15}$$

que é resolvido usando uma solução de mínimos quadrados que faz uso da pseudoinversa de Moore-Penrose [24]:

$$l = (\boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T c. \tag{4.16}$$

O vector l determina a matriz $\boldsymbol{L}$ através da equação (4.13), donde, por decomposição em valores próprios, se pode obter a matriz $\boldsymbol{N}$ ($\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \Longrightarrow \boldsymbol{N} = \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}$). Conhecendo a matriz de normalização $\boldsymbol{N}$, é possível determinar a matriz de rotação e a matriz de forma, através da equação (4.11) acima.

Uma vez estimado o vector $\boldsymbol{T}$ relacionado com a diferença entre o centro dos sistemas de coordenadas da câmara e do objecto, a matriz de rotação, $\boldsymbol{R}$, e a matriz de forma, $\boldsymbol{S}$, está resolvido o problema *structure from motion*, de acordo com a formulação, na equação (4.7). Uma vez que, na prática, há defeitos no seguimento dos pontos, é desejável que haja o maior número possível de pontos e de imagens na sequência de vídeo, uma vez que a solução é obtida com base em métodos de mínimos quadrados, que têm maior imunidade ao ruído quando existe mais informação disponível, como será visto no capítulo experimental que aborda este método, na segunda parte desta tese.

A referência [1] propõe um algoritmo que factoriza uma matriz de característica 1, em vez da factorização de característica 3 usada no método de Tomasi e Kanade, o que permite resolver o problema SFM com um custo computacional menor. Quando se pretende estimar um modelo tridimensional denso, o custo computacional torna-se num factor especialmente relevante, uma vez que a matriz de observação é de grandes dimensões.

Capítulo 5

FACTORIZAÇÃO COM INFORMAÇÃO DESCONHECIDA

5.1 Formulação do problema

Ao longo duma sequência de video, existem bocados da estrutura tridimensional da cena filmada que aparecem numas imagens mas não aparecem noutras, devido à oclusão. Assim, as trajectórias dos pontos só são conhecidas enquanto estes aparecem na imagem, sendo incompletas caso estes não apareçam em toda a sequência de video. Este facto faz com que, em casos práticos em que existe oclusão, não sejam conhecidos todos os valores da matriz de observação do método de factorização de Tomasi e Kanade, $\boldsymbol{W}$, o que impede o cálculo da matriz de observação registada, $\widetilde{\boldsymbol{W}}$, e em particular impede o cálculo da SVD, como foi visto no capítulo anterior. Este facto faz com que, em casos práticos em que existe oclusão, não seja possível usar o método de factorização de Tomasi e Kanade, descrito no capítulo anterior.

Neste capítulo e no seguinte são apresentados novos algoritmos que permitem estimar a matriz de observação completa a partir da informação conhecida desta matriz, o que permite usar o método de factorização mesmo na presença de oclusão. A generalidade destes algoritmos tornam-nos também adequados para recuperar informação desconhecida em matrizes singulares com uma característica qualquer.

No método de factorização de Tomasi e Kanade, descrito no capítulo anterior, foi visto que a matriz de observação não-registada, $\boldsymbol{W}$, é dada por

$$\boldsymbol{W} = \boldsymbol{R}\boldsymbol{S}^T + \boldsymbol{T}.e_p = [\boldsymbol{R}|\boldsymbol{T}] \begin{bmatrix} \boldsymbol{S}^T \\ \hline e_p \end{bmatrix} = \boldsymbol{R}_4\boldsymbol{S}_4^T, \tag{5.1}$$

em que $\boldsymbol{R}_4$ contém a matriz de rotação $\boldsymbol{R}$ e a matriz $\boldsymbol{T}$ relacionada com a translação, e $\boldsymbol{S}_4$ contém a matriz de forma $\boldsymbol{S}$ e um vector linha $e_p^T = [1\ 1...1]^T$ com P uns. O produto $\boldsymbol{R}\boldsymbol{S}^T$ tem característica 3 uma vez que as matrizes $\boldsymbol{R}$ e $\boldsymbol{S}$ têm 3 colunas independentes [37]. Uma vez que as matrizes $\boldsymbol{R}_4$ e $\boldsymbol{S}_4$ têm 4 colunas independentes, o

produto $\boldsymbol{R}_4\boldsymbol{S}_4^T$ tem característica 4, o que implica que a matriz $\boldsymbol{W}$ tem característica 4, na ausência de ruído.

O novo método aqui proposto para estimar a informação desconhecida na matriz, faz uso de três algoritmos que têm como função determinar a estimativa de máxima verosimilhança, $\widehat{\boldsymbol{W}}$. Esta estimativa é a que resulta da minimização

$$\min_{\widehat{\boldsymbol{W}}} \|(\boldsymbol{W} - \widehat{\boldsymbol{W}}) \odot \boldsymbol{M}\|_F, \tag{5.2}$$

em que $\widehat{\boldsymbol{W}}$ é uma matriz sem pontos desconhecidos com as mesmas dimensões de $\boldsymbol{W}$, $2F \times P$, e de característica 4, que melhor aproxima a matriz $\boldsymbol{W}$ nos pontos conhecidos; $\boldsymbol{M}$ é uma matriz de máscara de dimensão $2F \times P$ em que $m_{ij} = 1$ se o elemento w_{ij} for conhecido e $m_{ij} = 0$ caso contrário; e $\odot$ representa o produto ponto-a-ponto, também conhecido como produto de Hadamard.

O primeiro algoritmo do novo método aqui proposto calcula, duma forma eficiente, uma estimativa inicial sub-óptima da matriz $\widehat{\boldsymbol{W}}$. Se a matriz $\boldsymbol{W}$ não tiver ruído, esta estimativa retorna imediatamente a matriz $\widehat{\boldsymbol{W}}$ ideal. Os restantes dois algoritmos são algoritmos iterativos óptimos, com um modo de funcionamento complementar, que convergem para $\widehat{\boldsymbol{W}}$ após uma inicialização adequada. O primeiro destes métodos iterativos é um algoritmo de *Expectation-Maximization* (EM) e o segundo, denominado *Row-Column*, generaliza o *power method* para o caso em que existe informação desconhecida e inclui a generalização do método dos mínimos quadrados para o caso em que há informação que não é conhecida. Assim, o método total que permite estimar a matriz $\widehat{\boldsymbol{W}}$ que melhor aproxima $\boldsymbol{W}$ nos pontos conhecidos, para o caso geral em que as trajectórias dos pontos foram estimadas com algum erro, consiste em calcular uma estimativa inicial sub-óptima de $\widehat{\boldsymbol{W}}$, seguido de um dos dois algoritmos iterativos, que refinam a estimativa inicial de forma óptima. Os algoritmos iterativos podem ser usados independentemente, desde que sejam correctamente inicializados. O algoritmo que calcula a estimativa inicial é descrito em seguida, e os algoritmos iterativos são descritos no capítulo seguinte.

5.2 Estimativa inicial

Uma vez que as coordenadas xx e yy de um ponto sofrem oclusão ou inclusão ao mesmo tempo, é conveniente agregá-las, para aumentar a eficácia do algoritmo que calcula a estimativa inicial. Assim, definem-se as matrizes $\boldsymbol{W}$, $\boldsymbol{R}$ e $\boldsymbol{T}$ de forma

equivalente à anterior, através de

$$\boldsymbol{W} = \begin{bmatrix} x_{11} & \dots & x_{1P} \\ y_{11} & \dots & y_{1P} \\ x_{21} & \dots & x_{2P} \\ y_{21} & \dots & y_{2P} \\ \vdots & \dots & \vdots \\ x_{F1} & \dots & x_{FP} \\ y_{F1} & \dots & y_{FP} \end{bmatrix}, \boldsymbol{R} = \begin{bmatrix} i_1 \\ j_1 \\ i_2 \\ j_2 \\ \vdots \\ i_F \\ j_F \end{bmatrix}, \boldsymbol{T} = \begin{bmatrix} t_{x1} \\ t_{y1} \\ t_{x2} \\ t_{y2} \\ \vdots \\ t_{xF} \\ t_{yF} \end{bmatrix}, \tag{5.3}$$

mantendo-se válida a relação (5.1), que redefine a matriz $\boldsymbol{R}_4$.

5.2.1 Ilustração para um caso simples

Considerando inicialmente, para facilitar a compreensão do algoritmo total, o caso mais simples em que todos os pontos estão presentes nas primeiras $n/2$ imagens (n é o número de linhas conhecidas) e m pontos estão presentes em todas as imagens, a matriz de observação $\boldsymbol{W}$ pode ser representada da forma representada na figura 5.1, em que a zona escura representa os elementos conhecidos e a zona branca representa os elementos desconhecidos. Uma vez que esta matriz tem elementos que não são conhecidos, não é possível aplicar a decomposição em valores singulares.

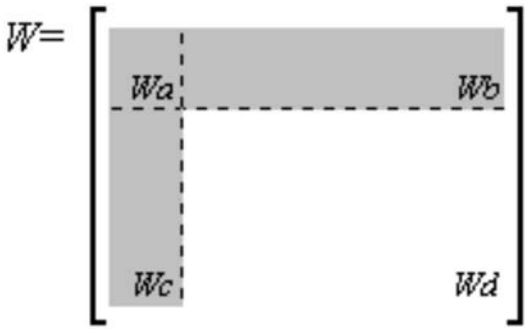


Figura 5.1: Ilustração da matriz $\boldsymbol{W}$ simplificada com informação desconhecida.

Aplicando a SVD de característica 4 (i.e., estimando os primeiros 4 vectores e valores singulares) à matriz $\boldsymbol{W}_{ac}$ dada por

$$\boldsymbol{W}_{ac} = \begin{bmatrix} \boldsymbol{W}_a \\ \hline \boldsymbol{W}_c \end{bmatrix} = \boldsymbol{U}_{ac}\boldsymbol{\Sigma}_{ac}\boldsymbol{V}_{ac}^T = \begin{bmatrix} \boldsymbol{U}_a \\ \hline \boldsymbol{U}_c \end{bmatrix} \boldsymbol{\Sigma}_{ac}\boldsymbol{V}_{ac}^T, \tag{5.4}$$

obtém-se, entre outras, uma matriz $\boldsymbol{U}_{ac}$ de dimensão $2F \times 4$, que pode ser relacionada com o movimento de rotação e translação da câmara, $\boldsymbol{R}_4$, através da relação

$$\boldsymbol{R}_4 = \boldsymbol{U}_{ac}\boldsymbol{N}, \tag{5.5}$$

em que $\boldsymbol{N}$ é uma matriz de normalização (esta matriz $\boldsymbol{N}$ não é a matriz de normalização do método de factorização de Tomasi e Kanade, apesar de partilharem a

designação). A matriz $\boldsymbol{U}_a$ corresponde às primeiras n linhas da matriz $\boldsymbol{U}_{ac}$. Assim, a primeira restrição deste método é a de que o número de pontos com a trajectória totalmente conhecidas (i.e., o número de colunas totalmente conhecidas), m, seja maior ou igual à característica da matriz do problema, neste caso 4, de modo que a matriz $\boldsymbol{U}_{ac}$ tenha dimensão $2F \times 4$. Definindo $\boldsymbol{R}_{n4}$ como as primeiras n linhas de $\boldsymbol{R}_4$, é válida a relação

$$\boldsymbol{W}_{ab} = [\boldsymbol{W}_a|\boldsymbol{W}_b] = \boldsymbol{R}_{n4}\boldsymbol{S}_4^T = (\boldsymbol{U}_a\boldsymbol{N})\left(\boldsymbol{N}^{-1}\boldsymbol{V}_{ab}^T\right) = \boldsymbol{U}_a\boldsymbol{V}_{ab}^T, \tag{5.6}$$

em que $\boldsymbol{S}_4^T = \boldsymbol{N}^{-1}\boldsymbol{V}_{ab}^T$. Uma vez que as matrizes $\boldsymbol{W}_{ab}$ e $\boldsymbol{U}_a$ são conhecidas, basta aplicar a pseudoinversa de Moore-Penrose para determinar a matriz $P \times 4$, $\boldsymbol{V}_ab$, através de

$$\boldsymbol{V}_{ab} = \boldsymbol{W}_{ab}^T\boldsymbol{U}_a\left(\boldsymbol{U}_a^T\boldsymbol{U}_a\right)^{-1}, \tag{5.7}$$

o que impõe a restrição de que tem de haver pelo menos um número de linhas conhecidas maior ou igual à característica da matriz. Neste caso $n > 4$, i.e., todos os pontos têm de aparecer em pelo menos duas imagens, o que está de acordo com o problema SFM. Assim, a primeira estimativa da matriz $\boldsymbol{W}$, $\widehat{\boldsymbol{W}}$, é calculada através de

$$\boldsymbol{W} = \boldsymbol{R}_4\boldsymbol{S}_4^T = (\boldsymbol{U}_{ac}\boldsymbol{N})\left(\boldsymbol{N}^{-1}\boldsymbol{V}_{ab}^T\right) = \boldsymbol{U}_{ac}\boldsymbol{V}_{ab}^T, \tag{5.8}$$

que, caso não haja ruído, é imediatamente a matriz que melhor aproxima $\boldsymbol{W}$. Como o cálculo da cada uma das matrizes $\boldsymbol{U}_{ac}$ e $\boldsymbol{V}_{ab}$ não faz uso de toda a informação conhecida, este algoritmo, apesar de retornar estimativas bastante próximas da matriz $\widehat{\boldsymbol{W}}$ ideal (por ser baseado numa estratégia de mínimos quadrados, que tem como resultado a filtragem do ruído, como será visto mais adiante), não é óptimo, pelo que terá de ser complementado com um dos algoritmos iterativos que são descritos em detalhe no capítulo seguinte, para o caso geral em que $\boldsymbol{W}$ tem ruído.

5.2.2 Solução geral para o caso típico

No caso mais simples, em que a informação conhecida da matriz tem a forma ilustrada na figura 5.1, só é executado um único cálculo da decomposição em valores singulares. Apesar do caso considerado, na prática, não ser provável, os casos gerais são reduzidos a este caso, uma vez que também se irá calcular uma matriz relacionada com $\boldsymbol{R}_4$, que, através do métodos dos mínimos quadrados, irá permitir calcular uma matriz relacionada com $\boldsymbol{S}_4$, o que permite determinar uma estimativa de $\widehat{\boldsymbol{W}}$, através da relação (5.8).

Assim, considerando um caso mais próximo da realidade, a informação conhecida da matriz $\boldsymbol{W}$ tem uma forma semelhante à ilustrada na figura 5.2.
Aplicando a decomposição em valores singulares de característica 4 à matriz $\boldsymbol{W}_{bcd}$ formada por $\boldsymbol{W}_b$, $\boldsymbol{W}_c$ e $\boldsymbol{W}_d$, e à matriz $\boldsymbol{W}_{fg}$ formada por $\boldsymbol{W}_f$ e $\boldsymbol{W}_g$, obtém-se as

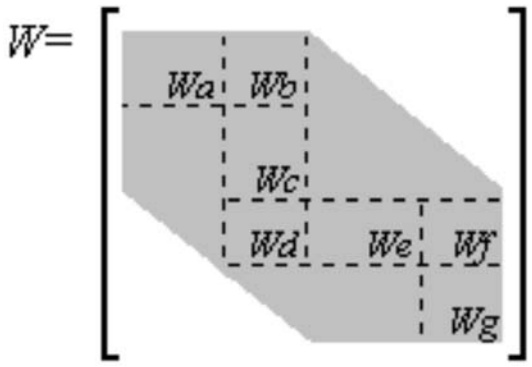


Figura 5.2: Ilustração da matriz $\boldsymbol{W}$ típica com informação desconhecida.

matrizes $\boldsymbol{U}_{bcd}$ e $\boldsymbol{U}'_{fg}$, que estão relacionadas com partes da matriz $\boldsymbol{R}_4$,

$$\begin{cases} \boldsymbol{W}_{bcd} = \begin{bmatrix} \boldsymbol{W}_b \\ \hline \boldsymbol{W}_c \\ \hline \boldsymbol{W}_d \end{bmatrix} = \boldsymbol{U}_{bcd}\boldsymbol{\Sigma}_{bcd}\boldsymbol{V}_{bcd}^T = \begin{bmatrix} \boldsymbol{U}_b \\ \hline \boldsymbol{U}_c \\ \hline \boldsymbol{U}_d \end{bmatrix} \boldsymbol{\Sigma}_{bcd}\boldsymbol{V}_{bcd}^T \\ \boldsymbol{W}_{fg} = \begin{bmatrix} \boldsymbol{W}_f \\ \hline \boldsymbol{W}_g \end{bmatrix} = \boldsymbol{U}'_{fg}\boldsymbol{\Sigma}'_{fg}\boldsymbol{V}'^T_{fg} = \begin{bmatrix} \boldsymbol{U}'_f \\ \hline \boldsymbol{U}'_g \end{bmatrix} \boldsymbol{\Sigma}'_{fg}\boldsymbol{V}'^T_{fg} \end{cases} \tag{5.9}$$

Como as matrizes $\boldsymbol{U}_{bcd}$ e $\boldsymbol{U}'_{fg}$ estão relacionadas com a matriz $\boldsymbol{R}_4$, estão também relacionadas entre si. A relação entre as matrizes é obtida através das parcelas que pertencem às mesmas linhas, i.e., às mesmas imagens, neste caso $\boldsymbol{U}_d$ e $\boldsymbol{U}'_f$, através de

$$\boldsymbol{U}_d \approx \boldsymbol{U}'_f\boldsymbol{N} \tag{5.10}$$

onde $\boldsymbol{N}$ é uma matriz 4×4 que relaciona aproximadamente as duas matrizes, que pode ser obtida pelo método dos mínimos quadrados. Assim, a matriz $\boldsymbol{U}$ total é dada por

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{U}_{bcd} \\ \hline \boldsymbol{U}'_g\boldsymbol{N} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_{bcd} \\ \hline \boldsymbol{U}_g \end{bmatrix}. \tag{5.11}$$

Este procedimento pode ser repetido tantas vezes quantas forem necessárias para que sejam contabilizadas todas as linhas da matriz $\boldsymbol{W}$. Para obter a matriz relacionada com $\boldsymbol{S}_4$, basta usar o método dos mínimos quadrados de forma semelhante à da equação (5.7), nas porções conhecidas de $\boldsymbol{W}$. Neste caso, uma das soluções é dada por

$$\begin{cases} \boldsymbol{V}_{ab} = [\boldsymbol{W}_a|\boldsymbol{W}_b]^T \boldsymbol{U}_b \left(\boldsymbol{U}_b^T\boldsymbol{U}_b\right)^{-1} \\ \boldsymbol{V}_{ef} = [\boldsymbol{W}_e|\boldsymbol{W}_f]^T \boldsymbol{U}_d \left(\boldsymbol{U}_d^T\boldsymbol{U}_d\right)^{-1} \end{cases} \tag{5.12}$$

de onde resulta que a estimativa de $\boldsymbol{W}$, $\widehat{\boldsymbol{W}}$, é dada por

$$\widehat{\boldsymbol{W}} = \boldsymbol{U}\boldsymbol{V}^T = \boldsymbol{U}\begin{bmatrix} \boldsymbol{V}_{ab} \\ \hline \boldsymbol{V}_{ef} \end{bmatrix}^T. \tag{5.13}$$

As restrições deste método são as mesmas que surgiram naturalmente pela simples manipulação matemática para o caso em que a informação conhecida está disposta de uma forma mais simples, apenas com mais uma restrição. Assim, as restrições do algoritmo que calcula a estimativa inicial são: *i*) as matrizes $\boldsymbol{U}$ só podem ser extraídas de porções de $\boldsymbol{W}$ onde pelo menos quatro pontos percorrem todas as imagens contempladas pela respectiva matriz; *ii*) as matrizes $\boldsymbol{V}$ só podem ser extraídas de porções de $\boldsymbol{W}$ onde o mesmo ponto é seguido em, pelo menos, duas imagens; *iii*) para efectuar a união entre matrizes $\boldsymbol{U}$, a primeira matriz tem de partilhar pelo menos duas imagens com a matriz $\boldsymbol{U}$ seguinte. As restrições indicadas partem do princípio que $\boldsymbol{W}$ tem informação suficientemente *variada*, ou seja, que a variação da câmara é suficiente para que o movimento detectado na imagem contenha informação que permita recuperar a estrutura tridimensional da cena e o movimento da câmara. No entanto, para minorar o efeito do ruído na matriz $\boldsymbol{W}$, é desejável usar mais informação do que a que é imposta pelas restrições referidas.

A referência [37] sugere um método sub-óptimo semelhante ao método original aqui proposto para calcular a estimativa inicial, embora com grandes desvantagens face ao último. O método em [37] é computacionalmente muito pesado por requerer $O(\max(F, P))$ utilizações da SVD para preencher, de forma sequencial, linha-a-linha ou coluna-a-coluna, as posições desconhecidas de $\boldsymbol{W}$ com estimativas [18]. A forma sequencial como são estimadas as posições desconhecidas da matriz completa a partir de uma porção de $\boldsymbol{W}$ sem dados desconhecidos faz com que o método em [37] tenha problemas de propagação do erro, e introduz o problema complexo de escolher o melhor bloco de partida usado para estimar os dados desconhecidos [18]. Estes factos impossibilitam a sua utilização em matrizes de grandes dimensões e com grandes percentagens de dados desconhecidos. O método original aqui proposto é mais simples, geral e requer poucas utilizações da SVD.

Capítulo 6

ALGORITMOS *EXPECTATION-MAXIMIZATION* E *ROW-COLUMN*

Neste capítulo descrevem-se dois algoritmos iterativos com estratégias de funcionamento complementares, que estimam de forma óptima a matriz singular completa que melhor aproxima a matriz de observação incompleta, nos pontos conhecidos.

6.1 Algoritmo *Expectation-Maximization*

A abordagem de *Expectation-Maximization* (EM) nos problemas de estimação funciona aumentando o número de parâmetros a estimar, em que os dados desconhecidos são estimados juntamente com os outros parâmetros. A estimação é calculada, iterativa e alternadamente, em dois passos: *i*) no passo de *expectation* estimam-se os dados desconhecidos a partir da estimativa anterior dos parâmetros; *ii*) no passo de *maximization* estimam-se os parâmetros a partir dos dados desconhecidos [24].

Neste caso, dada uma estimativa inicial para a matriz de observação completa, $\widehat{\boldsymbol{W}}_{(0)}$, o algoritmo EM estima alternadamente: *i*) as posições desconhecidas da matriz de observação incompleta, $\boldsymbol{W}$; *ii*) a matriz $\widehat{\boldsymbol{W}}$ de característica 4 que melhor aproxima a matriz $\boldsymbol{W}$ com estimativas das posições desconhecidas. O algoritmo executa alternadamente estes passos até se atingir a convergência, i.e., até o erro

$$\min_{\widehat{\boldsymbol{W}}} \|(\boldsymbol{W} - \widehat{\boldsymbol{W}}) \odot \boldsymbol{M}\|_F, \tag{6.1}$$

estabilizar. Em 6.1, $\widehat{\boldsymbol{W}}$ é uma matriz sem pontos desconhecidos com as mesmas dimensões de $\boldsymbol{W}$, $2F \times P$, e característica 4, que melhor aproxima a matriz $\boldsymbol{W}$ nos pontos conhecidos; $\boldsymbol{M}$ é uma matriz de máscara de dimensão $2F \times P$ em que $m_{ij} = 1$ se o elemento w_{ij} for conhecido e $m_{ij} = 0$ caso contrário; e $\odot$ representa o produto ponto-a-ponto, também conhecido como produto de Hadamard.

6.1.1 Passo *expectation* - estimação dos dados desconhecidos

Dada a matriz $\widehat{\boldsymbol{W}}_{(k-1)}$, que é a estimativa da iteração $(k-1)$ da matriz completa de característica 4 que melhor aproxima $\boldsymbol{W}$ nos pontos conhecidos (de acordo com a equação (6.1)), define-se uma matriz $\overline{\boldsymbol{W}}_{(k)}$, de dimensão $2F \times P$, em que cada valor é dado por

$$\overline{w}_{(k)ij} = \begin{cases} w_{ij} & \text{se } w_{ij} \text{ for conhecido,} \\ \widehat{w}_{(k-1)ij} & \text{caso contrário.} \end{cases} \tag{6.2}$$

Este passo está ilustrado na figura 6.1, em que os pontos claros correspondem aos valores conhecidos da matriz de observação e os pontos escuros são estimativas dos valores das posições desconhecidas.

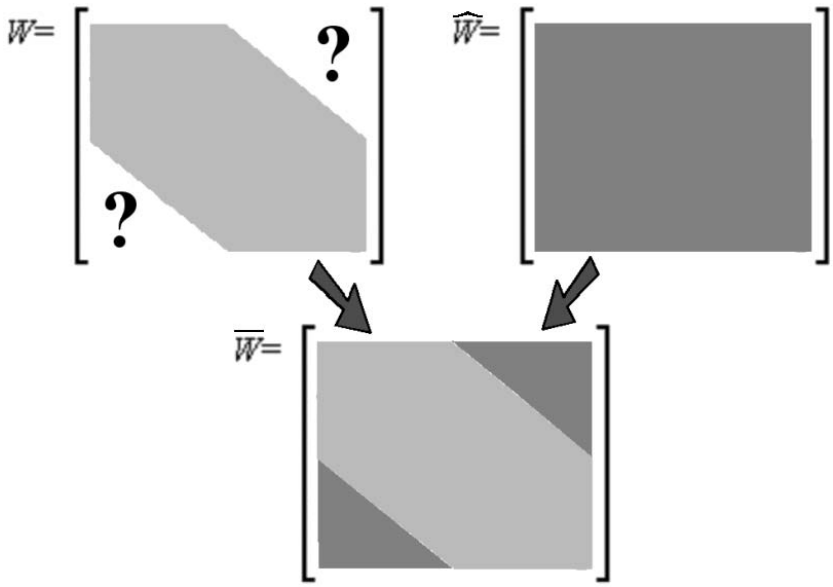


Figura 6.1: Ilustração do passo de *expectation* do algoritmo EM.

Assim, obtém-se uma matriz $\overline{\boldsymbol{W}}_{(k)}$ com todos os elementos conhecidos mas característica desconhecida. Para executar este algoritmo é necessário uma estimativa inicial da matriz de observação completa, $\widehat{\boldsymbol{W}}_{(0)}$. Esta estimativa pode ser calculada pelo algoritmo que calcula a estimativa inicial do capítulo anterior, ou pode ter origem aleatória, desde que respeite as condições que se deduzem teoricamente mais abaixo neste capítulo, que são comprovados experimentalmente na segunda parte desta tese.

Na forma matricial, a definição de $\overline{\boldsymbol{W}}_{(k)}$ é dada por

$$\overline{\boldsymbol{W}}_{(k)} = \boldsymbol{W} \odot \boldsymbol{M} + \widehat{\boldsymbol{W}}_{(k-1)} \odot [1 - M], \tag{6.3}$$

em que $\boldsymbol{M}$ é, tal como foi definido anteriormente, uma matriz de máscara em que $m_{ij} = 1$ se o elemento ij da matriz $\boldsymbol{W}$ for conhecido e $m_{ij} = 0$, caso contrário.

6.1.2 Passo *maximization* - estimação da matriz de característica deficiente

A matriz $\overline{\boldsymbol{W}}_{(k)}$ obtida no passo *expectation* não tem informação desconhecida mas tem, em geral, característica superior à esperada teoricamente para a matriz $\boldsymbol{W}$. Para determinar a matriz $\widehat{\boldsymbol{W}}_{(k)}$ de característica 4 que melhor aproxima a matriz $\boldsymbol{W}$ nas posições conhecidas, são eliminados os valores singulares além dos quatro primeiros, o que é conseguido calculando a decomposição em valores singulares [11] de característica 4 da matriz $\overline{\boldsymbol{W}}_{(k)}$ e definindo-se $\widehat{\boldsymbol{W}}_{(k)}$ como o produto das matrizes resultantes, tal como está escrito em

$$\begin{aligned} \overline{\boldsymbol{W}}_{(k)} &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \\ \widehat{\boldsymbol{W}}_{(k)} &= \boldsymbol{U}_4\boldsymbol{\Sigma}_4\boldsymbol{V}_4^T, \end{aligned} \tag{6.4}$$

em que $\boldsymbol{U}_4$ e $\boldsymbol{V}_4$ são as primeiras quatro colunas das matrizes $\boldsymbol{U}$ e $\boldsymbol{V}$ resultantes da SVD, e $\boldsymbol{\Sigma}_4$ é uma matriz 4×4 diagonal com os primeiros 4 valores singulares da matriz $\boldsymbol{\Sigma}$ na sua diagonal.

Não são usados os valores singulares não-dominantes (considera-se que os valores singulares dominantes são os 4 valores singulares de maior módulo, e que os restantes são não-dominantes) porque assume-se que estes se devem apenas a erro na estimação dos valores desconhecidos da matriz $\boldsymbol{W}$ ou ao ruído na estimação da trajectória dos pontos nas imagens. No entanto, quando a ordem de grandeza dos valores da inicialização é maior do que a dos dados conhecidos da matriz de observação, os valores singulares devidos aos dados são não-dominantes face aos da inicialização e, por serem retirados no passo de *maximization* do algoritmo EM, fazem com que este algoritmo não consiga convergir. Assim, $\overline{\boldsymbol{W}}_{(0)} = 0$ é uma boa inicialização para este algoritmo porque: *i*) 0 é a menor ordem de grandeza possível; *ii*) por ser constante, o seu impacto está confinado essencialmente a um valor singular.

A restrição de que os valores singulares dos dados têm de ser dominantes face aos valores singulares devidos ao ruído tem fortes implicações na sensibilidade deste algoritmo em relação à forma como é inicializado e à percentagem de informação desconhecida. De facto, grandes percentagens de informação desconhecida tornam este algoritmo ainda mais dependente da estratégia de inicialização, uma vez que estes valores têm uma influência maior nos valores singulares da matriz $\overline{\boldsymbol{W}}_{(k)}$. No entanto, o algoritmo deverá sempre convergir desde que a ordem de grandeza da inicialização seja inferior à dos dados, esperando-se que a percentagem de informação desconhecida apenas influencie a velocidade de convergência.

Para os casos normais em que este algoritmo é bem inicializado, quer através do algoritmo de inicialização do capítulo anterior quer tendo em conta as considerações acima, é muito eficaz face ao ruído. Isto acontece porque a sua influência faz-se

sentir essencialmente nos valores singulares não-dominantes, que são retirados neste passo de *maximization*, a menos que o ruído seja de tal forma grande que contamine os valores e vectores singulares dominantes ao ponto de prejudicar a velocidade de convergência ou mesmo impedir a convergência do algoritmo. O estudo experimental das condições para a convergência deste algoritmo é feito na segunda parte desta tese.

Do ponto de vista computacional, a execução deste algoritmo iterativo implica apenas o cálculo da SVD de característica 4 da matriz $\overline{\boldsymbol{W}}_{(k)}$ e a multiplicação das três matrizes resultantes, por cada iteração executada.

Este algoritmo iterativo é semelhante ao algoritmo EM em [23], com a diferença de que nesta referência a translação é tratada à parte, sendo um método feito para o caso específico do problema SFM com informação desconhecida. O método novo aqui proposto, além de ser geral, i.e., de determinar informação desconhecida em matrizes singulares de qualquer característica (apesar de ser apresentado no contexto de SFM, que tem característica 4), é mais simples de implementar do que [23], por não necessitar de inverter matrizes no passo *expectation*. Para usar os algoritmos iterativos apresentados nesta tese para determinar informação desconhecida em matrizes singulares com qualquer característica, basta substituir as referências à característica 4 do problema SFM pela característica desejada, já que os algoritmos se deduzem da mesma forma para qualquer característica.

6.2 Algoritmo *Row-Column*

Neste algoritmo, em vez de se aumentar o número de parâmetros, como no algoritmo EM da secção anterior, parametriza-se a matriz de observação como sendo o produto do espaço das linhas pelo espaço das colunas, estimando-se alternadamente estes dois espaços.

Com este algoritmo pretende-se estimar as matrizes $\boldsymbol{U}$, de dimensão $2F \times 4$, e $\boldsymbol{V}$, $P \times 4$, que resolvem o problema de minimização

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|(\boldsymbol{W} - \boldsymbol{U}\boldsymbol{V}^T) \odot \boldsymbol{M}\|_F, \tag{6.5}$$

em que $\boldsymbol{M}$ é uma matriz de máscara definida como anteriormente. Ou seja, pretende-se estimar matrizes $\boldsymbol{U}$ e $\boldsymbol{V}$ tal que $\boldsymbol{U}\boldsymbol{V}^T$ é a melhor aproximação possível da matriz de observação, $\boldsymbol{W}$, nas posições conhecidas.

Dada uma estimativa inicial da matriz $\boldsymbol{U}$, $\boldsymbol{U}_{(0)}$, o algoritmo RC minimiza a equação (6.5) estimando alternadamente: *i*) $\boldsymbol{V}$ conhecendo $\boldsymbol{U}$, no passo R; *ii*) $\boldsymbol{U}$ conhecendo $\boldsymbol{V}$, no passo C. O algoritmo executa alternadamente estes passos até se atingir a convergência, i.e., até o erro da equação (6.5) estabilizar.

Quando não há informação desconhecida na matriz de observação $\boldsymbol{W}$, i.e., quando $\boldsymbol{M} = 1_{2F \times P}$, os passos R e C são obtidos pela simples aplicação da pseudo-inversa

[24] a

$$\boldsymbol{W} = \boldsymbol{U}\boldsymbol{V}^T, \tag{6.6}$$

obtendo-se então

$$\boldsymbol{U}_{(k)} = \left(\boldsymbol{V}_{(k-1)}^T \boldsymbol{V}_{(k-1)}\right)^{-1} \boldsymbol{V}_{(k-1)}^T \boldsymbol{W} \tag{6.7}$$

$$\boldsymbol{V}_{(k)} = \boldsymbol{W}\boldsymbol{U}_{(k)}^T \left(\boldsymbol{U}_{(k)}\boldsymbol{U}_{(k)}^T\right)^{-1}, \tag{6.8}$$

em que (6.7) calcula o passo R e (6.8) calcula o passo C. Substituindo (6.7) em (6.8), compacta-se este algoritmo num único passo, dado por

$$\boldsymbol{U}_{(k)} = \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{U}_{(k-1)} \left(\boldsymbol{U}_{(k-1)}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{U}_{(k-1)}\right)^{-1} \boldsymbol{U}_{(k-1)}^T\boldsymbol{U}_{(k-1)}, \tag{6.9}$$

que mostra que o algoritmo RC implementa a aplicação do *power method* [24] à matriz $\boldsymbol{W}\boldsymbol{W}^T$ (o factor $\left(\boldsymbol{U}_{(k-1)}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{U}_{(k-1)}\right)^{-1} \boldsymbol{U}_{(k-1)}^T\boldsymbol{U}_{(k-1)}$ é a normalização). O *power method* é muito usado para calcular a melhor aproximação de baixa característica de uma dada matriz, evitando o cálculo da SVD [11]. Assim, o algoritmo RC implementa uma generalização do *power method* e da pseudo-inversa de Moore-Penrose para o caso em que existe informação desconhecida, em que ambos os passos do algoritmo RC continuam a admitir solução em forma fechada e são muito simples.

6.2.1 Estimação do espaço das linhas

Da equação (6.6) obtém-se que a relação entre a coluna p da matriz de observação, w_p, a matriz $\boldsymbol{U}$ e a linha p da matriz $\boldsymbol{V}$, v_p, é dada por

$$w_p = \boldsymbol{U}v_p^T, \tag{6.10}$$

para todos os $p = 1..P$. Quando não são conhecidas todas as linhas do vector coluna w_p, a equação (6.10) não pode ser resolvida pelo método dos mínimos quadrados habitual. Multiplicando por zero, i.e., mascarando as linhas que não são conhecidas no vector w_p, no vector v_p e na matriz $\boldsymbol{U}$, obtém-se

$$w_p \odot m_p = (\boldsymbol{U} \odot \boldsymbol{M}_p)v_p^T, \tag{6.11}$$

onde m_p é a coluna p da matriz de máscara $\boldsymbol{M}$ definida anteriormente, e $\boldsymbol{M}_p$ é uma matriz $2F \times 4$ onde todas as colunas são iguais a m_p, para todo os $p = 1...P$. Uma vez que as parcelas da equação (6.11) já são conhecidas, é possível aplicar o método dos mínimos quadrados a esta equação, obtendo-se a generalização do método dos mínimos quadrados dado por

$$v_p^T = \left[(\boldsymbol{U} \odot \boldsymbol{M}_p)^T(\boldsymbol{U} \odot \boldsymbol{M}_p)\right]^{-1} (\boldsymbol{U} \odot \boldsymbol{M}_p)^T (w_p \odot m_p), \tag{6.12}$$

que é simplificada eliminando repetições dos mascaramentos binários,

$$v_p^T = \left[\boldsymbol{U}^T(\boldsymbol{U} \odot \boldsymbol{M}_p)\right]^{-1} \boldsymbol{U}^T (w_p \odot m_p). \tag{6.13}$$

Verifica-se que cada coluna é obtida de forma semelhante ao *power method*, anulando-se as linhas de w_p e $\boldsymbol{U}$ onde existe informação desconhecida. Determina-se a matriz $\boldsymbol{V}$ usando a equação (6.13) para todos os $p \in [1, P]$.

6.2.2 Estimação do espaço das colunas

A equação que relaciona cada linha f da matriz de observação $\boldsymbol{W}$, w_f, com a linha f da matriz $\boldsymbol{U}$, u_f e a matriz $\boldsymbol{V}$, é dada por

$$w_f = u_f \boldsymbol{V}^T, \tag{6.14}$$

para todo $f = 1..2F$. Com informação desconhecida na matriz w_f, é necessário anular as colunas correspondentes, resultando em

$$w_f \odot m_f = (u_f \odot \boldsymbol{M}_f)\boldsymbol{V}^T, \tag{6.15}$$

onde m_f é a linha f da matriz $\boldsymbol{M}$ e $\boldsymbol{M}_f$ é uma matriz $4 \times P$ onde todas as linhas são iguais a m_f, para todo os $f = 1...2F$. Resolvendo esta equação usando o método dos mínimos quadrados e eliminando as repetições nos mascaramentos de forma análoga a (6.13), obtém-se

$$u_f = (w_f \odot m_f)\boldsymbol{V}[(\boldsymbol{V}^T \odot \boldsymbol{M}_f)\boldsymbol{V}]^{-1}. \tag{6.16}$$

Determina-se a matriz $\boldsymbol{U}$ usando a equação (6.16) para todos os $f \in [1, 2F]$.

Ao contrário do método dos mínimos quadrados sem informação desconhecida, os vectores v_p e u_f são determinadas um-a-um, uma vez que a distribuição da informação desconhecida da matriz de observação (representada nas matrizes $\boldsymbol{M}_p$ e m_p e nas matrizes $\boldsymbol{M}_f$ e m_f) variam, em geral, para cada ponto e cada linha de $\boldsymbol{M}$, respectivamente. Caso hajam várias linhas ou colunas iguais na matriz de máscara $\boldsymbol{M}$, é possível poupar tempo de computação notando que o resultado das pseudo-inversas das equações (6.13) e (6.16) é o mesmo de vector para vector, quando a distribuição da informação desconhecida é a mesma.

Tal como no *power method* ou no algoritmo EM da secção 6.1, este algoritmo requer uma inicialização, que pode provir do algoritmo que calcula a estimativa inicial de $\widehat{\boldsymbol{W}}$ do capítulo anterior (nesse caso, não é necessário calcular a matriz $\boldsymbol{V}$ final desse algoritmo, uma vez que essa informação não é usada por este algoritmo iterativo) ou pode ser aleatória.

Uma vez que este algoritmo estima sempre as matrizes $\boldsymbol{U}$ e $\boldsymbol{V}$ de forma a que o seu produto seja $\boldsymbol{W}$, ao contrário do algoritmo EM descrito acima, a ordem de grandeza da inicialização não é determinante para a sua convergência, uma vez que a ordem de grandeza duma matriz é compensada pela ordem de grandeza da outra. Contudo, ao contrário do algoritmo EM, este algoritmo exige que a informação conhecida das matrizes $\boldsymbol{U}$, $\boldsymbol{V}$ e $\boldsymbol{W}$ tenha pelo menos a característica que o problema

exige (neste caso, característica 4), para que as matrizes de que se calculam as pseudo-inversas nas equações (6.13) e (6.16) não estejam próximas da singularidade, o que dá origem a problemas computacionais e poderá impedir a convergência do algoritmo. Este facto mostra que, caso não seja usado o algoritmo que calcula a estimativa inicial, a inicialização ideal para este algoritmo seria uma matriz $\boldsymbol{U}_{(0)}$ com valores aleatórios.

A quantidade de dados desconhecidos é também um factor de grande importância para a convergência deste algoritmo. De facto, se existir muita informação conhecida, as matrizes que são invertidas estão mais distantes da singularidade e a convergência é mais rápida. A existência de pouca informação conhecida poderá dar origem a problemas de precisão numérica no cálculo das pseudo-inversas, o que poderá atrasar ou mesmo impedir a convergência deste algoritmo.

Para os casos normais em que este algoritmo é bem inicializado, quer através do algoritmo de inicialização do capítulo anterior quer tendo em conta as considerações acima, e a matriz $\boldsymbol{W}$ é suficientemente variada para que as matrizes a inverter não estejam próximas da singularidade, a abordagem de mínimos quadrados que está na sua base torna-o muito eficaz face ao ruído. O estudo experimental da condições para a sua convergência é feito na segunda parte desta tese.

Do ponto de vista computacional, por cada iteração, são calculadas, no máximo, $P + 2F$ inversas de matrizes 4×4, que são multiplicadas pelas matrizes indicadas P e $2F$ vezes.

Parte II

Experiências e Aplicações

Capítulo 7

ESTIMAÇÃO DE MOVIMENTO

7.1 Estimação do mapa denso de correspondências

Para ilustrar a estimação do movimento na imagem, usaram-se pares de imagens obtidos em [48], em que a localização das câmaras é desconhecida. Nas figuras 7.1-7.4, (a,g) e (b,h) são as imagens originais da esquerda e da direita, respectivamente. Para validar as correspondências estimadas, a figura (c,i) mostra a estrutura 3D dada pelo método do capítulo 4, a partir dos movimentos estimados; e as figuras (d-f,j-l) mostram projecções dos modelos 3D estimados de vários ângulos.

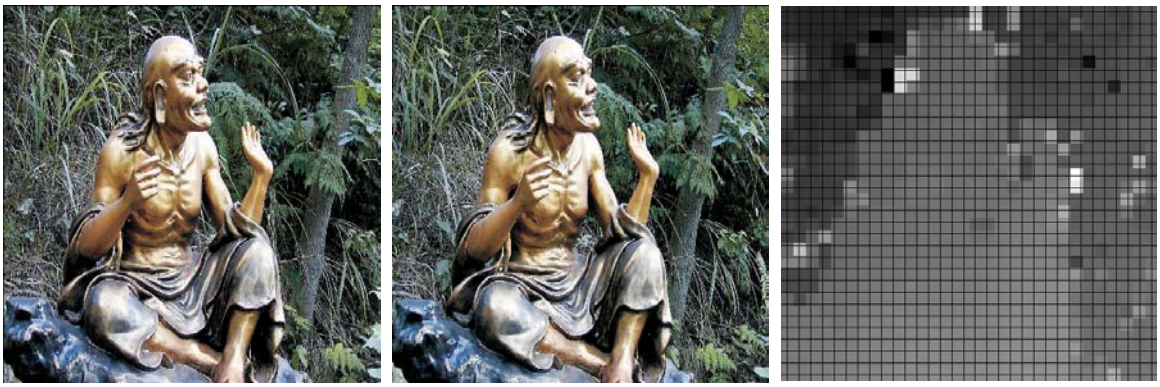
A figura 7.1 mostra figuras relativas à estátua de um dragão da Indonésia.

(a,b) imagens da esquerda e da direita;(c) estrutura 3D (pontos mais escuros = mais distantes)

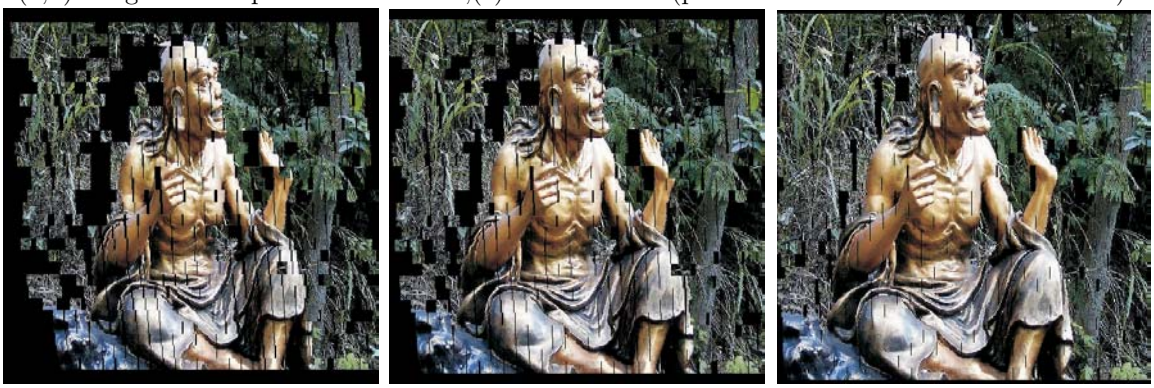
(d-f) projecções do modelo tridimensional obtido

Figura 7.1: Modelo 3D da estátua dum dragão (Bali, Indonésia).

A figura 7.2 mostra figuras relativas à estátua de Buda, num jardim de Taipei, na Republica da China. As figuras 7.1.(c) e 7.2.(c) mostram que, à parte de alguns *outliers*, as estruturas tridimensionais das cenas foram obtidas correctamente, como se confirma através das vistas dos modelos 3D das figuras 7.1.(d-f) e 7.2.(d-f). O modelo 3D do dragão tem pior aparência do que o de buda devido à oclusão.



(a,b) imagens da esquerda e da direita;(c) estrutura 3D (pontos mais escuros = mais distantes)



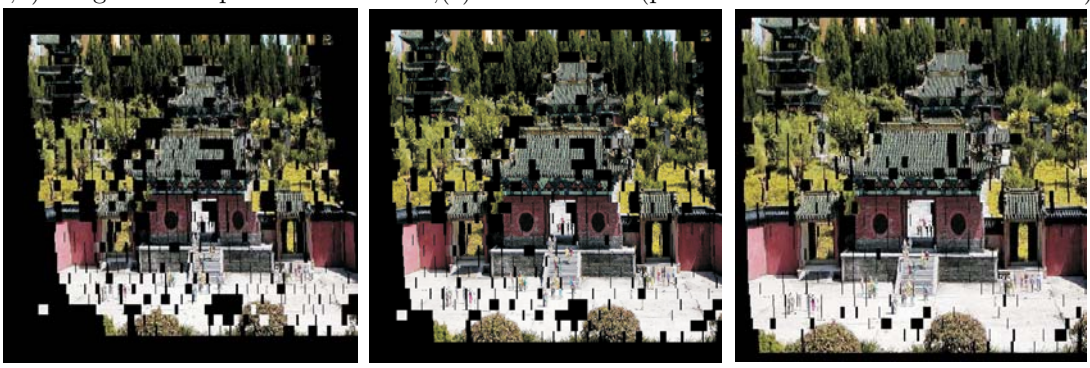
(d-f) projecções do modelo tridimensional obtido

Figura 7.2: Modelo 3D da estátua de buda (Taipei, Republica da China).

As figuras 7.3 e 7.4 mostram figuras relativas a miniaturas de alguns dos monumentos mais conhecidos do mundo. Estas miniaturas estão presentes num jardim denominado *mundo em miniatura*, localizado em Lungtan Shiang, Taoyuan, Taiwan. As figuras 7.3 e 7.4 mostram que a estrutura tridimensional das miniaturas dos monumentos e da cena onde estes estão inseridos foram obtidos, em geral, correctamente. No entanto, as figuras 7.3.(d-f,j-l) e 7.4.(j-l) mostram também algumas zonas onde a correspondência não foi estimada correctamente, devido à ausência de textura ou porque a textura disponível apenas restringe um dos graus de liberdade, como acontece nas escadas das figuras 7.4.(g-h).

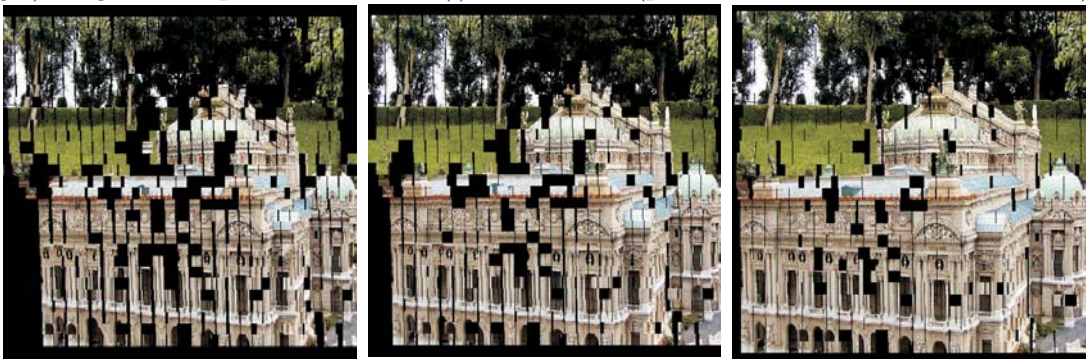
Nos modelos 3D estimados nesta subsecção existem zonas a preto, tal como é visível na figura 7.2.(d) à esquerda de Buda. Estas zonas correspondem a porções da estrutura 3D dos objectos que não estão presentes nas imagens originais. Este problema pode ser resolvido com sequências de video maiores e com os algoritmos que permitem lidar com oclusão e inclusão, apresentados nos capítulos 5 e 6.

(a,b) imagens da esquerda e da direita;(c) estrutura 3D (pontos mais escuros = mais distantes)



(d-f) projecções do modelo tridimensional obtido

(g,h) imagens da esquerda e da direita;(i) estrutura 3D (pontos mais escuros = mais distantes)



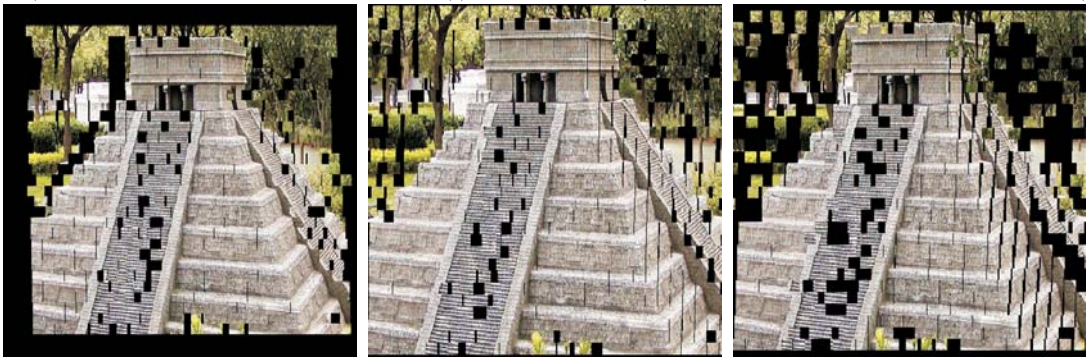
(j-l) projecções do modelo tridimensional obtido

Figura 7.3: Modelos 3D de miniaturas de monumentos (1 de 2)

(a,b) imagens da esquerda e da direita;(c) estrutura 3D (pontos mais escuros = mais distantes)

(d-f) projecções do modelo tridimensional obtido

(g,h) imagens da esquerda e da direita;(i) estrutura 3D (pontos mais escuros = mais distantes)



(j-l) projecções do modelo tridimensional obtido

Figura 7.4: Modelos 3D de miniaturas de monumentos (2 de 2)

7.2 Estimação de homografia

Para ilustrar o desempenho do estimador de homografias, estimaram-se, de forma automática, mosaicos a partir de várias sequências de video. Os mosaicos são construídos sobrepondo as imagens da sequência de video, afectadas das correspondentes homografias estimadas. A sobreposição das imagens é dada, em cada *pixels*, pela média das intensidades dos pontos que correspondem a esse *pixels*. Nesta subsecção são mostrados os resultados obtidos.

Nesta experiência usou-se uma sequência de video com apenas 3 imagens, de resolução 640×480, onde é visível várias porções de uma toalha de praia exposta na montra de uma loja. Estas imagens são apresentadas nas figuras 7.5.(a-c). Apesar dos objectos filmados terem uma estrutura um pouco tridimensional, o facto do centro óptico da câmara de filmar ser aproximadamente o mesmo permite a aproximação de homografia.



(a) imagem 1; (b) imagem 2; (c) imagem 3.

(d) mosaico estimado a partir da sequência de video da toalha de praia.

Figura 7.5: Estimação dum mosaico a partir duma sequência de video com uma toalha de praia.

A figura 7.5.(d) mostra que o estimador de homografias projectado conseguiu estimar um mosaico correcto a partir das imagens da sequência de video, mesmo quando o

movimento de uma imagem para a imagem seguinte é bastante grande, como é visível nas figuras 7.5.(a-c).

Na experiência seguinte usou-se uma sequência de video com 6 imagens, de resolução 640 × 480, onde se vêm várias porções de uma toalha com o desenho de Portugal e vários símbolos representativos das suas regiões. As figuras 7.6.(a-c) mostram algumas destas imagens. O facto da quarta imagem da sequência de video estar um pouco desfocada (ver figura 7.6.(b), que corresponde à região central de Portugal), impede à partida uma estimação precisa dum mosaico a partir desta sequência de video, não só porque será construído com imagens desfocadas, mas porque o pressuposto de brilho constante, que está na base do funcionamento deste algoritmo, não é respeitado em rigor. Obteve-se o mosaico da figura 7.6.(d).

(a) imagem 1; (b) imagem 4 (desfocada); (c) imagem 6.

(d) mosaico estimado a partir da sequência de video da toalha de Portugal

Figura 7.6: Estimação dum mosaico a partir duma sequência de video de um desenho de Portugal.

Para testar o funcionamento do estimador de homografias em sequências de video mais longas, usou-se uma sequência de video com 26 imagens, de resolução 512×512, onde se vêm várias porções de um tapete com letras e alguns móveis à sua volta. As figuras 7.7.(a-h) mostram algumas destas imagens. Como é visível nas figuras

7.7.(a-h), o princípio de brilho constante é neste caso uma aproximação ainda mais grosseira do que na figura 7.6, como se vê por exemplo pela deformação das letras E, F e G nas imagens 7.7.(e,f) face à imagem 7.7.(d). Estas deformações devem-se às técnicas de compressão de video e ao movimento da câmara de filmar.

(a) imagem 1; (b) imagem 5; (c) imagem 9; (d) imagem 13

(e) imagem 16; (f) imagem 19; (g) imagem 22; (h) imagem 26

(i) mosaico estimado.

Figura 7.7: Estimação dum mosaico a partir duma sequência de video com um tapete com letras.

Tendo em conta que o principio de brilho constante não pode ser aplicado em rigor nesta sequência de video, a boa qualidade do mosaico estimado da figura 7.7.(i) ilustra o bom comportamento do algoritmo estimador de homografia aqui apresentado.

Capítulo 8

FACTORIZAÇÃO MATRICIAL

Neste capítulo é feito um estudo aprofundado dos métodos de factorização estudados neste trabalho, nomeadamente o método de Tomasi e Kanade e os novos algoritmos que permitem estimar informação desconhecida em matrizes singulares. Os novos algoritmos são estudados no contexto de SFM, para resolver o problema da oclusão e inclusão, mas também de forma geral, demonstrando-se a sua eficácia em outros problemas em que existem observações incompletas.

8.1 Estimação de dados desconhecidos em matrizes singulares

8.1.1 Introdução

Nesta secção é estudado o comportamento dos novos algoritmos que estimam informação desconhecida em matrizes singulares, descritos nos capítulos 5 e 6, em situações mais gerais em que as matrizes $\boldsymbol{W}$ podem ter origem em qualquer problema de engenharia. As matrizes $\boldsymbol{W}$ têm característica r e podem ter qualquer quantidade ou distribuição da informação desconhecida, desde que satisfaçam os requisitos mínimos indicados nos capítulos 5 e 6. Foram feitas várias experiências para estudar o seu comportamento em diversas situações, das quais se mostram as mais significativas, tal como uma análise do custo computacional. O código que implementa estes algoritmos está disponível em [49].

Foram criadas várias matrizes de observação não-registadas $\boldsymbol{W}$ de várias características, com informação desconhecida e ruído, donde se determinaram estimativas das matrizes de observação completas $\widehat{\boldsymbol{W}}$ e estudou-se o erro médio por ponto, de forma a compreender o comportamento experimental dos métodos. Em particular, os algoritmos iterativos foram testados com inicializações de várias ordens de grandeza e diversidade. As dimensões das matrizes de observação usadas variaram desde 2×2 a 200×100, com informação desconhecida de 0 a 80% e característica de 1 a 6 e ruído com várias médias e desvios padrões.

O erro médio por ponto para o caso em que existe informação desconhecida é

dado pela equação (8.1)

$$E\{erro/ponto\} = \frac{\|(\boldsymbol{W}_{\sigma=0} - \widehat{\boldsymbol{W}}) \odot \boldsymbol{M}\|_F}{\sqrt{\#M}}, \tag{8.1}$$

em que $\boldsymbol{M}$ é uma matriz de máscara de dimensão $2F \times P$ em que $m_{ij} = 1$ se o elemento w_{ij} for conhecido e $m_{ij} = 0$ caso contrário; $\odot$ representa o produto ponto-a-ponto, também conhecido como produto de Hadamard; e $\#\boldsymbol{M} = \sum_{i,j} m_{ij}$, i.e., $\#\boldsymbol{M}$ é o número de entradas conhecidas da matriz $\boldsymbol{W}$.

8.1.2 Matrizes 2×2

De forma a abordar nesta secção experimental um caso mais simples que ainda permita uma representação gráfica ilustrativa do comportamento dos algoritmos, foram criadas matrizes $\boldsymbol{W}$ de dimensão 2×2, característica 1 e com 1 valor desconhecido.

Definindo a matriz $\widehat{\boldsymbol{W}}$ em termos do espaço das linhas e das colunas através da equação (8.2),

$$\widehat{\boldsymbol{W}} = \boldsymbol{a}_{2\times 1}\boldsymbol{b}_{1\times 2}, \tag{8.2}$$

e considerando, sem perda de generalidade, que o vector $\boldsymbol{b}$ tem norma unitária e é descrito em termos de um ângulo θ, de acordo com a equação (8.3),

$$\boldsymbol{b} = [\cos\theta \ \sin\theta], \tag{8.3}$$

o erro de Frobenius pode ser expresso em termos de um único parâmetro, θ, através da equação (8.4),

$$erro(\theta) = \| \left(\boldsymbol{W} - a_{(\boldsymbol{W},\theta)}[\cos\theta \ \sin\theta] \right) \odot \boldsymbol{M}\|_F. \tag{8.4}$$

Pretende-se estimar a matriz $\widehat{\boldsymbol{W}}$ de característica 1 que melhor aproxima a matriz de observação $\boldsymbol{W}$ nas posições conhecidas. Note-se que para cada conjunto de 3 entradas duma matriz 2×2 existe sempre um quarto valor que torna a característica igual a 1, ou seja, é sempre possível encontrar uma matriz $\widehat{\boldsymbol{W}}$ de característica 1 e exactamente igual a W nas posições conhecidas, i.e., cujo erro, dado por (8.4), é 0.

Os exemplos seguintes ilustram o impacto da inicialização no comportamento dos algoritmos para este caso mais simples, com experiências que usam as mesmas observações, que estão indicadas na equação (8.5),

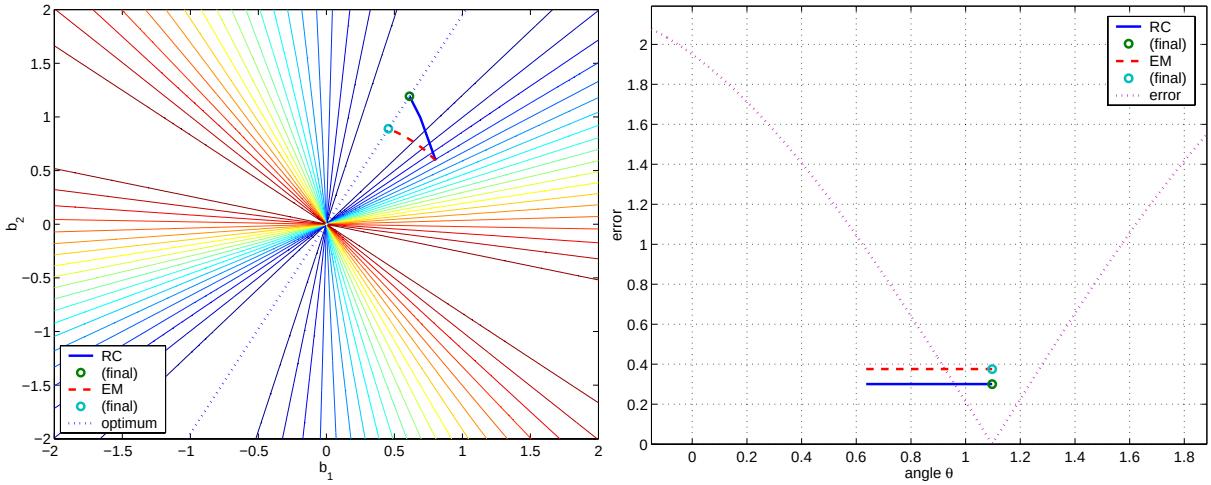
$$\widehat{\boldsymbol{W}}_{ideal} = \begin{bmatrix} -1 & -1.95 \\ 2 & 3.9 \end{bmatrix}, \quad \boldsymbol{M} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad \boldsymbol{W} = \begin{bmatrix} -1 & -1.95 \\ 2 & ? \end{bmatrix}, \tag{8.5}$$

onde *?* representa a entrada desconhecida w_{22} da matriz de observação $\boldsymbol{W}$.

Usando a estimativa inicial para $\widehat{\boldsymbol{W}}$, dada pela equação (8.6),

$$\widehat{\boldsymbol{W}}_{(0)} = \begin{bmatrix} -1 & -1.95 \\ 2 & 0 \end{bmatrix}, \tag{8.6}$$

descreve-se a evolução da estimativa $\widehat{\boldsymbol{W}}_{(k)}$ mostrando duas representações equivalentes: (i) o espaço das linhas $\boldsymbol{b} = [b_1\ b_2]$; e (ii) o ângulo correspondente, definido da forma descrita acima ($\theta = \arctan(b_2/b_1)$). A figura 8.1.(a) mostra as curvas de nível da função de erro em função do vector $\boldsymbol{b} = [b_1\ b_2]$, às quais se sobrepõe a evolução das estimativas de $\boldsymbol{b}$ para os algoritmos iterativos EM (linha a tracejado) e RC (linha sólida). Nesta figura, a linha a pontilhado (óptima) são aos vectores a que corresponde erro de estimação nulo. A figura 8.1.(b) representa a mesma função de erro, agora em função de θ, tal como definida em (8.4), à qual se sobrepõe os ângulos θ estimados pelos algoritmos EM (linha a tracejado) e RC (linha sólida).



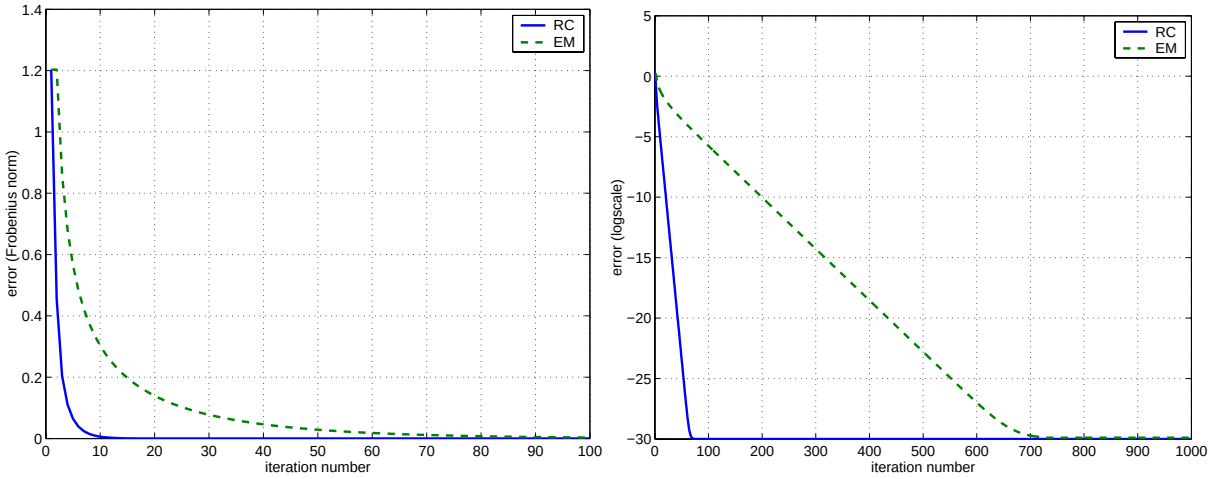
(a) trajectória do vector das linhas estimado; (b) trajectória do ângulo estimado

Figura 8.1: Trajectórias de convergência para funcionamento típico dos algoritmos EM e RC.

Analisando a figura 8.1.(a), vemos que as trajectórias de ambos algoritmos iterativos começam no mesmo ponto (por terem a mesma inicialização), e convergem para pontos na linha óptima. Tal como esperado, as estimativas do espaço das colunas dadas pelo algoritmo EM têm norma unitária (devido à normalização efectuada pela SVD, pois o vector $\boldsymbol{b}$ é igual ao vector $\boldsymbol{V}_1^T$ do passo de *maximization*) enquanto as estimativas dadas pelo algoritmo RC não têm. O bom comportamento de ambos os algoritmos é confirmado pela figura 8.1.(b), em que os ângulos θ estimados pelos algoritmos convergem para o ângulo θ que minimiza a função de erro da equação (8.4).

De forma a avaliar a velocidade de convergência, a figura 8.2 mostra o erro de estimação em função da iteração de ambos os algoritmos, tanto em escala linear (figuras 8.2.(a)) como em escala logarítmica (figuras 8.2.(b)). A figura 8.2 mostra uma convergência mais rápida do algoritmo RC face ao algoritmo EM.

De forma a ilustrar uma limitação na convergência do algoritmo EM e considerando-se os dados definidos pela equação (8.5), define-se a estimativa inicial, $\widehat{\boldsymbol{W}}_{(0)}$ através



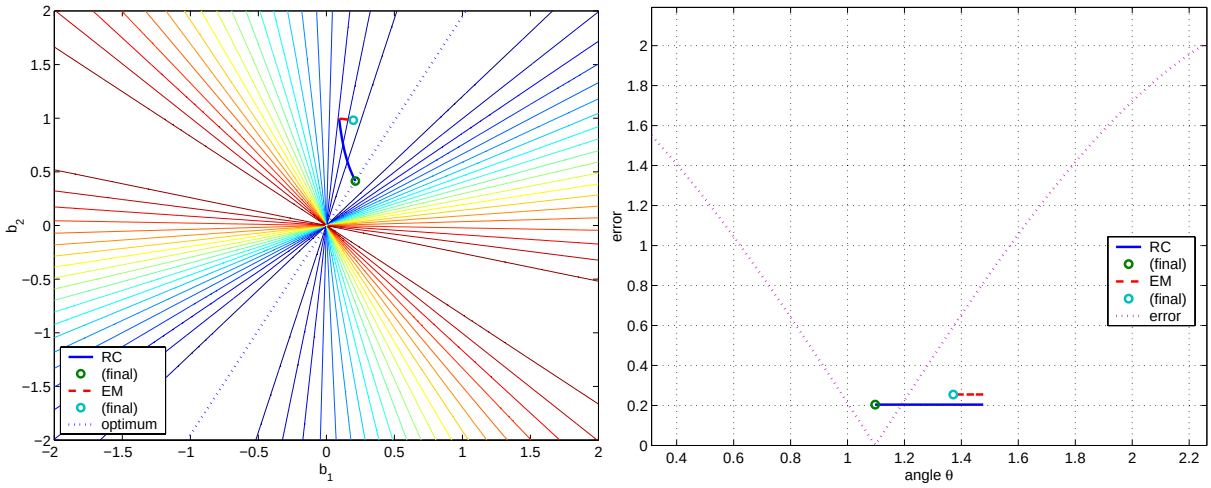
(a) escala linear; (b) escala logaritmica

Figura 8.2: Erro de estimação para funcionamento típico dos algoritmos EM e RC.

da equação (8.7),

$$\widehat{\boldsymbol{W}}_{(0)} = \begin{bmatrix} -1 & -1.95 \\ 2 & 22 \end{bmatrix}. \tag{8.7}$$

Executando 1000 iterações de ambos algoritmos iterativos, descreve-se a sua evolução na figura 8.3, da mesma forma do que na figura 8.1.

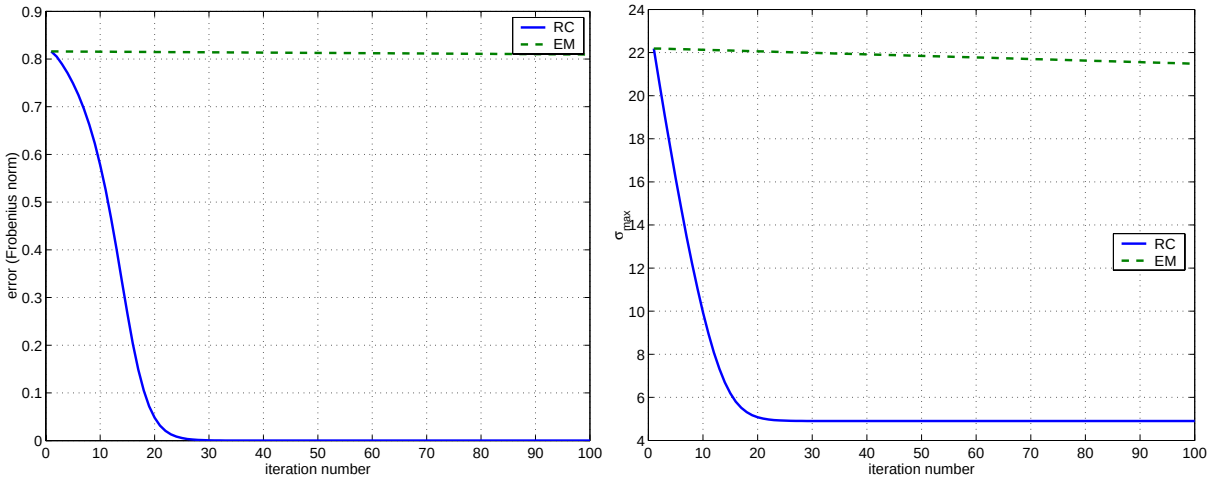


(a) trajectória do vector das linhas estimado; (b) trajectória do ângulo estimado

Figura 8.3: Trajectórias para grandes inicializações dos algoritmos EM e RC.

Analisando a figura 8.3 vê-se que, enquanto a estimativa dada pelo algoritmo RC converge para a linha óptima, i.e., para o ângulo θ tal que $erro(\theta) = 0$, a estimativa dada pelo algoritmo EM praticamente não se altera ao longo das iterações, não convergindo para a solução óptima. Este resultado é confirmado pela figura 8.4.(a), que mostra o erro de estimação em função do número de iterações, onde se vê que o

erro da estimativa dada pelo algoritmo RC decresce para zero em poucas iterações, enquanto o erro da estimativa do algoritmo EM praticamente não decresce ao longo das 1000 iterações dos algoritmos.



(a) erro de estimação; (b) maior valor singular matrizes $\widehat{\boldsymbol{W}}$

Figura 8.4: Análise para grandes inicializações dos algoritmos EM e RC.

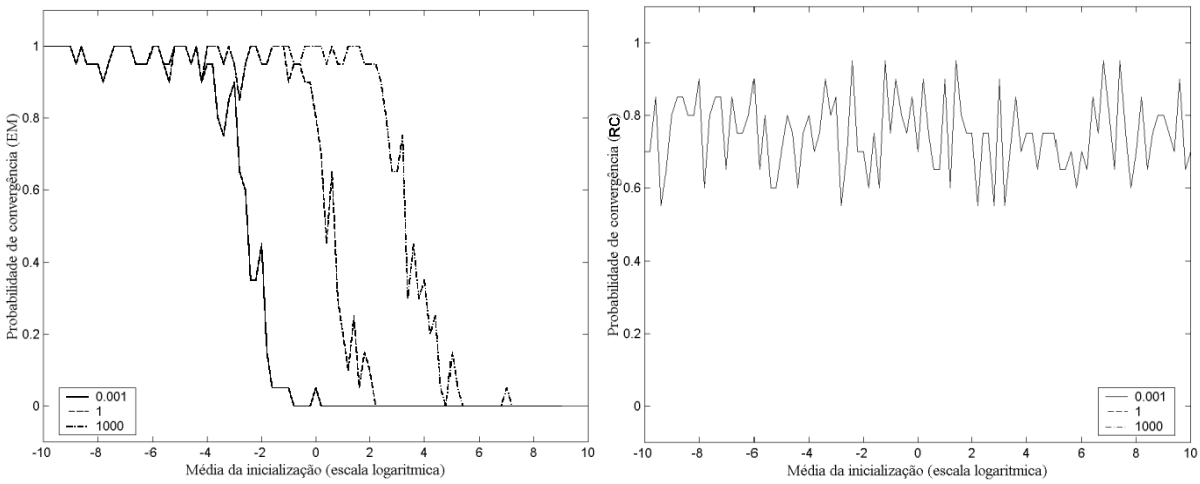
O mau comportamento do algoritmo EM, previsto teoricamente, deve-se à estimativa inicial de $\widehat{w}_{22}$, que tem uma ordem de grandeza maior do que a dos dados. Isto acontece porque a matriz de característica 1 estimada pelo algoritmo EM na primeira iteração está associada ao maior valor singular de $\widehat{\boldsymbol{W}}_{(0)}$, que é dominado pelo alto valor da inicialização $\widehat{w}_{22} = 22$ (note-se que enquanto os valores singulares da matriz $\widehat{\boldsymbol{W}}_{ideal}$ são $\sigma_1(\widehat{\boldsymbol{W}}_{ideal}) \simeq 4.9$ e $\sigma_2(\widehat{\boldsymbol{W}}_{ideal}) \simeq 0$, os valores singulares da estimativa inicial são $\sigma_1(\widehat{\boldsymbol{W}}_{(0)}) \simeq 22.2$ e $\sigma_2(\widehat{\boldsymbol{W}}_{(0)}) \simeq 0.8$ devido ao valor da inicialização, $\widehat{w}_{22} = 22$). Em seguida, o algoritmo EM substitui a estimativa inicial de $\widehat{w}_{22}$ por outra mais recente, obtendo-se outra estimativa muito próxima da anterior. Para melhor ilustrar a lenta convergência do algoritmo EM, a figura 8.4.(b) mostra a evolução do maior valor singular das estimativas $\widehat{\boldsymbol{W}}_{(k)}$ dadas por ambos algoritmos iterativos, em função da iteração. Tal como seria de esperar, o maior valor singular da estimativa dada pelo algoritmo RC converge rapidamente para o maior valor singular da solução, $\sigma_1(\widehat{\boldsymbol{W}}_{ideal}) \simeq 4.9$; enquanto o maior valor singular da estimativa dada pelo algoritmo EM decresce muito lentamente a partir do seu valor inicial, $\sigma_1(\widehat{\boldsymbol{W}}_{(0)}) \simeq 22.2$.

O comportamento descrito não ocorre apenas em casos em que os valores da inicialização são muito grandes face aos dados. De facto, este comportamento também foi observado em situações em que a matriz de observação tem valores muito grandes em linhas ou colunas em que existem entradas desconhecidas. Nestas condições, devido aos altos valores observados, mesmo pequenos valores da inicialização têm um grande impacto nos vectores do espaço das linhas e das colunas estimados com o algoritmo EM. Estes levam a uma convergência lenta semelhante à ilustrada nas figuras 8.3 e 8.4.

Nas subsecções seguintes estuda-se o comportamento dos algoritmos apresentados nos capítulos 5 e 6 para matrizes de maiores dimensões e características, em casos mais próximos das aplicações reais para estes algoritmos.

8.1.3 Influência da inicialização

Nesta experiência, foram criadas matrizes $\boldsymbol{W}$, 24×24, de característica 4, com 70% de informação desconhecida, cuja média do módulo dos dados é 0.001, 1 e 1000. Os algoritmos iterativos EM e RC foram inicializados com matrizes aleatórias cuja média do módulo é variável. A figura 8.5 mostra a probabilidade de convergência estimada experimentalmente de cada um dos algoritmos em função da média do módulo das inicializações aleatórias.



(a) algoritmo iterativo EM; (b) algoritmo iterativo RC

Figura 8.5: Probabilidade de convergência estimada dos algoritmos EM e RC, em função da média do módulo de inicializações aleatórias.

Analisando a figura 8.5 é possível retirar várias conclusões. Verifica-se que o algoritmo EM deixa de convergir quando o módulo da inicialização é da mesma ordem de grandeza que os dados em $\boldsymbol{W}$, ou superior, o que está de acordo com que o que foi previsto teoricamente, uma vez que os cálculos da SVD no passo de *maximization* passam a ser dominados pela inicialização, o que faz com que os dados conhecidos sejam desprezados. A probabilidade de convergência do algoritmo RC situa-se na ordem dos 75%, sendo independente na média da inicialização, o que também está de acordo com o que foi previsto teoricamente. O algoritmo RC não converge sempre porque as matrizes de inicialização não têm sempre informação suficientemente diversificada nas posições conhecidas da matriz $\boldsymbol{W}$ para evitar a inversão de matrizes próximas da singularidade, o que provoca a divergência do algoritmo.

Esta experiência mostra que o diferente modo de funcionamento dos dois algoritmos iterativos tem importantes consequências no seu desempenho. A probabilidade de convergência estimada experimentalmente na figura 8.5 ilustra o desempenho dos

algoritmos do ponto de vista da fiabilidade em funcionamento independente (i.e., sem o algoritmo de inicialização), em função da média da inicialização e da diversidade dos dados. Deste ponto de vista, o algoritmo EM é o mais indicado, por convergir quase sempre se a média da inicialização for menor do que a média dos dados, o que acontece se a matriz de inicialização $\widehat{\boldsymbol{W}}_{(0)}$ for nula. A convergência deste algoritmo não depende da diversidade dos dados ou da inicialização e não é tão dependente da quantidade de informação desconhecida como o algoritmo RC. O algoritmo RC funciona bem para os casos em que a matriz dos dados e de inicialização tem muita diversidade, sendo independente das ordens de grandeza das respectivas matrizes.

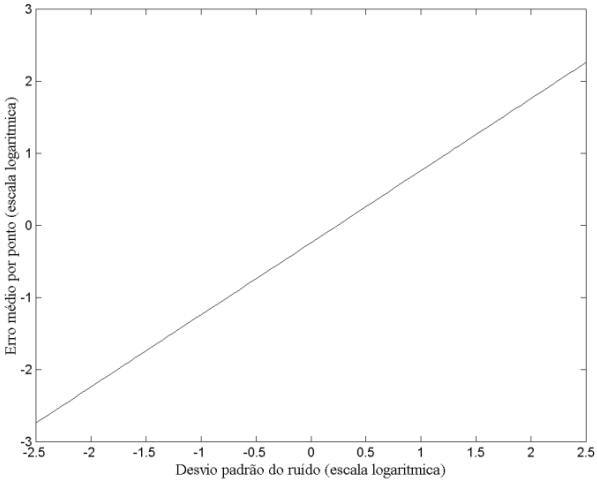
Inicializando ambos os algoritmos iterativos com o algoritmo que calcula a estimativa inicial da matriz $\widehat{\boldsymbol{W}}$ ideal, descrito no capítulo 5, é notória a influência da inicialização no desempenho dos algoritmos iterativos. Em todas as experiências efectuadas, o erro médio por ponto (obtido através da equação (8.1)) é da ordem da resolução numérica do computador quando a matriz de observação não tem ruído. Quando esta tem ruído, a estimativa sub-óptima dada pelo algoritmo que calcula a estimativa inicial de $\widehat{\boldsymbol{W}}$ está bastante próxima do resultado final, fazendo com que os algoritmos iterativos convirjam 100% das vezes e para a solução óptima, em muitos poucas iterações - tipicamente menos de 10. Os resultados obtidos nesta experiência ilustram os resultados obtidos noutras experiências semelhantes, com matrizes de outras dimensões, características e percentagens de informação desconhecida.

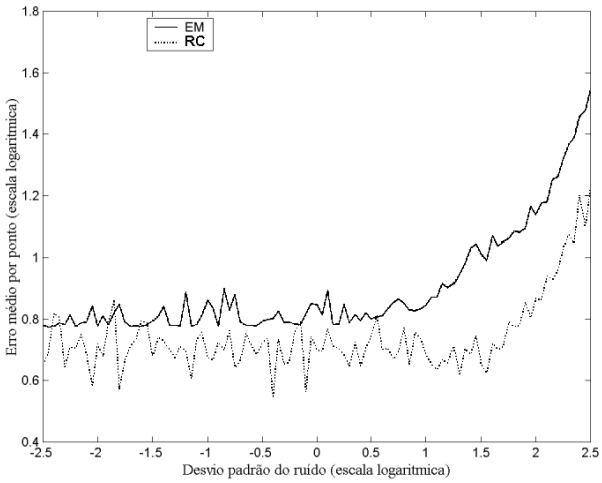
8.1.4 Sensibilidade ao ruído

Define-se uma matriz de observação não-registada sem ruído $\boldsymbol{W}_{\sigma=0}$ de dimensão 24×24 e característica 4, com 70% de informação desconhecida e em que a média do módulo dos dados é 1. De forma a simular situações da vida real em que a matriz de observação está contaminada com ruído, soma-se ruído gaussiano de média nula e desvio padrão σ à matriz de observação $\boldsymbol{W}_{\sigma=0}$, obtendo-se a matriz $\boldsymbol{W}_{\sigma}$.

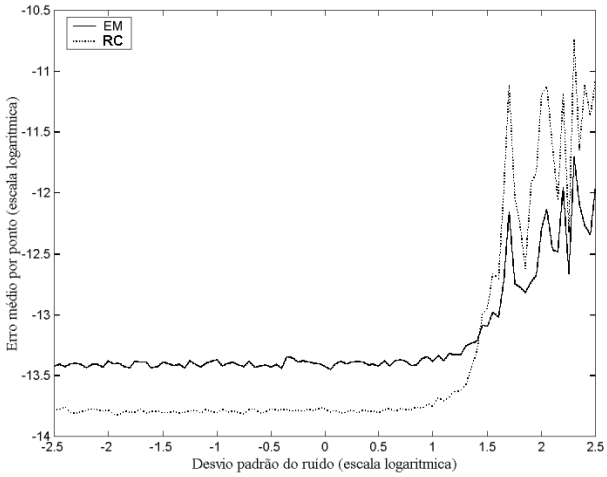
A figura 8.6 mostra o erro médio por ponto dado pela equação (8.1), em função de σ. O erro médio por ponto é obtido: (a) usando apenas o algoritmo que calcula a estimativa inicial; (b) usando apenas os algoritmos iterativos inicializados aleatoriamente com matrizes cuja média do módulo é 0.05 (valor em que ambos os algoritmos convergem), após 20 iterações; (c) usando o algoritmo completo, i.e., usando o algoritmo que calcula a estimativa inicial seguido de 20 iterações de ambos os algoritmos iterativos.

Notando que a inclinação da recta na figura 8.6.(a) é 1, verifica-se que o erro médio por ponto da estimativa inicial depende linearmente do desvio padrão do ruído σ, como seria de esperar dado que o funcionamento deste algoritmo é baseado numa estratégia de mínimos quadrados linear. Analisando a figura 8.6.(b), verifica-se um erro médio por ponto aproximadamente constante ao fim de 20 iterações para cada um dos algoritmos iterativos, até se verificar um atraso na convergência para valores mais altos do desvio padrão do ruído, que varia entre $10^{-2.5}$ e $10^{2.5}$. Na figura





(a) algoritmo da estimativa inicial; (b) algoritmos iterativos, com inicialização aleatória



(c) algoritmo completo: algoritmo da estimativa inicial + algoritmos iterativos

Figura 8.6: Erro médio por ponto em função do σ do ruído.

8.6.(c) verifica-se que a integração do algoritmo que calcula a estimativa inicial com os algoritmos iterativos permite erros médios por ponto extremamente reduzidos, neste caso da ordem de 10^{-11} com apenas 20 iterações dos algoritmos iterativos, o que confirma a eficácia do algoritmo total, mesmo quando o desvio padrão do erro adicionado é $10^{2.5}$ vezes maior do que a média do módulo dos próprios dados. Naturalmente, num caso real os resultados seriam inferiores uma vez que o ruído no seguimento dos pontos só pode ser considerado como gaussiano por aproximação. A figura 8.6.(b,c) mostra também que o erro médio por ponto do algoritmo RC é em geral menor do que o erro do algoritmo EM.

Os testes efectuados com outras matrizes apenas mostraram um decréscimo na velocidade de convergência, face a $\boldsymbol{W}_{\sigma=0}$, para matrizes $\boldsymbol{W}_{\sigma}$ com valores muito altos de σ, em que σ é o desvio padrão do ruído somado a $\boldsymbol{W}_{\sigma=0}$. Para valores normais do desvio padrão do ruído, a velocidade de convergência é aproximadamente a mesma de quando não existe ruído, como se verifica também através da figura 8.6(b), em

que o erro médio por ponto é aproximadamente constante para valores baixos do desvio padrão do ruído somado à matriz $\boldsymbol{W}_{\sigma=0}$.

A importância da inicialização é ainda maior na presença de ruído, uma vez que, além deste piorar a velocidade de convergência dos algoritmos iterativos, pode reduzir a sua probabilidade de convergência. A filtragem ao ruído efectuada pela estratégia de mínimos quadrados do algoritmo que calcula a estimativa inicial torna-o ideal para este caso.

8.1.5 Sensibilidade à informação desconhecida

Define-se uma matriz de observação não-registada sem ruído $\boldsymbol{W}_{\sigma=0}$ de dimensão 24×24 e característica 4, em que a média do módulo dos dados é 1, em que a quantidade de informação desconhecida é variável. A esta matriz é somado ruído gaussiano de média nula e desvio padrão 1, obtendo-se a matriz $\boldsymbol{W}_{\sigma=1}$. A porção desconhecida da matriz varia desde a máxima possível, $(24-r)\times(24-r) = 20\times 20$ (70%), até à situação em que não existem posições desconhecidas, variando-se o número de linhas conhecidas de $\boldsymbol{W}$, N. Ou seja, a parte desconhecida da matriz $\boldsymbol{W}_{\sigma=1}$ tem tamanho $(24-N)\times 20$, em que $N = 4,..,24$. A figura 8.7mostra o erro médio por ponto após 20 iterações dos algoritmos iterativos EM e RC, inicializados com matrizes cuja média do módulo é 0.05, em função do número de linhas conhecidas de $\boldsymbol{W}_{\sigma=1}$.

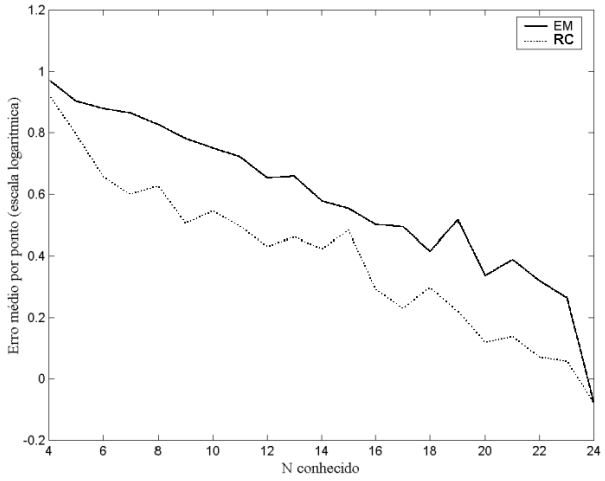


Figura 8.7: Erro médio por ponto dos algoritmos EM e RC em função do número de linhas conhecidas N da matriz $\boldsymbol{W}_{\sigma=1}$.

Como seria de esperar, a figura 8.7 mostra a diminuição do erro médio por ponto para ambos os algoritmos iterativos à medida que o número de linhas conhecidas aumenta, ou melhor, à medida que a informação desconhecida diminui. Verifica-se mais uma vez que o erro médio por ponto do algoritmo RC é inferior ao algoritmo RC.

A quantidade de informação desconhecida afecta ambos os algoritmos. Para o algoritmo EM, uma maior quantidade de informação desconhecida faz com que a

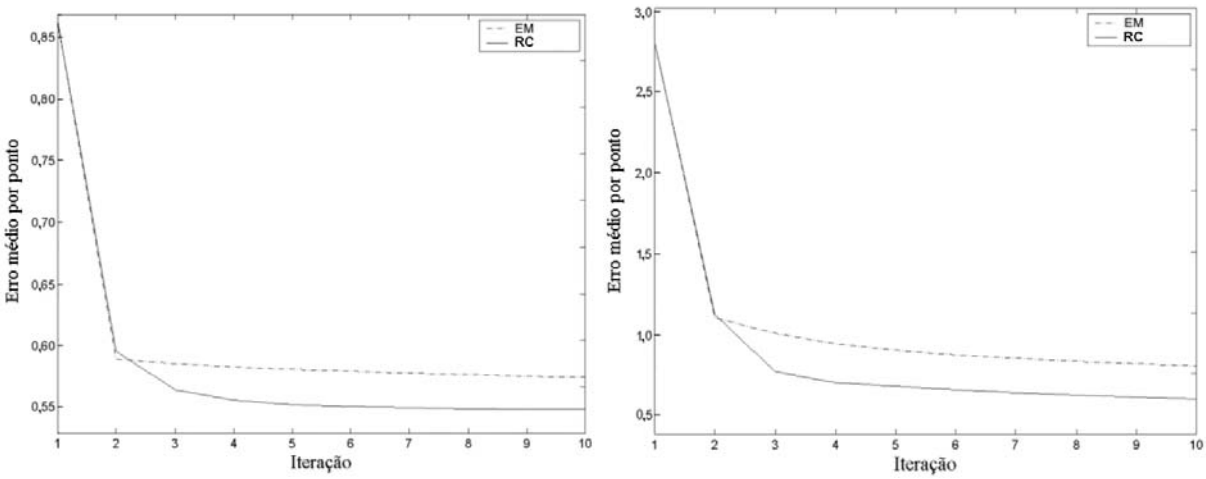
inicialização tenha maior peso nos valores próprios dominantes da matriz $\overline{\boldsymbol{W}}$, de onde se extraem os valores singulares dominantes no passo de *maximization*, o que pode piorar a velocidade de convergência ou mesmo fazer com que o algoritmo divirja. Para o algoritmo RC, uma maior quantidade de informação desconhecida implica uma maior quantidade de linhas e/ou colunas anuladas nas matrizes do algoritmo, o que reduz a diversidade das matrizes que são invertidas, dando origem a problemas numéricos que pioram a velocidade de convergência ou fazem o algoritmo divergir.

Apesar da percentagem de informação conhecida ser importante no desempenho dos algoritmos, a experiência mostrou que a forma da matriz binária $\boldsymbol{M}$, que indica a disposição da informação conhecida nas matrizes $\boldsymbol{W}_\sigma$ também é relevante no sucesso deste método, tal como é abordado no capítulo 5 no contexto do algoritmo que calcula a estimativa inicial de $\widehat{\boldsymbol{W}}$. No caso prático de *structure from motion*, em os pontos entram e saem da imagem duma forma continua, a matriz $\boldsymbol{M}$ tem tipicamente a forma representada na figura 5.2, que permite estimar informação desconhecida com facilidade.

8.1.6 Influência da característica

As experiências efectuadas acerca da influência da característica da matriz de entrada no desempenho dos novos algoritmos aqui apresentados mostram que esta está relacionada com a influência da redundância dos dados duma matriz. Ou seja, para matrizes com a mesma dimensão e percentagem de informação desconhecida, maiores características da matriz de entrada implicam menor redundância dos seus dados, o que é equivalente a considerar que houve uma diminuição da percentagem de informação conhecida. Assim, tendo em conta que maiores características na matriz de entrada equivalem a maiores percentagens de informação desconhecida, o estudo da influência da característica reduz-se ao estudo efectuado na secção 8.1.5.

Para exemplificar o funcionamento destes algoritmos com uma característica diferente de 4, definiu-se uma matriz $\boldsymbol{W}_{\sigma=0}$ de dimensão 40×40, característica 6 e cuja média do módulo é 1, com 56% de informação desconhecida. A esta matriz foi somado ruído gaussiano de média nula e variância 1, obtendo-se $\boldsymbol{W}_{\sigma=1}$. A figura 8.8 mostra o erro médio por ponto em função do número de iterações, obtido: (a) com o método total; (b) apenas com os algoritmo iterativos, inicializados aleatoriamente com matrizes cuja média do módulo é 0.05.
Na figura 8.8, a iteração 1 representa a inicialização que, em 8.8(a), é dada pelo algoritmo que calcula a estimativa inicial; enquanto em 8.8(b) é obtida aleatoriamente. Assim, estão representadas 9 iterações dos algoritmos iterativos. A figura 8.8(a) mostra que o erro médio por ponto da estimativa sub-óptima obtida pelo algoritmo da estimativa inicial é já bastante reduzido, o que comprova a eficácia deste algoritmo, para qualquer característica. Esta figura mostra também a convergência dos algoritmos iterativos após um reduzido número de iterações. A figura 8.8(b) mostra também a convergência de ambos algoritmos iterativos, embora mais lenta,



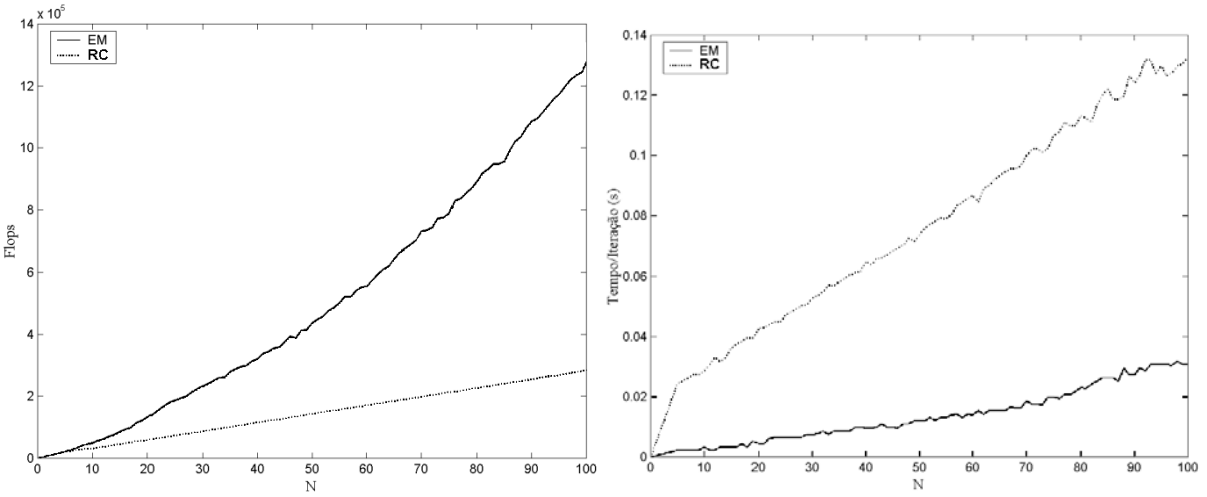
(a) método completo; (b) algoritmos iterativos, com inicialização aleatória

Figura 8.8: Erro médio por ponto em função do número de iterações, para $\boldsymbol{W}$ de característica 6.

por terem sido inicializados aleatoriamente.

8.1.7 Custo computacional

Define-se uma matriz de observação não-registada sem ruído, $\boldsymbol{W}_{\sigma=0}$, de dimensão $N \times 24$ e característica 4, em que só é conhecido o valor de 4 linhas e 4 colunas. Foi medido o número de operações de vírgula flutuante e o tempo por iteração, em função de N, para cada algoritmo iterativo. O resultado é apresentado na figura 8.9.



(a) operações em vírgula flutuante (*FLOating Point operationS*) por iteração; (b) tempo por iteração

Figura 8.9: Custo computacional dos algoritmos iterativos.

A figura 8.9(a) mostra que o número de operações de vírgula flutuante do algoritmo EM cresce exponencialmente com o número de linhas da matriz de observação. O

algoritmo RC necessita de um número de operações em vírgula flutuante menor do que o algoritmo EM, crescendo de forma linear com o número de linhas da matriz de observação. A figura 8.9(b) mostra que, apesar do número de operações de vírgula flutuante do algoritmo EM crescer de forma exponencial e ser sempre superior ao número de operações em vírgula flutuante do algoritmo RC, o software $MatLab^{©}$ necessita de menos tempo para calcular uma iteração do algoritmo EM do que uma iteração do algoritmo RC. Isto acontece por várias razões. Por um lado, a simples contagem do número de operações em vírgula flutuante não contabiliza a diferença de tempo no cálculo dos diferentes tipos de operações (por exemplo, a divisão é bastante mais lenta do que uma adição). Por outro lado, apesar do cálculo da SVD do algoritmo EM requerer muitas operações em vírgula flutuante, o seu cálculo está bastante optimizado no software $MatLab^{©}$, de forma a tirar grande partido dos recursos de hardware disponíveis. Assim, conclui-se que o algoritmo EM é computacionalmente mais pesado do que o algoritmo RC.

8.2 Método de factorização de Tomasi e Kanade.

Implementou-se o método de factorização de Tomasi e Kanade em $MatLab^{©}$ e fizeram-se várias experiências para estudar o seu comportamento, das quais se mostram as mais significativas.

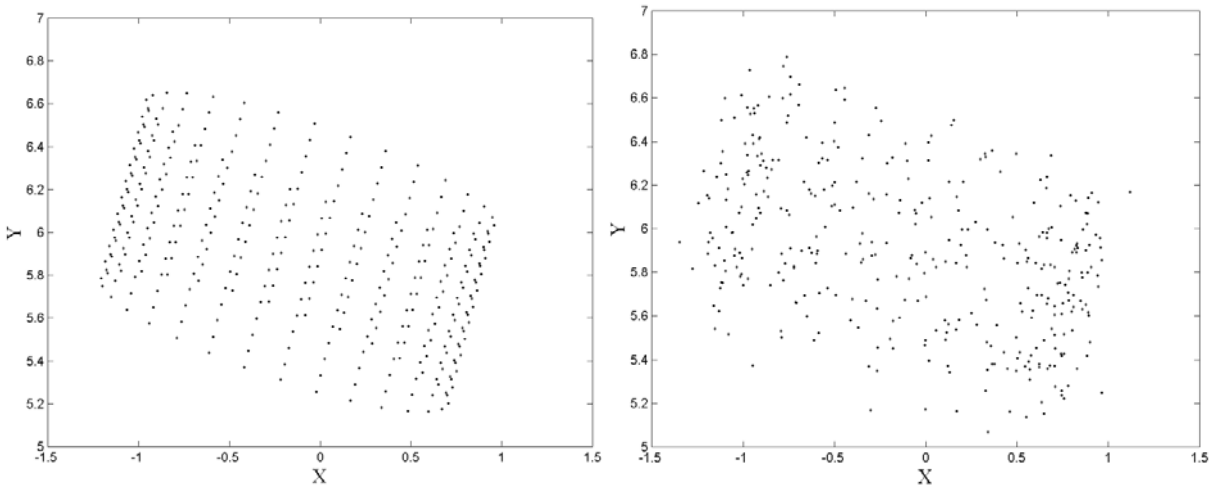
Nesta experiência usou-se uma sequência de vídeo artificial, em que as projecções 2D de 372 pontos da face de um cilindro em movimento no espaço são armazenadas numa matriz de observação $\boldsymbol{W}_{\sigma=0}$, sem ruído. Somando ruído gaussiano de média nula e desvio padrão $\sigma = 0.1$ à matriz $\boldsymbol{W}_{\sigma=0}$ obtém-se $\boldsymbol{W}_{\sigma=0.1}$. Executou-se o método de factorização às matrizes resultantes e obteve-se duas estimativas das matrizes de rotação e forma. A figura 8.10 mostra uma imagem da sequência de vídeo sem ruído, em (a), e com ruído de $\sigma = 0.1$, em (b); e as matrizes de forma resultante da factorização da matriz $\boldsymbol{W}_{\sigma=0}$ sem ruído, em (c), e da matriz $\boldsymbol{W}_{\sigma=0.1}$, em (d).

A figura 8.10 mostra o correcto funcionamento do método de factorização de Tomasi e Kanade na recuperação da estrutura tridimensional dos objectos filmados (e, consequentemente, do movimento tridimensional da câmara de video), mesmo na presença de bastante ruído face à ordem de grandeza dos dados, como é visível na figura 8.10(b) e 8.10(d).

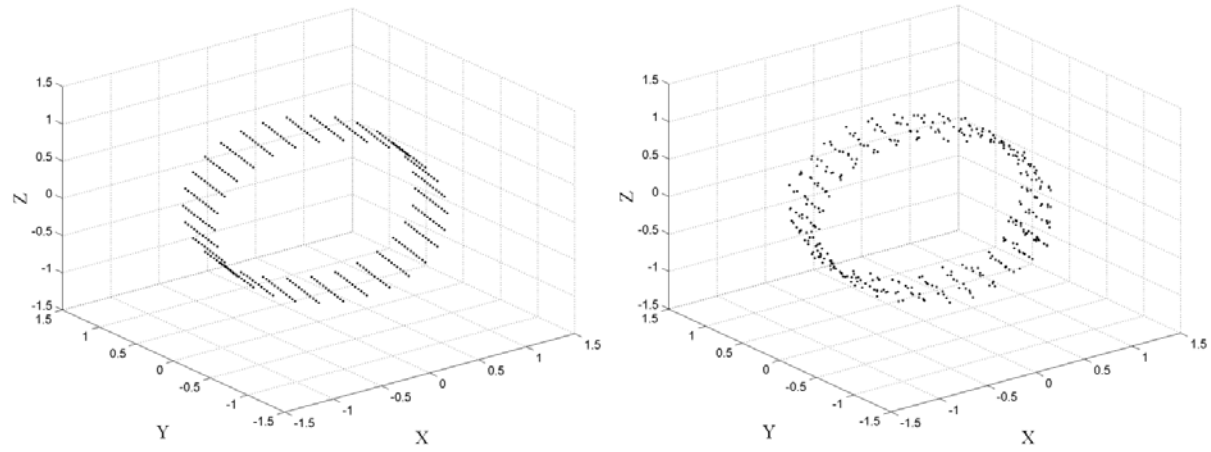
Definindo o erro médio por ponto através da equação (8.8),

$$E\{erro/ponto\} = \frac{\|\boldsymbol{W}_{\sigma=0} - \boldsymbol{R}\boldsymbol{S}^T\|_F}{\sqrt{2F \times P}}, \tag{8.8}$$

a figura 8.11 mostra a diminuição do erro médio por ponto em função do número de imagens da sequência de video, para vários valores do desvio padrão σ do ruído de média nula que é somado à matriz $\boldsymbol{W}_{sigma=0}$ da sequência de video da experiência



(a) imagem sem ruído; (b) imagem com ruído $\sigma = 0.1$

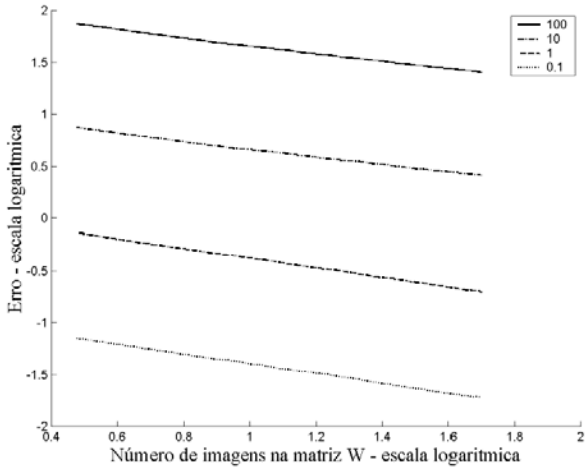


(c) reconstrução sem ruído; (d) reconstrução com ruído

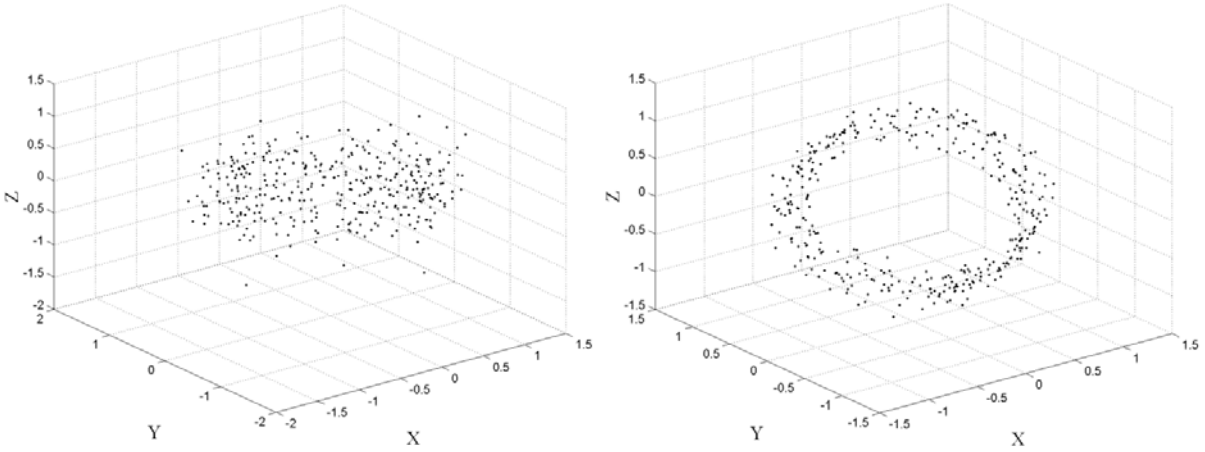
Figura 8.10: Reconstrução de um cilindro artificial.

anterior (o desvio padrão do ruído que foi somado à matriz $\boldsymbol{W}_{\sigma=0}$ está indicado na legenda das figuras).

Esta experiência comprova, tal como seria de esperar, que existe uma maior imunidade ao ruído com o aumento do número de imagens, por haverem $2P$ equações novas por cada imagem, com um acréscimo de apenas 8 novas incógnitas por imagem, correspondentes ao movimento da câmara. Este facto torna os cálculos ainda mais sobre-determinados, o que permite uma melhor filtragem do ruído no cálculo da SVD e nos outros passos do método de factorização. Em particular, as figuras 8.11(b) e (c) mostram uma melhoria significativa na matriz de forma quando se considera a mesma sequência de vídeo com 40 ou 400 imagens, com ruído branco de média nula e desvio padrão de um. A figura 8.11(a) mostra também uma diminuição linear do logaritmo do erro médio por ponto, em função do logaritmo do número de imagens na sequência de vídeo. Assim, é possível estimar o erro médio por ponto da imagem f, $E\{erro/ponto\}_f$, a partir da estimativa do erro médio por ponto da



(a) erro médio por ponto 8.8 em função do número de imagens da sequência de video.



(b) matriz de forma (40 imagens, $\sigma = 1$); (c) matriz de forma (400 imagens, $\sigma = 1$)

Figura 8.11: Imunidade ao ruído em função do número de imagens da sequência de video.

imagem f_0, $E\{erro/ponto\}_{f0}$, através da equação (8.9),

$$\begin{cases} \log\left(E\{erro/ponto\}_f\right) = \log\left(E\{erro/ponto\}_{f0}\right) + m\log\left(f - f_0\right) \\ m \approx -0.5 \end{cases} \iff \tag{8.9}$$

$$\iff E\{erro/ponto\}_f = \frac{E\{erro/ponto\}_{f0}}{\sqrt{f - f_0}},$$

em que m é a inclinação da recta da figura 8.11(a).

A aplicação deste método a sequências de video reais é feita na secção 8.3 e no capítulo 9. Em ambos os casos, este método é precedido dos novos algoritmos apresentados nesta tese, que estimam as trajectórias desconhecidas dos pontos seguidos nas imagens, que têm demonstrado serem vitais para a aplicação do método de factorização de Tomasi e Kanade a sequências de video reais, onde existe muita oclusão. No capítulo 9 comparam-se resultados obtidos apenas com o método de Tomasi e Kanade, onde se unem matrizes de forma obtidas independentemente; e

com este método precedido dos novos algoritmos, onde se faz uma estimação global das matrizes de forma e rotação.

8.3 Factorização com informação desconhecida

8.3.1 Dados artificiais

Os novos algoritmos que estimam informação desconhecida em matrizes singulares foram usados em sequências de video artificiais em que existe oclusão, para estimar matrizes de observação completas $\widehat{\boldsymbol{W}}$ a partir de matrizes de observação incompletas $\boldsymbol{W}$ obtidas das sequências de video, de forma a poder-se usar o método de factorização de Tomasi e Kanade para obter as matrizes de forma e de rotação.

A figura 8.12 mostra uma das 50 imagens duma sequência de video artificial (semelhante à sequência em 8.2), em que se vê a projecção bidimensional de um total de 372 pontos localizados na face de um cilindro em movimento num espaço tridimensional, mas com oclusão. A matriz $\boldsymbol{W}_{\sigma=0.1}$ total, de dimensão 100×372, contém 9724 elementos desconhecidos (26%) e ruído gaussiano de média nula e desvio padrão $\sigma = 0.1$. Na figura 8.12, os círculos representam pontos sem ruído e as pintas representam pontos com ruído usados para obter SFM.

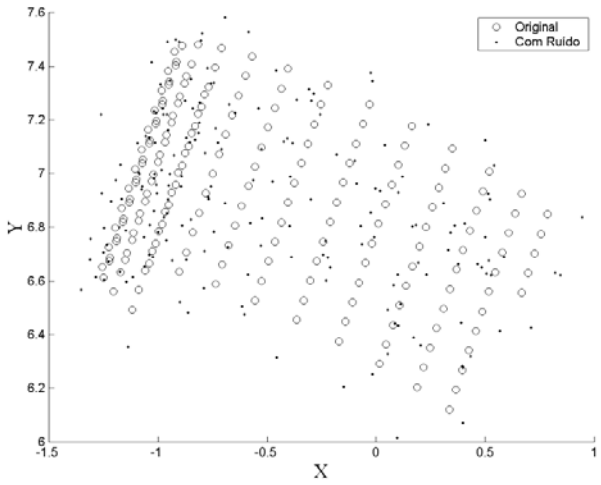
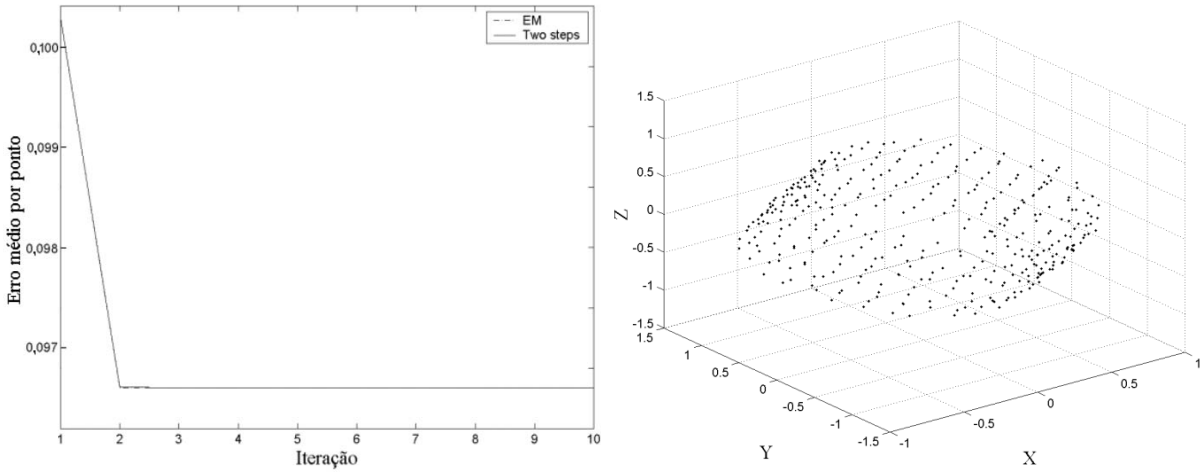


Figura 8.12: Imagem duma sequência de video artificial dum cilindro, com oclusão.

Usando o método completo desenvolvido, i.e., o algoritmo que calcula a estimativa inicial e os dois algoritmos iterativos, obtiveram-se duas estimativas de $\boldsymbol{W}$ completa, $\widehat{\boldsymbol{W}}_{EM}$ e $\widehat{\boldsymbol{W}}_{RC}$, oriundos de cada um dos algoritmos iterativos, cujo erro médio por ponto ao longo das iterações está indicado na figura 8.13.(a). A partir desta estimativa foi executado o método de factorização descrito no capítulo 4, recuperando-se a forma indicada na figura 8.13.(b).

A figura 8.13.(a), em que a primeira iteração representa novamente o algoritmo que calcula a estimativa inicial, volta a sublinhar o importante papel da estimativa



(a) erro médio por ponto, por iteração; (b) matriz de forma obtida

Figura 8.13: SFM com informação desconhecida duma sequência de video artificial dum cilindro.

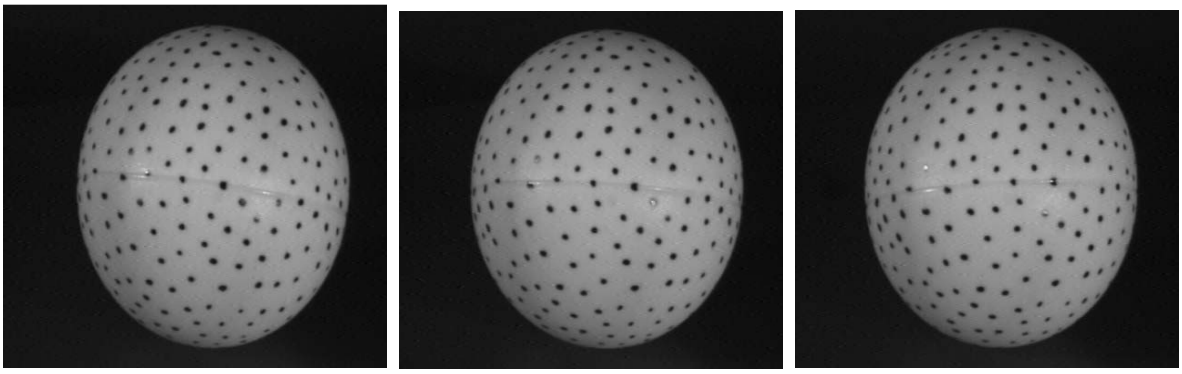
inicial na estimação final –tão relevante que basta apenas uma só iteração dos algoritmos iterativos para que estes convirjam para a matriz $\widehat{\boldsymbol{W}}$ que melhor aproxima a matriz de observação $\boldsymbol{W}$ nas posições conhecidas. Na figura 8.13.(b), é visível que o método completo, que transforma o problema SFM com informação desconhecida no problema SFM, em conjunto com o método de factorização de Tomasi e Kanade, estimam correctamente a forma tridimensional do cilindro (e, consequentemente, o movimento da câmara), mesmo na presença de ruído.

8.3.2 Dados reais

Para testar a aplicação dos novos algoritmos a caso real em que exista oclusão, foram usadas várias sequências de vídeo. A primeira foi obtida na página de visão por computador [47] e contém 52 imagens obtidas por uma câmara a rodar na horizontal em torno duma bola de *ping-pong* com pequenos pontos pintados, como é visível na figura 8.14.

Nesta sequência de video foi usado um seguidor de pontos notórios para seguir a trajectória dos pontos pretos pintados na bola de *ping-pong*. Ao longo das 52 imagens da sequência de vídeo foram seguidos no total 64 pontos notórios. Devido à rotação da câmara, foi obtida uma matriz de observação não-registada $\boldsymbol{W}$ com 41% de elementos desconhecidos. Usando o método completo descrito nos capítulos 5 e 6 para estimar uma matriz de observação completa, $\widehat{\boldsymbol{W}}$, aplicou-se o método de factorização de Tomasi e Kanade à matriz obtida e estimou-se uma matriz de forma. A figura 8.15 mostra a forma tridimensional recuperada, em que os pontos foram acopulados através de um procedimento de *mesh*.

A figura 8.15 mostra que a forma esférica da bola de *ping-pong* foi recuperada com sucesso. A superfície recuperada está incompleta porque a sequência de video filmou



(a) imagem 1; (b) imagem 5; (c) imagem 9

Figura 8.14: Imagens duma sequência de video real com uma bola de *ping-pong*.

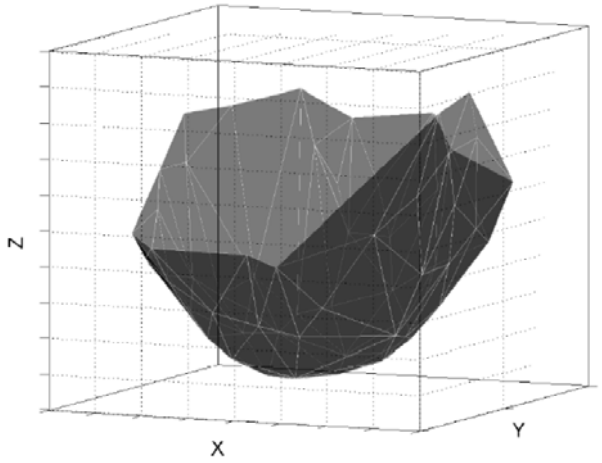


Figura 8.15: Superfície esférica obtida da bola de *ping-pong*.

apenas cerca de metade da bola de *ping-pong*. A estimação da estrutura tridimensional da bola de *ping-pong*, usando o método de factorização de Tomasi e Kanade, seria bastante dificultada sem a utilização de uma estratégia global para lidar com a oclusão, uma vez que os pontos notórios entram e saem da imagem continuamente.

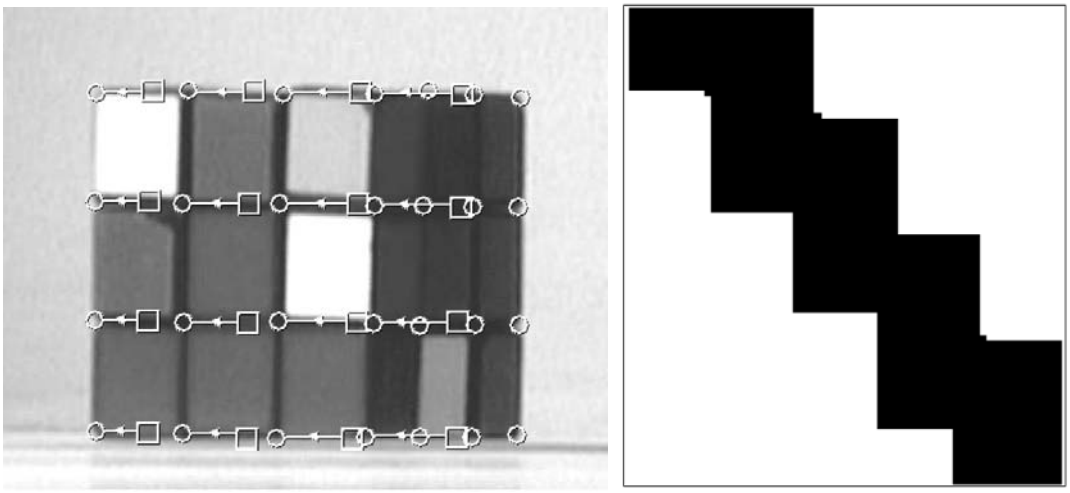
Capítulo 9

APLICAÇÕES

9.1 Modelos 3D para realidade virtual

Para demonstrar a aplicação prática dos novos algoritmos aqui apresentados, foram seguidos pontos notórios numa sequência de video com 85 imagens que mostra um *cubo mágico* a rodar na horizontal. A figura 9.1.(a) ilustra o seguimento de pontos notórios nesta sequência de video, em que os quadrados correspondem às posições anteriores dos pontos notórios e os circulos correspondem à posição dos mesmos pontos nótorios na imagem actual.

As projecções bidimensionais dos pontos notórios seguidos são armazenados na matriz de observação $\boldsymbol{W}$. Devido à entrada e saída de pontos notórios da imagem, devido à oclusão, a matriz $\boldsymbol{W}$ tem informação desconhecida, assumindo a forma da figura 9.1.(b), em que os pontos escuros correspondem a entradas conhecidas da matriz $\boldsymbol{W}$ e os claros, a posições desconhecidas. A matriz $\boldsymbol{W}$, de dimensão 170×64, foi obtida com 85 imagens, tendo sido seguidos um total de 64 pontos notórios. A matriz de observação tem um total de 62% de informação desconhecida.

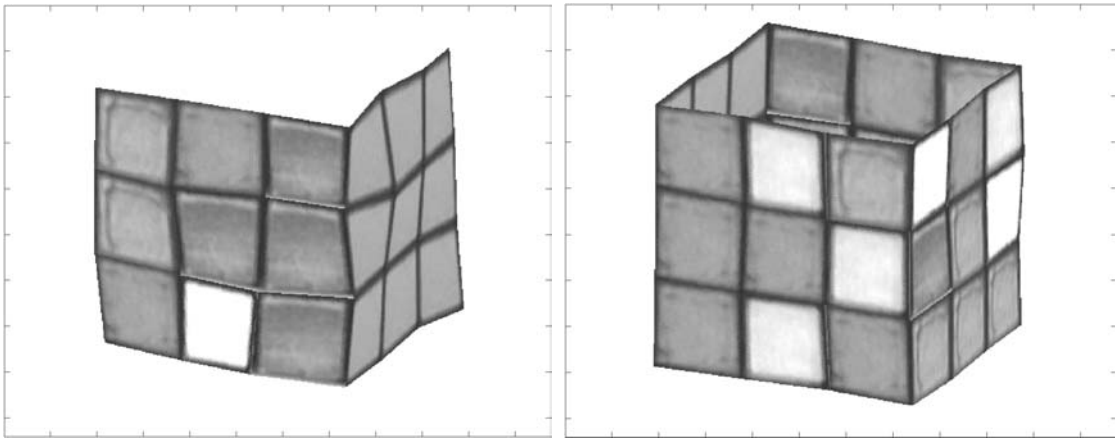


(a) trajectórias de pontos notórios;(b) forma da matriz $\boldsymbol{W}$ (pontos escuros=dados conhecidos)

Figura 9.1: Ilustração da sequência de video com o *cubo mágico*.

Aplicando o novo método que permite estimar as trajectórias completas a partir de trajectórias parciais à matriz de observação $\boldsymbol{W}$, obteve-se uma estimativa da matriz completa, $\widehat{\boldsymbol{W}}$, que permite aplicação o método de Tomasi e Kanade duma forma global. Aplicando este método, obteve-se a matriz de forma e de movimento do *cubo mágico* filmado, a partir da qual foi estimado o respectivo modelo tridimensional do cubo, cuja projecção bidimensional é visível na figura 9.2.(b).

A figura 9.2 mostra os modelos tridimensionais obtidos: (a) apenas com o método de Tomasi e Kanade, a partir de 28 pontos notórios seguidos ao longo de 18 imagens; (b) com os algoritmos que estimam as trajectórias completas dos pontos, seguido do método de factorização de Tomasi e Kanade.



(a) método de T & K;(b) novos algoritmos + método de T & K (pontos escuros = conhecidos)

Figura 9.2: Modelos tridimensionais do *cubo mágico*.

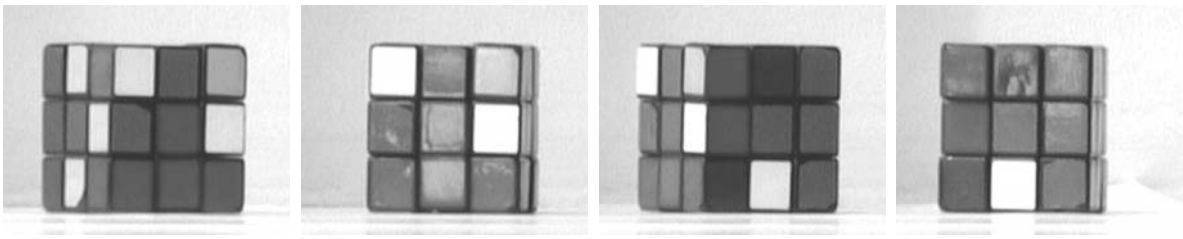
Na figura 9.2, os modelos tridimensionais do *cubo mágico* estão projectadas bidimensionalmente num ângulo de que este não tinha sido filmado, uma vez que este só rodava na horizontal. Este facto faz com que a sua face superior nunca tenha sido filmada e, portanto, não esteja presente nos modelos 3D do cubo. A comparação entre a qualidade dos modelos tridimensionais das figuras 9.2.(a) e 9.2.(b) ilustra bem as vantagens da utilização do novo método aqui apresentado, que permite usar toda a informação disponível duma forma global e óptima na recuperação do modelo tridimensional do cubo, resultando numa maior imunidade ao ruído e numa maior simplicidade na recuperação da forma total. Para recuperar a forma total do *cubo mágico* usando apenas o modelo de Tomasi e Kanade, seria necessário estimar pequenos modelos tridimensionais (como os da figura 9.2.(a)) a partir de pequenas porções das sequências de video em que não existe oclusão. Estes pequenos modelos teriam de ser fundidos posteriormente, numa operação de complexidade muito superior à dos novos algoritmos aqui apresentados, em especial para formas mais complexas.

Além de realidade virtual, a construção de modelos tridimensionais tem aplicação em diversas áreas da industria, tal como a observação/registo de espaços, design, etc.

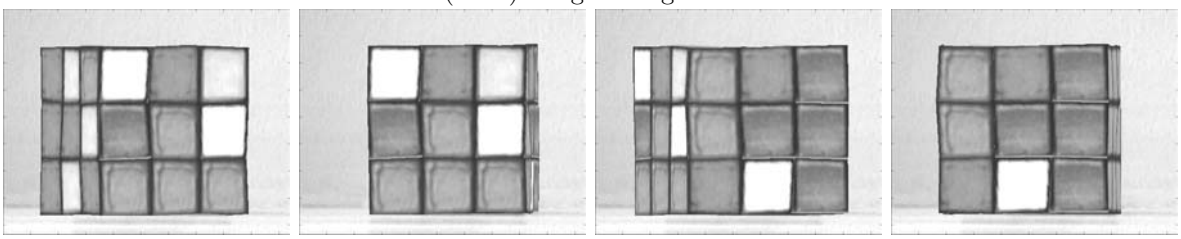
9.2 Compressão de video

Outra aplicação do novo método aqui apresentado, que permite transformar o problema SFM com informação desconhecida no problema SFM, é a compressão de imagem. Neste caso, em vez de se transmitir uma sequência de video completa, imagem após imagem, apenas é transmitido o modelo tridimensional dos objectos e o movimento da câmara, o que é equivalente. Este processo pode resultar numa grande poupança de largura de banda, pois, uma vez transmitido o modelo 3D dos objectos na sequência de video, é apenas necessário enviar o movimento de translação e rotação da câmara, o que pode ser feito com poucos *bytes* por imagem.

Com esta metodologia, a sequência de video do *cubo mágico*, que contém 2161 imagens, foi comprimida obtendo-se uma taxa de compressão de aproximadamente 10^3, relativamente às imagens originais em formato JPEG. A figura 9.3 mostra imagens originais desta sequência (em cima) e as respectivas imagens comprimidas (em baixo).



(cima) imagens originais



(baixo) imagens comprimidas

Figura 9.3: Compressão da sequência de video com o *cubo mágico*.

Os exemplos de imagens resultantes do processo de SFM com informação desconhecida duma sequência de video real com um *cubo mágico* da figura 9.3 mostram ligeiras distorções face às originais, tanto na tonalidade como na forma. As distorções na tonalidade devem-se ao efeito de sombras, porque, nesta sequência de video, em vez de ter sido a câmara de filmar a rodar em torno do *cubo mágico*, foi este que rodou enquanto a câmara se manteve estática (o que é formalmente equivalente), o que fez com que as faces mudem ligeiramente de tonalidade ao longo da sequência de video - efeito que não é contabilizado na construção do modelo tridimensional. As distorções devem-se, naturalmente, a erros no seguimento dos pontos, agravados pelo facto de não ter sido filmada a face superior, que serviria como elemento de ligação entre as faces, reduzindo a distorção na forma do cubo.

Capítulo 10

CONCLUSÃO

10.1 Sumário

Nesta tese propõem-se um conjunto de novos algoritmos que, juntamente com o conhecido método de factorização de Tomasi e Kanade, permitem estimar modelos tridimensionais densos de estruturas rígidas, a partir de grandes sequências de video com oclusão e inclusão.

As correspondências estimadas pelo novo estimador de correspondências permitiram determinar correctamente a estrutura tridimensional de várias cenas, somente a partir de duas imagens. A utilização das correspondências entre os vários pares de imagens duma sequência de video mais longa reduz ainda mais o erro de estimação.

Propôs-se também um novo algoritmo de fluxo óptico baseado em correlação que permite estimar a homografia entre duas ou mais imagens (uma estratégia habitual que facilita o passo de correspondência de SFM), com precisão *sub-pixels* e uma estratégia multi-resolução de grande eficiência. Mostrou-se que a estratégia de funcionamento deste algoritmo apresenta importantes vantagens face às estratégias alternativas baseadas em pontos notórios ou fluxo óptico, com uma estratégia especial que permite estimar mosaicos de grande qualidade em sequências de video longas. Este método permitiu estimar mosaicos de grande resolução a partir de sequências de video reais com 3 a 26 imagens.

Descreve-se e estuda-se o método de factorização de Tomasi e Kanade, reconhecido pela comunidade de visão por computador como um dos métodos fundamentais para a resolução do problema *structure from motion* [23]. Os testes efectuados a este método demonstram a sua eficácia na recuperação da estrutura 3D dos objectos filmados e o movimento da câmara, mesmo quando os pontos notórios são seguidos com bastante ruído. Foi também mostrado que a imunidade ao ruído é tanto maior quanto mais imagens tiver a sequência de vídeo, o que advém da estratégia de minimização de mínimos quadrados e das fortes restrições impostas à matriz de movimento, que ajudam a eliminar o ruído. Mostra-se também que, na presença de oclusão e inclusão, a matriz de observação tem entradas desconhecidas, o que impossibilita a sua aplicação em casos práticos em que existe muita oclusão,

por não ser possível aplicar a SVD.

Nesta tese são também apresentados novos métodos que permitem estimar informação desconhecida em matrizes singulares, o que permite transformar o problema SFM com informação desconhecida no problema SFM, com importantes vantagens face aos métodos existentes quanto à generalidade, fiabilidade, custo computacional, imunidade ao ruído e simplicidade. Estes novos algoritmos resolvem o problema da oclusão e inclusão, permitindo a utilização sem restrições do método de factorização de Tomasi e Kanade, tendo sido testados em sequência de video artificiais e reais. Os testes efectuados em sequências de video artificiais com o método completo (estimativa inicial + algoritmos iterativos) mostram erros médios por ponto da ordem de 10^{-11}, para ruído com desvio padrão $10^{2.5}$ vezes maior do que o dos dados, com probabilidade de convergência de 100%, para qualquer característica e qualquer percentagem de informação desconhecida que satisfaça condições muito gerais. Além da grande eficácia do algoritmo que calcula a estimativa inicial, os testes efectuados mostram também o bom funcionamento dos algoritmos iterativos, em que as condições de convergência são bem conhecidas. Do ponto de vista do custo computacional, foi visto que a estimativa inicial é calculada com um número reduzido de cálculos da SVD (dependendo da disposição dos dados conhecidos na matriz de observação) e os algoritmos iterativos têm aproximadamente o mesmo número de operações em vírgula flutuante do que a SVD (algoritmo EM) ou menor (algoritmo RC), por iteração. Qualquer um destes algoritmos apresenta claras vantagens face às soluções alternativas existentes.

Finalmente, foram apresentadas duas aplicações para o problema SFM com informação desconhecida, que são resolvidos com grande eficácia pelos novos métodos aqui introduzidos, demonstrando a sua importância em aplicações de grande relevância para a industria. Foi mostrada a construção de modelos 3D para utilização em aplicações de realidade virtual e observação/registo de espaços, e foi apresentado um poderoso compressor de video, que apresentou uma taxa de compressão de aproximadamente 10^3.

Apesar dos algoritmos propostos nesta tese serem descritos no contexto de SFM com informação desconhecida, podem ser aplicados em várias áreas da engenharia. De facto, a estimação de homografia pode ser usada em qualquer aplicação em que seja necessário efectuar o alinhamento entre duas ou mais imagens. Os novos algoritmos que estimam informação desconhecida em matrizes singulares funcionam com qualquer característica, encontrando aplicação em áreas tais como o reconhecimento de objectos [4, 40], aplicações em fotometria [33, 27, 5] e o alinhamento de imagem [45, 46], etc.

10.2 Direcções futuras

Nesta secção são apresentadas algumas ideias para estudos futuros acerca de problemas que os algoritmos propostos nesta tese deixam em aberto.

No capítulo 6 desta tese foram propostos algoritmos iterativos que, segundo mostram as experiências realizadas, quando adequadamente inicializados, convergem sempre. O objectivo de um estudo futuro poderá ser a demonstração da convergência destes algoritmos. Assim, poder-se-ia determinar as condições que os dados (no caso de SFM, o movimento da câmara, a forma 3D e o ruído) e a inicialização terão de verificar para que os algoritmos iterativos convirjam para a matriz completa de característica definida que melhor aproxima a matriz de observação incompleta, nas posições conhecidas.

Outra possível direcção de trabalho é a comparação dos algoritmos iterativos EM e RC com algoritmos iterativos desenvolvidos usando ferramentas de Geometria Diferencial, usando directa e explicitamente todas as restrições impostas pelo problema para encontrar o algoritmo óptimo do ponto de vista da convergência.

Nesta tese, entre outros assuntos, tratou-se o problema de estimar a matriz que melhor aproxima a matriz de observação com dados desconhecidos, com uma dada característica, como uma solução natural para SFM com informação desconhecida. Contudo, em SFM com oclusão e inclusão, é frequente que uma região esteja visível, depois deixe de o estar (por desaparecer do campo de visão ou ficar tapada por outras partes do objecto), para ficar visível novamente. Nesta tese, essa região é tratada como uma nova região, não sendo associada à que foi vista anteriormente. Um outro tópico de trabalho futuro é a identificação de regiões comuns. Este problema poderia ser resolvido identificando, de forma automática, duas colunas da matriz de observação como pertencendo à mesma coluna da matriz completa que a aproxima. Esta identificação permitiria constringir mais o problema de estimação e, consequentemente, obter melhores estimativas de forma e movimento 3D.

Bibliografia

[1] Aguiar, P.M.Q., Moura, J.M.F.: Factorization as a rank 1 problem. In: IEEE Computer Vision and Pattern Recognition, Fort Collins CO, USA (1999)

[2] P. M. Q. Aguiar and J. M. F. Moura, "Three-dimensional modeling from two-dimensional video," *IEEE Transactions on Image Processing*, vol. 10, no. 10, 2001.

[3] Serge Ayer, *Sequential and competitive methods for estimation of multiple motions*, Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 1995.

[4] Basri, R., Ullman, S.: The alignment of objects with smooth surfaces. Computer Graphics, Vision, and Image Processing: Image Understanding **57** (1988)

[5] Belhumeur, P., Kriegman, D.: What is the set of images of an object under all possible lighting conditions ? In: IEEE Computer Vision and Pattern Recognition. (1996)

[6] Cipolla, R., Okamoto, Y., Kuno, Y.: Robust Structure form Motion Using Motion Parallax. In: IEEE International Conference on Computer Vision. (1993)

[7] Branco, C., Costeira, J.P.: A 3D Mosaic System Using the Factorization Method. In: IEEE International Symposium on Industrial Electronics, Pretoria, South Africa (1998)

[8] Costeira, J.P., Kanade, T.: A Multi-Body Factorization Method for Motion Analysis. In: IEEE International Conference on Computer Vision, MIT, Cambrige, Massachussets, USA (1995)

[9] Forsyth, D.: Shape from texture without boundaries. Proc. of the European Conference on Computer Vision, Copenhaga, 2002.

[10] Geiger,D., Ladendorf,B., Yuille,A.L., Occlusions and Binocular Stereo. In: IEEE European Conference on Computer Vision (1995).

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 1996.

[12] Guerreiro, R.F.C., Aguiar, P.M.Q.: 3D structure from video streams with partially overlapping images. In: IEEE International Conference on Computer Vision, New York, USA (2002)

[13] Guerreiro, R.F.C., Aguiar, P.M.Q.: Factorization with missing data for 3D structure recovery. In: IEEE Workshop on Multimedia and Signal Processing, St. Thomas, USA (2002)

[14] Guerreiro, R.F.C., Aguiar, P.M.Q.: Estimamation of rank deficient matrices from partial observations: two-step iterative algorithms. In: IEEE International Workshop on Energy Minimization Methods on Computer Vision and Pattern Recognition, Lisbon, Portugal (2003)

[15] Horn, B., Brooks, M.J.: Shape from shading. MIT Press, Cambridge University, USA, 1989.

[16] Irani, M., Anandan, P.: All about direct methods. In: IEEE International Conference on Computer Vision, Kerkyra, Corfu, Greece (1999)

[17] Irani, M., Anandan, P.: Factorization with uncertainty. In: Proc. of European Conference on Computer Vision, Dublin, Ireland (2000)

[18] Jacobs, D.: Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In: IEEE Computer Vision and Pattern Recognition, St Barbara, USA (1997)

[19] Kang,S., Szeliski,R., Chai,J., Handling Occlusions in Dense Multi-view Stereo. In: IEEE Computer Vision and Pattern Recognition (2001).

[20] Lee, M., Philippos, M., Inference of Segmented Overlapping Surfaces from Binocular Stereo. In: IEEE Transactions on Pattern Analysis and Machine Inteligence (2002).

[21] McLachlan, G., Krishnan, T.: The EM Algorithm and Extensions. John Wiley & Sons, New York (1997)

[22] Mann, S., Picard, R.: Video Orbits of the Projective Group: a Simple Approach to Featureless Estimation of Parameters. IEEE Transactions on Image Processing, 1997.

[23] Maruyama, M. and Kurumi, S., "Bidirectional optimization for reconstructing 3D shape from an image sequence with missing data," in *IEEE International Conference on Computer Vision*, Kobe, Japan, 1999.

[24] Moon, T. K., Stirling, W. C. *Mathematical Methods and Algoritms for Signal Processing*, Prentice Hall, 1999.

[25] Morris, D., Kanade, T.: A unified factorization algorithm for points, line segments and planes with uncertainty models. In: IEEE International Conference on Computer Vision. (1998)

[26] Morita, T., Kanade, T.: A sequential factorization method for recovering shape and motion from image streams. IEEE Transactions on Pattern Analysis and Machine Inteligence **19** (1997)

[27] Moses, Y.: Face recognition: generalization to novel images. PhD thesis, Weizmann Institute of Science (1993)

[28] Okutomi, M., Kanade, T.: A Locally Adaptive Window for Signal Matching. In: IEEE International Conference on Computer Vision, Osaka, Japan (1990).

[29] C. J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol. 19, no. 3, 1997.

[30] Quan, L., Kanade, T.: A factorization method for affine structure from line correspondences. In: IEEE Computer Vision and Pattern Recognition, San Francisco CA, USA (1996)

[31] Shapiro, L.: Affine Analysis of Image Sequences. Cambridge University Press, Cambridge, UK (1995)

[32] Shapiro, L.G., Stockman, G. C.: Computer Vision. Prentice Hall, New Jersey (2001).

[33] Shashua, A.: Geometry and photometry in 3D visual recognition. MIT TR 1401, Massachussets Institute of Technology, MA, USA (1992)

[34] Shum, H., Ikeuchi, K., Reddy, R.: Principal component analysis with missing data and its applications to polyhedral object modeling. IEEE Transactions on Pattern Analysis and Machine Inteligence **17** (1995)

[35] Sturn, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: Procedings of European Conference on Computer Vision, Cambridge, UK (1996)

[36] Sturn,P., Triggs,B.: "A Factorization Based Algorithm for Multi-Image Projective Structure and Motion", IEEE Transactions on Pattern Analysis and Machine Inteligence, Vol. 19, No. 8, Aug. 1997.

[37] Tomasi, C., Kanade,T.: "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, 1992.

[38] Torr, P., Zisserman,A.: "Feature Based Methods for Structure and Motion Estimation." In: IEEE International Conference on Computer Vision, Kerkyra, Corfu, Greece (1999)

[39] Ullman, S., "The interpretation of Visual Motion", MIT Press, Cambridge University, USA, 1979.

[40] Ullman, S., Basri, R.: Recognition by linear combinations of models. IEEE Transactions on Pattern Analysis and Machine Inteligence **13** (1991)

[41] Wiberg, T.: Computation of principal components when data are missing. Procedings of the Second Symposium Computational Statistics, Berlin, Germany (1976)

[42] Watt, A.: 3D Computer Graphics. Addison-Wesley, 3rd Edition.

[43] Xiong, Y.,Shafer, S. A.: Depth from Focusing and Defocusing. IEEE International Conference on Computer Vision, New York, USA, 1993.

[44] Xu, G., Tsuji, S.: Recovering surface shape from boundary. International Joint Conference on Artificial Inteligence, Milão, Itália, 1987.

[45] Zelnik-Manor, L., Irani, M.: Multi-frame alignement of planes. In: IEEE Computer Vision and Pattern Recognition, Fort Collins CO, USA (1999)

[46] Zelnik-Manor, L., Irani, M.: Multi-view subspace constraints on homographies. In: IEEE International Conference on Computer Vision, Kerkyra, Greece (1999)

[47] "http://www-2.cs.cmu.edu/∼cil/vision.html"

[48] "http://www.photo3D.net"

[49] "http://www.isr.ist.utl.pt/˜aguiar/code.html"