**Reconstrução de Faces a Partir
de Múltiplas Imagens Utilizando
Um Modelo de Referência e um Conjunto
de Pontos de Controle.**

Bruno Pera

**DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO DE MESTRE
EM
CIÊNCIAS**

**Área de Concentração: Ciência da Computação
Orientador: Prof. Dr. Carlos Hitoshi Morimoto**

Durante a elaboração deste trabalho o autor recebeu auxílio financeiro da FAPESP

**SÃO PAULO
2006**

Reconstrução de Faces a Partir de Múltiplas Imagens Utilizando Um Modelo de Referência e um Conjunto de Pontos de Controle.

Este exemplar corresponde à redação final da dissertação devidamente corrigida e apresentada por Bruno Pera e aprovada pela Comissão Julgadora.

São Paulo, 18 de Setembro de 2006

Banca Examinadora :

- Prof. Dr. Carlos Hitoshi Morimoto (Orientador) - IME/USP
- Prof. Dr. Marcel Parolin Jackowski - IME/USP
- Prof. Dr. Manuel Menezes de Oliveira Neto - UFRGS

à minha mãe Cristina, ao meu pai José Antônio e aos meus irmãos Daniel e Alex.

Agradecimentos ¹

Ao meu orientador Prof. Carlos Hitoshi Morimoto.

¹Este projeto foi financiado pela FAPESP

Resumo

Modelos realistas da face humana podem ser utilizados em diversas áreas como animação para filmes e comerciais, jogos eletrônicos, realidade virtual, tele-conferência, agentes para interface com o usuário, planejamento de cirurgias faciais, etc. O processo de criação de tais modelos ainda é extremamente trabalhoso e deve ser realizado por técnicos especializados.

Este trabalho apresenta um resumo sobre as diversas técnicas existentes atualmente para a criação automática de modelos de faces, com destaque para os métodos baseados em múltiplas imagens. O texto apresenta também uma revisão dos conceitos de geometria projetiva que podem ser utilizados para lidar com o problema de reconstrução de objetos rígidos.

Implementamos um sistema para reconstrução de faces que ajusta um modelo genérico de faces a múltiplas imagens de uma face real através da correspondência entre pontos marcados nas imagens de entrada. A única tarefa do usuário é localização de um pequeno conjunto de pontos de controle sobre cada uma das imagens.

Combinamos as imagens de entrada em um mapa de textura em coordenadas esféricas que aplicado ao modelo ajustado aumenta consideravelmente o realismo do resultado. Problemas encontrados no mapa de texturas devido a falhas no ajuste do modelo são corrigidos com um processamento estatístico baseado em cor de pele.

Os resultados obtidos com nosso protótipo demonstram a simplicidade da interação e a boa qualidade do sistema, que ajusta bem o modelo às características internas da face como olhos, boca e nariz, porém em alguns casos distorce levemente o contorno do rosto. Implementamos ainda uma interface simples que permite a criação de animações com diferentes expressões faciais a partir dos modelos reconstruídos.

Abstract

Realistic human face models can be used in different applications such as animation for films and advertisements, video games, virtual reality, teleconference, user interface agents, etc. Currently, the creation process of such models are mostly manual and must be done by people with special training.

This work presents a survey of the current techniques available for automatic reconstruction of face models, with emphasis on the reconstruction from multiple images technique. The text also presents a revision of the main projective geometry concepts that we have used to deal with the problem of reconstruction of three dimensional rigid objects.

We have implemented a face reconstruction system prototype that adjusts a generic model to several images of a real face using the feature correspondences previously marked on the input images. The user is only required to manually locate a small set of control points on each image.

The input images are combined in a spherical coordinate texture map that improves the realism of the result when applied onto the adjusted model. Problems in the generation of texture maps due to innaccurate model adjustments are treated with a statistic skin color filter.

The results show that the system prototype can accurately adjust the model to the inner face features such as the mouth, the eyes and the nose, but in some cases the model presents small innaccuracies to the face outline. We have also implemented a simple interface that allows the creation of animations with different facial expressions from the reconstructed face models.

Sumário

1	Introdução	1
1.1	Motivação	3
1.2	Objetivos	4
1.3	Organização da Dissertação	5
2	Reconstrução de Faces	7
2.1	O Problema Clássico de Reconstrução	7
2.2	O Problema de Reconstrução de Faces	9
2.3	Modelamento de Faces Utilizando <i>Morphing</i> 3D	12
2.3.1	O Banco de Faces	12
2.3.2	O Modelo	13
2.3.3	Características Faciais	13
2.3.4	Ajuste do Modelo a Imagens	14
2.4	Modelos de Faces a Partir de Múltiplas Imagens	15
2.4.1	Visão Geral	15
2.4.2	Ajuste do Modelo	16
2.4.3	Extração dos Mapas de Texturas	19
2.4.4	Transições Entre Expressões Faciais	22

2.5	Comentários	22
3	Geometria Projetiva	25
3.1	Geometria Projetiva 2D	25
3.1.1	O Plano Projetivo	26
3.1.2	Transformações Projetivas do Plano	27
3.2	Geometria projetiva 3D	30
3.2.1	Pontos, Planos e o Espaço Projetivo	30
3.2.2	Transformações Projetivas Do Espaço	30
3.3	Câmeras	31
3.3.1	Câmeras Finitas	31
3.3.2	Câmeras no Infinito	36
3.4	Geometria Epipolar	36
3.4.1	A Matriz Fundamental	39
3.4.2	Obtenção das Câmeras a Partir das Matriz Fundamental	39
3.5	Reconstrução	40
3.5.1	A matriz Fundamental a Partir das Correspondências	41
3.5.2	Reconstrução das Câmeras e da Cena	41
3.5.3	Reconstrução Afim	42
4	O Sistema de Reconstrução de Faces	45
4.1	O Modelo Genérico de Face	45
4.2	Pontos de Controle	47
4.3	Posição e Orientação das Câmeras	50
4.3.1	Câmeras Projetivas	50
4.3.2	Câmeras Calibradas	52

4.4	Ajuste do Modelo	52
4.5	Mapeamento de Textura	54
4.6	Processamento Baseado em Cor de Pele	57
4.6.1	Fase de Treinamento	58
4.6.2	Fase de Classificação	59
4.7	Calibragem da Câmera	62
4.7.1	Homografia Entre o Plano do Modelo e Sua Imagem	62
4.7.2	Estimando a Homografia H	63
4.7.3	Restrições Sobre os Parâmetros Internos	64
4.8	Implementação	66
4.9	Métodos Numéricos	69
4.9.1	Decomposição de Valores Singulares (SVD)	69
4.9.2	Método de Powell Para Minimização Não Linear	70
5	Resultados	71
5.1	Um Exemplo Detalhado	72
5.2	Outros Exemplos	75
6	Conclusão	83

Lista de Figuras

2.1	Diagrama do sistema.	17
3.1	Câmera <i>pinhole</i> : Centro de projeção O_c , plano principal X_cY_c , plano da imagem x_cy_c , ponto principal p , ponto no espaço X e sua projeção x	32
3.2	Geometria epipolar: Centro das câmeras C_1 e C_2 , os epípolos c_1 e c_2 , as retas epipolares l_1 e l_2 , o ponto no espaço X e suas projeções q_1 e q_2	38
4.1	O modelo genérico de faces: (a) <i>wire frame</i> : vértices e músculos, (b) <i>flat</i> : polígonos e efeitos de iluminação, (c) <i>smooth</i> : polígonos suavizados, efeito mais realista.	46
4.2	Algumas expressões faciais pré-definidas do modelo: (a) felicidade, (b) raiva, (c) surpresa, (d) medo, (e) repulsa, (f) tristeza.	48
4.3	Pontos de controle mostrados sobre o modelo de face.	49
4.4	Conjunto estendido de pontos de controle mostrados sobre o modelo de face.	54
4.5	A textura gerada usando projeção cilíndrica não cobre a parte inferior do queixo.	55
4.6	Quando o modelo não se ajusta perfeitamente à face da imagem de entrada (a) a textura apresenta pixels que não correspondem à pixels da face (b). A textura é corrigida usando um processamento baseado em cor de pele (c).	56
4.7	Extração de regiões de cor de pele para conjunto de treinamento.	59

4.8	Resultado do cálculo da probabilidade de cor de pele para cada pixel.	60
4.9	Exemplo de histograma particionado em intervalos de 10/100.	61
4.10	Resultado da limiarização aplicada ao exemplo dado na figura 4.8.	61
4.11	Padrão utilizado para calibrar a câmera.	62
4.12	Interface para localização de pontos de controle. Na parte esquerda o usuário carrega a coleção de imagens de entrada. O modelo genérico é apresentado do lado direito, a correspondência de pontos é feita clicando-se primeiramente no vértice do modelo e em seguida na posição correspondente na imagem de entrada.	67
4.13	Diagrama do funcionamento do sistema.	68
5.1	Exemplo de imagens usadas para calibrar a câmera.	72
5.2	Exemplo 1	73
5.3	Exemplo 2	76
5.4	Exemplo 3	77
5.5	Exemplo 4	78
5.6	Exemplo 5	79
5.7	Exemplo 6	80
5.8	Exemplo 7	81

Lista de Tabelas

4.1	Lista de pontos de controle.	49
-----	--------------------------------------	----

Capítulo 1

Introdução

A modelagem e animação de faces humanas é um dos problemas clássicos de computação gráfica. Esse assunto já vem recebendo atenção da comunidade científica pelo menos desde o início da década de 70, sendo que um dos trabalhos pioneiros foi apresentado no SIGGRAPH em 1972 por Parke [Par72]. Nesse trabalho ele utilizou uma malha de polígonos para representar a geometria da face e interpolação não linear dos vértices do modelo para a animação baseada em quadros-chave, simplificando o processo tradicional de animação e produzindo resultados mais realistas.

Modelos realistas de face podem ser aplicados em diversas áreas como animação para filmes e comerciais, jogos eletrônicos, realidade virtual, tele-conferência, agentes para interface com o usuário, planejamento de cirurgias faciais, etc.

Os primeiros trabalhos, como o de Parke, tratavam simplesmente de malhas de polígonos e a animação era feita através da interpolação geométrica entre os vértices do modelo ao longo dos quadros. Essa estratégia simplista traz consigo grandes dificuldades. Como a estrutura geométrica de uma face é bastante complicada, alterar manualmente cada um dos vértices para se obter uma determinada expressão facial, por exemplo, é uma tarefa extremamente

trabalhosa. Além disso, tratar apenas da geometria do modelo, não levando em consideração outras características como a textura, não é suficiente para gerar modelos faciais realistas.

Para amenizar as dificuldades relacionadas com a animação de modelos geométricos baseado puramente em vértices e polígonos, foram introduzidos métodos que simulam a atuação dos músculos faciais para alterar a geometria do modelo e assim gerar as expressões faciais [Wat87]. Ainda assim, não é trivial determinar qual a configuração dos músculos faciais que geram uma dada expressão, mas já torna significativamente mais fácil a tarefa de animar os modelos faciais.

Essas técnicas podem ser utilizadas para animar um modelo de face específico, mas com a grande demanda por modelos de faces para animação, jogos, filmes e anúncios torna-se interessante a elaboração de métodos capazes de gerar um modelo 3D de um indivíduo em particular. Nesse sentido existem dois grandes grupos de técnicas: as que são baseadas em dados provenientes de scanners 3D [BV99] ou outros equipamentos que disponibilize algum tipo de informação de profundidade e as que são baseadas apenas em uma ou mais imagens [PHL⁺98, FM99, D'A02], incluindo seqüências de vídeo. *Alto é mais caro!*

As técnicas baseadas em scanners podem gerar resultados muito bons mas têm como ponto fraco o alto custo dos equipamentos envolvidos. As técnicas baseadas em conjuntos de imagens ou seqüências de vídeo não requerem equipamentos tão caros mas apresentam uma precisão menor e, em geral, requerem o auxílio do usuário para detectar informações de interesse nas imagens.

Podemos encontrar na literatura diferentes técnicas para reconstrução de faces a partir de múltiplas imagens que utilizam diversos tipos de informação. Algumas utilizam a correspondência de pontos de controle nas várias imagens, outras o contorno do rosto ou mesmo o sombreamento gerado pela iluminação e outras que usam a localização de características como olhos, boca, nariz, etc.

Outro método bastante utilizado atualmente para a animação facial é baseado em *motion capture* no qual os movimentos de uma face real são capturados e podem ser reproduzidos por um modelo de face digital. Para tanto, coloca-se no rosto da pessoa em questão uma coleção de sensores que são utilizados para medir as deformações que ocorrem na face por conta das expressões faciais.

Mesmo com todo o esforço já realizado nessas áreas, criar um modelo de face realista para animação ainda é um processo altamente trabalhoso que, em geral, é feito por pessoas muito bem treinadas. Sendo assim, um método automático e de baixo custo para a criação de modelos de face para animação facilitaria muito o processo de modelagem poupando uma enorme quantidade de tempo e trabalho.

1.1 Motivação

Dentre os principais problemas para gerar animações de faces podemos citar a dificuldade de criar modelos realistas pois a face possui uma geometria extremamente complexa com um grande número de rugas e vincos, além disso, apresenta variações de cor e textura ao longo de toda sua extensão.

Outro fator agravante é que os movimentos não rígidos da face (as expressões faciais) são resultado de uma complexa interação entre os músculos, a pele e o esqueleto, sendo que a forma e as propriedades mecânicas dessas estruturas são diferentes para cada indivíduo e não são constantes ao longo de uma determinada face.

Vale notar também que a enorme capacidade que todos nós temos de reconhecer faces e expressões faciais e detectar facilmente qualquer desvio, por menor que seja, de uma face sintética com relação à realidade contribui significativamente para tornar o trabalho ainda mais complicado.

Uma consequência das dificuldades envolvidas na criação de modelos de face é que o processo torna-se bastante trabalhoso. Como exemplo podemos citar o caso do longa metragem *Shrek* produzido pela *Dream Works SDK*. O trabalho todo contou com a participação de mais de 275 pessoas [dWc], sendo a maioria composta por desenhistas, modeladores e animadores que trabalharam por mais 4 anos. A quantidade de informação utilizada durante o processo de renderização também é assustadora, cerca de 2,5GiB por dia em mais de 200 milhões de polígonos[dWa].

Este exemplo demonstra que mesmo atualmente, apesar da existência de uma vasta coleção de ferramentas para auxiliar o processo de criação de modelos tridimensionais, o processo ainda é bastante trabalhoso, exigindo a participação intensa de vários artistas especializados.

1.2 Objetivos

O objetivo do trabalho é estudar alguns métodos de reconstrução de faces a partir de múltiplas imagens e a implementação de um sistema de baixo custo para reconstrução que seja o mais automático possível requerendo o mínimo de interação com o usuário e que gere modelos faciais realistas.

O trabalho foi desenvolvido dentro do Laboratório de Tecnologias de Interação do Departamento de Ciência da Computação do Instituto de Matemática e Estatística da Universidade de São Paulo (LaTIn), que tem como objetivo o desenvolvimento de formas alternativas de interação com o computador.

Modelos de faces têm um grande potencial para serem utilizadas diretamente na interação com computadores em ambientes virtuais ou de realidade aumentada, por exemplo na forma de agentes para interface com o usuário. Além disso, ter um modelo 3D de face de uma dada pessoa poderia facilitar a implementação de tarefas como a identificação do usuário e

reconhecimento de expressões faciais. Essas informações também poderiam ser usadas para melhorar o processo de interação homem-computador.

O sistema implementado é baseado nos trabalhos de Pighin *et. al.* [PHL⁺98] e em princípios da geometria projetiva. Para estimar a geometria da face o sistema usa como principal informação um pequeno conjunto de 12 pontos de controle que devem ser localizados sobre cada uma das imagens de entrada. Por enquanto a localização dos pontos ainda é feita de forma manual, mas em breve o sistema poderá ser integrado com o trabalho do aluno Régis de Abreu Barbosa que estuda técnicas para detecção e rastreamento de características faciais. Uma vez feita a integração teremos um sistema completamente automático para recuperação de modelos 3D de faces.

As imagens de entrada são combinadas para formar um mapa de textura que aumenta consideravelmente o realismo do modelo obtido. Para eliminar falhas na geração da textura devido à problemas de ajuste do modelo às imagens de entrada, usamos um processamento baseado em cor de pele [PdABM04] que foi resultado das etapas iniciais do nosso trabalho.

Diferente de Pighin, que realiza a reconstrução do modelo de face para diferentes expressões faciais e faz a animação do modelo através de *morphing*, nós realizamos a reconstrução apenas para a expressão neutra e fazemos a animação através de um sistema de simulação de músculos. Para isso, implementamos uma interface para combinar diferentes expressões faciais pré-definidas não sendo necessário definir o estado de cada músculo para obter uma dada expressão.

1.3 Organização da Dissertação

No capítulo 2 apresentamos uma definição para o problema de reconstrução 3D de faces e apresentamos algumas técnicas utilizadas para tanto encontradas na literatura.

No capítulo 3 apresentamos uma introdução à geometria projetiva, dando ênfase aos tópicos que se relacionam diretamente com o problema de reconstrução, como transformações projetivas, projeções (câmeras), geometria epipolar, etc.

No capítulo 4 apresentamos em detalhes o nosso sistema, desenvolvido com base nos trabalhos encontrados na literatura e discutidos no capítulo 2 e os tópicos de geometria projetiva discutidos no capítulo 3.

No capítulo 5 apresentamos e discutimos alguns resultados obtidos com a utilização do sistema implementado.

No capítulo 6 encerramos esse trabalho apresentando as conclusões e alternativas de trabalhos futuros que poderiam melhorar os resultados obtidos.

Capítulo 2

Reconstrução de Faces

2.1 O Problema Clássico de Reconstrução

O problema clássico de reconstrução de objetos rígidos em visão computacional consiste em determinar a estrutura tridimensional de um objeto a partir de uma ou mais imagens.

Uma técnica bastante utilizada para reconstrução é o método de fatoração proposto por Tomasi e Kanade [TK92].

A entrada para esse método é uma coleção de correspondências de pontos 2D entre um conjunto de imagens de uma dada cena. É possível representar essas informações por uma matriz W de dimensão $2F \times P$ que é composta pelas coordenadas x e y de P pontos em F imagens. É possível provar que se a projeção é ortográfica então a matriz W possui posto 3.

Como consequência disto, pode-se mostrar que a matriz W pode ser fatorada na forma $W = RS$, onde R é uma matriz $2F \times 3$ que representa a rotação das câmeras e S é uma matriz $3 \times P$ que representa as coordenadas 3D dos pontos reconstruídos.

Esse método pode lidar com correspondências parciais, isto é, correspondências entre algumas imagens de entrada, mas não todas, que pode ocorrer devido à oclusão ou erro por parte

do algoritmo de rastreamento.

Infelizmente para utilização de tal método é necessário supor que a projeção oriunda da câmera é ortográfica, que em geral é uma hipótese bastante forte. Além disso, o método trata apenas de objetos rígidos não podendo ser aplicada diretamente na reconstrução de objetos deformáveis.

Outros trabalhos estendem o método original para lidar com mais de um objeto em movimento na cena, objetos deformáveis e a hipótese de projeção perspectiva fraca (*weak perspective*) [Had05].

O caso geral de reconstrução utilizando correspondência de pontos entre imagens pode ser tratado de forma bastante eficaz e elegante através da geometria projetiva, como veremos em detalhes no capítulo 3.

Outros métodos existentes diferem quanto ao tipo de dados utilizados, a precisão dos resultados gerados e a facilidade de uso. A seguir apresentamos uma lista resumida das técnicas mais comuns:

Triangulação Laser/Luz Estruturada um feixe laser desenha uma linha que se move pela superfície do objeto sendo modelado. Uma câmera, colocada em um ângulo específico em relação à origem do feixe laser, registra as variações na curvatura da linha gerada, estimando a topologia do objeto [BBEHC01].

Profundidade Laser um feixe laser é direcionado diretamente ao objeto, e através de variações de frequência e comprimento de onda, o tempo de retorno do feixe à câmera é calculado, e a distância do ponto em relação à câmera é encontrada [DC01].

Sonda Mecânica uma espécie de “caneta” presa a um braço mecânico com sensores em cada junta é movida pela superfície do objeto. Pelo movimento das juntas, é possível encontrar a posição tridimensional, em relação à base do braço, da ponta da “caneta”

em diversos momentos. Esse conjunto de pontos é utilizado para definir a estrutura do objeto. A operação de um equipamento deste porte exige grande tempo e paciência. Em contrapartida, os resultados são em geral extremamente precisos.

Reconstrução a partir de sombras um conjunto de técnicas que utilizam as propriedades de reflexão da superfície do objeto sendo analisado para estimar sua estrutura tridimensional [ZTCS99]. Essa família de técnicas é aplicada há muitos anos, com grande sucesso, para estimar a topologia de terrenos a partir de fotos de satélites.

Reconstrução Estéreo a partir de duas câmeras cuja posição relativa é conhecida, podemos utilizar um ramo da geometria conhecido como *geometria epipolar* para estruturar a relação entre as imagens capturadas pelas câmeras [Fau92]. Isto nos permite estimar a posição de um ponto no espaço que aparece em cada uma das imagens e, a partir dessa informação, podemos encontrar a distância deste ponto em relação às câmeras.

Reconstrução a partir de Movimento a estimação da estrutura a partir do movimento é equivalente ao caso de reconstrução estéreo, porém utilizando apenas uma câmera. O desafio agora é encontrar a movimentação tridimensional da câmera entre duas posições distintas.

2.2 O Problema de Reconstrução de Faces

O problema de reconstruir faces é mais complicado por lidar com um objeto deformável e que possui uma geometria extremamente complicada. Os métodos tradicionais de reconstrução devem ser adaptados para gerar modelos de face que possam ser animados facilmente. Em geral a solução encontrada na literatura é ajustar um modelo de face genérico à face que deve ser reconstruída.

Assim como no caso de objetos rígidos os métodos existentes utilizam diferentes fontes de dados. No nosso trabalho decidimos empregar as técnicas de reconstrução a partir várias imagens devido ao baixo custo dos equipamentos necessários (câmera), a facilidade de obtenção dos dados (imagens) e a grande flexibilidade de tais métodos. A seguir apresentamos um resumo de algumas técnicas:

Lee *et. al.* [LKMT97] apresenta um método semi-automático capaz de reconstruir modelos 3D de faces a partir de duas fotografias tiradas de direções ortogonais, uma frontal e outra de perfil. O método é baseado na extração dos contornos do cabelo e do rosto e na detecção características internas como olhos, boca, etc. A detecção dos contornos e das características de interesse são feitas utilizando-se uma técnica conhecida como *structured snakes* [KWT88].

Um processo de deformação é então utilizado para ajustar um modelo 3D de face genérico ou pré-definido à face das imagens. Finalmente um mapa de textura baseado em projeção cilíndrica gerado a partir da combinação das duas imagens de entrada é aplicado ao modelo para aumentar o realismo.

Fua e Miccio [FM99] mostram um método eficiente para ajustar modelos complexos a dados extraídos de imagens de faces. O método é baseado no ajuste de mínimos quadrados, usando pontos e contornos no ajuste. Os dados de entrada são marcados manualmente pelo usuário. O resultado é um modelo completo, incluindo orelhas e cabelo, mas o método não prevê uma forma simples para animar o modelo resultante.

A técnica de reconstrução proposta por D'Apuzzo [D'A02] consiste de 5 passo básicos:

- Aquisição de múltiplas imagens,
- calibragem do sistema,
- detecção de pontos correspondentes nas imagens,
- cálculo de suas coordenadas 3D,

- geração da superfície do modelo.

As imagens são capturadas por 5 câmeras dispostas em frente da face, o sistema é calibrado com o uso de um objeto de referência que pode ser medido automaticamente.

Para facilitar a detecção de pontos correspondentes nas imagens, texturas na forma de padrões aleatórios podem ser projetadas em duas direções diferentes sobre a face. O processo de detecção de correspondências baseado no algoritmo de mínimos quadrados com restrições geométricas produz um conjunto denso de correspondências. Filtros de vizinhança são aplicados para remover os erros.

Após a filtragem, as coordenadas tridimensionais dos pontos correspondentes são determinados, por triangulação, usando o resultado da calibragem. Por fim, uma superfície é gerada a partir da nuvem de pontos 3D e um mapa de textura pode ser projetado sobre o modelo para aumentar o realismo.

A técnica descrita por Ivanov *et. al.* [ILS⁺03] permite realizar a reconstrução a partir de duas imagens ortogonais (frontal e de perfil), de uma coleção de imagens de pontos de vista arbitrários ou de uma sequência de vídeo mostrando a face de um lado ao outro. O processo é implementado como um sequência de estágios que registram as imagens disponíveis num sistema de coordenadas associado à face, ajustam um modelo genérico de face às imagens de entrada e geram um mapa de textura consistente usando mapeamento inverso.

A qualidade visual do modelo obtido depende da resolução espacial do modelo genérico e da resolução das imagens iniciais. Para realizar o ajuste do modelo são utilizadas correspondências entre pontos, contornos de estruturas como olhos e boca e silhuetas que devem ser localizadas manualmente pelo usuário.

2.3 Modelamento de Faces Utilizando *Morphing* 3D

Um método que mostrou resultados muito interessantes, como apresentado por Blanz e Vetter [BV99], tem como base um banco de dados de modelos de face 3D, a partir do qual constrói-se um modelo de face mutável transformando a forma e a textura dos exemplos do banco de dados numa representação de espaço vetorial. Novas faces e expressões faciais podem ser geradas como combinação linear dos exemplos.

Restrições quanto à forma e à textura definidas a partir de estatísticas dos exemplos são usadas para guiar a modelagem manual ou automática. É possível ajustar o modelo mutável a uma ou mais imagens para realizar a reconstrução 3D de faces. Também é possível manipular o modelo com relação a parâmetros complexos como idade, gênero, etc. A maior dificuldade dessa técnica é a geração do banco de dados inicial cujos modelos são provenientes de scanners 3D e devem estar registrados entre si.

2.3.1 O Banco de Faces

Dado um conjunto de modelos faciais 3D de treinamento, a idéia é derivar um modelo mutável de face transformando a geometria e a textura dos exemplos numa representação de espaço vetorial.

Novas faces e expressões faciais podem ser modeladas como a combinação linear dos exemplos, além disso, estatísticas a respeito da geometria e da textura podem ser utilizadas para guiar a criação de novas faces, evitando assim a obtenção de faces muito distorcidas que não correspondam a faces reais.

Na maioria dos casos esse banco inicial de modelos é obtido a partir de *laser scanners*, como descrito em [BV99]. Neste caso o banco de dados é constituído pelo *laser scan* de 200 jovens adultos, sendo 100 homens e 100 mulheres.

Em geral, cada *laser scan* produz uma estrutura de dados em representação cilíndrica, com raios $r(h, \theta)$ dos pontos da superfície (face) amostrados em 512 ângulos equiespaçados θ e 512 alturas equiespaçadas h . Além disso, os valores RGB das cores em cada ponto $R(h, \theta)$, $G(h, \theta)$, $B(h, \theta)$ também são armazenados na mesma resolução espacial com 8 bits por canal.

2.3.2 O Modelo

Sejam $S = (X_1, Y_1, Z_1, X_2, Y_2, Z_2, \dots, X_n, Y_n, Z_n)$ a representação da geometria de uma face e $T = (R_1, G_1, B_1, R_2, G_2, B_2, \dots, R_n, G_n, B_n)$ a representação do mapa de textura que contém a cor RGB para cada um dos n vértices de S .

Representado os elementos do banco na forma descrita podemos obter novas formas S_{mod} e novas texturas T_{mod} como a combinação linear das m formas e texturas de exemplo:

$$S_{mod} = \sum_{i=1}^m a_i S_i, \quad T_{mod} = \sum_{i=1}^m b_i T_i, \quad \sum_{i=1}^m a_i = \sum_{i=1}^m b_i = 1 \quad (2.1)$$

O modelo de faces é definido como $(S_{mod}(a), T_{mod}(b))$, parametrizado pelos coeficientes $a = (a_1, a_2, \dots, a_m)$ e $b = (b_1, b_2, \dots, b_m)$. Novas faces arbitrárias podem ser obtidas variando-se os parâmetros a e b que controlam a forma e a textura respectivamente.

A expressividade do modelo pode ser aumentada dividindo-se as faces em sub regiões independentes que são tratadas separadamente, como a região dos olhos, da boca e do nariz, por exemplo.

2.3.3 Características Faciais

Os coeficientes a e b do modelo de faces não correspondem a nenhuma característica facial utilizada comumente para descrever faces, como por exemplo gênero, tamanho da boca, tamanho do nariz, cor da pele, cor dos olhos, etc.

Baseado num conjunto de faces (S_i, T_i) com rótulos μ_i definidos manualmente que quantificam uma dada característica, é possível calcular as somas ponderadas:

$$\Delta S = \sum_{i=1}^m \mu_i (S_i - E(S)), \quad \Delta T = \sum_{i=1}^m \mu_i (T_i - E(T)) \quad (2.2)$$

Múltiplos de $(\Delta S, \Delta T)$ podem ser adicionados ou subtraídos de uma dada face (S, T) , tendo o efeito de tornar mais ou menos pronunciada a característica facial em questão.

Como exemplo desta os autores definem $\mu_i = 1$ para toda face masculina e $\mu_i = 0$, para toda face feminina. Neste caso $(\Delta S, \Delta T)$ é um vetor que aponta na direção do desvio médio entre os vetores que representam faces masculinas. Adicionar um múltiplo positivo deste vetor a uma certa face teria o efeito de enfatizar as características faciais normalmente encontradas em faces masculinas.

2.3.4 Ajuste do Modelo a Imagens

Uma aplicação interessante do modelo descrito é a possibilidade de ajustar o modelo a uma dada imagem (fotografia) de uma face que não esteja representada no banco de treinamento.

Para tanto, os parâmetros do modelo 3D devem ser otimizados juntamente com um conjunto de parâmetros de renderização de modo que produzam uma imagem tão próxima quanto possível da imagem de entrada.

Os parâmetros do modelo são os coeficientes a e b já descritos, os parâmetros de renderização ρ incluem posição e orientação da câmera e os coeficientes de iluminação ambiente e direcional.

A partir dos parâmetros (a, b, ρ) imagens coloridas $I_{mod}(x, y)$ são renderizadas usando projeção perspectiva de modo a minimizar a distância euclidiana:

$$E_I = \sum_{x,y} ||I_{input}(x,y) - I_{model}(x,y)||^2 \quad (2.3)$$

Utilizando esse procedimento é possível obter um modelo 3D de face a partir de uma ou mais fotografias de um dado indivíduo.

2.4 Modelos de Faces a Partir de Múltiplas Imagens

Nesta seção vamos descrever de forma detalhada as técnicas de reconstrução de faces a partir de múltiplas imagens apresentadas por Pighin [PHL⁺98, PSS99] e que servem como base para o nosso trabalho.

O objetivo é estimar um modelo 3D de face a partir de uma coleção de imagens (fotografias ou seqüência de vídeo) de uma face real de diferentes pontos de vista. O modelo obtido deve ser o mais realista possível e capaz de apresentar diferentes expressões faciais.

2.4.1 Visão Geral

O processo pode ser resumido da seguinte maneira:

- A partir de uma coleção de imagens de uma face real, a posição e a orientação das câmeras correspondentes a cada uma das imagens e também as coordenadas 3D de um conjunto de pontos de controle escolhidos sobre a face são recuperados de forma manual.
- Técnicas de interpolação de dados esparsos [Nie93] são utilizadas para deformar um modelo de face genérico para que este se ajuste à geometria da face presente nas imagens de entrada.

- Uma vez obtidas as câmeras e a geometria da face, um ou mais mapas de textura para o modelo são extraídos das imagens.
- Todo o processo é repetido para diferentes expressões faciais de uma dada pessoa. Para gerar transições entre essas expressões é utilizado *morphing* 3D entre os modelos geométricos e os mapas de textura correspondentes.

A figura 2.1 mostra um diagrama que ilustra a sequência de passos executados. A entrada para o sistema é uma coleção de imagens de uma face, um modelo genérico de face e as estimativas iniciais da posição e orientação das câmeras. A saída é o modelo 3D ajustado à face e um mapa de textura.

2.4.2 Ajuste do Modelo

Esta etapa visa deformar um modelo de face genérico para que este se ajuste à face e à expressão facial de um dado indivíduo. Como entrada para este processo tomamos um conjunto de imagens de uma face com a mesma expressão facial, de diferentes pontos de vista e um modelo 3D genérico de face.

Alguns pontos pré determinados da face (chamados de pontos de controle) são manualmente selecionados em cada uma das imagens, esses pontos são utilizados para recuperar a posição e a orientação das câmeras em cada uma das imagens. Os pontos também são utilizados para refinar o modelo genérico para que se ajuste à face presente nas imagens.

Determinação da Posição e da Orientação das Câmeras

A partir de uma estimativa grosseira da posição e orientação das câmeras e da forma 3D da face (dada pelo modelo genérico) o método refina iterativamente tais estimativas de modo a

Figura 2.1: Diagrama do sistema.

minimizar a diferença entre os pontos de controle observados nas imagens e os pontos previstos pelo modelo.

A formulação é baseada no algoritmo não linear de mínimos quadrados. Para formular o problema, associa-se a cada câmera uma matriz de rotação R e um vetor de translação t (e denota-se as linhas de R por r_x , r_y e r_z e os elementos de t por t_x , t_y e t_z). Sejam p_i os pontos de controle 3D e (x_i, y_i) suas coordenadas 2D.

Assumindo que a origem do sistema de coordenadas da imagem está no centro óptico de cada imagem, a equação tradicional de projeção para uma câmera com distância focal f pode ser dada por:

$$x_i = f \frac{r_x \cdot p_i + t_x}{r_z \cdot p_i + t_z} \quad y_i = f \frac{r_y \cdot p_i + t_y}{r_z \cdot p_i + t_z} \quad (2.4)$$

O problema é resolvido pelo método de otimização contínua de Levenberg-Marquardt. A idéia é encontrar matrizes R , vetores t , distâncias focais f e pontos 3D p_i , que minimizem a soma das diferenças entre a projeção dos pontos de controle prevista pela equação acima e os pontos marcados pelo usuário em cada imagem.

Interpolação de Dados Esparsos

Uma vez determinadas as coordenadas 3D dos pontos p_i de controle, tais pontos são usados para deformar os demais vértices do modelo genérico. Para tanto, constrói-se uma função suave de interpolação que fornece o deslocamento entre a posição original e a nova posição adaptada para todos os vértices do modelo.

Construir tal função de interpolação é um problema clássico de interpolação de dados esparsos: Dados um conjunto conhecido de deslocamentos $u_i = p_i - p_i^{(0)}$, (onde $p_i^{(0)}$ denota a posição original do ponto e p_i denota a nova posição depois do deslocamento) para cada vértice de controle i , construir uma função que nos forneça o deslocamento u_j para todos os

vértices j do modelo.

Existem diversos fatores a serem considerados ao se escolher tal função. A primeira consideração é com relação ao domínio da função. Neste caso os autores utilizam as coordenadas 3D dos pontos de controle como domínio. Uma alternativa seria utilizar alguma parametrização da superfície, ou coordenadas cilíndricas, por exemplo.

Sendo assim, o problema é encontrar uma função vetorial $f(p)$ ajustada nos pontos conhecidos de modo que $u_i = f(p_i)$, a partir da qual seja possível calcular $u_j = f(p_j)$ para todo j .

Existem várias escolhas possíveis de como construir essa interpolação. A opção utilizada por Pighin é baseada em funções de base radial (*radial basis function* [Nie93]), que são funções da forma:

$$f(p) = \sum_i c_i \phi(\|p - p_i\|) \quad (2.5)$$

onde $\phi(r)$ é uma função de base radialmente simétrica. Para determinar os coeficientes c_i , basta resolver o sistema de equações lineares $u_i = f(p_i)$. Vale notar que as coordenadas x, y, z de c_i podem ser determinadas independentemente usando o sistema de equações. A função utilizada no trabalho em questão é $\phi(r) = e^{-r/64}$ que, segundo os autores, apresenta resultados satisfatórios.

2.4.3 Extração dos Mapas de Texturas

Os mapas de texturas são fundamentais para a renderização realista do modelo reconstruído. Em [PHL⁺98, PSS99] encontramos um método para obter um único mapa de textura independente do ponto de vista, que pode ser utilizado para renderizar o modelo rapidamente, e também uma técnica de mapeamento de textura dependente do ponto de vista que é mais

lenta mas gera resultados mais precisos.

O processo de extração de textura segue os passos abaixo:

1. Um mapeamento entre as coordenadas 3D dos pontos do modelo de face e o espaço 2D de textura é definido através de uma projeção cilíndrica.
2. Para cada pixel da textura centrado nas coordenadas cilíndricas (u, v) :
 - Calcula-se o ponto 3D p na superfície da face cuja projeção cilíndrica seja (u, v) ;
 - Para cada câmera j , calcula-se (x_j, y_j) , a projeção de p na imagem correspondente e determina-se a cor do pixel naquele ponto $I_j(x_j, y_j)$;
 - A cor da textura no ponto (u, v) é definida como uma combinação linear das cores dos pontos correspondentes nas imagens:

$$T(u, v) = \sum_j w_j(u, v) I_j(x_j, y_j) \quad (2.6)$$

onde $w_j(u, v)$ é a contribuição relativa da j -ésima imagem para o pixel (u, v) do mapa de textura, tal que $\sum_j w_j(u, v) = 1$.

Mapas de Pesos

Construir os mapas de pesos para combinar as imagens de entrada num mapa de textura é provavelmente a parte mais interessante da técnica de mapeamento de textura. Existem várias propriedades que um mapa de pesos deve ter:

- Se o ponto p na superfície correspondente ao pixel (u, v) da textura não é visível na j -ésima imagem então o peso $w_j(u, v)$ deve ser zero.

- Os pesos em cada um dos mapas devem variar de modo suave, para garantir uma transição imperceptível entre as várias imagens de entrada.
- O produto escalar entre o vetor normal à superfície no ponto p e a direção de projeção, chamado de *certeza posicional*, também influencia no valor final de w .
- Finalmente, para produzir um mapa de textura dependente do ponto de vista, o peso $w_j(u, v)$ deve depender também da similaridade entre a direção de projeção de p na j -ésima imagem e a direção de projeção no ponto de vista que está sendo renderizado.

Para determinar mapas de peso que satisfaçam as propriedades citadas acima procede-se da seguinte maneira: Primeiramente, define-se cada entrada de w_j como 1 se o ponto correspondente p é visível na j -ésima imagem e 0 caso contrário. Em seguida, interpola-se suavemente os valores nas vizinhanças das fronteiras do mapa binário obtido. Finalmente, cada $w_j(u, v)$ é multiplicado pela *certeza posicional* e os pesos são normalizados para somarem 1.

Textura Dependente do Ponto de Vista

Em vez de combinar todas as imagens de entrada num único mapa de textura independente do ponto de vista, pode-se adiar a combinação até que o ponto de vista a partir do qual o modelo será renderizado seja especificado.

Cada uma das imagens de entrada é mapeada no espaço cilíndrico de textura e armazenada juntamente com o seu mapa de pesos. Uma vez que o ponto de vista seja especificado, aumenta-se o peso relativo das imagens cujas câmeras correspondentes sejam mais parecidas com a câmera referente ao ponto de vista a partir do qual o modelo será renderizado.

Combinar os mapas individuais dessa forma resulta num mapa de textura dependente do ponto de vista. Este mapeamento é mais caro pois o mapa deve ser recalculado uma vez a cada quadro, no caso de uma animação.

2.4.4 Transições Entre Expressões Faciais

Um dos objetivos ao modelarmos faces é a possibilidade de gerar transições contínuas e realistas entre diferentes expressões faciais. É possível atingir este objetivo através do *morphing* dos modelos obtidos para cada expressão facial.

Em geral o problema de *morphing* entre superfícies poligonais genéricas é um problema complexo, uma vez que é necessária a correspondência entre os vértices das superfícies, que podem possuir topologias bem diferentes.

Neste caso, a topologia das duas superfícies é a mesma. Existe uma correspondência natural entre os vértices. Sendo assim, é possível atingir um efeito bastante realista através de *morphing* 3D usando uma simples interpolação linear entre os vértices das duas superfícies.

Junto com a interpolação da geometria do modelo, é necessário combinar os mapas de textura correspondentes. Novamente, em geral o *morphing* entre duas imagens requer a correspondência entre pontos característicos nas imagens. No caso em questão a correspondência está implícita nas coordenadas cilíndricas das superfícies associadas.

Em vez de combinar as duas texturas para formar uma intermediária, o modelo geométrico intermediário (obtido com *morphing* 3D) é renderizado primeiro com uma textura e depois com a outra. As duas imagens resultantes são então combinadas.

2.5 Comentários

Analisando as várias técnicas existentes decidimos nos concentrar nas que são baseadas em múltiplas imagens. Dentre os motivos para tal escolha podemos citar o baixo custo do equipamento envolvido, a facilidade de obtenção das imagens e os resultados promissores que já podem ser encontrados na literatura.

A partir das imagens de entrada extraímos um conjunto de correspondências de pontos

característicos que servirão como principal fonte de informação para estimar a geometria da face. Esta estimativa é feita utilizando-se os conceitos da geometria projetiva que é capaz de lidar de forma bastante geral e elegante com o problema de reconstrução baseado em correspondência entre pontos.

Capítulo 3

Geometria Projetiva

A geometria projetiva [HZ04] nos dá uma base teórica bem estruturada e ferramentas convenientes para tratarmos de problemas como o da reconstrução 3D de forma simples e elegante. Usando os conceitos de tal geometria podemos modelar de forma direta o conceito de *câmera* (projeções do espaço 3D no plano) e estudar de maneira clara as relações geométricas inerentes de várias imagens de uma cena obtidas por diferentes câmeras.

Encontramos na literatura vários trabalhos que utilizam geometria projetiva para reconstrução de objetos rígidos como prédios e arquitetura em geral [DTM96, DYB98, FCZ98, PKG99, PKVG00], no entanto, sua utilização não é relatada com frequência para reconstrução de objetos não rígidos como a face.

3.1 Geometria Projetiva 2D

Nesta seção apresentaremos os conceitos básicos de geometria projetiva plana, como a representação de pontos e retas em coordenadas homogêneas e as transformações projetivas do plano, também chamadas de homografias.

3.1.1 O Plano Projetivo

Uma reta no plano pode ser representada por uma equação do tipo $ax + by + c = 0$, sendo que diferentes escolhas de a , b e c levam a diferentes retas. Desse modo uma reta pode ser naturalmente representada pelo vetor $(a, b, c)^T$.

No entanto a correspondência entre vetores $(a, b, c)^T$ e retas não é injetora uma vez que as retas $ax + by + c = 0$ e $(ka)x + (kb)y + (kc) = 0$ são a mesma para todo k não nulo. Conseqüentemente os vetores $(a, b, c)^T$ e $k(a, b, c)^T$ representam a mesma reta para todo k não nulo.

De fato, dois vetores que diferem apenas por um fator de escala serão então considerados equivalentes. A classe de equivalência de vetores sob essa relação é conhecida como *vetor homogêneo*, sendo que qualquer vetor é representante de sua classe. O conjunto de classes de equivalência dos vetores de $R^3 - (0, 0, 0)^T$ forma o plano projetivo P^2 .

Vejamos agora como podemos representar, além de retas, pontos com coordenadas homogêneas. Um ponto $\mathbf{x} = (x, y)^T$ pertence à reta $\mathbf{l} = (a, b, c)^T$ se e somente se $ax + by + c = 0$ ou em notação vetorial $(x, y, 1)(a, b, c)^T = (x, y, 1)\mathbf{l} = 0$. Representamos o ponto (x, y) por um vetor 3D com a última coordenada (coordenada homogênea) igual a 1.

Notemos que para todo escalar não nulo k e reta $\mathbf{l}$, a equação $(kx, ky, k)\mathbf{l} = 0$ se e somente se $(x, y, 1)\mathbf{l} = 0$, sendo portanto natural considerar que todos os vetores (kx, ky, k) , para diferentes valores de k , sejam representações do mesmo ponto (x, y) .

Sendo assim, do mesmo modo que as retas, os pontos são representados por vetores homogêneos e são elementos de P^2 . Um vetor homogêneo arbitrário representando um ponto tem a forma $(x_1, x_2, x_3)^T$ e denota o ponto $(x_1/x_3, x_2/x_3)$ do plano real.

Com as noções apresentadas até aqui já é possível derivar alguns resultados:

- Pertinência de pontos em retas: Como já mencionado, para qualquer ponto $\mathbf{x}$ e qualquer reta $\mathbf{l}$ ambos em P^2 , vale que o ponto $\mathbf{x}$ pertence a reta $\mathbf{l}$ se e somente se $\mathbf{x}^T \mathbf{l} = 0$.

- Intersecção entre duas retas: Dadas duas retas $\mathbf{l} = (a, b, c)^T$ e $\mathbf{l}' = (a', b', c')^T$, podemos verificar facilmente pelas propriedades de produto misto que $\mathbf{x} = \mathbf{l} \times \mathbf{l}'$ é a intersecção das retas. De fato $\mathbf{l} \cdot (\mathbf{l} \times \mathbf{l}') = \mathbf{l}' \cdot (\mathbf{l} \times \mathbf{l}') = 0$ e portanto $\mathbf{x}$ pertence às duas retas.
- Reta por dois pontos: De forma análoga, dados dois pontos $\mathbf{x}$ e $\mathbf{x}'$, a reta $\mathbf{l}$ que liga os dois pontos pode ser dada por $\mathbf{l} = \mathbf{x} \times \mathbf{x}'$.

Pontos Ideais e a Reta no Infinito:

Considere duas retas paralelas $\mathbf{l} = (a, b, c)^T$ e $\mathbf{l}' = (a, b, c')^T$ para as quais as duas primeiras coordenadas são iguais. Podemos calcular a intersecção dessas retas utilizando o resultado apresentado há pouco. A intersecção é $\mathbf{x} = \mathbf{l} \times \mathbf{l}' = (c' - c)(b, -a, 0)^T$, ignorando o fator de escala, temos $\mathbf{x} = (b, -a, 0)^T$.

Se tentarmos encontrar a representação não homogênea deste ponto obtemos $(b/0, -a/0)$, o que não faz sentido, apenas sugere que o ponto teria coordenadas infinitamente grandes. Em geral, vetores homogêneos $(x, y, 0)$ não correspondem a nenhum ponto finito do R^2 . Esta observação está de acordo com a idéia usual de que retas paralelas se encontram no infinito.

Pontos da forma $(x, y, 0)$ são chamados de *pontos ideais* ou *pontos no infinito*. Tais pontos pertencem a uma reta comum chamada de *reta no infinito*, denotada pelo vetor $\mathbf{l}_{\text{inf}} = (0, 0, 1)$.

3.1.2 Transformações Projetivas do Plano

Uma transformação projetiva ou *projetividade* é um mapeamento invertível h de P^2 em P^2 , tal que pontos $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ são colineares se e somente se $h(\mathbf{x}_1), h(\mathbf{x}_2), h(\mathbf{x}_3)$ são colineares. Em outras palavras, é um mapeamento invertível que leva retas em retas.

Um mapeamento $h : P^2 \rightarrow P^2$ é uma projetividade se e somente se existe uma matriz 3×3 não singular H tal que para qualquer ponto $\mathbf{x} \in P^2$ vale que $h(\mathbf{x}) = H\mathbf{x}$. Isto significa

que qualquer projetividade pode ser representada por uma transformação linear invertível em coordenadas homogêneas.

Vejamos agora diferentes classes de transformações projetivas e suas propriedades geométricas.

Transformações Isométricas ou Isometrias

Transformações isométricas são mapeamentos do plano real que preservam a distância euclidiana. Tais transformações podem ser representadas por:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \epsilon \cos \theta & -\sin \theta & t_x \\ \epsilon \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.1)$$

onde $\epsilon = \pm 1$. Se $\epsilon = 1$ então a transformação preserva a orientação e é chamada de transformação euclidiana (composição de translação e rotação). Se $\epsilon = -1$ então a transformação inverte a orientação e pode ser interpretada como a composição de uma reflexão e uma transformação euclidiana.

Transformações de Similaridade

Uma transformação de similaridade é uma isometria composta com um escalamento isotrópico (igual em todas as direções). Podemos representar o caso de uma transformação euclidiana composta com escalamento da seguinte forma:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s \cos \theta & s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.2)$$

onde s representa o fator de escala isotrópico. Uma transformação de similaridade preserva

o ângulo entre retas e a razão entre distâncias.

Transformações Afins ou Afinidades

Uma transformação afim é uma transformação linear não singular seguida de uma translação. Possui a seguinte representação matricial:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.3)$$

onde a sub matriz 2x2 A_{ij} formada pelos elementos a_{ij} é não singular. Uma maneira de interpretar tal transformação é notar que podemos decompor a transformação A em função de rotações e escalamento não isotrópico. A sub matriz A pode sempre ser decomposta na forma:

$$A = R(\theta)R(-\phi)DR(\phi) \quad (3.4)$$

onde $R(\theta)$ e $R(\phi)$ são matrizes de rotação por θ e ϕ respectivamente e D é a matriz diagonal: $\text{diag}(\lambda_1, \lambda_2)$, que representa o escalamento não isotrópico.

Em uma transformação afim linhas paralelas são mapeadas em linhas paralelas, a razão entre segmentos de retas paralelas é preservada e também a razão entre áreas.

Transformações Projetivas ou Projetividades

Como já mencionado, uma transformação projetiva é qualquer transformação invertível que leva retas em retas, podendo ser representada por uma matriz não singular 3x3. Além de preservar a colinearidade, tais transformações preservam também o *cross ratio*, que é a razão das razões das distâncias entre pontos.

3.2 Geometria projetiva 3D

As idéias apresentadas na seção anterior podem ser facilmente generalizadas para o caso 3D. Vamos nos concentrar apenas nas diferenças e propriedades geométricas adicionais que são introduzidas pela dimensão extra.

3.2.1 Pontos, Planos e o Espaço Projetivo

Analogamente ao caso 2D, pontos e planos são duais em P^3 , ambos podem ser representados por vetores homogêneos da forma $\mathbf{x} = (x_1, x_2, x_3, x_4)$. Quando $x_4 \neq 0$, podemos interpretar $\mathbf{x}$ como o ponto finito $(x_1/x_4, x_2/x_4, x_3/x_4) \in R^3$. Quando x_1, x_2 e x_3 não são todos nulos podemos interpretar $\mathbf{x}$ como o plano real $ax + by + cz + d = 0$.

Pontos da forma $(x_1, x_2, x_3, 0)$ não representam pontos finitos reais e são denominados pontos ideais (pontos no infinito). O conjunto de tais pontos pertencem a um mesmo plano, chamado plano no infinito, representado pelo vetor $\pi_{\text{inf}} = (0, 0, 0, 1)$.

Pode-se verificar facilmente que um dado ponto $\mathbf{x} \in P^3$ pertence ao plano $\pi \in P^3$ se e somente se $\pi^T \mathbf{x} = 0$. Tal verificação segue diretamente da equação que define um plano.

3.2.2 Transformações Projetivas Do Espaço

De modo análogo ao caso 2D uma transformação projetiva do espaço é um mapeamento invertível que leva retas em retas, isto é, preserva a colinearidade dos pontos. Prova-se que qualquer transformação projetiva pode ser representada por uma matriz 4x4 não singular H e que qualquer matriz 4x4 não singular H é a representação de alguma transformação projetiva.

A mesma hierarquia de transformações que definimos no caso 2D pode ser considerada no caso 3D também. Denotando as transformações euclidianas por H_e , as de similaridade por H_s , as afins por H_a e as projetivas em geral por H , tais transformações podem ser representadas

pelas seguintes matrizes:

$$H_e = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad H_s = \begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad H_a = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad H = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix}$$

Onde A é uma matriz 3x3 não singular, R é uma matriz 3x3 de rotação, $\mathbf{t} = (t_x, t_y, t_z)^T$ é um vetor de translação, $\mathbf{v}$ é um vetor 3D arbitrário, v e s são escalares e $\mathbf{0} = (0, 0, 0)^T$.

3.3 Câmeras

Uma câmera projetiva é um mapeamento (projeção) do espaço projetivo 3D para o plano projetivo 2D. As câmeras podem ser divididas em câmeras finitas cujo centro de projeção é um ponto finito, e câmeras ideais ou câmeras no infinito cujo centro de projeção é um ponto ideal, que generalizam a idéia de projeção ortográfica.

3.3.1 Câmeras Finitas

Começaremos com o modelo de câmera mais especializado e mais simples que é conhecido como modelo de câmera *pinhole* e progressivamente generalizamos tal modelo em alguns passos até o modelo de câmeras projetivas no caso mais geral.

O Modelo *Pinhole*

Considere a projeção central de pontos do espaço num plano. Seja o centro de projeção a origem de um sistema de coordenadas euclidianas e o plano $z = f$, chamado de *plano da imagem* ou *plano focal*.

Sob o modelo de câmera pinhole, um ponto no espaço com coordenadas $\mathbf{X} = (x, y, z)^T$ é mapeado no ponto de intersecção do plano da imagem com a reta que liga $\mathbf{X}$ ao centro de

Figura 3.1: Câmera *pinhole*: Centro de projeção O_c , plano principal $X_c Y_c$, plano da imagem $x_c y_c$, ponto principal p , ponto no espaço X e sua projeção x .

projeção. Por semelhança de triângulos podemos verificar que o mapeamento é dado por:

$$(x, y, z)^T \rightarrow (fx/z, fy/z) \quad (3.5)$$

O centro de projeção é também chamado de *centro da câmera* ou *centro óptico*. A reta que passa pelo centro da câmera e é perpendicular ao plano da imagem é chamada de *eixo principal* ou *raio principal*. O ponto de intersecção do eixo principal com o plano da imagem é chamado de *ponto principal*. O plano que passa pelo centro da câmera e é paralelo ao plano da imagem é chamado de *plano principal*.

A figura 3.1 mostra as principais entidades geométricas envolvidas no modelo de câmera pinhole.

Representação Homogênea:

Se os pontos do espaço e do plano forem representados por vetores homogêneos, então a projeção pode ser expressa de forma bastante simples como um mapeamento linear em coordenadas homogêneas:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fx \\ fy \\ z \end{pmatrix} = \begin{bmatrix} f & 0 \\ & f & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (3.6)$$

Se denotarmos os pontos do espaço por $\mathbf{X}$, os pontos do plano por $\mathbf{x}$ e a matriz de projeção da câmera por P , podemos reescrever a equação como $\mathbf{x} = P\mathbf{X}$.

Deslocamento do Ponto Principal

As equações anteriores assumem que a origem do sistema de coordenadas do plano da imagem é o ponto principal. Na prática isso pode não acontecer, e de forma mais geral teríamos o seguinte mapeamento:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fx + zp_x \\ fy + zp_y \\ z \end{pmatrix} = \begin{bmatrix} f & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (3.7)$$

onde $(p_x, p_y)^T$ são as coordenadas do ponto principal. Se escrevermos:

$$K = \begin{bmatrix} f & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \quad (3.8)$$

o mapeamento pode ser reescrito como $\mathbf{x} = K[I|0]\mathbf{X}$.

Rotação e Translação da Câmera

Até o momento assumimos que o centro da câmera está na origem do sistema de coordenadas e que a direção do eixo principal é a do eixo z , de modo que o sistema de coordenadas da câmera e da cena sejam o mesmo. Em geral, as coordenadas dos pontos no espaço são dadas em termos de um sistema de coordenadas da cena que não coincide com o da câmera.

Os dois sistemas estão relacionados por uma rotação e uma translação, de tal modo que se $\mathbf{X}$ denota um ponto no sistema de coordenadas da cena, $\mathbf{X}_{\text{cam}}$ o mesmo ponto no frame de coordenadas da câmera e $\mathbf{C}$ o centro da câmera no sistema de coordenadas da cena, então podemos escrever:

$$\mathbf{X}_{\text{cam}} = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix} \mathbf{X} \quad (3.9)$$

Onde R é a matriz de rotação que relaciona os dois frames de coordenadas. Juntando este resultado com a equação do mapeamento da câmera obtemos a seguinte fórmula:

$$\mathbf{x} = KR[I|-\mathbf{C}]\mathbf{X} \quad (3.10)$$

onde $\mathbf{X}$ é dado em função do frame de coordenadas da cena. Este é o mapeamento dado

por uma câmera pinhole no caso geral. A matriz K é chamada de matriz de calibragem e seus elementos são chamados de parâmetros internos da câmera, uma câmera cuja matriz K é conhecida é dita calibrada. Os elementos de R e as coordenadas $\mathbf{C}$ que definem a orientação e posição da câmera são chamados de parâmetros externos.

Câmeras CCD

O modelo de câmera pinhole assume que o sistema de coordenadas da imagem possui a mesma escala em ambas as direções x e y . No caso de câmeras CCD existe a possibilidade de termos pixels retangulares, que implica fatores de escala diferentes em cada eixo.

Em particular, se o número de pixels por unidade de distância nas coordenadas da imagem é m_x e m_y nas direções x e y respectivamente, então o mapeamento definido pela câmera é obtido multiplicando-se a matriz de calibragem pela matriz diagonal $(m_x, m_y, 1)$, o que nos leva a seguinte matriz de calibragem:

$$K = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \quad (3.11)$$

Onde $\alpha_x = fm_x$, $\alpha_y = fm_y$ representam a distância focal da câmera em termos das dimensões dos pixels nas direções x e y respectivamente. Similarmente, (x_0, y_0) representa a posição do ponto principal em termos das dimensões dos pixels.

Câmeras Projetivas Finitas

Câmeras projetivas finitas na sua forma mais geral são aquelas cuja matriz de calibragem é da seguinte forma:

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \quad (3.12)$$

O parâmetro extra s é chamado de parâmetro de cizalhamento (*skew*). Quando $s \neq 0$, os eixos x e y do sistema de coordenadas da imagem não são perpendiculares, de modo que para a maioria das câmeras reais teremos $s = 0$.

3.3.2 Câmeras no Infinito

Como vimos, as câmeras projetivas finitas podem ser representadas por matrizes P da forma: $P = KR[I] - C$. Pode-se verificar que a sub-matriz 3x3 da esquerda KR é não singular. Qualquer matriz 3x4 cuja sub-matriz 3x3 da esquerda seja não singular é a matriz de alguma câmera projetiva finita.

O último passo para generalização das câmeras projetivas é eliminar a restrição de não singularidade da sub-matriz 3x3 da esquerda, de modo que uma câmera projetiva é qualquer matriz 3x4 de posto 3. A restrição de posto 3 é necessária pois caso contrário a imagem da transformação não será um plano e sim uma reta ou mesmo um único ponto.

Quando a sub-matriz 3x3 é singular significa que o centro de projeção da câmera é um ponto ideal. Entre tais câmeras as mais importantes são as chamadas câmeras afins cujo plano da imagem é paralelo ao plano no infinito π_{inf} .

3.4 Geometria Epipolar

A geometria epipolar é a geometria projetiva inerente de duas projeções. Isto é, relaciona os pontos de duas imagens obtidas por duas câmeras. A geometria epipolar é independente da

estrutura da cena e depende apenas dos parâmetros internos das câmeras envolvidas e suas posições e orientações relativas.

Suponha que um ponto $\mathbf{X}$ do espaço seja projetado em duas imagens. Em $\mathbf{x}$ na primeira imagem e em $\mathbf{x}'$ na segunda. Vejamos que relações existem entre tais pontos.

Primeiramente, notamos que os pontos $\mathbf{x}$ e $\mathbf{x}'$ nas imagens, o ponto $\mathbf{X}$ no espaço e os centros $\mathbf{C}$ e $\mathbf{C}'$ das câmeras são todos coplanares. Vamos denotar este plano por π . Não é difícil perceber que a projeção inversa de $\mathbf{x}$ e $\mathbf{x}'$ são retas que se encontram em $\mathbf{X}$ e pertencem a π .

Supondo agora que conhecemos apenas o ponto $\mathbf{x}$ (e as câmeras), o plano π pode ser definido pela *baseline* (reta que une os centros das câmeras) e a projeção inversa de $\mathbf{x}$. Como $\mathbf{x}'$ também pertence a π , segue que $\mathbf{x}'$ deve estar na reta l' que é a intersecção do plano π com o plano da segunda imagem.

As entidades envolvidas na geometria epipolar são:

- **Epipolo:** É o ponto de intersecção da *baseline* com o plano da imagem, em outras palavras, é a projeção do centro de uma câmera no plano de imagem da outra.
- **Plano epipolar:** É um plano que contém a *baseline*.
- **Reta epipolar:** É a intersecção de um plano epipolar com o plano da imagem. Todas as linhas epipolares se interceptam no epipolo. Um plano epipolar intersepta as duas imagens em retas epipolares, o que define a correspondência entre pontos nas duas imagens.

A figura 3.2 apresenta os elementos envolvidos na geometria epipolar inerente da projeção de um ponto $\mathbf{X}$ por duas câmeras projetivas.

Figura 3.2: Geometria epipolar: Centro das câmeras C_1 e C_2 , os epipolos e_1 , e e_2 , as retas epipolares l_1 e l_2 , o ponto no espaço X e suas projeções q_1 e q_2 .

3.4.1 A Matriz Fundamental

A matriz fundamental é a representação algébrica da geometria epipolar. Dado um par de imagens, vimos que para cada ponto $\mathbf{x}$ na primeira imagem existe uma reta epipolar correspondente l' na segunda imagem. O ponto $\mathbf{x}'$ na segunda imagem correspondente a $\mathbf{x}$ deve pertencer a reta l' .

Desse modo existe um mapeamento $\mathbf{x} \rightarrow l'$, que leva um ponto $\mathbf{x}$ de uma imagem na sua reta epipolar correspondente na outra. Ocorre que este mapeamento é linear e singular que pode ser representado por uma matriz 3×3 F , chamada de matriz fundamental.

Dadas duas câmeras P e P' a matriz fundamental pode ser dada por:

$$F = [\mathbf{e}']_{\times} P' P^+ \quad (3.13)$$

onde P^+ é a pseudo-inversa de P (tal que $PP^+ = I$) e $\mathbf{e}'$ é o epipolo na segunda imagem e se $\mathbf{e}' = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$, então $[\mathbf{e}]_{\times}$ denota a matriz:

$$\begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}$$

Segue que se dois pontos $\mathbf{x}$ e $\mathbf{x}'$ na primeira e na segunda imagem respectivamente são correspondentes (isto é, são a imagem do mesmo ponto do espaço) então $\mathbf{x}'^T F \mathbf{x} = 0$.

3.4.2 Obtenção das Câmeras a Partir das Matriz Fundamental

Vejamos agora uma das propriedades mais importantes da matriz fundamental F , que ela pode ser usada para determinar as matrizes das câmeras correspondentes às duas imagens a menos de uma transformação projetiva.

Ambigüidade projetiva: Vimos que dadas duas câmeras podemos determinar unicamente a matriz fundamental correspondente. Porém, esse mapeamento entre pares de câmeras e matrizes fundamentais não é injetor uma vez que pares de câmeras que estejam relacionadas por uma transformação projetiva possuem a mesma matriz fundamental.

Em outras palavras, se F é a matriz fundamental correspondente ao par de câmeras (P, P') , então F é também a matriz fundamental correspondente ao par $(PH, P'H)$, para qualquer transformação projetiva H .

Desse modo, dada a matriz fundamental F , podemos determinar as câmeras (P, P') apenas a menos de uma transformação projetiva. Sendo assim, é comum a utilização de uma forma canônica para o par de câmeras, em que a primeira matriz P é da forma $P = [I|\mathbf{0}]$ e a segunda pode ter a forma genérica $P' = [M|\mathbf{m}]$, com isso é possível mostrar que dada F , as câmeras podem ser escolhidas como:

$$(P = [I|\mathbf{0}], P' = [[\mathbf{e}']_{\times} F | \mathbf{e}']) \quad (3.14)$$

3.5 Reconstrução

Vejamos agora como e até que ponto é possível recuperar a estrutura 3D da cena e das câmeras dadas duas imagens. Suponha que um conjunto de correspondências $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ seja dado. Supomos que tais correspondências venham do mesmo conjunto desconhecido de pontos 3D. Da mesma forma, a posição, orientação e parâmetros internos das câmeras são desconhecidos. O problema de reconstrução é encontrar câmeras P e P' e um conjunto de pontos 3D $\mathbf{X}_i$, tais que:

$$\mathbf{x}_i = P\mathbf{X}_i, \quad \mathbf{x}'_i = P'\mathbf{X}_i, \quad \forall i \quad (3.15)$$

Se forem dadas correspondências suficientes, podemos determinar a matriz fundamental unicamente e então obter a estrutura da cena e das câmeras a menos de uma transformação projetiva. Essa ambigüidade pode ser reduzida se tivermos mais alguma informação a respeito das câmeras ou da cena.

Uma reconstrução a menos de uma transformação projetiva é chamada de reconstrução projetiva, analogamente a menos de uma transformação afim é chamada de reconstrução afim e a menos de uma similaridade de reconstrução de similaridade ou reconstrução métrica.

3.5.1 A matriz Fundamental a Partir das Correspondências

Vimos que dadas correspondências $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ em duas imagens, a matriz fundamental F satisfaz a condição $\mathbf{x}'_i{}^T F \mathbf{x}_i = 0$ para todo i . Conhecendo os pontos $\mathbf{x}_i$ e $\mathbf{x}'_i$ essa equação é linear nas entradas de F . De fato, cada correspondência gera uma equação linear e portanto, dadas pelo menos 8 correspondências é possível determinar F a menos de escala (como F é uma matriz homogênea, a escala é irrelevante). Quando mais de 8 correspondências são dadas podemos achar uma solução de mínimos quadrados.

3.5.2 Reconstrução das Câmeras e da Cena

Podemos determinar facilmente um par de câmeras (P, P') correspondentes à matriz fundamental F utilizando os resultados apresentados na seção anterior.

Uma vez determinadas as câmeras podemos determinar os pontos 3D $\mathbf{X}_i$ lembrando que os pontos $\mathbf{x}_i$, $\mathbf{x}'_i$, $\mathbf{X}_i$, e os centros das câmeras pertencem a um mesmo plano. Sendo assim, podemos determinar o ponto $\mathbf{X}_i$ como sendo a intersecção da reta que liga o centro $\mathbf{C}$ da primeira câmera ao ponto $\mathbf{x}_i$ e da reta que liga o centro $\mathbf{C}'$ da segunda câmera ao ponto $\mathbf{x}'_i$.

Os únicos pontos que não podem ser determinados a partir de suas imagens são os pontos da *baseline* entre as duas câmeras, pois a projeção inversa de tais pontos é a própria *baseline*.

Tais pontos são projetados nos epipolos.

3.5.3 Reconstrução Afim

Como mencionado anteriormente, sem nenhuma informação adicional a respeito da cena ou das câmeras a reconstrução das câmeras e da estrutura 3D da cena é possível apenas a menos de uma transformação projetiva. Para várias aplicações, como a reconstrução 3D de faces, esse nível de precisão não é suficiente, devido às deformações como o escalamento não isotrópico e o fator de cisalhamento que são características das transformações afins.

Podemos refinar a reconstrução para uma reconstrução afim se pudermos determinar de algum modo as coordenadas do plano no infinito no frame de coordenadas reconstruído, pois uma transformação é afim se e somente se ela preserva o plano no infinito.

Desse modo, se determinarmos que as coordenadas do plano no infinito no frame reconstruído são dadas por $\pi = (a, b, c, d)$, pode-se verificar que existe uma transformação H que leva o plano π nas suas coordenadas canônicas $(0, 0, 0, 1)$. Se aplicarmos essa transformação H nas câmeras e nos pontos reconstruídos obtemos uma reconstrução que difere da ideal por no máximo uma transformação afim, pois preserva as coordenadas do plano no infinito.

A transformação H é dada por:

$$H = \begin{bmatrix} I & 0 \\ \pi^T & 1 \end{bmatrix} \quad (3.16)$$

A maneira mais direta de se determinar o plano no infinito é identificar três conjuntos de retas paralelas (cada conjunto com uma direção diferente), pois a intersecção de cada conjunto de retas é um ponto do plano no infinito e com três desses pontos podemos determinar o plano.

Reconstrução métrica

De forma análoga à reconstrução afim, a chave para a reconstrução métrica é determinar uma cônica especial chamada de *cônica absoluta* Ω_{inf} . Esta é uma cônica de pontos puramente imaginários pertencentes π_{inf} , definida por:

$$x_1^2 + x_2^2 + x_3^2 = 0, \quad x_4 = 0 \quad (3.17)$$

cuja representação matricial no sistema de coordenadas canônicas é a identidade: $\Omega_{inf} = I_3$.

Identificar tal curva é suficiente para refinar a reconstrução pois, analogamente ao caso da reconstrução afim, uma transformação é de similaridade se e somente se ela preserva Ω_{inf} e, conseqüentemente, a projeção de tal curva nas imagens. A imagem da cônica absoluta é também uma cônica no plano da imagem, representada por ω .

Sendo assim se determinarmos um mapeamento que leve a cônica absoluta em suas coordenadas canônicas $\Omega_{inf} = I$ e aplicarmos tal mapeamento às câmeras e aos pontos reconstruídos obtemos uma reconstrução métrica.

Tal mapeamento é dado por:

$$H = \begin{bmatrix} A^{-1} & \\ & 1 \end{bmatrix} \quad (3.18)$$

Onde A é obtida por decomposição de Cholesky a partir da equação $AA^T = (M^T \omega M)^{-1}$. Várias informações podem ser utilizadas para restringir ω e na prática a combinação de tais restrições são utilizadas para determinar ω . Entre tais restrições podemos destacar:

- Dois conjuntos ortogonais de retas paralelas: sejam v_1 e v_2 as intersecções de cada um dos conjuntos de retas, temos: $v_1^T \omega v_2 = 0$.

- *skew*: se o parâmetro de skew é $s = 0$ então $\omega_{12} = \omega_{21} = 0$. Se além disso $\alpha_x = \alpha_y$, então $\omega_{11} = \omega_{22}$.
- Câmera calibrada: se a matriz K de uma das câmeras é conhecida, então $\omega = K^{-T}K^{-1}$.

Reconstrução a Partir de Várias Imagens

Existem generalizações das idéias apresentadas para o caso de 3, 4 ou mais imagens. Um resultado muito importante é o fato que podemos obter uma reconstrução métrica a partir da correspondência $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$ de pelo menos 5 pontos em pelo menos 3 imagens, se os parâmetros internos das câmeras forem constantes em todas as imagens. Este pode ser o caso em que obtemos as imagens de pontos de vista diferentes mas utilizando a mesma câmera. Este resultado ainda é válido mesmo em casos um pouco menos restritivos, como quando mudamos a distância focal da câmera de uma imagem para outra.

Capítulo 4

O Sistema de Reconstrução de Faces

Neste capítulo descrevemos o sistema para reconstrução 3D de faces a partir de múltiplas imagens que foi desenvolvido ao longo do mestrado. O sistema é baseado nos trabalhos de Pighin [PHL⁺98, PSS99] descritos na seção 2.4, no entanto algumas modificações foram feitas visando a possível automação do processo de reconstrução. As seções seguintes mostram os detalhes do sistemas dando ênfase às modificações e aos elementos que não foram cobertos no capítulo anterior.

4.1 O Modelo Genérico de Face

Como descrito na seção 2.4 o processo de reconstrução baseia-se no ajuste de um modelo 3D genérico de face às imagens de entrada. O modelo de face que utilizamos no nosso trabalho é baseado no modelo paramétrico apresentado por Waters [Wat87].

A geometria do modelo é composta por 512 vértices que definem um malha de 876 polígonos. O modelo é capaz de simular a ação de um conjunto de 18 músculos faciais que agem independentemente deformando a malha, possibilitando assim a síntese de inúmeras expressões

Figura 4.1: O modelo genérico de faces: (a) *wire frame*: vértices e músculos, (b) *flat*: polígonos e efeitos de iluminação, (c) *smooth*: polígonos suavizados, efeito mais realista.

faciais. A figura 4.1 mostra o modelo com a expressão neutra, podemos ver os vértices, os polígonos e os músculos sintéticos.

O modelo de músculo consiste de um vetor que representa a direção e sentido no qual ele atua, a posição do músculo dada por um ponto fixo, um ângulo de abertura que define um cone sobre o qual o músculo age e um raio de ação. Todos os vértices na superfície do cone têm deslocamento e esse deslocamento em direção ao centro do cone, sendo que no eixo do cone o deslocamento é máximo [Wat87].

Codificar cada expressão facial em função da situação dos 18 músculos é consideravelmente trabalhoso, tornando difícil a síntese de uma determinada expressão. Amenizamos esse problema com base no trabalho de Ekman [ER97], que diz que os seres humanos apresentam 6 tipos de expressões básicas (raiva, repulsa, medo, felicidade, surpresa e tristeza) que podem ser chamadas de expressões primitivas pois são apresentadas por todos, independentemente de raça ou cultura. Além disso, as demais expressões podem ser geradas como combinação

das expressões primitivas.

Sendo assim, associamos 6 configurações diferentes dos músculos a cada uma das expressões básicas e geramos novas expressões através de combinação linear, isto é, para cada expressão base E_i temos um vetor $v_i \in \mathbb{R}^{18}$ que determina o grau de contração de cada um dos 18 músculos, geramos uma expressão intermediária $\alpha E_i + \beta E_k$ associando a ela o vetor $\alpha v_i + \beta v_k$. A figura 4.2 mostra as 6 expressões pré-determinadas.

Implementamos uma interface simples que permite controlar com o mouse o peso de cada expressão base e dessa forma gerar animações da face de forma muito mais intuitiva do que controlando diretamente cada um dos músculos.

4.2 Pontos de Controle

Dadas as imagens de entrada, o primeiro passo é localizar em tais imagens um conjunto de pontos pré-determinados chamados de pontos de controle. Esses pontos são usados para determinar a posição e a orientação das câmeras correspondentes a cada imagem e também para deformar o modelo genérico para que se ajuste à face presente nas imagens.

Tendo em vista que queremos tornar o processo o mais automático possível, escolhemos como pontos de controle um pequeno conjunto de 12 pontos característicos da face como os cantos da boca, cantos dos olhos, etc. Cada um desses pontos é correspondente à um vértice do modelo 3D de face.

A lista completa de pontos de controle utilizados para a determinação da posição e orientação das câmeras é dada na tabela 4.1 e uma imagem do modelo com os pontos de controle em destaque é mostrada na figura 4.3.

É importante notar que não é necessário que todos os pontos estejam visíveis e sejam localizados em todas as imagens. Para câmeras não calibradas 6 pontos em cada imagem são

Figura 4.2: Algumas expressões faciais pré-definidas do modelo: (a) felicidade, (b) raiva, (c) surpresa, (d) medo, (e) repulsa, (f) tristeza.

N.	Descrição	Rótulo
01	canto esquerdo do olho esquerdo	CEOE
02	canto direito do olho esquerdo	CDOE
03	canto esquerdo do olho direito	CEOD
04	canto direito do olho direito	CDOD
05	parte superior do nariz	PTSN
06	ponta do nariz	PTNR
07	parte inferior do nariz	PTIN
08	canto esquerdo da boca	CEBC
09	canto direito da boca	CDBC
10	lábio superior	LBSP
11	lábio inferior	LBIF
12	parte superior do queixo	PTSQ

Tabela 4.1: Lista de pontos de controle.

Figura 4.3: Pontos de controle mostrados sobre o modelo de face.

suficientes para determinar os parâmetros internos e externos. Para câmeras calibradas apenas 4 pontos são necessários para determinar unicamente os parâmetros externos. Naturalmente quanto mais pontos disponíveis melhor a qualidade da estimativa dos parâmetros das câmeras.

4.3 Posição e Orientação das Câmeras

O método para determinar a posição e a orientação das câmeras correspondentes a cada imagem de entrada descrito na seção 2.4.2 requer uma aproximação inicial e simplesmente executa uma rotina de minimização não linear sobre os parâmetros de posição e orientação das câmeras.

Esse é um dos fatores que dificultam a automação do processo, no entanto existem formas de se determinar tais parâmetros sem a necessidade de uma estimativa inicial. Para câmeras não calibradas podemos utilizar uma solução linear baseada na geometria projetiva e para câmeras calibradas utilizamos o algoritmo POSIT apresentado por Dementhon [DDDS04].

4.3.1 Câmeras Projetivas

Quando não conhecemos nenhum dos parâmetros internos da câmera, podemos determinar os parâmetros internos e externos a partir de correspondências entre pontos 3D e suas projeções 2D [HZ04].

Dada uma coleção de correspondências $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ entre pontos 3D $\mathbf{X}_i$ e pontos 2D $\mathbf{x}_i$, queremos determinar uma câmera P , isto é uma matriz 3×4 de posto 3 tal que $\mathbf{x}_i = P\mathbf{X}_i$, para todo i . Primeiramente devemos notar que a equação $\mathbf{x}_i = P\mathbf{X}_i$ é uma equação envolvendo vetores homogêneos, de modo que os vetores $\mathbf{x}_i$ e $P\mathbf{X}_i$ não são iguais mas têm mesma direção.

Sendo assim a equação pode ser expressa em forma de produto vetorial como $\mathbf{x}_i \times P\mathbf{X}_i = 0$. Com isso podemos derivar uma solução linear para P . Denotando a j -ésima linha da matriz

P pelo vetor $\mathbf{p}^{jT}$ podemos escrever:

$$P\mathbf{X}_i = \begin{pmatrix} \mathbf{p}^{1T}\mathbf{X}_i \\ \mathbf{p}^{2T}\mathbf{X}_i \\ \mathbf{p}^{3T}\mathbf{X}_i \end{pmatrix}$$

Escrevendo $\mathbf{x}_i = (x_i, y_i, w_i)^T$, o produto vetorial pode ser dado explicitamente por:

$$\mathbf{x}_i \times P\mathbf{X}_i = \begin{pmatrix} y_i\mathbf{p}^{3T}\mathbf{X}_i - w_i\mathbf{p}^{2T}\mathbf{X}_i \\ w_i\mathbf{p}^{1T}\mathbf{X}_i - x_i\mathbf{p}^{3T}\mathbf{X}_i \\ x_i\mathbf{p}^{2T}\mathbf{X}_i - y_i\mathbf{p}^{1T}\mathbf{X}_i \end{pmatrix}$$

Sabendo que $\mathbf{p}^{jT}\mathbf{X}_i = \mathbf{X}_i^T\mathbf{p}^j$, obtemos 3 equações nas entradas da matriz P , que podem ser escritas da seguinte forma:

$$\begin{bmatrix} 0 & -w_i\mathbf{X}_i^T & y_i\mathbf{X}_i^T \\ w_i\mathbf{X}_i^T & 0 & -x_i\mathbf{X}_i^T \\ -y_i\mathbf{X}_i^T & x_i\mathbf{X}_i^T & 0 \end{bmatrix} \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{pmatrix} = 0 \quad (4.1)$$

Essas equações têm a forma $A_i\mathbf{p} = \mathbf{0}$, onde A_i é uma matriz 3×9 e $\mathbf{p}$ é um vetor de dimensão 9 formado pelas entradas da matriz P :

$$\mathbf{p} = \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{pmatrix}, \quad H = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \end{bmatrix}$$

com p_i sendo i -ésimo elemento do vetor $\mathbf{p}$. Embora existam 3 equações em 4.1, apenas duas delas são linearmente independentes. Sendo assim, cada correspondência de pontos nos

dá duas equações independentes nas entradas de P . Como P tem 12 entradas e (ignorando a escala) 11 graus de liberdade, são necessárias 11 equações para determiná-la, de modo que com $n \geq 6$ correspondências podemos determinar P resolvendo um sistema de equações lineares da forma $A\mathbf{p} = 0$, onde A é uma matriz $2n \times 9$ (ou $3n \times 9$, se usarmos as 3 equações para cada correspondência). Um vez determinada a matriz de projeção P , podemos encontrar os parâmetros internos e externos decompondo P da seguinte forma: $P = KR[I] - \mathbf{c}$ (seção 3.3).

4.3.2 Câmeras Calibradas

Quando os parâmetros internos das câmeras já são conhecidos, precisamos determinar apenas os 6 parâmetros externos (3 de rotação e 3 de translação). Esse problema não pode ser resolvido apenas por métodos lineares, uma maneira bastante conhecida para se resolver esse problema é o algoritmo POSIT apresentado por Dementhon [DDDS04], um procedimento iterativo cuja implementação pode ser encontrada na biblioteca do openCV e também baseia-se na correspondência de pontos 3D e suas projeções 2D.

4.4 Ajuste do Modelo

Na seção anterior vimos como determinar os parâmetros de uma câmera a partir de uma série de correspondências entre pontos 3D e suas projeções. Vejamos agora como podemos aplicar esses métodos ao nosso problema e determinar a posição e orientação das câmeras e então ajustar o modelo de face.

Considere uma determinada imagem da face de um dado indivíduo com os pontos de controle (sec. 4.2) devidamente localizados. Cada um dos pontos marcados na imagem possui um vértice correspondente no modelo 3D de face (que é usado como estimativa inicial da geometria da face presente na imagem). Desse modo temos uma coleção de correspondências entre

pontos 3D (vértices do modelo) e suas projeções (pontos marcados na imagem de entrada).

Aplicando os métodos descritos na seção 4.3 determinamos então uma estimativa para a câmera correspondente à imagem considerada e da mesma forma determinamos as câmeras correspondentes às demais imagens de entrada.

O modelo genérico de face não corresponde exatamente à face presente nas imagens, e portanto a posição e a orientação das câmeras obtidas podem diferir significativamente dos valores corretos.

Para minimizar esse problema aplicamos um método iterativo para ajustar o modelo e reestimar as câmeras. Nesse ponto ajustamos apenas a escala do modelo de face para que se ajuste melhor às imagens. Para fazer esse ajuste de escala no modelo procedemos da seguinte maneira:

Seja $\mathbf{X}_i$ o i -ésimo ponto de controle no modelo 3D, P^j a câmera correspondente à j -ésima imagem e $\mathbf{x}_i^j$ o i -ésimo ponto de controle marcado na j -ésima imagem. Queremos determinar os fatores de escala α_x , α_y , α_z que minimizem a distância entre a projeção dos pontos de controle do modelo escalado e os pontos de controle marcados na imagem. Em outras palavras, queremos minimizar a função f dada por:

$$f(\alpha_x, \alpha_y, \alpha_z) = \sum_i \sum_j \|P^j(S\mathbf{X}_i) - \mathbf{x}_i^j\|, \quad S = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & \alpha_z \end{bmatrix} \quad (4.2)$$

Existem diversos métodos para determinar numericamente o mínimo de funções não lineares multi variadas dada uma estimativa inicial, utilizamos para tanto o método de Powell como apresentado em [PFTV92] e como estimativa inicial o modelo sem mudança de escala, isto é, $\alpha_x = \alpha_y = \alpha_z = 1$.

Podemos então iterativamente estimar as câmeras e ajustar a escala do modelo até obter

Figura 4.4: Conjunto estendido de pontos de controle mostrados sobre o modelo de face.

convergência, isto é, até que não exista mais melhora significativa na projeção dos pontos do modelo sobre os pontos marcados nas imagens.

Supondo que neste ponto já temos uma boa estimativa das câmeras fazemos o ajuste fino do modelo como descrito na seção 2.4.2 mantendo fixos os parâmetros das câmeras e utilizando opcionalmente um conjunto estendido de pontos de controle para que o ajuste seja mais detalhado. A figura 4.4 mostra uma imagem do modelo com o conjunto estendido de pontos de controle em destaque.

4.5 Mapeamento de Textura

Tendo estimado as câmeras e a geometria da face combinamos as imagens de entrada num mapa de textura usando uma adaptação do método apresentado na seção 2.4.3. A primeira modificação que percebemos ser necessária foi alterar o mapeamento de cilíndrico para esférico pois o mapeamento cilíndrico não cobre certas partes do modelo de face deixando buracos na textura final, mesmo que todas as partes da face estejam visíveis nas imagens de entrada,

como mostra a figura 4.5.

Figura 4.5: A textura gerada usando projeção cilíndrica não cobre a parte inferior do queixo.

Além disso, como usamos um conjunto bastante reduzido de pontos de controle (visando a automação do processo), o modelo não se ajusta perfeitamente às imagens de entrada (principalmente nas bordas) e conseqüentemente teremos contribuições de pixels que não pertencem à face na textura final. A figura 4.6 mostra uma imagem de entrada com o modelo sobreposto e o resultado da textura correspondente à essa imagem, podemos notar que pixels que não pertencem à face aparecem como parte da textura.

Para corrigir as texturas aplicamos um filtro baseado na cor da pele. No trabalho descrito brevemente na seção 4.6 e apresentado com detalhes em [PdABM04] mostramos que podemos caracterizar de forma confiável a cor de pele usando apenas a componente T da representação de cor TSL (*tint, saturation, luminance* [TA00]). Sendo assim, supondo que a textura gerada é formada principalmente por pixels que correspondem a cor de pele, simplesmente calculamos o T médio da imagem e eliminamos os pixels que estejam muito distantes da média. O resultado desse processamento pode ser vista na figura 4.6.

Figura 4.6: Quando o modelo não se ajusta perfeitamente à face da imagem de entrada (a) a textura apresenta pixels que não correspondem à pixels da face (b). A textura é corrigida usando um processamento baseado em cor de pele (c).

4.6 Processamento Baseado em Cor de Pele

A cor de pele tem sido a característica mais utilizada para rastreamento devido a sua simplicidade. Esta técnica consiste em obter a distribuição de cor de pele em um espaço cromático determinado a partir de um conjunto de treinamento formado por amostras de cor de pele.

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

Apesar de uma dupla (r, g) possuir menos informação do que uma equivalente tripla (R, G, B) , a primeira é normalmente útil nas aplicações em que queremos justamente considerar apenas a informação de crominância e ignorar a luminância.

Terrillon *et al.* [TA00] propôs um novo espaço cromático chamado TSL, que consiste em uma variação do HSV (*Hue-Saturation-Value*). Neste espaço cada cor é representada por uma tripla (T, S, L) onde cada componente tem o seguinte significado: Matiz (T - Tint): informação da cor; Saturação (S): diluição da cor original com a luz branca; e Luminância (L): quantidade de energia de uma fonte de luz percebida por um observador.

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

$$S = [9/5(r'^2 + g'^2)]^{1/2} \quad (4.3)$$

$$T = \begin{cases} \arctan(r'/g')/2\pi + 1/4, & \text{se } g' > 0 \\ \arctan(r'/g')/2\pi + 3/4, & \text{se } g' < 0 \\ 0 & \text{se } g' = 0 \end{cases} \quad (4.4)$$

$$L = 0.299R + 0.587G + 0.114B \quad (4.5)$$

onde $r' = (r - 1/3)$, $g' = (g - 1/3)$, $r = R/(R + G + B)$ e $g = G/(R + G + B)$.

A distribuição de cor de pele para este modelo de cor foi obtida utilizando uma normalização do TSL pelas componentes T e S, o chamado TS-normalizado.

No artigo [PdABM04] foi feita uma análise da distribuição de cor de pele neste espaço e verificou-se que a componente T possuía uma variância significativamente menor que a componente S, o que resultou na definição do espaço T. Este espaço foi capaz de gerar uma boa segmentação de cor de pele se comparado aos outros espaços citados anteriormente, embora apresente grande quantidade de falsos positivos em regiões da imagem de tonalidade vermelha.

4.6.1 Fase de Treinamento

A primeira etapa deste algoritmo consiste na fase de treinamento. Esta fase é responsável por descrever a distribuição das cores no espaço cromático determinado.

A partir de imagens coloridas de face, extraímos manualmente regiões de cor de pele para formar o banco de imagens utilizadas no treinamento. A figura 4.7 exemplifica a obtenção de uma imagem do conjunto de treinamento.

Supondo que a distribuição de cor de pele pode ser representada por um modelo de distribuição gaussiana, tal distribuição pode ser determinada pela média $\mathbf{m}$ e pela matriz de covariância C das amostras $\mathbf{x}$ de cores de pele.

Figura 4.7: Extração de regiões de cor de pele para conjunto de treinamento.

4.6.2 Fase de Classificação

A distribuição da cor de pele obtida na fase de treinamento é utilizada para atribuir uma probabilidade para cada pixel da imagem conforme a sua cor. Esta probabilidade é calculada através da distância de Mahalanobis, utilizando, no caso geral, a média e a covariância calculadas na fase de treinamento.

A probabilidade é dada por:

$$P(x) = \exp[-0.5(x - m)^T C^{-1}(x - m)] \quad (4.6)$$

No nosso caso utilizamos apenas a componente T do espaço TSL, a probabilidade pode ser obtida de forma mais simplificada através da média m e pela variância σ^2 :

$$P(x) = \exp\left[-\frac{(x - m)^2}{2\sigma^2}\right] \quad (4.7)$$

Este cálculo resulta em uma nova imagem em níveis de cinza indicando a probabilidade do pixel correspondente ser cor de pele, como vemos na figura 4.8. Nesta, as regiões mais escuras

correspondem à maior probabilidade de cor de pele.

Figura 4.8: Resultado do cálculo da probabilidade de cor de pele para cada pixel.

O próximo passo do algoritmo consiste em um processo de limiarização. O problema neste passo é definir um limiar aplicável a diferentes imagens. A variação de cor de pele dificulta a escolha de um limiar fixo para limiarizar a imagem. A solução é aplicar um limiar adaptativo.

Analizando o histograma da imagem de probabilidade, temos as regiões de cor de pele representadas pelos maiores valores de probabilidade. Mas geralmente temos várias outras regiões da imagem que apresentam valores intermediários de probabilidade. O objetivo é encontrar um limiar ótimo que separe essas duas regiões.

Primeiramente, precisamos definir qual a faixa aceitável a conter o limiar. Após alguns testes verificamos que um limiar no intervalo entre 50% e 90% seria aceitável para diferentes cores de pele.

Um limiar ótimo é aquele que inclui o maior número de pontos da face e exclui pontos que não pertencem a face. Supondo que a face encontra-se na região superior do histograma e que outros objetos encontram-se em regiões inferiores do histograma, deve haver uma separação caracterizada por conter o menor número de pontos nesta região do histograma.

Com base nesta observação, o processo de limiarização adaptativa consiste em particionar o histograma da imagem de probabilidade em pequenos intervalos, e encontrar o intervalo de menor área dentre aqueles que são candidatos a conter um limiar adequado. Escolhemos então

o limiar dentro deste intervalo.

Figura 4.9: Exemplo de histograma particionado em intervalos de 10/100.

A figura 4.9 mostra um exemplo de um histograma particionado em intervalos de 10%.

Para este histograma, considerando somente o intervalo entre 50% e 90%, escolhemos o limiar dentro do intervalo de menor área, no caso, entre 51% e 60%. A figura 4.10 mostra o resultado da aplicação deste limiar para a figura 4.8.

Figura 4.10: Resultado da limiarização aplicada ao exemplo dado na figura 4.8.

Figura 4.11: Padrão utilizado para calibrar a câmera.

4.7 Calibragem da Câmera

Apesar de ser possível determinar os parâmetros internos e externos de uma determinada câmera dado um conjunto correspondências de pontos, como mostra a seção 4.3.1, usar câmeras calibradas diminui o número de parâmetros que precisam ser estimados tornando o processo mais tolerante a erros nos dados e o resultado final mais satisfatório.

Existem várias formas conhecidas para estimar os parâmetros internos de uma dada câmera, escolhemos utilizar o método de calibragem apresentado por Zhang [Zha98] pois trata-se de um método bastante flexível que pode ser facilmente implementado e que apresenta bons resultados.

Para utilizar esse método são necessárias pelo menos três imagens de um objeto plano (chamado de modelo) com pontos cujas coordenadas sejam conhecidas com bastante precisão. Utilizamos para isso o padrão quadriculado mostrado na figura 4.11.

4.7.1 Homografia Entre o Plano do Modelo e Sua Imagem

Considere uma câmera P decomposta na forma $P = K[R|\mathbf{c}]$, com K a matriz de calibragem, R a matriz de rotação e $\mathbf{c}$ o centro da câmera.

Sem perda de generalidade, vamos assumir que o modelo está no plano $Z = 0$ do sistema de coordenadas da cena. Vamos denotar a i -ésima coluna da matriz de rotação por $\mathbf{r}_i$. Podemos escrever a equação de projeção do ponto 3D $\mathbf{X} = (X, Y, Z, 1)$ no ponto 2D $\mathbf{x} = (x, y, 1)$ como:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3 \ \mathbf{c}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{c}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (4.8)$$

Dessa forma, um ponto $\mathbf{X}$ no modelo e sua imagem $\mathbf{x}$ estão relacionados por uma homografia H :

$$\mathbf{x} = H\hat{\mathbf{X}}, \quad \text{com} \quad H = K[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{c}] \quad \text{e} \quad \hat{\mathbf{X}} = (X, Y, 1) \quad (4.9)$$

com H definida a menos de escala.

4.7.2 Estimando a Homografia H

Dado um conjunto de correspondências em coordenadas homogêneas $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, podemos determinar a homografia H tal que $\mathbf{x}'_i = H\mathbf{x}_i$ da seguinte maneira:

1. Para eliminar o fator de escala homogênea, a equação $\mathbf{x}'_i = H\mathbf{x}_i$ pode ser escrita na forma de produto vetorial como $\mathbf{x}'_i \times H\mathbf{x}_i = 0$.
2. Denotando a j -ésima linha da matriz H por $\mathbf{h}^{jT}$, temos:

$$H\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1T} \mathbf{x}_i \\ \mathbf{h}^{2T} \mathbf{x}_i \\ \mathbf{h}^{3T} \mathbf{x}_i \end{pmatrix}$$

3. Escrevendo $\mathbf{x}'_i = (x'_i, y'_i, w'_i)^T$, o produto vetorial pode ser dado explicitamente por:

$$\mathbf{x}'_i \times H\mathbf{x}_i = \begin{pmatrix} y'_i \mathbf{h}^{3T} \mathbf{x}_i - w'_i \mathbf{h}^{2T} \mathbf{x}_i \\ w'_i \mathbf{h}^{1T} \mathbf{x}_i - x'_i \mathbf{h}^{3T} \mathbf{x}_i \\ x'_i \mathbf{h}^{2T} \mathbf{x}_i - y'_i \mathbf{h}^{1T} \mathbf{x}_i \end{pmatrix}$$

4. Isso nos dá um conjunto de 3 equações lineares nas entradas de H que podem ser escritas na forma matricial:

$$\begin{bmatrix} \mathbf{0}^T & -w'_i \mathbf{x}_i^T & y'_i \mathbf{x}_i^T \\ w'_i \mathbf{x}_i^T & \mathbf{0}^T & -x'_i \mathbf{x}_i^T \\ -y'_i \mathbf{x}_i^T & x'_i \mathbf{x}_i^T & \mathbf{0}^T \end{bmatrix} = \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix}$$

5. Das 3 equações obtidas apenas 2 são linearmente independentes, sendo assim, combinando as equações provenientes de pelo menos 4 correspondências podemos determinar H unicamente (a menos de escala), resolvendo um sistema da forma $A\mathbf{h} = \mathbf{0}$.

4.7.3 Restrições Sobre os Parâmetros Internos

Dada uma imagem do modelo, podemos estimar uma homografia $H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3]$ e da equação 4.9 segue que:

$$[\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3] = K[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}]$$

sabendo que $\mathbf{r}_1$ e $\mathbf{r}_2$ são ortonormais, temos que:

$$\begin{aligned} \mathbf{h}_1^T K^{-T} K^{-1} \mathbf{h}_2 &= 0 \\ \mathbf{h}_1^T K^{-T} K^{-1} \mathbf{h}_1 &= \mathbf{h}_2^T K^{-T} K^{-1} \mathbf{h}_2 \end{aligned} \tag{4.10}$$

Essas são as duas restrições básicas sobre os parâmetros internos da câmera dada uma homografia. Podemos usar essas equações para determinar a matriz K .

Seja

$$B = K^{-T}K^{-1} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{12} & b_{22} & b_{23} \\ b_{13} & b_{23} & b_{33} \end{bmatrix}$$

B é uma matriz simétrica, definida pelo vetor $\mathbf{b} = [b_{11}, b_{12}, b_{22}, b_{13}, b_{23}, b_{33}]^T$.

Denotando a i -ésima coluna de H por $\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T$, temos que:

$$\mathbf{h}_i^T B \mathbf{h}_j = \mathbf{v}_{ij} \mathbf{b}, \quad \text{onde}$$

$$\mathbf{v}_{ij} = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{bmatrix}$$

Sendo assim as restrições dadas pelas equações 4.10 a partir de uma dada homografia podem ser reescritas como equações lineares em $\mathbf{b}$:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} \quad (4.11)$$

Combinando as restrições provenientes de n imagens do modelo obtemos um sistema equações lineares da forma $V\mathbf{b} = \mathbf{0}$, com V uma matriz de dimensão $2n \times 6$. Com pelo

menos 3 imagens do modelo podemos determinar $\mathbf{b}$ de forma única. Uma vez conhecido $\mathbf{b}$ (e conseqüentemente B) podemos determinar K a partir da equação $B = K^{-T}K^{-1}$ usando a decomposição de Cholesky [PFTV92].

4.8 Implementação

O sistema foi implementado em plataforma linux usando a linguagem C++, para as rotinas numéricas utilizamos basicamente os algoritmos descritos em *Numerical Recipes* [PFTV92], para a interface com o usuário usamos o pacote GTK+ [dWb] e para as rotinas de visualização 3D do modelo de face utilizamos as bibliotecas do *OpenGL* [dWd].

A primeira parte do sistema é a interface para localização manual de pontos de controle, como ela o usuário pode carregar uma coleção de fotos da face de interesse e localizar manualmente um conjunto de pontos de controle que serão utilizados posteriormente para determinar as câmeras e deformar o modelo genérico de face. A figura 4.12 mostra a interface para localização de pontos de controle.

Através da interface marcamos cada uma das imagens de entrada e gravamos as correspondência num arquivo de texto que descreve a posição de cada ponto de controle em cada imagem. Esse arquivo de texto é usado como entrada para o algoritmo de reconstrução propriamente dito.

A figura 4.13 mostra um diagrama que ilustra a seqüência de passos executados. A entrada para o sistema é uma coleção de imagens de uma face e a saída é o modelo 3D ajustado à face e um mapa de textura.

Figura 4.12: Interface para localização de pontos de controle. Na parte esquerda o usuário carrega a coleção de imagens de entrada. O modelo genérico é apresentado do lado direito, a correspondência de pontos é feita clicando-se primeiramente no vértice do modelo e em seguida na posição correspondente na imagem de entrada.

Figura 4.13: Diagrama do funcionamento do sistema.

4.9 Métodos Numéricos

4.9.1 Decomposição de Valores Singulares (SVD)

A decomposição de valores singulares é uma técnica muito poderosa que usamos ao longo de todo o sistema para lidar com conjunto de equações lineares ou matrizes que sejam singulares ou numericamente muito próximas de serem singulares, e também no caso de sistemas lineares sobre-determinados.

A SVD é baseada no seguinte teorema da álgebra linear: Qualquer matriz $A \in \mathbb{R}^{M \times N}$ pode ser escrita como produto de uma matriz ortogonal $U \in \mathbb{R}^{M \times N}$, uma matriz diagonal $W \in \mathbb{R}^{N \times N}$ cujos elementos são todos maiores ou iguais a zero (os valores singulares) e a transposta de uma matriz ortogonal $V \in \mathbb{R}^{N \times N}$. Isto é:

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} = \begin{bmatrix} & & \\ & U & \\ & & \end{bmatrix} \cdot \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{bmatrix} \cdot \begin{bmatrix} & & \\ & V^T & \\ & & \end{bmatrix}$$

Na nossa aplicação utilizamos a SVD para resolver sistemas de equações lineares homogêneas (sistemas da forma $A\mathbf{x} = \mathbf{0}$). Pode-se provar que uma base ortonormal para o espaço de soluções do sistema é dada pelos vetores- coluna de V correspondentes ao valores singulares (elementos da matriz diagonal W) iguais a zero.

Detalhes sobre a implementação e outras utilidades da SVD podem ser encontradas em [PFTV92].

4.9.2 Método de Powell Para Minimização Não Linear

O método de Powell é um método iterativo que utilizamos para encontrar o mínimo (local) de funções $f : \mathbb{R}^n \mapsto \mathbb{R}$. De forma resumida o método funciona da seguinte maneira:

- Inicialize um conjunto de direções $\mathbf{u}_i$ com os vetores de uma base de $\mathbb{R}^n$, isto é $\mathbf{u}_i = \mathbf{e}_i$, $i = 1, \dots, n$.
- Chame a posição inicial de $\mathbf{P}_0$.
- Execute os passos a seguir até alcançar convergência:
 1. Para $i = 1 \dots n$, encontre o mínimo local a partir de $\mathbf{P}_{i-1}$ na direção $\mathbf{u}_i$ e chame o ponto encontrado de $\mathbf{P}_i$. Isto pode ser feito usando-se o conhecido método de minimização unidimensional de *Newton*, por exemplo.
 2. Para $i = 1 \dots n - 1$, $\mathbf{u}_i \leftarrow \mathbf{u}_{i+1}$.
 3. $\mathbf{u}_n \leftarrow \mathbf{P}_n - \mathbf{P}_0$.
 4. Encontre o mínimo a partir de $\mathbf{P}_n$ na direção de $\mathbf{u}_n$ e chame o ponto encontrado de $\mathbf{P}_0$.

Maiores informações sobre o método de Powell e outros métodos de minimização podem ser encontrados em [PFTV92].

Capítulo 5

Resultados

Neste capítulo apresentamos detalhadamente alguns exemplos de reconstrução de faces usando o sistema implementado como descrito no capítulo 4. Em todos os exemplos apresentados as imagens de entrada possuem 720 pixels de largura por 528 de altura.

Os primeiros testes foram realizados sem a calibragem da câmera, utilizando-se as técnicas apresentadas na seção 4.3.1, no entanto os resultados não foram satisfatórios. A rotina mostrou-se muito instável para tratar o erro na medida dos pontos de controle e lidar com o grande número de variáveis a serem minimizadas simultaneamente.

Sendo assim, passamos a utilizar uma câmera calibrada. Para realizar a calibragem utilizamos um conjunto de 6 imagens do padrão quadriculado citado na seção 4.7, algumas dessas imagens são apresentadas na figura 5.1. Aplicando o método descrito na seção 4.7 seguido de uma rotina de minimização não linear sobre o erro de reprojeção obtivemos os seguintes parâmetros internos:

Figura 5.1: Exemplo de imagens usadas para calibrar a câmera.

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} = \begin{bmatrix} 756.09 & 9.68504 & 9.80734 \\ & 763.965 & -25.1164 \\ & & 1 \end{bmatrix}$$

Com isso reduzimos consideravelmente o número de variáveis a serem minimizadas simultaneamente, controlando a propagação do erro nas medidas dos pontos de controle. Os resultados obtidos são bastante satisfatórios como veremos nas próximas seções.

5.1 Um Exemplo Detalhado

Os resultados referentes ao nosso primeiro exemplo são apresentados na figura 5.2. Neste primeiro exemplo mostramos a reconstrução feita a partir de uma coleção de 5 imagens, sendo uma frontal, duas de perfil e duas de semi perfil. As imagens de entrada podem ser vistas na primeira linha da figura.

Com o auxílio da interface apresentada na seção 4.8 localizamos sobre cada uma das imagens de entrada o conjunto de pontos de controle mostrados na seção 4.2. Como nem todos os 12 pontos estão visíveis ou podem ser localizados com precisão em todas as imagens, o número

Figura 5.2: Exemplo 1

de pontos marcados em cada imagem varia, nesse caso, de 7 a 11. As imagens de entrada com os pontos de controle localizados podem ser na segunda linha da figura.

Com os pontos de controle devidamente localizados determinamos a posição e orientação das câmeras correspondentes a cada uma das imagens de entrada usando o algoritmo POSIT (seção 3.3) e deformamos o modelo genérico para que se ajuste melhor a face presente nas imagens. Os vértices do modelo ajustado sobrepostos às imagens de entrada são apresentados na terceira linha da figura.

Usando a posição e orientação das câmeras e o modelo ajustado extraímos as texturas referentes a cada uma das imagens usando a projeção esférica, como descrito na seção 4.5. As imagens correspondentes após o processamento baseado na cor de pele e a normalização da luminância são apresentadas na quarta linha da figura.

Combinando as imagens parciais geramos uma textura que cobre todas as partes do rosto que estejam visíveis nas imagens de entrada. Partes do rosto que não estão visíveis deixam buracos na textura combinada, tais buracos são preenchidos usando as cores dos pixels vizinhos. A quinta linha da figura mostra a combinação das texturas parciais e o resultado do processamento para fechar buracos.

Temos então um modelo geométrico de face ajustado às imagens de entrada e também um mapa de textura construído a partir de tais imagens. Podemos agora renderizar o modelo de face de qualquer ponto de vista e usar o sistema de simulação de músculos para gerar diferentes expressões faciais como podemos ver na última linha da figura.

Este primeiro exemplo mostra resultados muito bons, notamos que o modelo se ajustou muito bem aos detalhes como contorno dos olhos, boca e nariz. Além disso o rosto presente nas imagens de entrada possui uma forma alongada, característica que foi reproduzida corretamente pelo modelo resultante.

5.2 Outros Exemplos

As figuras 5.3, 5.4, 5.5, 5.6, 5.7 e 5.8 mostram exemplos de reconstrução usando coleções de 12, 10, 5, 5, 3, e 2 imagens de entrada respectivamente.

A primeira coluna da tabela mostra algumas imagens de entrada, a coluna do meio mostra imagens do modelo obtido sem expressões faciais e a última coluna mostra imagens do modelo com diferentes expressões faciais geradas sinteticamente.

Em todos os exemplos, que contempla pessoas de ambos os sexos e diferentes etnias, podemos perceber que o modelo se ajusta muito bem às características internas da face como os olhos a boca e o nariz. Isso já era esperado uma vez que utilizamos um conjunto de pontos de controle concentrado nessas regiões.

Nas figuras 5.5, 5.6 e 5.7 vemos que o modelo gerado mostra-se mais alongado do que as respectivas faces originais. O que acontece é que não utilizamos pontos de controle no contorno do rosto, pois notamos que tais pontos eram difíceis de serem localizados mesmo manualmente.

Sendo assim, os modelos gerados possuem um contorno muito parecido com o contorno do modelo genérico que é mais alongado e conseqüentemente se ajusta melhor aos exemplos 5.2, 5.3, 5.4 e 5.8.

Outro fator que devemos levar em consideração é com relação às expressões faciais. O processo de reconstrução foi realizado apenas com a expressão neutra e utilizamos o sistema de simulação de músculos para animar as faces reconstruídas. Como o sistema de simulação é o mesmo para todas os modelos gerados é possível perceber que as expressões obtidas com os diversos modelos são praticamente iguais.

Uma possível solução para esse problema seria aplicar o processo de reconstrução para cada expressão básica e realizar a animação através da interpolação dos modelos obtidos, com a desvantagem de tornar o processo muito mais trabalhoso.

Figura 5.3: Exemplo 2

Figura 5.4: Exemplo 3

Figura 5.5: Exemplo 4

Entrada	Sem Expressão	Com Expressão

Figura 5.6: Exemplo 5

Figura 5.7: Exemplo 6

Entrada	Sem Expressão	Com Expressão

Figura 5.8: Exemplo 7

Capítulo 6

Conclusão

Durante este projeto estudamos diversas técnicas utilizadas para modelagem e animação de faces. Dentre tais técnicas estudamos de forma mais aprofundada as técnicas de modelagem usando reconstrução a partir de múltiplas imagens e animação através da simulação da ação de músculos faciais.

Verificando os principais métodos encontrados na bibliografia percebemos que, apesar de existir muito trabalho nessa área, ainda não está disponível um sistema completamente automático que seja capaz estimar um modelo realista de face a partir de uma coleção de imagens. São vários os fatores que tornam esse um problema bastante complexo: A face possui uma geometria extremamente complexa, apresentando variações de cor e textura ao longo de toda sua extensão. Além disso, as expressões faciais são resultado de uma complexa interação entre os músculos, a pele e o esqueleto e reproduzir tal interação é uma tarefa por si só bastante complicada.

Em nosso trabalho propusemos e implementamos um sistemas para reconstrução de faces que utiliza um conjunto de correspondência de pontos para ajustar um modelo genérico de face à um conjunto de imagens. De forma resumida o sistema funciona da seguinte maneira:

- Um conjunto de pontos de controle é localizado sobre cada uma das imagens da face a ser reconstruída. Não existem restrições sobre a posição e orientação relativa da câmera com relação à face, embora seja necessário o uso de uma câmera calibrada.
- Para cada imagem, usando as correpondências dos pontos 2D marcados na imagem e os vértices 3D de um modelo genérico de face, estimamos a posição e a orientação relativa da câmera com relação a face.
- Conhecendo as posições e orientações das câmeras, desenvolvemos um algoritmo iterativo que ajusta o modelo genérico minimizando a distância entre a projeção de seus vértices e os pontos localizados nas imagens. Em geral, o algoritmo converge rapidamente.
- Combinamos as imagens de entrada em um mapa de textura em coordenadas esféricas que é projetado sobre o modelo ajustado à face presente nas imagens. Problemas encontrados no mapa de texturas devido a falhas no ajuste do modelo são corrigidos com um processamento estatístico baseado em cor de pele.
- Usando o sistema de simulação de músculos do modelo de face, podemos animar o modelo obtido gerando diferentes expressões faciais.

De forma resumida podemos dizer que o sistema que propusemos e implementamos possui os seguintes pontos positivos:

- Ajusta um modelo genérico de face à um número arbitrário de imagens.
- Quanto maior o número de imagens de entrada, melhor a qualidade visual do resultado.
- As imagens de entrada podem ser obtidas de qualquer ponto de vista, não sendo necessária nenhuma informação a priori sobre a posição e orientação da câmera.

- O sistema é quase automático, sendo que a única interação com o usuário é para a localização dos pontos de controle sobre as imagens de entrada, tarefa que em geral pode ser feita em poucos minutos.
- O resultado é um modelo 3D de face realista que pode ser facilmente animado usando o sistema de simulação muscular.
- O sistema pode ser utilizado para reconstrução de outras estruturas complexas que não a face, bastando para tanto trocar o modelo genérico e redefinir o conjunto de pontos de controle.

Analisando os resultados obtidos verificamos que existem alguns pontos que ainda precisam ser trabalhados para melhorar a qualidade do modelo resultante:

- A localização dos pontos de controle é a única parte do sistema que requer o auxílio do usuário, a localização automática desses pontos tornaria o sistema completamente automático. Existem no LaTIn trabalhos relacionados à localização e rastreamento de características faciais sendo desenvolvidos e que deverão ser integrados ao nosso sistema.
- O modelo resultante não possui olhos, dentes, língua, orelha nem cabelo. A utilização de um modelo mais completo e o ajuste das estruturas adicionais poderiam aumentar o realismo. O modelo de face possuía originalmente olhos e dentes mas o ajuste de tais estruturas não foi levado em conta nesse trabalho, e fica como uma sugestão para trabalhos futuros.
- Devido ao fato de utilizamos um conjunto bastante limitado de pontos de controle e concentrados no interior da face (olhos, boca, nariz), o resultado do ajuste é muito bom nessas estruturas mas não apresenta a mesma qualidade no contorno do rosto, por exemplo. Para contornar essa situação utilizamos um conjunto estendido de pontos de

controle, mas notamos que é difícil localizar com precisão os pontos adicionais. Outra possibilidade seria utilizar outras informação além de pontos, como o contorno do rosto como um todo, mas essa técnica não foi implementada em nosso trabalho.

- Outro ponto que dificulta a tentativa de realizar um ajuste mais fino é o fato de o modelo utilizado possuir relativamente poucos vértices, não podendo ser deformado a ponto de capturar os detalhes finos de uma determinada face. A geração de malhas mais finas a partir do nosso modelo é relativamente simples, e um método mais refinado de ajuste do modelo às imagens, que leve em conta mais regiões além dos pontos de controle, poderia ser implementado.
- Até o momento não realizamos nenhum experimento para medir de forma quantitativa o erro de reconstrução. Uma possibilidade seria realizar a reconstrução de modelos de faces virtuais, e comparar o modelo reconstruído com o modelo virtual.

Tanto o sistema de reconstrução como a interface para animação de faces serão disponibilizados como software de código aberto para a comunidade para que mais avanços possam ser realizados. As imagens utilizadas também serão disponibilizadas para que testes comparativos possam ser realizados.

Referências Bibliográficas

- [AS96] S. Avidan and A. Shashua. Unifying two-view and three-view geometry. Technical Report 96-21, Institute of Computer Science, The Hebrew University, 1996.
- [BBEHC01] F. Blais, J.-A. Beraldin, S. F. El-Hakim, and L. Cournoyer. Real-time geometrical tracking and pose estimation using laser triangulation and photogrammetry. *3dim*, 00:205, 2001. 8
- [BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999. 2, 12
- [CHT03] Ya Chang, Changbo Hu, and Matthew Turk. Manifold of facial expression. In *AMFG '03: Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, page 28. IEEE Computer Society, 2003.
- [D'A02] Nicola D'Apuzzo. Measurement and modeling of human faces from multi images. *International Archives of Photogrammetry and Remote Sensing*, 34:241–246, 2002. 2, 10

- [DC01] J. Davis and X. Chen. A laser range scanner designed for minimum calibration complexity. *3dim*, 00:91, 2001. 8
- [DD95] Daniel F. Dementhon and Larry S. Davis. Model-based object pose in 25 lines of code. *Int. J. Comput. Vision*, 15(1-2):123–141, 1995.
- [DDDS04] Philip David, Daniel Dementhon, Ramani Duraiswami, and Hanan Samet. Softposit: Simultaneous pose and correspondence determination. *Int. J. Comput. Vision*, 59(3):259–284, 2004. 50, 52
- [DTM96] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: a hybrid geometry and image-based approach. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20. ACM Press, 1996. 25
- [dWa] Site da Web. flarg.com - shrek rendering statistics. <http://www.flarg.com/Graphics/ShrekRenderfarm.html>. 4
- [dWb] Site da Web. Gtk+ the gimp tool kit. <http://www.gtk.org>. 66
- [dWc] Site da Web. imdb.com - trivia for shrek (2001). <http://www.imdb.com/title/tt0126029/trivia>. 4
- [dWd] Site da Web. Opengl - the industry's foundation for high performance graphics. <http://www.opengl.org>. 66
- [DYB98] Paul Debevec, Yizhou Yu, and George Boshokov. Efficient view-dependent image-based rendering with projective texture-mapping. Technical report, University of California at Berkeley, 1998. 25

- [EGP02] Tony Ezzat, Gadi Geiger, and Tomaso Poggio. Trainable videorealistic speech animation. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 388–398. ACM Press, 2002.
- [EP97] Irfan A. Essa and Alex P. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern, Analysis and Machine Intelligence*, 19(7):757–763, 1997.
- [ER97] P. Ekman and E.L. Rosenberg. *WHAT THE FACE REVEALS: BASIC AND APPLIED STUDIES OF SPONTANEOUS EXPRESSION USING THE FACIAL ACTION CODING SYSTEM (FACS)*. Oxford University Press, ISBN: 0195104471, second edition, 1997. 46
- [Fau92] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 563–578, London, UK, 1992. Springer-Verlag. 9
- [FCZ98] Andrew W. Fitzgibbon, Geoff Cross, and Andrew Zisserman. Automatic 3d model construction for turn-table sequences. *Lecture Notes in Computer Science*, 1506:155, jan 1998. 25
- [FM99] P. Fua and C. Miccio. Animated heads from ordinary images: A least squares approach. *Computer Vision and Image Understanding*, 1999. 2, 10
- [GHMQ97] P. Gros, R. Hartley, R. Mohr, and L. Quan. How useful is projective geometry? *Comput. Vis. Image Underst.*, 65(3):442–446, 1997.
- [Had05] L. Hadjer. An iterative improvement of the tomasi-kanade factorization. *Third Hugarian Conference on Computer Graphics and Geometry*, 2005. 8

- [HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 25, 50
- [ILS⁺03] D. Ivanov, V. Lempitsky, A. Shokurov, A. Khropov, and Ye. Kuzmin. Creating personalized head models from image series. In *Proceedings of the GraphiCon*, pages 93–99, 2003. 11
- [KWT88] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. 10
- [LKMT97] W. Lee, P. Kalra, and N. Magnenat-Thalmann. Model based face reconstruction for animation. In *Proc. MMM'97 (World Scientific Press), Singapore*, pages 323–338, 1997. 10
- [LSZ01] Zicheng Liu, Ying Shan, and Zhengyou Zhang. Expressive expression mapping with ratio images. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 271–276. ACM Press, 2001.
- [LTW95] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. Realistic modeling for facial animation. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 55–62. ACM Press, 1995.
- [LV96] Q.-T. Luong and T. Vieville. Canonical representations for the geometries of multiple projective views. *Comput. Vis. Image Underst.*, 64(2):193–229, 1996.
- [Nie93] Gregory M. Nielson. Scattered data modeling. *IEEE Computer Graphics and Applications*, 13(1):60–70, 1993. 15, 19

- [Par72] Frederick I. Parke. Computer generated animation of faces. In *ACM'72: Proceedings of the ACM annual conference*, pages 451–457. ACM Press, 1972. 1
- [PdABM04] Bruno Pera, Régis de Abreu Barbosa, and Carlos Hitoshi Morimoto. Análise comparativa de diferentes espaços cromáticos para detecção de cor de pele. In *SVR 2004: Proceedings of the VII Symposium on Virtual Reality*, 2004. 5, 55, 58
- [PFTV92] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge (UK) and New York, 2nd edition, 1992. 53, 66, 69, 70
- [PHL⁺98] Frédéric Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, and David H. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 75–84. ACM Press, 1998. 2, 5, 15, 19, 45
- [PKG99] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7, 1999. 25
- [PKVG00] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Automated reconstruction of 3d scenes from sequences of images. *Isprs Journal Of Photogrammetry And Remote Sensing*, 55(4):251–267, 2000. 25
- [PSS99] Frédéric Pighin, Richard Szeliski, and David H. Salesin. Resynthesizing facial animation through 3d model-based tracking. In *International Conference on Computer Vision-Volume 1*, pages 143–150. IEEE Computer Society, 1999. 15, 19, 45

- [QL99] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):774–780, 1999.
- [RTZ03] Greg Reid, Jianliang Tang, and Lihong Zhi. A complete symbolic-numeric linear method for camera pose determination. In *ISSAC '03: Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 215–223, New York, NY, USA, 2003. ACM Press.
- [TA00] J. Terrillon and S. Akamatsu. Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images, 2000. 55, 57
- [TK92] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992. 7
- [TP91] M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition*, pages 586–591. IEEE Computer Society, 1991.
- [TZ95] C. Tomasi and J. Zhang. Is structure-from-motion worth pursuing? In *In Proceedings of the Seventh International Symposium on Robotics Research*, pages 391–400, 1995.
- [Wat87] Keith Waters. A muscle model for animating three-dimensional facial expression. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 17–24. ACM Press, 1987. 2, 45, 46
- [Zha98] Zheng Zhang. A new flexible technique for camera calibration. Technical report, Microsoft Research, Microsoft Corporation, 1998. 62

- [ZPS02] Yu Zhang, Edmond C. Prakash, and Eric Sung. Hierarchical face modeling and fast 3d facial expression synthesis. In *SIBGRAPI '02: Proceedings of the 15th Brazilian Symposium on Computer Graphics and Image Processing*, pages 357–364. IEEE Computer Society, 2002.
- [ZTCS99] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999. 9