

Métodos Fotométricos para Visão Computacional

Rafael Felipe Veiga Saracchini

Este exemplar corresponde à redação final da Tese devidamente corrigida e defendida por Rafael Felipe Veiga Saracchini e aprovada pela Banca Examinadora.

Campinas, 18 de Abril de 2012.



Jorge Stolfi
IC-Unicamp (Orientador)

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

FICHA CATALOGRÁFICA ELABORADA POR
MARIA FABIANA BEZERRA MULLER - CRB8/6162
BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E
COMPUTAÇÃO CIENTÍFICA - UNICAMP

Saracchini, Rafael Felipe Veiga, 1984-
Sa71m Métodos fotométricos para visão computacional / Rafael Felipe
Veiga Saracchini. – Campinas, SP : [s.n.], 2012.

Orientador: Jorge Stolfi.
Tese (doutorado) – Universidade Estadual de Campinas,
Instituto de Computação.

1. Processamento de imagens. 2. Visão por computador. 3.
Ciência da computação. 4. Imagem tridimensional. I. Stolfi,
Jorge, 1950-. II. Universidade Estadual de Campinas. Instituto de
Computação. III. Título.

Informações para Biblioteca Digital

Título em inglês: Photometric methods for computer vision

Palavras-chave em inglês:

Image processing

Computer vision

Computer science

Three-dimensional imaging systems

Área de concentração: Ciência da Computação

Titulação: Doutor em Ciência da Computação

Banca examinadora:

Jorge Stolfi [Orientador]

José Ricardo de Almeida Torreão

Roberto Marcondes Cesar Junior

Hélio Pedrini


Anamaria Gomide

Data de defesa: 18-04-2012

Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

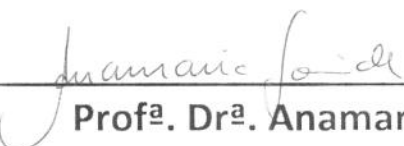
Tese Defendida e Aprovada em 18 de Abril de 2012, pela Banca
examinadora composta pelos Professores Doutores:




Prof. Dr. José Ricardo de Almeida Torreão
IC - UFF



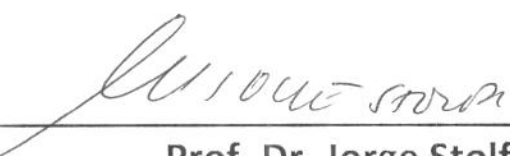
Prof. Dr. Roberto Marcondes César Junior
IME – USP



Profª. Drª. Anamaria Gomide
IC – UNICAMP



Prof. Dr. Hélio Pedrini
IC- UNICAMP



Prof. Dr. Jorge Stolfi
IC – UNICAMP

Métodos Fotométricos para Visão Computacional

Rafael Felipe Veiga Saracchini¹

Abril de 2012

Banca Examinadora:

- Jorge Stolfi
IC-Unicamp (Orientador)
- José R. A. Torreão
IC-UFF
- Roberto M. Cesar Jr
IME-USP
- Hélio Pedrini
IC-Unicamp
- Anamaria Gomide
IC-Unicamp

¹Suporte financeiro de: Bolsa FAPESP (processo 07/59509-9) 2007–2009 e Universidade Estadual de Campinas 2007–2009.

Resumo

O principal objetivo desta tese de doutorado é o desenvolvimento e implementação de técnicas inovadoras para *visão estereoscópica* ou *estéreo* — a reconstrução de um modelo tridimensional de uma cena, a partir de duas ou mais imagens bidimensionais da mesma. Especificamente, estudamos e aprimoramos métodos de *estéreo fotométrico*, que utilizam imagens tomadas com mesmo ponto de vista sob várias condições de iluminação e determinam a normal à superfície da cena em cada pixel das imagens. Estas normais podem ser então integradas para gerar um mapa de alturas que fornece a terceira dimensão da cena. Consideramos principalmente métodos em que as condições de iluminação e o acabamento da superfície da cena são determinadas pela análise de imagens de objetos de referência (*gabaritos*) com forma conhecida.

As contribuições desta tese consistem em: (1) desenvolvimento de *gabaritos virtuais* para eliminar o ruído presente nas imagens de gabaritos reais; (2) um método de grade bidimensional para determinação rápida das normais da cena; (3) uma abordagem robusta baseada em inferência bayesiana para determinação das normais na presença de anomalias de tonalização; e (4) dois métodos para integração multi-escala para mapas de normais que podem ser usados em mapas com lacunas e ruído. Investigamos também uma aplicação prática dessas técnicas: a captura da geometria da face de uma pessoa em movimento.

Abstract

The main goal of this PhD thesis is the development and implementation of innovative techniques for *stereo vision* or *stereo* — the reconstruction of a tridimensional model of a scene from two or more two dimensional images. Specifically, we studied and improved *photometric stereo* methods, which use images taken from the same viewpoint under different lighting conditions, and determine the surface normal of the scene at each pixel. We mainly studied methods that determine the lighting conditions and surface finish by the analysis of images of reference objects (*gauges* or *example objects*) with known geometry.

The contributions of this thesis consist of: (1) development of *virtual example objects* in order to eliminate noise and defects present in images of real example objects; (2) a 2D hashing grid method for fast normal computation; (3) a robust approach based in Bayesian inference for surface normal computation in the presence of shading anomalies, and (4) two methods for multi-scale normal map integration that can be used in maps with missing or noisy data. We also investigated a practical application of such methods: the capture of the 3D facial geometry of a person in movement.

Agradecimentos

Gostaria de agradecer primeiramente a Deus, que sempre esteve ao meu lado, pela grande conquista que foi poder escrever esta tese e concluir meu doutorado.

Agradeço aos meus pais Paulina e Francisco e aos meus irmãos Paulo Tiago e Paula Gabrielle, e a minha namorada Ewa Piotrowska, que estiveram ao meu lado, me apoiaram e fizeram-me acreditar nos meus sonhos, principalmente nos momentos mais difíceis, estando tão distantes fisicamente e ao mesmo tempo tão próximos. Agradeço ao meu tio Josene Marcos, que foi para mim, desde cedo, um modelo de inspiração e dedicação. A todos eles agradeço a paciência e compreensão que tiveram comigo, e principalmente o carinho que me ofereceram.

Agradeço aos meus vários amigos e colegas que conheci em Campinas e em Bristol, que durante estes anos, de uma forma ou outra, trilharam o mesmo caminho que eu, e que compartilharam comigo muitas alegrias e dificuldades.

Agradeço ao professor Melvyn Smith, que muito fez por mim, oferecendo-me uma oportunidade de conhecer o ambiente de pesquisa internacional, e que durante minha estadia na Inglaterra, fez de tudo para me sentir bem-vindo.

Finalmente, agradeço aos professores Jorge Stolfi e Helena Leitão por me orientar em todos estes anos e me oferecer a incrível oportunidade de conhecer o mundo da pesquisa acadêmica. Muita desta obra é parte tanto do esforço deles, quanto foi do meu, sem o qual esta não seria possível.

Sumário

Resumo	vii
Abstract	ix
Agradecimentos	xi
1 Introdução	1
1.1 Captura de modelos tridimensionais	1
1.2 Técnicas de estéreo computacional	2
1.3 Aplicações de estéreo fotométrico	3
1.4 Vantagens e desvantagens do estéreo fotométrico	4
1.5 Objetivos da tese	5
I Determinação das normais e albedos	7
2 Fundamentos do Estéreo Fotométrico	9
2.1 Geometria da formação das imagens	10
2.1.1 Imagem e pixels	10
2.1.2 Sistema de coordenadas da cena	11
2.1.3 A direção de vista	12
2.2 Princípios de Fotometria	12
2.2.1 Valores dos pixels	13
2.2.2 Radiância da superfície	13
2.2.3 Fluxo de iluminação	13
2.3 Função de distribuição de radiância	14
2.3.1 Albedo	15
2.3.2 Acabamento	16
2.3.3 Acabamento lambertiano	16
2.3.4 Acabamento lustroso e o modelo de Phong	17

2.4	Equação de tonalização	19
2.5	Hipóteses simplificadoras	19
2.5.1	Iluminação uniforme	19
2.5.2	Albedo isotrópico	21
2.5.3	Acabamento uniforme	21
2.5.4	Hipótese de luz unidirecional	21
2.6	Função de tonalização	22
2.6.1	Vetores de observação e tonalização	22
2.6.2	Assinaturas de observação e tonalização	23
2.7	Hipótese de invertibilidade	23
2.7.1	Número mínimo de imagens	24
2.8	Métodos algébricos para cálculo de normais	25
2.8.1	Superfície lambertiana com iluminação unidirecional	25
2.8.2	Acabamento lambertiano com luzes extensas	26
2.9	Estéreo fotométrico baseado em gabaritos	26
2.10	Estéreo fotométrico com luz policromática	28
2.10.1	Equação de tonalização policromática	28
2.10.2	Imagens coloridas	28
2.10.3	Hipóteses de uniformidade espectral	29
2.10.4	Hipótese de separação espectral	30
2.10.5	Estéreo fotométrico multi-espectral	31
3	Trabalhos Relacionados	33
3.1	Conceitos básicos	33
3.2	Métodos com função de tonalização conhecida	33
3.3	Métodos com função de tonalização desconhecida	34
3.3.1	Abordagens com iluminação conhecida	35
3.3.2	Abordagens com iluminação desconhecida	37
3.4	Outros trabalhos	39
3.5	Sumário	40
4	Modelagem dos gabaritos	43
4.1	Trabalhos relacionados	43
4.2	As imagens do gabarito	44
4.2.1	Qualidade das imagens	44
4.2.2	Critério dos quadrados mínimos	45
4.3	Geometria de projeção do gabarito	46
4.3.1	Normais amostradas pelo gabarito	46
4.3.2	Sistema de coordenadas do gabarito	47

4.3.3	Forma da projeção do gabarito	47
4.3.4	Calculando as normais do gabarito	48
4.3.5	Exclusão dos pixels da borda	49
4.4	Modelos lineares genéricos	49
4.4.1	Harmônicos esféricos	50
4.4.2	Bases radiais	51
4.5	Modelo de fonte de luz compacta única	55
4.5.1	Tonalização lambertiana com iluminação unidirecional	55
4.5.2	Tonalização lambertiana com luz difusa	56
4.5.3	Modelo para superfície parcialmente lustrosa	59
4.5.4	Iluminação ambiente difusa	60
4.5.5	Modelo geral para fonte compacta única	62
4.6	Ajustando o modelo de fonte única	62
4.6.1	Determinando \vec{u} com K, ρ conhecidos	62
4.6.2	Determinando \vec{u}, ρ, K e ξ por otimização não linear	66
4.7	Testes	66
4.7.1	Gabarito sintético lambertiano com luz unidirecional	67
4.7.2	Gabarito sintético lustroso com luz compacta	69
4.7.3	Esfera pintada	72
4.7.4	Esfera de látex	75
5	Inversão por busca em tabela	79
5.1	Tabela de assinaturas	79
5.2	Trabalhos anteriores	80
5.3	Busca rápida com grade de baldes 2D	80
5.3.1	A grade 2D	81
5.3.2	O algoritmo de busca	83
5.3.3	Condição de varredura do balde	83
5.3.4	Ordem de busca de baldes e retorno precoce	84
5.3.5	Análise	85
5.4	Testes	86
5.4.1	Testes com imagens sintéticas de cena lambertiana	87
5.4.2	Testes com imagens sintéticas de cena lustrosa	93
5.4.3	Testes com fotos de uma cena real	97
5.5	Comparação com algoritmo $k-d$ tree	105
5.6	Comparação com ANN e LSH	107
5.7	Efeito do ruído na imagem	108
5.7.1	Efeito no custo da busca.	108

5.7.2	Efeito na precisão do resultado	110
6	Sombras projetadas e realces lustrosos	113
6.1	Anomalia de radiância e pontos canônicos	114
6.2	A hipótese de dados suficientes	115
6.3	Soluções anteriores	115
6.3.1	Descarte de imagens anômalas	115
6.3.2	Outros métodos	116
6.4	Nossa abordagem: inferência bayesiana	117
6.4.1	Análise bayesiana	119
6.4.2	Simplificação da fórmula bayesiana	120
6.4.3	Modelo probabilístico das anomalias	121
6.5	Maximização da probabilidade	123
6.5.1	Enumeração exaustiva	124
6.5.2	Enumeração exaustiva de subconjuntos	124
6.5.3	Consenso com amostragem aleatória	125
6.5.4	Refinamento da solução	125
6.6	Testes	126
6.6.1	Testes com cena sintética lambertiana	126
6.6.2	Testes com cena sintética lustrosa	128
6.6.3	Teste com cena real lambertiana	130
6.6.4	Teste com cena real lustrosa	133
II	Integração do mapa de normais	135
7	Integração multi-escala de mapas de gradiente	137
7.1	Definição do Problema	137
7.1.1	Integração de gradientes obtidos por estéreo fotométrico	138
7.1.2	Mapas de peso	139
7.1.3	Calculando gradientes a partir das normais	139
7.2	Trabalhos relacionados	140
7.2.1	Métodos de integração por caminhos	140
7.2.2	Métodos de filtragem via Fourier	141
7.2.3	Métodos de Poisson com resolução direta	141
7.2.4	Métodos iterativos locais	142
7.2.5	Outros métodos	143
7.3	Equações ponderadas de Poisson	144
7.3.1	Forma contínua das equações de Poisson	144

7.3.2	Discretizando as alturas	145
7.3.3	Equações discretas para pesos uniformes	146
7.3.4	Equações axiais para pesos gerais	147
7.3.5	Equações diagonais para pesos gerais	149
7.3.6	Alturas desconexas	149
7.3.7	Montando o sistema	150
7.4	Estimando as derivadas de aresta	150
7.5	Solução multi-escala	152
7.5.1	O algoritmo	153
7.5.2	Reduzindo os mapas de gradiente e pesos	154
7.5.3	Expandindo o mapa de alturas	155
7.5.4	Ordenando as equações	156
7.5.5	Termo linear indeterminado	157
7.5.6	Análise	157
7.6	Testes	158
7.6.1	Algoritmos de integração	160
7.6.2	Robustez e precisão	161
7.6.3	Tempo e memória	167
7.7	Calculando pesos a partir dos gradientes	169
7.8	Limitações do método multi-escala	170
8	Integração de mapas de gradiente por malhas não regulares	173
8.1	Malha de diferenças ponderadas	173
8.1.1	O problema de integração da malha	174
8.1.2	Integração por caminhos	175
8.1.3	Equação de equilíbrio dos vértices	175
8.1.4	Formulação matricial	176
8.2	Conversão de grade para malha	177
8.3	Integração multi-escala de malha	178
8.3.1	Dizimação de malha	179
8.3.2	Interpolação	181
8.3.3	Ajuste iterativo	182
8.4	Análise do algoritmo	182
8.4.1	Corretude	182
8.4.2	Tempo e espaço	182
8.4.3	Velocidade de convergência	183
8.5	Testes	184
8.5.1	Conjunto de testes	184

8.5.2	Precisão e robustez	185
8.5.3	Custo	190
III	Aplicação	193
9	Captura da face humana	195
9.1	Trabalhos anteriores	195
9.1.1	Considerações gerais	196
9.2	O projeto <i>Photoface</i>	198
9.2.1	Captura das imagens	199
9.2.2	Calibração do dispositivo	199
9.2.3	Cálculo das normais	201
9.3	Melhorando o Photoface	202
9.3.1	Calibração da fonte de luz	202
9.3.2	Cálculo das normais	204
9.3.3	Integração do mapa de normais	204
10	Conclusões	207
10.1	Contribuições	207
10.1.1	Interpolação e suavização de gabaritos	207
10.1.2	Aceleração de estéreo fotométrico	207
10.1.3	Estéreo fotométrico robusto	207
10.1.4	Integração eficiente e robusta	208
10.2	Aplicações	208
10.3	Publicações	208
10.4	Trabalhos Futuros	209
10.4.1	Inversão analítica	209
10.4.2	Aceleração de estéreo fotométrico robusta	209
10.4.3	Variação espacial de iluminação	210
10.4.4	Associação com outras abordagens	210
	Bibliografia	211

Capítulo 1

Introdução

Esta tese trata do uso de estéreo fotométrico para a captura de modelos tridimensionais de objetos reais. Este é um problema computacional com inúmeras aplicações, como por exemplo a documentação e restauração de monumentos e obras de artes [72, 86, 66], a engenharia reversa de peças ou prédios [3, 41], segurança [94], controle de qualidade [91], jogos [45], medicina [99] e biometria [48, 14].

1.1 Captura de modelos tridimensionais

Uma maneira direta de capturar a geometria de uma cena tridimensional é usando *scanners tridimensionais* [110, 103], aparelhos que obtêm a terceira dimensão medindo o tempo de trânsito de pulsos *laser* dirigidos ao objeto. Entretanto, tais equipamentos ainda são caros e volumosos, e seu campo de aplicação ainda é relativamente restrito.

Outra maneira é a técnica de *luz estruturada* [26, 63, 1] em que uma textura conhecida (linha, grade, rede de pontos, etc.) é projetada obliquamente sobre o objeto, que é fotografado com uma câmera comum. O deslocamento lateral aparente da textura na imagem permite determinar a profundidade. Esta técnica, embora mais barata que *scanners* tridimensionais, exige um projetor de boa resolução, e é limitada a objetos de baixo relevo.

Uma terceira abordagem é a *visão estereoscópica* (ou *estéreo*) *computacional*, que permite obter a terceira dimensão a partir de duas ou mais fotos digitais comuns, dispensando equipamentos especializados. Apesar de exigir algoritmos mais complexos que as outras duas, esta abordagem tem a vantagem de baixo custo e alta disponibilidade do equipamento (câmeras digitais comuns), rapidez de aquisição, e grande gama de aplicações — tendo sido usada para objetos de todo tipo, escala e tamanho, desde planetas [10] até micro-organismos [67].

1.2 Técnicas de estéreo computacional

As técnicas de visão estereoscópica computacional podem ser divididas em duas classes principais: *estéreo geométrico* e *estéreo fotométrico*.

No estéreo geométrico, utilizam-se duas ou mais fotos do objeto, tiradas a partir de posições diferentes. O deslocamento da câmera entre uma foto e outra dá origem a *paralaxes*, deslocamentos aparentes das várias partes da cena na imagem, que variam conforme a profundidade dessas partes. Assim, é possível recuperar a terceira dimensão da cena identificando os pontos das imagens dadas que correspondem ao mesmo ponto na cena, e analisando suas paralaxes. O resultado pode ser apresentado na forma de um *mapa de alturas* que dá a terceira dimensão do ponto da superfície visível em cada pixel. Veja a figura 1.1.

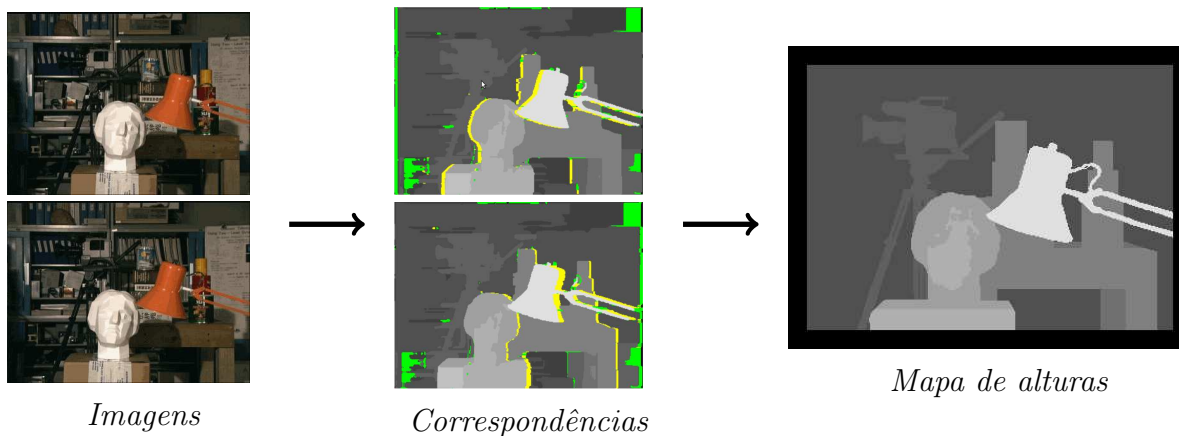


Figura 1.1: Sequência de passos para reconstrução tridimensional por estéreo geométrico. As correspondências foram calculadas pelo método de DiStefano *et al.* [70]. Nos mapas de correspondências (centro), regiões com a mesma cor nas duas imagens são consideradas correspondentes, regiões em verde indicam falha no método e regiões em amarelo indicam a presença de oclusão.

No estéreo fotométrico, por outro lado, as fotos do objeto são todas tiradas do mesmo ângulo e distância, mas com iluminações diferentes. Em princípio, os valores de um mesmo pixel em todas essas fotos permitem determinar a direção normal da superfície visível do objeto nesse pixel. É esta técnica que consideramos nesta tese.

O resultado imediato do estéreo fotométrico é apenas um *mapa de normais*. Numa etapa separada, estas inclinações podem ser integradas para se obter o mapa de alturas. Veja a figura 1.2.

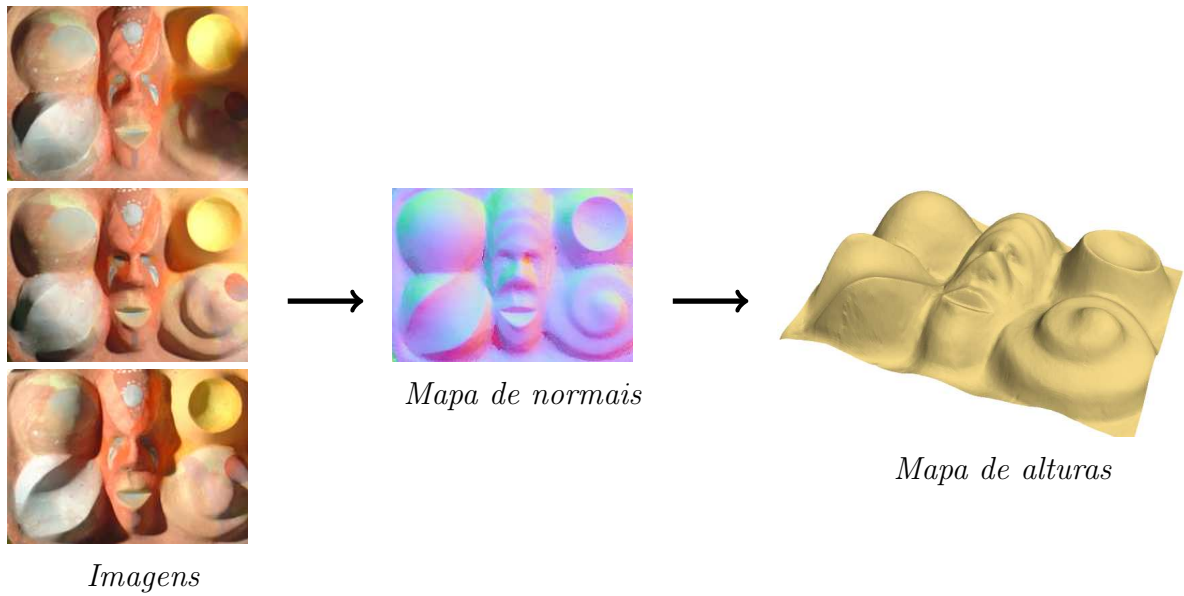


Figura 1.2: Seqüência de passos para reconstrução tridimensional por estéreo fotométrico. No mapa de normais, para fins de visualização, cada normal $\vec{n} = (\vec{n}.x, \vec{n}.y, \vec{n}.z)$ está codificada como uma cor RGB com $R = (\vec{n}.x + 1)/2$, $G = (\vec{n}.y + 1)/2$ e $B = \vec{n}.z$.

1.3 Aplicações de estéreo fotométrico

As aplicações onde estéreo fotométrico foi usado com sucesso incluem, entre outras, captura da geometria facial [12, 87], segurança [94], arqueologia [60, 86, 72] e dermatologia [95]. Veja a figura 1.3.

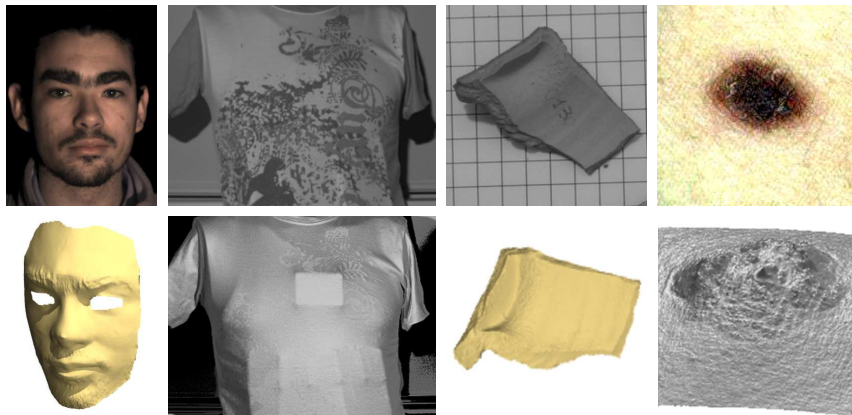


Figura 1.3: Da esquerda para direita: aplicações de estéreo fotométrico em captura de geometria facial, inspeções de segurança, arqueologia e dermatologia.

1.4 Vantagens e desvantagens do estéreo fotométrico

Uma vantagem do estéreo fotométrico é sua resolução potencial. No estéreo geométrico, a identificação de pontos correspondentes nas imagens dadas exige a comparação de janelas com vários pixels de largura. Neste processo, detalhes em relevo da superfície com tamanho menor que essa janela são geralmente ignorados ou incorretamente identificados. Portanto, a resolução lateral do estéreo geométrico (nas direções paralelas ao plano da imagem) é geralmente bem menor que a resolução das imagens de entrada [20, 3]. Além disso, o erro da profundidade calculada pelo estéreo geométrico é inversamente proporcional ao ângulo entre as direções de visão. Portanto, em situações típicas, a resolução em profundidade é bem menor que a resolução lateral.

Em comparação, métodos de estéreo fotométrico permitem obter a inclinação da superfície da cena em cada pixel individualmente. A resolução lateral é portanto limitada pela resolução das imagens digitais da cena. Por exemplo, uma câmera digital com sensor de 1000×1000 pixels (1 megapixel) permite resolver detalhes de 0.5 mm de diâmetro em um objeto de 50cm.

Métodos de estéreo fotométrico também funcionam muito bem para objetos com superfícies lisas, uniformes e sem textura ou texturas periódicas, onde o estéreo geométrico não pode ser usado.

A principal desvantagem do estéreo fotométrico é que, no estado atual da arte, ele exige iluminação bastante uniforme e superfícies com acabamento não muito complexo. Por exemplo, o estéreo fotométrico não pode ser usado em superfícies muito rugosas ou esponjosas como tecidos e cascalho ou superfícies com texturas anisotrópicas, como aço escovado. Sombras, penumbras, inter-reflexões, reflexos e iluminação não uniforme também dificultam bastante o estéreo fotométrico e limitam a precisão dos resultados.

Mesmo nas melhores condições a inclinação calculada por estéreo fotométrico está sujeita a erros de várias naturezas, tipicamente da ordem de 5 a 10 graus [3]. A etapa de integração das inclinações calculadas pode amplificar ainda mais esses erros. Além disso, devido à perspectiva fixa da câmera, as imagens dadas não contêm informação suficiente para determinar as diferenças abruptas de profundidade que ocorrem ao longo das silhuetas dos objetos da cena.

Em vista dessas considerações, pode-se dizer que o estéreo geométrico e fotométrico são complementares mais que competidores [106]. A integração das duas técnicas é um assunto atual de pesquisa.

1.5 Objetivos da tese

Os tópicos desta tese de doutorado são a análise dos fundamentos de estéreo fotométrico, o desenvolvimento de métodos computacionais inovadores para esse problema e sua aplicação a problemas práticos selecionados. Em particular, consideramos o uso de *gabaritos de iluminação*, objetos de forma e cor conhecida, para determinar as características da iluminação usada em cada foto.

A estrutura da tese é a seguinte:

- **Fundamentos.** No capítulo 2 apresentamos os conceitos fundamentais do estéreo fotométrico, baseados na física de formação das imagens digitais.
- **Trabalhos relacionados.** No capítulo 3 descrevemos as diversas metodologias presentes na literatura no campo de estéreo fotométrico, apresentamos uma classificação para estas abordagens e discutimos seus pontos fortes e fracos.
- **Modelagem de gabaritos de iluminação.** No capítulo 4 descrevemos como extrair e representar matematicamente a informação contida nos gabaritos de iluminação visando eliminar os erros devidos a defeitos de formato, irregularidades na sua superfície, ruído da câmera e erros de quantização.
- **Aceleração do estéreo fotométrico.** As m intensidades de um mesmo pixel em todas as m fotos da cena constituem o *vetor de observação* deste pixel. O problema central do estéreo fotométrico é determinar a normal no pixel a partir do vetor de observação.

No estéreo fotométrico baseado em gabaritos, este problema pode ser visto como um caso particular do problema de encontrar o ponto mais próximo a um ponto dado em um conjunto de pontos no espaço \mathbb{R}^m . Os métodos descritos na literatura para efetuar esta busca são extremamente lentos. No capítulo 5 apresentamos um método original para acelerar esta etapa. Este método fundamenta-se na nossa descoberta de que o conjunto de vetores de observação do gabarito forma uma variedade bidimensional relativamente plana no espaço \mathbb{R}^m .

- **Tratamento de sombras e reflexos.** Muitos métodos de estéreo fotométrico, incluindo o descrito no capítulo 5, são extremamente sensíveis a sombras, reflexos e outras anomalias de iluminação. No capítulo 6 apresentamos uma abordagem que utiliza um modelo estatístico bimodal (*inlier/outlier*) e análise bayesiana para tratar estas anomalias.
- **Integração de mapas de gradiente.** Nos capítulos 7 e 8 desenvolvemos métodos eficientes e robustos para a integração dos mapas de normais produzidos por estéreo

fotométrico. Mostramos que nossos algoritmos são mais eficientes e precisos que os encontrados na literatura. O integrador descrito no capítulo 7 trabalha em uma malha regular e usa equações de diferenças finitas que são resolvidas por uma versão multi-escala do método iterativo de Gauss-Seidel. Este método é muito rápido mas falha quando o mapa de inclinação apresenta regiões conectadas apenas por corredores estreitos, que são perdidos quando o mapa é reduzido para escalas menores. No capítulo 8 desenvolvemos outro método, também iterativo e multi-escala mas usando malhas irregulares, que não sofre desta limitação.

- **Captura da geometria facial.** Finalmente, no capítulo 9 discutimos o uso dos nossos algoritmos para uma aplicação específica, a captura da geometria tridimensional da face humana. Neste capítulo, descrevemos o projeto *Photoface* da University of West England, para o qual contribuímos durante nosso estágio-sanduíche.

Parte I

Determinação das normais e albedos

Capítulo 2

Fundamentos do Estéreo Fotométrico

Neste capítulo vamos apresentar os princípios do estéreo fotométrico de iluminação variável (EFIV). O objetivo desta técnica é determinar a geometria tridimensional de uma cena a partir de uma lista de $m \geq 3$ imagens digitais monocromáticas S_1, S_2, \dots, S_m , obtidas em diferentes condições de iluminação, com mesmo ponto de vista. Veja a figura 2.1.

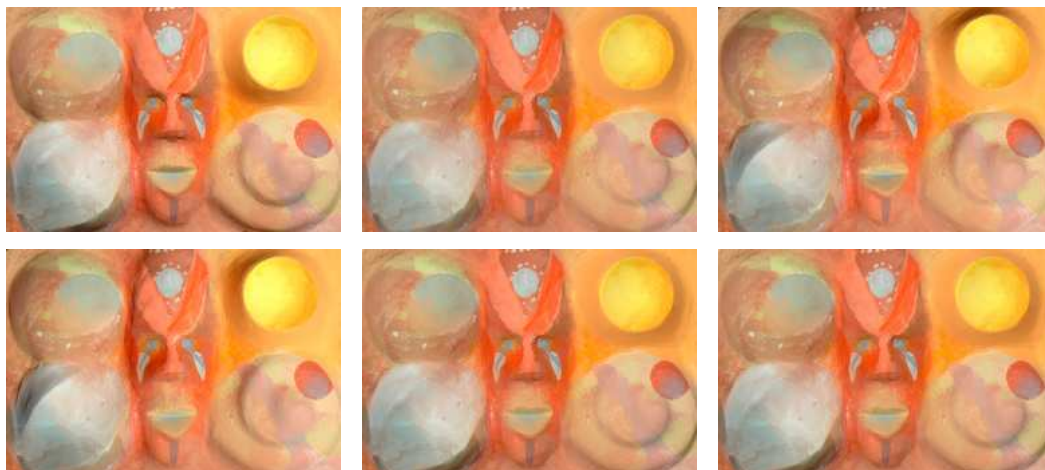


Figura 2.1: Seis imagens de uma mesma cena sob diferentes condições de iluminação.

A técnica EFIV foi introduzida em 1980, por R. J. Woodham [106]. Ele mostrou que, analisando-se as m intensidades de pixel $S_i[p]$ em um ponto p da imagem, é possível em certos casos recuperar a *normal* $\vec{n}[p]$, o vetor unitário perpendicular ao elemento de superfície da cena que é visível no ponto p . Este resultado fundamental foi bastante discutido e elaborado nos anos seguintes [18, 53, 88, 116, 16, 108].

2.1 Geometria da formação das imagens

As imagens dadas S_i podem ser consideradas *projeções cônicas* (ou *de perspectiva*) da luz emitida pela cena sobre uma *matriz de sensores* situada atrás da lente da câmera. Vamos supor que cada imagem está geometricamente retificada, de modo que podemos imaginar que ela é formada por linhas retas que ligam os pontos visíveis da cena a suas projeções nos sensores da câmera, passando por um ponto comum no interior da lente, o *centro ótico da câmera*. A reta E que passa pelo centro ótico e é perpendicular ao plano de projeção é chamada *eixo ótico* da câmera. Veja a figura 2.2(a)

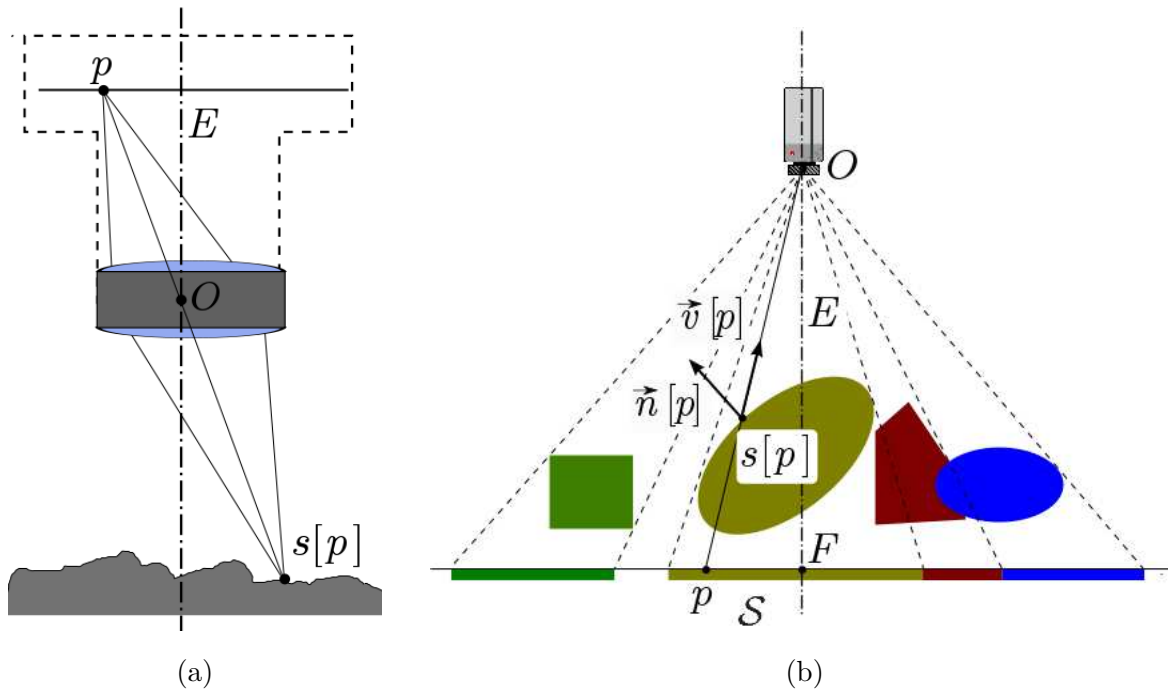


Figura 2.2: (a) Esquema da formação de uma imagem digital por projeção cônica da cena sobre a matriz de sensores, através do centro ótico O da lente. A reta E é o eixo ótico da câmera. (b) Projeção idealizada da cena a partir do centro ótico em um plano de imagem situado abaixo dela, mostrando o ponto visível da cena $s[p]$, a direção normal $\vec{n}[p]$ e a direção de vista $\vec{v}[p]$ correspondente a um ponto p da imagem. O ponto F é o centro ótico da imagem.

2.1.1 Imagem e pixels

Embora numa câmera real a projeção seja feita pela luz que sai da cena e atinge a matriz de sensores situada atrás da lente, para fins de cálculo podemos imaginar também que a projeção ocorre no sentido oposto, por linhas retas a partir do centro ótico da câmera

passando pelos pontos visíveis da cena até um *plano da imagem*, perpendicular ao eixo ótico e situado em alguma posição arbitrária além da cena. Veja a figura 2.2(b).

Na maioria das câmeras, os sensores estão dispostos em uma grade retangular com os espaçamentos iguais nas duas direções principais. Se a matriz de sensores tem n_x colunas e n_y linhas, a imagem terá n_x colunas e n_y linhas de elementos, denominados *pixels*. Conceitualmente, cada pixel é um quadrado no plano de projeção; a união desses quadrados é o *domínio* \mathcal{S} da imagem. O ponto F onde o eixo ótico da câmera encontra o plano de projeção é o *centro ótico da imagem*, que, em geral, coincide com o centro geométrico do domínio \mathcal{S} das imagens.

Nesta tese vamos supor que todas as imagens S_1, \dots, S_m estão cortadas e alinhadas de tal forma que elas tem o mesmo domínio \mathcal{S} , e que cada ponto p de \mathcal{S} corresponde ao mesmo ponto $s[p]$ na parte visível da superfície da cena.

A direção (vetor unitário de \mathbb{R}^3) do ponto p (ou $s[p]$) para o centro ótico da câmera é denominada *direção de vista* desse ponto e denotada $\vec{v}[p]$. As normais $\vec{n}[p]$, de todos os pixels p de \mathcal{S} , podem ser apresentadas na forma de uma imagem, o *mapa de normais*. Veja a figura 2.3.



Figura 2.3: Mapa de normais \vec{n} da cena da figura 2.1.

2.1.2 Sistema de coordenadas da cena

Para simplificar os cálculos, vamos supor um *sistema de coordenadas da cena* cuja origem é o centro ótico F da imagem. Neste sistema, o eixo x é paralelo ao eixo horizontal da imagem, apontando para a direita; o eixo y é paralelo ao eixo vertical da imagem, apontando para cima; e o eixo z coincide com o eixo ótico, apontando na direção da câmera. A unidade de medida para os três eixos é o espaçamento entre os pixels no plano de projeção; de modo que cada pixel da imagem poderá ser identificado com um quadrado de lado 1 no plano de projeção (onde $z = 0$). Veja a figura 2.4.

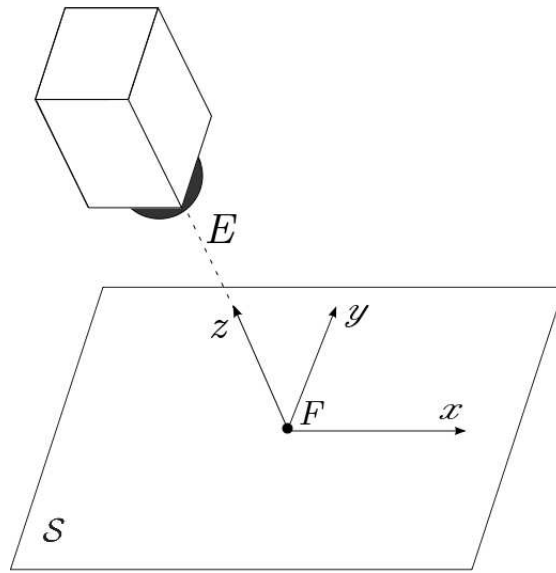


Figura 2.4: O sistema de coordenadas da cena. O quadrilátero é o domínio \mathcal{S} da imagem, situado no plano de projeção.

2.1.3 A direção de vista

Em situações típicas, a direção de vista $\vec{v}[p]$ varia sensivelmente de ponto para ponto, e esta variação não pode ser ignorada. Ela pode ser calculada pela fórmula

$$\vec{v}[p] = \frac{V[p]}{|V[p]|} \quad (2.1)$$

onde $V[p] = (-p.x, -p.y, O.z)$, sendo que todas as coordenadas são medidas em pixels no sistema da câmera. Note que $O.z$ é a distância entre o centro ótico da câmera O e o plano da imagem. Note também que a direção $\vec{v}[p]$ depende do ponto projetado p , mas não depende da geometria da cena (ou seja, da altura do ponto $s[p]$ em relação ao plano de projeção).

2.2 Princípios de Fotometria

Nesta seção vamos rever os princípios básicos de fotometria necessários para esta tese. Vamos inicialmente supor que todas as imagens são obtidas sob luz *monocromática* (com um único comprimento b de onda).

2.2.1 Valores dos pixels

Com iluminação monocromática, podemos supor que a informação capturada pela câmera, para cada pixel p de cada imagem S_i , é um valor real (*amostra*) $S_i[p]$ que mede a intensidade da luz que a lente projeta no sensor da câmera correspondente ao pixel p . A lista de valores $(S_1[p], S_2[p], \dots, S_m[p])$ é o *vetor de observação* do pixel p denotado por $\mathbf{S}[p]$.

2.2.2 Radiância da superfície

Vamos supor também que a codificação das amostras é linear, ou seja, que o valor de cada amostra $S_i[p]$ é diretamente proporcional à energia luminosa medida pelo elemento sensor correspondente. Ou seja, o valor de cada amostra é proporcional à *radiância* (brilho aparente) média do fragmento da superfície da cena que é visível nesse pixel [56].

Para definir melhor este conceito, vamos considerar um elemento infinitesimal da superfície da cena em torno do ponto $s[p]$, observado de uma direção \vec{v} . A radiância desse elemento, vista de \vec{v} , é

$$\frac{dP}{d\vec{v} dA} \quad (2.2)$$

onde dA é a área do elemento projetada perpendicularmente a \vec{v} , e dP é a potência (watts) de toda luz que é emitida por este elemento de superfície e cuja direção de propagação está dentro de um cone com eixo \vec{v} e ângulo sólido infinitesimal $d\vec{v}$. Veja a figura 2.5.

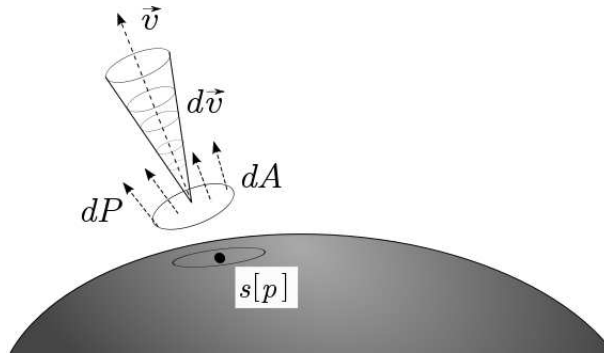


Figura 2.5: Definição da radiância.

2.2.3 Fluxo de iluminação

O estéreo fotométrico exige conhecimento (direto ou indireto) do *campo de iluminação* de cada imagem S_i . Trata-se de uma função $\Phi_i[p](\vec{u})$, que fornece, para cada ponto p de \mathcal{S} e cada direção \vec{u} na esfera unitária \mathbb{S}^2 , a densidade de fluxo de potência luminosa (watts por

metro quadrado por estereorradiano) logo acima do ponto $s[p]$ da cena, vinda de direções próximas a \vec{u} [56].

Considere um círculo de área infinitesimal dA perpendicular a \vec{u} , logo acima de $s[p]$. O valor de $\Phi_i[p](\vec{u})$ é dado por

$$\Phi_i[p](\vec{u}) = \frac{dP}{d\vec{u} dA} \quad (2.3)$$

onde dP é a potência luminosa que atravessa esse círculo, devida a luz cuja direção de origem está dentro de um cone em torno de \vec{u} com um ângulo sólido infinitesimal $d\vec{u}$. Veja a figura 2.6.

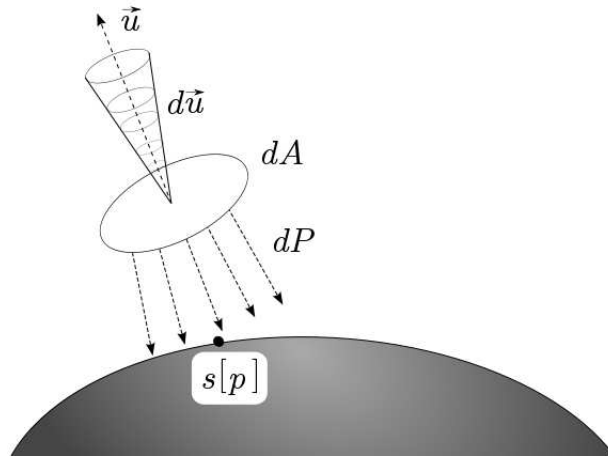
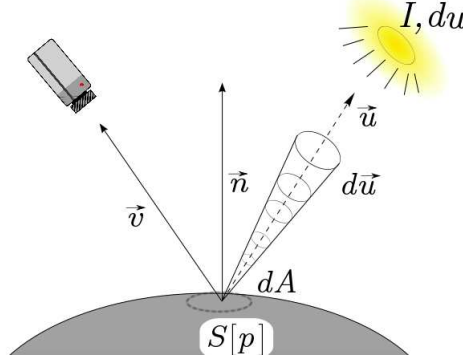


Figura 2.6: Definição do fluxo luminoso $\Phi_i[p](\vec{u})$.

2.3 Função de distribuição de radiância

Outra informação importante para estéreo fotométrico é a maneira como a superfície da cena espalha ou absorve a luz que incide sobre ela.

Vamos supor um elemento compacto da superfície da cena de área infinitesimal dA e normal \vec{n} em torno do ponto $s[p]$, iluminado por luz proveniente de uma fonte com radiância I que, vista de $s[p]$, está localizada na direção \vec{u} e cobre um ângulo sólido infinitesimal de $d\vec{u}$ estereorradianos. Suponha que esse elemento é observado de uma direção \vec{v} . Veja a figura 2.7.

Figura 2.7: Definição da FBDR $\sigma[p](\vec{n}, \vec{u}, \vec{v})$.

A radiância do elemento dA , devida à luz desse elemento da fonte luminosa, pode ser escrita como um produto

$$\sigma[p](\vec{n}, \vec{u}, \vec{v}) |\vec{n} \cdot \vec{u}| I d\vec{u} \quad (2.4)$$

onde $\sigma[p](\vec{n}, \vec{u}, \vec{v})$ é um fator que em geral depende das três direções $\vec{n}, \vec{u}, \vec{v}$. O termo $|\vec{n} \cdot \vec{u}|$ nesta fórmula leva em conta o fato de que o fluxo de luz vindo na direção \vec{u} se espalha sobre uma área proporcional a $1/|\vec{n} \cdot \vec{v}|$ e portanto a intensidade efetiva da fonte é reduzida por esse fator.

A função $\sigma[p]$ é por definição a *função bidirecional de distribuição da radiância* (FBDR) da superfície da cena no ponto $s[p]$. Observe que para superfícies de objetos sólidos opacos, como as consideradas nesta tese, o valor $\sigma[p](\vec{n}, \vec{u}, \vec{v})$ da FBDR é zero se $\vec{n} \cdot \vec{u} < 0$ ou se $\vec{n} \cdot \vec{v} < 0$.

2.3.1 Albedo

Para cada direção fixa \vec{u} de origem da luz, a energia luminosa capturada por um elemento da superfície de área dA e normal \vec{n} é

$$|\vec{n} \cdot \vec{u}| I dA d\vec{u} \quad (2.5)$$

A energia emitida pelo elemento dA na direção \vec{v} é

$$\sigma[p](\vec{n}, \vec{u}, \vec{v}) |\vec{n} \cdot \vec{u}| |\vec{n} \cdot \vec{v}| I dA d\vec{u} \quad (2.6)$$

onde o fator $|\vec{n} \cdot \vec{v}|$ captura o fato de que o elemento dA tem área $|\vec{n} \cdot \vec{v}| dA$ quando projetado na direção \vec{v} . Portanto a integral

$$a[p](\vec{n}, \vec{u}) = \int_{\mathbb{S}^2} \sigma[p](\vec{n}, \vec{u}, \vec{v}) |\vec{n} \cdot \vec{v}| d\vec{v} \quad (2.7)$$

dá a fração total da energia incidente vinda da direção \vec{u} que é devolvida em todas as direções $\vec{v} \in \mathbb{S}^2$ pela superfície, na vizinhança de $s[p]$. Esta fração $a[p](\vec{n}, \vec{u})$ é denominada o *albedo* da superfície para a direção de iluminação \vec{u} e normal \vec{n} .

Para materiais que tipicamente ocorrem na superfície de objetos, como tintas, plásticos, cerâmica e concreto, o albedo depende apenas do produto escalar $\vec{n} \cdot \vec{u}$. Madeira e certos materiais com textura complexa, como tecidos e pelos, podem ter comportamento mais complexo.

O valor $a[p](\vec{n}[p], \vec{u})$ amostrado em cada pixel p de \mathcal{S} pode ser apresentado na forma de uma imagem, o *mapa de albedo*. Veja a figura 2.8.

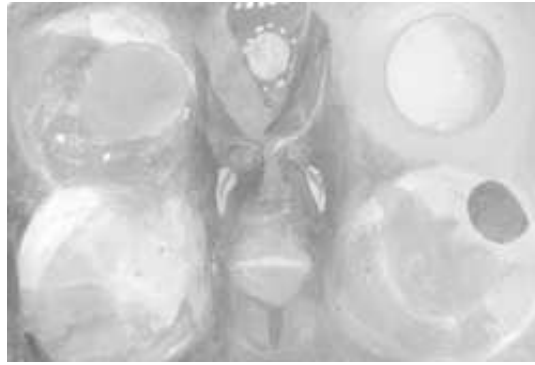


Figura 2.8: Mapa de albedo a obtido da cena da figura 2.1 para o canal de luminância Y.

2.3.2 Acabamento

Podemos então decompor uma FBDR genérica $\sigma[p]$ no produto dos dois fatores

$$\sigma[p](\vec{n}, \vec{u}, \vec{v}) = a[p](\vec{n}, \vec{u}) \beta[p](\vec{n}, \vec{u}, \vec{v}) \quad (2.8)$$

onde β é uma FBDR que devolve toda energia luminosa que incide sobre ela. Isto é, a FBDR β tem albedo unitário

$$\int_{\mathbb{S}^2} \beta[p](\vec{n}, \vec{u}, \vec{v}) |\vec{n} \cdot \vec{v}| d\vec{v} = 1 \quad (2.9)$$

Esta propriedade pode ser considerada a característica que define uma superfície “branca”. Diremos que a FBDR $\beta[p]$ que satisfaz as equações (2.8) e (2.9) é o *acabamento* da superfície no ponto $s[p]$.

2.3.3 Acabamento lambertiano

O exemplo canônico de acabamento é a FBDR de uma superfície branca opaca lambertiana:

$$\beta_L(\vec{n}, \vec{u}, \vec{v}) = \begin{cases} \frac{1}{2\pi} & \text{se } \vec{n} \cdot \vec{u} > 0 \text{ e } \vec{n} \cdot \vec{v} > 0 \\ 0 & \text{se } \vec{n} \cdot \vec{u} \leq 0 \text{ ou } \vec{n} \cdot \vec{v} \leq 0 \end{cases} \quad (2.10)$$

Observe que $\beta_L(\vec{n}, \vec{u}, \vec{v})$ é constante para todo \vec{n}, \vec{u} e \vec{v} , desde que o ponto seja iluminado ($\vec{n} \cdot \vec{u} > 0$) e visível ($\vec{n} \cdot \vec{v} > 0$). Este acabamento é característico de materiais cuja superfície é composta de partículas irregulares muito pequenas (cristais ou fibras) com orientações aleatórias que lhe dão uma aparência fosca. Gesso, papel sulfite, tinta fosca de parede e tinta guache são exemplos de superfícies com acabamento muito próximo do lambertiano. Veja a figura 2.9.

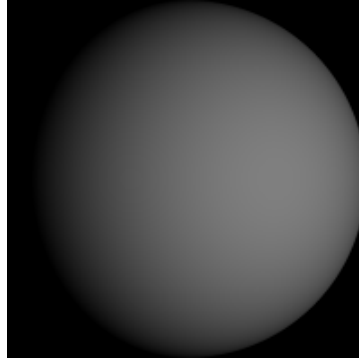


Figura 2.9: Imagem sintética de uma esfera, calculada supondo acabamento lambertiano e iluminação por uma fonte de luz pontual distante com direção $\vec{u} = (\sin 30^\circ, 0, \cos 30^\circ)$.

No nosso caso, a condição $\vec{n} \cdot \vec{v} > 0$ é automaticamente satisfeita pela definição do ponto $s[p]$; de modo que a radiância de uma superfície lambertiana não depende da direção de observação \vec{v} .

2.3.4 Acabamento lustroso e o modelo de Phong

Muitas superfícies reais têm aparência parcialmente lustrosa. Esse efeito pode ser devido a polimento ou compactação do material de modo a tornar sua superfície relativamente lisa na escala microscópica; ou a uma película lisa e transparente (como verniz, cera ou a componente resinosa de tintas esmalte) aplicada sobre a superfície natural do objeto. Nos dois casos, o efeito é transformar parte da superfície em um espelho rugoso, com irregularidades maiores que o comprimento de onda da luz, mas menores que um pixel.

Em uma superfície espelhada totalmente lisa, toda energia que incide na superfície vinda de uma direção \vec{u} é refletida na direção \vec{v} tal que a normal \vec{n} coincide com a bissetriz \vec{m} de \vec{u} e \vec{v} . Em uma superfície lustrosa mas não espelhada, as ondulações invisíveis da

superfície espalham a luz de maneira mais ampla. Visualmente, o resultado são manchas brilhantes (*realces* ou *highlights*) nas partes da cena onde a normal \vec{n} é próxima de \vec{m} (e não apenas nos pontos onde $\vec{n} = \vec{m}$).

O *acabamento de Phong* β_P [78] é um modelo matemático para a FBDR de tais superfícies, muito usado em computação gráfica, definido pela fórmula

$$\beta_P^K(\vec{n}, \vec{u}, \vec{v}) = \begin{cases} A_K(\vec{n} \cdot \vec{m})^K & \text{se } \vec{n} \cdot \vec{m} > 0 \text{ e } \vec{n} \cdot \vec{u} > 0 \\ 0 & \text{caso contrário} \end{cases} \quad (2.11)$$

Nesta fórmula o parâmetro K é um numero real não-negativo que determina o grau de polimento do objeto, e A_K é uma constante de normalização (que depende de K) necessária para garantir que o albedo de β_P^K seja unitário como exigido pela equação (2.9). Veja a figura 2.10. Quanto maior o valor de K , mais compacto será o realce; isto é, o valor de β_P^K diminuirá mais rapidamente à medida que \vec{n} se afastar de \vec{m} .

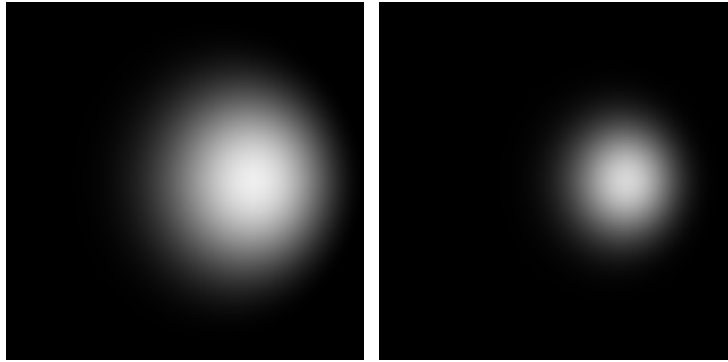


Figura 2.10: Imagens sintéticas da esfera da figura 2.9 calculadas supondo acabamento puramente lustroso de Phong com parâmetro $K = 4$ (esquerda) e $K = 10$ (direita), cada qual iluminada por uma fonte pontual distante com direção $\vec{u} = (\sin 30^\circ, 0, \cos 30^\circ)$. As intensidades das fontes foram ajustadas para melhor visualização.

Note que o acabamento lambertiano β_L é o caso limite de β_P^K quando o parâmetro K é zero. Por outro lado, o limite de β_P^K quando K tende para infinito é o acabamento de superfície lisa espelhada.

Há outros modelos mais sofisticados para acabamento lustroso, como o de Trowbridge-Reitz [101] e de Torrance-Sparrow [100]; mas as diferenças entre esses modelos são relativamente pequenas.

Em geral, superfícies reais são parcialmente lambertianas e parcialmente polidas. Seu acabamento então pode ser modelado por uma FBDR que é uma combinação convexa do acabamento lambertiano β_L com o acabamento β_P^K de Phong. Veja a figura 2.11.

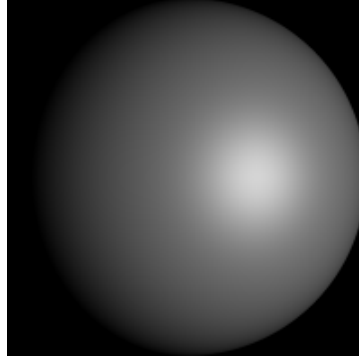


Figura 2.11: Imagem sintética da esfera da figura 2.9, calculada supondo acabamento misto (lambertiano e de Phong com $K = 10$) iluminada por uma fonte distante pontual com direção $\vec{u} = (\sin 30^\circ, 0, \cos 30^\circ)$.

2.4 Equação de tonalização

Com estes conceitos, podemos expressar a radiância (ou brilho aparente) $S_i[p]$ de um pixel p da imagem i pela *equação de tonalização monocromática* :

$$S_i[p] = \int_{\mathbb{S}^2} \Phi_i[p](\vec{u}) a[p](\vec{n}[p], \vec{u}) \beta[p](\vec{n}[p], \vec{u}, \vec{v}[p]) |\vec{n}[p] \cdot \vec{u}| d\vec{u} \quad (2.12)$$

2.5 Hipóteses simplificadoras

Com o modelo geral de tonalização descrito acima, o problema do EFIV é bastante difícil ou mesmo insolúvel; pois, se o albedo $a[p](\vec{n}, \vec{u})$ variar de maneira suficientemente complexa em função de \vec{n} e \vec{u} , o vetor de observação $\mathcal{S}[p]$ pode não conter informação suficiente para determinar a normal \vec{n} . Nesta tese (como na maioria dos trabalhos sobre EFIV) contornamos este obstáculo adotando certas hipóteses simplificadoras sobre a iluminação e sobre a FBDR da superfície: especialmente as hipóteses de *iluminação uniforme*, de *albedo isotrópico* e de *acabamento uniforme*. Iremos também enunciar a hipótese de *luz unidirecional*, que é adotada por muitos pesquisadores, mas que será adotada nesta tese apenas em alguns casos particulares.

2.5.1 Iluminação uniforme

A *hipótese da iluminação uniforme* diz que o fluxo luminoso $\Phi_i[p](\vec{u})$ que incide efetivamente na superfície (isto é, para direções \vec{u} que fazem ângulo agudo com $\vec{n}[p]$) não depende explicitamente do ponto p , mas apenas da direção \vec{u} ; e portanto pode ser escrito $\Phi_i(\vec{u})$. Para que seja possível adotar esta hipótese, a distância da cena às fontes de luz deve ser

bem maior que as dimensões da parte da cena capturada nas imagens. Se a fonte de luz é próxima à cena, a intensidade da iluminação varia com p tanto pelo fator $1/d^2$, onde d é a distância do ponto $s[p]$ à fonte, quanto pela mudança na direção desse ponto à fonte. Além disso, cada fonte deve ser razoavelmente *isotrópica*, isto é, deve emitir a mesma intensidade de fluxo em todas as direções que interceptam a cena.

A hipótese de iluminação uniforme pode parecer inviável na prática, pois a componente de fluxo $\Phi_i[p](\vec{u})$ que é devida a fontes externas à cena é bloqueada pela própria cena nos pontos onde a direção \vec{u} faz ângulo obtuso com a normal $\vec{n}[p]$. Diz-se que esses pontos estão na *sombra própria* para luz vinda de \vec{u} . Nesses pontos, o fluxo $\Phi_i[p](\vec{u})$ é efetivamente zero.

Porém, note que, na equação de tonalização (2.12), o valor de $\Phi_i[p](\vec{u})$ é irrelevante para essas direções \vec{u} . Ou seja, a hipótese de iluminação uniforme, para cada direção \vec{u} , só precisa valer na parte da cena onde $\vec{n}[p] \cdot \vec{u} > 0$.

Outra causa importante de não-uniformidade no fluxo $\Phi[p](\vec{u})$ são as *sombras projetadas*, que ocorrem quando uma *outra* parte da cena bloqueia, total ou parcialmente, a luz que de outra forma incidiria no ponto $s[p]$. Ao contrário das sombras próprias, sombras projetadas causam variações no fluxo de luz que não podem ser ignoradas. Portanto, para que seja possível a hipótese de iluminação uniforme, as fontes luminosas externas à cena devem ser posicionadas de modo a não produzir sombras projetadas. Isto é, o segmento que liga a posição da fonte de luz a qualquer ponto visível da cena não pode atingir nenhuma outra parte da cena. Esta restrição limita severamente a geometria da cena e o posicionamento das fontes de luz.

A hipótese de iluminação não uniforme também não vale quando há *iluminação secundária*, resultante de luz espalhada por outras partes da cena (um efeito conhecido na computação gráfica pelo nome de *radiosidade*). Adotar a hipótese de iluminação uniforme implica supor que variações do fluxo devidas à iluminação secundária são desprezíveis. Isto só é rigorosamente verdade se a parte visível da cena for totalmente convexa. Na prática esta condição dificilmente vale e a iluminação geralmente inclui uma componente perceptível de luz secundária, variável de ponto a ponto. Técnicas para tratar esse problema serão descritas no capítulo 6.

Por outro lado, observamos que o efeito da luz secundária é tanto menor quanto mais escura for a superfície; pois a intensidade da luz espalhada apenas uma vez é proporcional ao albedo no ponto de espalhamento, enquanto que a espalhada duas vezes é proporcional ao produto dos albedos nos dois pontos. Além disso, o efeito da luz secundária é reduzido se a superfície tiver apenas concavidades rasas, pois apenas uma pequena parte da luz espalhada nessas partes da superfície atinge novamente a cena. Por estas razões, mesmo quando a luz secundária é perceptível, podemos obter uma normal razoavelmente correta ignorando sua existência e considerando apenas a iluminação direta.

Na prática é difícil evitar sombras próprias e iluminação secundária. Técnicas para tratar estes casos serão discutidas no capítulo 6.

2.5.2 Albedo isotrópico

Outra hipótese simplificadora importante é a do *albedo isotrópico*, isto é, de que o albedo $a[p](\vec{n}, \vec{u})$ tem o mesmo valor para qualquer orientação \vec{n} da superfície e qualquer direção de iluminação \vec{u} . Com esta hipótese podemos denotar o albedo apenas por $a[p]$. Esta hipótese vale para materiais lambertianos, e pode ser suposta válida para muitos materiais lustrosos. Ela não vale, entretanto, para materiais com texturas tridimensionais complicadas, como madeira, aço escovado e certos tipos de tecido. Ainda não existem algoritmos EFVI que consigam trabalhar com tais materiais, exceto em situações extremamente restritas [53, 71].

2.5.3 Acabamento uniforme

A *hipótese de acabamento uniforme* diz que toda superfície da cena tem o mesmo acabamento; isto é, que $\beta[p](\vec{n}, \vec{u}, \vec{v})$ não depende de p , e portanto pode ser denotado por $\beta(\vec{n}, \vec{u}, \vec{v})$. Note que esta hipótese ainda permite que o *albedo* varie de ponto a ponto. Esta hipótese vale em muitos casos práticos, por exemplo, objetos de cerâmica multicoloridos sem vitrificação, papel colorido e esculturas pintadas.

A hipótese de acabamento uniforme é difícil de conseguir em cenas reais. Se a cena contém objetos de materiais diferentes (por exemplo, gesso e plástico), é necessário conhecer o acabamento de cada material, ou seja, tratar cada objeto como um problema separado. Mesmo objetos que consistem de um único material apresentarão acabamento variável se o grau de polimento variar de ponto a ponto. Técnicas gerais para tratar estes casos também serão discutidas no capítulo 6.

2.5.4 Hipótese de luz unidirecional

Uma suposição feita por muitos pesquisadores é que, em cada imagem, a iluminação é produzida por uma fonte de luz pontual a uma distância grande da cena. Neste caso o fluxo Φ_i é *unidirecional*, isto é, toda a luz vem de uma única direção \vec{u}_i . Nesta condição, podemos supor que Φ é uma *distribuição* (e não uma função) semelhante à distribuição de *Dirac*; isto é, é zero para qualquer direção \vec{u} , exceto $\vec{u} = \vec{u}_i$, onde é infinito, e tal que a integral da equação de tonalização (2.12) simplifica-se para

$$S_i[p] = 2\pi E_i a[p](\vec{n}[p], \vec{u}_i) \beta[p](\vec{n}[p], \vec{u}_i, \vec{v}[p]) |\vec{n}[p] \cdot \vec{u}_i| \quad (2.13)$$

onde E_i é uma constante positiva, a *intensidade da fonte*.

2.6 Função de tonalização

Com as três hipóteses simplificadoras da seção 2.5 (iluminação uniforme, acabamento uniforme e albedo isotrópico) podemos retirar o albedo $a[p]$ de dentro da integral da equação de tonalização (2.12). Essa equação então pode ser decomposta em dois termos,

$$S_i[p] = a[p] L_i(\vec{n}[p], \vec{v}[p]) \quad (2.14)$$

onde L_i é a *função de tonalização*

$$L_i(\vec{n}, \vec{v}) = \int_{\mathbb{S}^2} \Phi_i(\vec{u}) \beta(\vec{n}, \vec{u}, \vec{v}) |\vec{n} \cdot \vec{u}| d\vec{u} \quad (2.15)$$

para todo $\vec{n}, \vec{v} \in \mathbb{S}^2$. Ou seja, a intensidade $S_i[p]$ de um pixel em um ponto p do domínio \mathcal{S} é o produto do albedo $a[p]$, que depende apenas das propriedades do material da superfície no ponto $s[p]$, e do *fator de tonalização* $L_i(\vec{n}[p], \vec{v}[p])$, que depende apenas do índice i , da direção de vista $\vec{v}[p]$ e da normal $\vec{n}[p]$ da superfície no ponto p .

A função de tonalização L_i resume o efeito do campo de iluminação Φ_i no acabamento β da superfície. Ela mapeia cada vetor unitário \vec{n} para a radiância que uma superfície branca (de albedo unitário) com este acabamento, perpendicular a \vec{n} , teria se fosse iluminada pelo fluxo Φ_i e se observada de uma direção qualquer \vec{v} .

Para luz unidirecional, com direção \vec{u}_i e intensidade E_i , a fórmula (2.15) reduz-se a

$$L_i(\vec{n}, \vec{v}) = 2\pi E_i \beta(\vec{n}, \vec{u}_i, \vec{v}) |\vec{n} \cdot \vec{u}_i| \quad (2.16)$$

2.6.1 Vetores de observação e tonalização

Definimos a *função vetor de tonalização* como sendo a função $\mathbf{L} : \mathbb{S}^2 \times \mathbb{S}^2 \rightarrow \mathbb{R}^m$ tal que $\mathbf{L}(\vec{n}, \vec{v}) = (L_1(\vec{n}, \vec{v}), \dots, L_m(\vec{n}, \vec{v}))$ para todo \vec{n} .

Com as hipóteses simplificadoras da seção 2.5, o problema fundamental do estéreo fotométrico passa a ser encontrar uma direção \vec{n} tal que o vetor de observação $\mathbf{S}[p]$ seja múltiplo do vetor $\mathbf{L}(\vec{n}, \vec{v}[p])$. Neste caso, podemos deduzir que a normal $\vec{n}[p]$ é essa direção \vec{n} , e que o albedo $a[p]$ é o fator de escala tal que

$$\mathbf{S}[p] = a[p] \mathbf{L}(\vec{n}, \vec{v}[p]) \quad (2.17)$$

Lembramos que a normal $\vec{n}[p]$ deve fazer ângulo agudo com a direção local de vista $\vec{v}[p]$. Denotaremos por $\mathbb{H}^2[p]$ o conjunto das direções que satisfazem esta condição, ou seja, o hemisfério de \mathbb{S}^2 centrado na direção $\vec{v}[p]$.

2.6.2 Assinaturas de observação e tonalização

Definimos a *assinatura de observação* $\mathbf{s}[p]$ como sendo a direção do vetor $\mathbf{S}[p]$ no espaço \mathbb{R}^m , ou seja

$$\mathbf{s}[p] = \frac{\mathbf{S}[p]}{|\mathbf{S}[p]|} \quad (2.18)$$

Nesta fórmula, o denominador $|\mathbf{S}[p]|$ é a norma euclidiana m -dimensional do vetor $\mathbf{S}[p]$, isto é $|\mathbf{X}| = \sqrt{\sum_{i=1}^m X_i^2}$. Da mesma forma, definimos a *função assinatura de tonalização* $\mathbf{l} : \mathbb{S}^2 \times \mathbb{S}^2 \rightarrow \mathbb{S}^m$ por

$$\mathbf{l}(\vec{n}, \vec{v}) = \frac{\mathbf{L}(\vec{n}, \vec{v})}{|\mathbf{L}(\vec{n}, \vec{v})|} \quad (2.19)$$

para todos $\vec{n} \in \mathbb{S}^2$ e $\vec{v} \in \mathbb{S}^2$ tais que $\vec{n} \cdot \vec{v} \geq 0$. Denotaremos as componentes da função \mathbf{l} por l_1, l_2, \dots, l_m . Podemos então reformular o problema fundamental do estéreo fotométrico como sendo encontrar $\vec{n} \in \mathbb{H}^2[p]$ tal que $\mathbf{s}[p] = \mathbf{l}(\vec{n}, \vec{v}[p])$.

2.7 Hipótese de invertibilidade

Mesmo com as hipóteses simplificadoras da seção 2.5, o problema fundamental do estéreo fotométrico é solúvel somente se a função \mathbf{l} é inversível no primeiro argumento; isto é, se para cada ponto \mathbf{s} de \mathbb{S}^{m-1} e cada direção de vista \vec{v} que pode ocorrer, existe no máximo uma direção \vec{n} tal que $\mathbf{l}(\vec{n}, \vec{v}) = \mathbf{s}$. O cálculo da normal em um pixel p é impossível se existirem duas normais \vec{n}', \vec{n}'' distintas em $\mathbb{H}^2[p]$ com mesmas assinaturas de tonalização $\mathbf{l}(\vec{n}', \vec{v}[p]) = \mathbf{l}(\vec{n}'', \vec{v}[p])$; ou seja, com vetores de tonalização $\mathbf{L}(\vec{n}', \vec{v}[p])$ e $\mathbf{L}(\vec{n}'', \vec{v}[p])$ colineares. A hipótese de que isto não acontece será chamada de *hipótese da invertibilidade*.

De modo geral, para que a hipótese de invertibilidade possa ser adotada, o acabamento β deve ser dominado por termos difusivos; ou seja, a luz incidente vinda de uma direção \vec{u} qualquer deve ser distribuída sobre um leque amplo de direções \vec{v} . Esta exigência geralmente exclui superfícies espelhadas. Para superfícies com acabamento puramente lustroso (como a da figura 2.10), garantir a hipótese de invertibilidade pode exigir um número de imagens muito grande, tanto maior quanto mais polida for a superfície, pois nestes casos o vetor $\mathbf{L}(\vec{n}, \vec{v})$ será essencialmente nulo para a maioria das normais \vec{n} . A invertibilidade também fica prejudicada se o albedo for muito baixo, pois neste caso o vetor $\mathbf{S}[p]$ será dominado pelos erros de medida, e a assinatura $\mathbf{s}[p]$ será bastante aleatória.

Em particular, para que o problema seja sempre solúvel, para uma dada direção de vista \vec{v} , não pode haver nenhuma região não-trivial de \mathbb{S}^2 onde a função \mathbf{l} seja constante. Mais geralmente, considere uma normal qualquer \vec{n} e sejam $\partial_0 \mathbf{L}(\vec{n}, \vec{v})$ e $\partial_1 \mathbf{L}(\vec{n}, \vec{v})$ as

derivadas de $\mathbf{L}(\vec{n}, \vec{v})$ em relação a duas perturbações de \vec{n} em direções perpendiculares entre si. A mesma notação aplica-se para $\partial_0 \mathbf{l}(\vec{n}, \vec{v})$ e $\partial_1 \mathbf{l}(\vec{n}, \vec{v}[p])$. Para que a função \mathbf{l} tenha inversa nas vizinhanças de $\mathbf{l}(\vec{n}, \vec{v})$, a matriz $2 \times m$

$$\begin{pmatrix} \partial_0 l_1(\vec{n}, \vec{v}) & \partial_0 l_2(\vec{n}, \vec{v}) & \dots & \partial_0 l_m(\vec{n}, \vec{v}) \\ \partial_1 l_1(\vec{n}, \vec{v}) & \partial_1 l_2(\vec{n}, \vec{v}) & \dots & \partial_1 l_m(\vec{n}, \vec{v}) \end{pmatrix} \quad (2.20)$$

deve ter posto 2. Em termos da função de tonalização \mathbf{L}_i , a condição é que a matriz $3 \times m$

$$\begin{pmatrix} \partial_0 L_1(\vec{n}, \vec{v}) & \partial_0 L_2(\vec{n}, \vec{v}) & \dots & \partial_0 L_m(\vec{n}, \vec{v}) \\ \partial_1 L_1(\vec{n}, \vec{v}) & \partial_1 L_2(\vec{n}, \vec{v}) & \dots & \partial_1 L_m(\vec{n}, \vec{v}) \\ L_1(\vec{n}, \vec{v}) & L_2(\vec{n}, \vec{v}) & \dots & L_m(\vec{n}, \vec{v}) \end{pmatrix} \quad (2.21)$$

tenha posto 3; ou seja, é necessário que existam pelo menos 3 imagens S_i, S_j, S_k tais que

$$\begin{vmatrix} \partial_0 L_i(\vec{n}, \vec{v}) & \partial_0 L_j(\vec{n}, \vec{v}) & \partial_0 L_k(\vec{n}, \vec{v}) \\ \partial_1 L_i(\vec{n}, \vec{v}) & \partial_1 L_j(\vec{n}, \vec{v}) & \partial_1 L_k(\vec{n}, \vec{v}) \\ L_i(\vec{n}, \vec{v}) & L_j(\vec{n}, \vec{v}) & L_k(\vec{n}, \vec{v}) \end{vmatrix} \neq 0 \quad (2.22)$$

No caso de fontes de luz aproximadamente pontuais, para garantir a hipótese de invertibilidade, é necessário (mas não suficiente) que, para cada ponto p de \mathcal{S} , existam pelo menos três imagens S_i, S_j e S_k nas quais o ponto $s[p]$ esteja iluminado por três fontes distintas; e que as direções dessas três fontes não estejam todas num mesmo plano.

Esta restrição significa, em particular, que o estéreo fotométrico dificilmente será viável com fotos de uma cena ao ar livre iluminada pelo sol em várias horas do dia, pois a trajetória aparente do sol em geral é próxima a um grande círculo da esfera \mathbb{S}^2 . Entretanto pode ser possível aplicar o método com esses dados se as fotos forem obtidas em latitudes elevadas, ou em épocas diferentes do ano, ou incluírem uma foto tirada com céu nublado.

2.7.1 Número mínimo de imagens

Tagare e Figueiredo [96] demonstraram que com iluminações unidirecionais, são necessárias no mínimo $2k$ imagens com iluminações distintas para garantir que cada ponto de um objeto genérico seja iluminado em no mínimo k dessas imagens.

Para explicar esta conclusão, diremos que uma fonte unidirecional na direção \vec{u} *ilumina* uma normal \vec{n} se $\vec{n} \cdot \vec{u} \geq 0$. Vamos denotar por $\mathbb{E}[p]$ a borda de $\mathbb{H}^2[p]$, isto é, o conjunto

das normais \vec{n} tais que $\vec{n} \cdot \vec{v}[p] = 0$. Observamos que uma iluminação unidirecional em qualquer direção diferente de $\vec{v}[p]$ ilumina apenas metade do conjunto $\mathbb{E}[p]$. Logo são necessárias duas luzes para iluminar $\mathbb{E}[p]$ inteiramente, e $2k$ luzes para iluminar cada normal de $\mathbb{E}(p)$ com k luzes.

O número $2k$ pode ser reduzido para $2k - 1$ se uma das luzes estiver posicionada bem próxima à câmera. Caso as fontes de luz sejam extensas, este número pode ser menor.

Este resultado diz, em particular que, com iluminações unidirecionais, são necessárias 6 imagens (ou 5 se uma das luzes estiver perto da câmera) com iluminação distinta para que a função \mathbf{l} seja inversível.

2.8 Métodos algébricos para cálculo de normais

O cálculo da normal (isto é, a inversão da função \mathbf{l}) pode ser feita por fórmulas explícitas em certos casos. Em particular quando a superfície é lambertiana e as fontes de luz são simples. Entretanto, este cálculo é muito mais difícil quando as direções e intensidades das luzes não são conhecidas ou o acabamento não é lambertiano. As abordagens para estes casos serão discutidas nos capítulos 3, 5 e 6.

2.8.1 Superfície lambertiana com iluminação unidirecional

O cálculo da normal, ou seja, a inversão da função \mathbf{l} , pode ser feito por métodos algébricos quando a superfície têm acabamento lambertiano e as fontes de luz são simples. Se o pixel p é completamente iluminado, em cada imagem i , por uma única fonte de luz unidirecional na direção \vec{u}_i , então o produto escalar $\vec{n} \cdot \vec{u}_i$ é positivo, a fórmula do acabamento (2.10) reduz-se a $\beta_L(\vec{n}, \vec{u}, \vec{v}) = 1/(2\pi)$, e a fórmula (2.16) reduz-se a

$$L_i(\vec{n}, \vec{v}) = L_i(\vec{n}) = (\vec{n} \cdot \vec{u}_i) E_i \quad (2.23)$$

onde \vec{u}_i é a direção da fonte luminosa da imagem i , e E_i é um coeficiente proporcional à sua intensidade. Note que a equação (2.23) não é válida se p está na região de sombra própria de alguma fonte de luz. Podemos então determinar a normal $\vec{n}[p]$ resolvendo o sistema linear

$$\begin{cases} (E_1 \vec{u}_1.x) \vec{a}.x + (E_1 \vec{u}_1.y) \vec{a}.y + (E_1 \vec{u}_1.z) \vec{a}.z = S_1[p] \\ (E_2 \vec{u}_2.x) \vec{a}.x + (E_2 \vec{u}_2.y) \vec{a}.y + (E_2 \vec{u}_2.z) \vec{a}.z = S_2[p] \\ (E_3 \vec{u}_3.x) \vec{a}.x + (E_3 \vec{u}_3.y) \vec{a}.y + (E_3 \vec{u}_3.z) \vec{a}.z = S_3[p] \end{cases} \quad (2.24)$$

nas incógnitas $\vec{a}.x, \vec{a}.y, \vec{a}.z$ onde \vec{a} é o *vetor de albedo* $a[p] \cdot \vec{n}[p]$. Uma vez determinado o vetor \vec{a} podemos calcular a partir dele o albedo $a[p] = |\vec{a}|$ e a normal $\vec{n}[p] = \vec{a}/|\vec{a}|$. Para

aplicar este método é necessário o conhecer a direção \vec{u}_i e o coeficiente de intensidade E_i de cada luz, que podem ser obtidos, por exemplo, com os métodos da seção 4.5.

O sistema (2.24) é indeterminado se as direções $\vec{u}_1, \vec{u}_2, \vec{u}_3$ estão todas no mesmo plano, pois nesse caso as linhas da matriz são linearmente dependentes. Além disso se o albedo $a[p]$ é zero, as radiâncias $S_i[p]$ serão nulas, a solução do sistema (2.24) será um vetor \vec{a} nulo e a normal $\vec{n}[p]$ não poderá ser calculada.

Podemos generalizar o sistema (2.24) para $m > 3$ iluminações unidirecionais. Supondo que o pixel p seja iluminado em todas elas, valem as equações

$$\begin{cases} (E_1 \vec{u}_1.x) \vec{a}.x + (E_1 \vec{u}_1.y) \vec{a}.y + (E_1 \vec{u}_1.z) \vec{a}.z = S_1[p] \\ (E_2 \vec{u}_2.x) \vec{a}.x + (E_2 \vec{u}_2.y) \vec{a}.y + (E_2 \vec{u}_2.z) \vec{a}.z = S_2[p] \\ \vdots \\ (E_m \vec{u}_m.x) \vec{a}.x + (E_m \vec{u}_m.y) \vec{a}.y + (E_m \vec{u}_m.z) \vec{a}.z = S_m[p] \end{cases} \quad (2.25)$$

Este é um sistema sobredeterminado (m equações e 3 incógnitas), cuja a solução \vec{a} pode ser determinada pelo critério de quadrados mínimos [81]. Por usar toda a informação disponível, este método fornece um resultado geralmente mais preciso que o sistema (2.24).

Esta abordagem não funciona bem se o ponto p está na sombra própria ou projetada de alguma das luzes. Para excluir essas iluminações, uma possibilidade é usar para cada ponto p apenas as imagens com os valores S_i/E_i mais altos.

2.8.2 Acabamento lambertiano com luzes extensas

Os sistemas (2.24) e (2.25) são válidos mesmo quando as fontes de luz são extensas, desde que cada fonte seja inteiramente visível do ponto $s[p]$. Como veremos na seção 4.5.2, nessas condições a função de tonalização é equivalente à de uma única iluminação unidirecional com direção média \vec{u}_i .

2.9 Estéreo fotométrico baseado em gabaritos

Nesta tese estudaremos a variante específica do estéreo fotométrico, denominada *estéreo fotométrico baseado em gabaritos* (EFBG) [106, 107, 108]. Nesta abordagem, as funções de tonalização L_i são extraídas de m imagens de um objeto de forma e cor conhecida, o *gabarito de iluminação*, obtidas sob as mesmas condições de iluminação das imagens correspondentes da cena S_1, S_2, \dots, S_m . É conveniente usar gabaritos esféricos com albedo uniforme, preferivelmente branco ou cinza claro, e seu acabamento deve ser similar ao da cena a ser reconstruída. Veja a figura 2.12

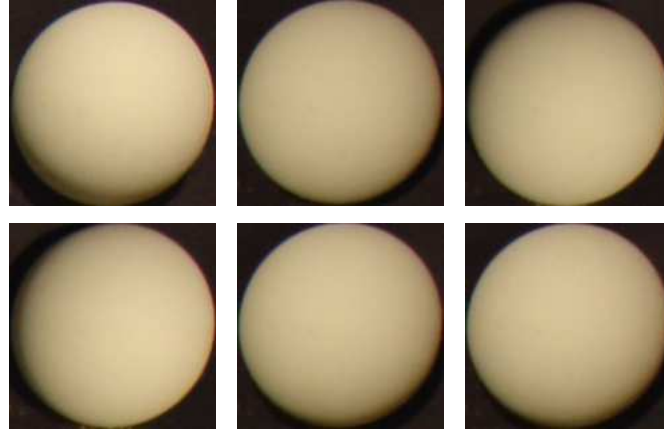


Figura 2.12: Imagens de gabaritos sob as mesmas condições de iluminação dos objetos da figura 2.1.

A justificativa para esta abordagem é que determinar e modelar diretamente a função de tonalização $L_i(\vec{n}, \vec{v})$ é mais simples do que determinar e modelar separadamente a FBDR do acabamento $\beta(\vec{n}, \vec{u}, \vec{v})$ e o fluxo de iluminação $\Phi_i(\vec{u})$. No aspecto teórico, além de ter menos parâmetros a determinar, a função L_i é uma convolução de Φ_i com β , que são ambas não negativas; portanto L_i é mais suave que essas duas funções separadamente. Em particular L_i só terá picos estreitos se tanto Φ_i quanto β tiverem picos estreitos (causados por fontes de luz pontuais e termos quase especulares, respectivamente). Por exemplo, se β é lambertiano e Φ_i é gerado por várias dezenas de fontes pontuais com direções espalhadas sobre \mathbb{H}^2 , então L_i pode ser aproximado com boa precisão por uma função suave com apenas 5—10 coeficientes a determinar.

No aspecto experimental, como veremos no capítulo 4, cada função de tonalização L_i pode ser determinada a partir de uma única foto do gabarito; enquanto que a determinação experimental da função β exige centenas de medições separadas com equipamento especializado.

Uma maneira simples de representar as funções L_i implícitas nas imagens do gabarito é construir uma *tabela de assinaturas*, uma lista de pares $(\vec{n}_k, \mathbf{l}_k)$ onde \vec{n}_k é uma normal e \mathbf{l}_k é a assinatura correspondente $\mathbf{l}(\vec{n}_k)$. O problema de inversão torna-se uma busca da entrada desta tabela que tem \mathbf{l}_k mais próxima a uma assinatura de observação \mathbf{s} dada.

Experimentos mostram que a abordagem EFBG é robusta mesmo em presença de acabamentos complicados e/ou fontes luminosas arbitrárias [52, 53]. Dentre as desvantagens da EFBG podemos enunciar a inconveniência de inserir um gabarito na cena, (ou de tirar um conjunto de m fotos separadas do mesmo); a dificuldade de construir gabaritos substancialmente precisos e o fato de que qualquer diferença entre o acabamento da cena e do gabarito introduz erros na normal calculada. Outra limitação desta abordagem é o fato de que são necessários diversos gabaritos quando a cena contém objetos de acabamento variado, por exemplo, mais ou menos lustrosos.

2.10 Estéreo fotométrico com luz policromática

Todos os conceitos vistos até agora pressupõem iluminação por luz monocromática (com um único comprimento de onda). Na prática, luz é quase sempre uma mistura de ondas eletromagnéticas com vários comprimentos de onda, que interagem com a cena e com a câmera de maneira diferente.

2.10.1 Equação de tonalização policromática

Para tratar luz policromática, precisamos acrescentar um parâmetro adicional λ , o comprimento de onda, a várias funções e fórmulas das seções anteriores. Especificamente:

- O fluxo de luz $\Phi[p](\vec{u})$ passa a ser $\Phi[p](\vec{u}, \lambda)$.
- A FBDR $\sigma[p](\vec{n}, \vec{u}, \vec{v})$, o albedo $a[p](\vec{n}, \vec{u})$ e o acabamento $\beta[p](\vec{n}, \vec{u}, \vec{v})$ passam a ser $\sigma[p](\vec{n}, \vec{u}, \vec{v}, \lambda)$, $a[p](\vec{n}, \vec{u}, \lambda)$ e $\beta[p](\vec{n}, \vec{u}, \vec{v}, \lambda)$, respectivamente.
- A função de tonalização $L_i[p](\vec{n}, \vec{v})$ passa a ser $L_i[p](\vec{n}, \vec{v}, \lambda)$ e o mesmo vale para as funções derivadas $\mathbf{L}(\vec{n}, \vec{v})$, $\mathbf{l}_i(\vec{n}, \vec{v})$ e $\mathbf{l}(\vec{n}, \vec{v})$.
- A radiância $S_i[p]$ da superfície no ponto p passa a ser $S_i[p](\lambda)$.

A equação (2.12) é então substituída pela *equação de tonalização policromática*

$$S_i[p](\lambda) = \int_{\mathbb{S}^2} \Phi_i[p](\vec{u}, \lambda) a[p](\vec{n}[p], \vec{u}, \lambda) \beta[p](\vec{n}[p], \vec{u}, \vec{v}[p], \lambda) |\vec{n}[p] \cdot \vec{u}| d\vec{u} \quad (2.26)$$

2.10.2 Imagens coloridas

Na grande maioria das câmeras, cada sensor mede simultaneamente radiação de vários comprimentos de onda, fornecendo uma única medida. Podemos supor que, em cada pixel p há um certo número n_c de sensores, cada qual sensível a uma certa faixa de comprimentos de onda (*canal de cor*). Tipicamente, $n_c = 1$ para imagens monocromáticas, e $n_c = 3$ para imagens coloridas no formato popular RGB. Denotaremos por $S_i[p, c]$ o valor medido pelo sensor do canal c no pixel p , sob iluminação Φ_i . Ou seja, cada imagem S_i fornece um vetor $S_i[p] = (S_i[p, 1], S_i[p, 2], \dots, S_i[p, n_c])$ de n_c valores numéricos. Cada valor é uma integral

$$S_i[p, c] = \int_0^\infty \tau[c](\lambda) S_i[p](\lambda) d\lambda \quad (2.27)$$

onde $\tau[c]$ é uma função não negativa de \mathbb{R} para \mathbb{R} , a *função de sensibilidade espectral* da câmera no canal c .

2.10.3 Hipóteses de uniformidade espectral

Com as hipóteses simplificadoras da seção 2.5 podemos denotar o fluxo de iluminação $\Phi[p](\vec{u}, \lambda)$ como $\Phi(\vec{u}, \lambda)$, o albedo $a[p](\vec{u}, \lambda)$ como $a[p](\lambda)$ e o acabamento $\beta[p](\vec{n}, \vec{u}, \vec{v}, \lambda)$ como $\beta(\vec{n}, \vec{u}, \vec{v}, \lambda)$. Temos então que

$$S_i[p, c] = \int_0^\infty \tau[c](\lambda) a[p](\lambda) \left[\int_{\mathbb{S}^2} \Phi_i(\vec{u}, \lambda) \beta(\vec{n}[p], \vec{u}, \vec{v}[p], \lambda) |\vec{n}[p] \cdot \vec{u}| d\vec{u} \right] d\lambda \quad (2.28)$$

Para separar estas integrais precisamos duas hipóteses simplificadoras adicionais.

A hipótese de *acabamento espectralmente uniforme* diz que o acabamento $\beta(\vec{n}, \vec{u}, \vec{v}, \lambda)$ não depende de λ , ou seja, a FBDR da superfície $\sigma[p](\vec{n}, \vec{u}, \vec{v}, \lambda)$ pode ser separada em dois fatores

$$\sigma[p](\vec{n}, \vec{u}, \vec{v}, \lambda) = a[p](\vec{n}[p], \vec{u}, \lambda) \beta(\vec{n}[p], \vec{u}, \vec{v}[p]) \quad (2.29)$$

sendo que o albedo a só depende de p , \vec{n} , \vec{u} e λ , e o acabamento β só depende de \vec{n} , \vec{u} e \vec{v} .

A hipótese de *luz espectralmente uniforme* diz que fluxo luminoso $\Phi_i[p](\vec{u}, \lambda)$ pode ser separado em dois termos: o *espectro da luz* $\zeta(\lambda)$ que só depende de λ (e não de i ou \vec{u}), e a *distribuição direcional da luz* $\Psi_i[p](\vec{u})$ que só depende de i , p e \vec{u} (mas não de λ). Ou seja, para cada pixel p , todas as m iluminações devem ter luz da mesma cor, determinada pelo espectro ζ , para qualquer direção de origem \vec{u} ; apenas a intensidade geral da luz $\Psi_i[p](\vec{u})$ pode variar com \vec{u} .

Com estas duas hipóteses adicionais podemos fatorar a integral (2.28) no produto de duas integrais

$$\begin{aligned} S_i[p, c] &= \left[\int_0^\infty \tau[c](\lambda) a[p](\lambda) \zeta(\lambda) d\lambda \right] \left[\int_{\mathbb{S}^2} \Psi_i(\vec{u}) \beta(\vec{n}[p], \vec{u}, \vec{v}[p]) |\vec{n}[p] \cdot \vec{u}| d\vec{u} \right] \\ &= a[p, c] L_i(\vec{n}[p], \vec{v}[p]) \end{aligned} \quad (2.30)$$

A primeira integral $a[p, c]$ é o *albedo efetivo no canal c* da superfície no ponto $s[p]$. A segunda integral $L_i(\vec{n}[p], \vec{v}[p])$ é a função de tonalização da imagem i dada pela equação (2.12); exceto que o fluxo Φ_i é substituído pelo seu fator Ψ_i .

Se as hipóteses de uniformidade espectral de iluminação e acabamento valem, podemos tratar cada canal separadamente como um problema de EFIV monocromático separado, obtendo um albedo diferente $a[p, c]$ para cada canal c , onde L_i é a função de tonalização monocromática (2.15) para o fluxo Ψ_i e acabamento β . Estes albedos descrevem a *cor intrínseca* ou *cor própria* da cena no ponto $s[p]$, independente da direção de iluminação

ou de vista (mas modificada pelo espectro da luz ζ e pela sensibilidade da câmera $\tau[c]$). Os valores de $a[p, c]$ podem ser apresentados como canais de cor de uma imagem com domínio \mathcal{S} , que denominaremos *mapa de cor intrínseca* da cena. Veja a figura 2.13.



Figura 2.13: Mapa de cor intrínseca $a[p, c]$, para $c = 1, 2, 3$ (R,G,B), obtido da cena da figura 2.1.

Quando valem as hipóteses de uniformidade espectral do acabamento e do fluxo luminoso, podemos também combinar linearmente os vários canais de cor para formar canais “sintéticos”, como por exemplo o canal de *luminância* $Y = 0.299 R + 0.587 G + 0.114 B$.

Este processo também fornece n_c estimativas $\vec{n}[p, c]$ da normal para cada pixel. Espera-se que todas estas normais sejam iguais ou muito próximas, e portanto podemos calcular a média dos vetores $a[p, c]\vec{n}[p, c]$, e normalizar essa média para obter uma única normal estimada $\vec{n}[p]$.

2.10.4 Hipótese de separação espectral

A rigor, as hipóteses de uniformidade espectral de luz e acabamento não precisam ser válidas para todo comprimento de onda λ , mas apenas para o conjunto $\Lambda[c]$ de valores de λ para os quais o produto de todos os fatores que dependem desta variável — a sensibilidade espectral $\tau[c](\lambda)$ do canal, o fluxo $\Phi_i[p](\vec{u}, \lambda)$ e a FBDR $\sigma[p](\vec{n}[p], \vec{u}, \vec{v}[p])$ — é significativo. Em particular, podemos supor que elas são satisfeitas se

1. Cada fluxo $\Phi_i[p](\vec{u}, \lambda)$ é significativo apenas para n_c comprimentos de onda específicos $\lambda^*[1], \lambda^*[2], \dots, \lambda^*[n_c]$.
2. Os sensores de cada canal medem apenas uma dessas componentes do fluxo; isto é, $\tau[c']\lambda^*[c'']$ é significativo apenas se $c' = c''$.

Estas condições podem ser satisfeitas usando-se fontes de luz monocromáticas, ou instalando-se filtros de banda estreita adequados em cada fonte de luz. Neste caso, as imagens podem

ser obtidas com uma câmera RGB comum. É possível garantir esta condição também com um filtro adequado na lente da câmera que deixe passar apenas n_c faixas estreitas centradas em $\lambda^*[1], \dots, \lambda^*[n_c]$.

Em qualquer destes dois casos, dizemos que vale a *hipótese de separação espectral*. Com ela, a integral (2.30) pode ser separada em

$$S_i[p, c] = a[p, c] L_i[c](\vec{n}[p], \vec{v}[p]) \quad (2.31)$$

onde $a[p, c]$ é o *albedo do canal*

$$a[p, c] = \tau[c](\lambda^*[c]) a[p](\lambda^*[c]) \quad (2.32)$$

e onde $L_i[c]$ é a *função de tonalização do canal c*

$$L_i[c](\vec{n}[p], \vec{v}[p]) = \int_{\mathbb{S}^2} \Phi_i(\vec{u}, \lambda^*[c]) \beta(\vec{n}, \vec{u}, \vec{v}, \lambda^*[c]) |\vec{n} \cdot \vec{u}| d\vec{u} \quad (2.33)$$

A diferença entre as equações (2.30) e (2.31) reside na função de tonalização, que na segunda é diferente para cada canal c .

Quando vale a hipótese de separação espectral, mas não as de uniformidade espectral de luz e albedo, ainda podemos aplicar os métodos do EFIV monocromático a cada canal, e combinar os resultados como descrito na seção 2.10.3; exceto que em cada canal precisamos usar uma função vetor de tonalização $\mathbf{L}[c]$ diferente. Entretanto, neste caso *não* é possível combinar os canais das imagens S_i (por exemplo, para trabalhar com o canal de luminância Y).

Nesse caso, por outro lado, podemos trabalhar com todos os canais simultaneamente. Para isso, definimos o *vetor de observação policromático* $\mathbf{S}[p] \in \mathbb{R}^{mn_c}$ como a concatenação dos n_c vetores de observação monocromáticos $\mathbf{S}[p, c]$, cada um com m amostras. Da mesma forma re-definimos a *função vetor de tonalização policromática* $\mathbf{L} : \mathbb{S}^2 \rightarrow \mathbb{R}^{mn_c}$ como sendo a concatenação das funções monocromáticas $(\mathbf{L}_1, \mathbf{L}[2], \dots, \mathbf{L}[n_c])$ dos n_c canais.

As *assinaturas policromáticas* $\mathbf{s}[p]$ e $\mathbf{l}[p]$ são obtidas a partir destes vetores policromáticos pela normalização separada de cada vetor monocromático. Portanto elas não pertencem ao espaço \mathbb{S}^{mn_c-1} , mas sim ao espaço $(\mathbb{S}^m - 1)^{n_c} \subseteq \mathbb{R}^{mn_c}$. Como no método monocromático, a normal da cena no ponto p será então o vetor \vec{n} que torna as assinaturas policromáticas $\mathbf{s}[p]$ e $\mathbf{l}(\vec{n})$ o mais próximas possível.

2.10.5 Estéreo fotométrico multi-espectral

Se vale a hipótese de separação espectral, podemos também usar a informação de cor para reduzir o tempo de captura das imagens. Nesta abordagem *multi-espectral*, proposta por

Woodham em 1994 [108], a cena é iluminada simultaneamente por n_c fontes de luz com cores distintas (por exemplo, vermelho, verde e azul) e posições bem distintas. Cada imagem colorida capturada essencialmente equivale a n_c imagens monocromáticas distintas. Esta técnica pode ser necessária em aplicações de tempo real, para garantir o alinhamento das imagens mesmo quando a cena contém objetos em movimento (Woodham supôs que a cena tinha acabamento e albedo uniformes, mas a separação espectral é suficiente).

Capítulo 3

Trabalhos Relacionados

3.1 Conceitos básicos

Os princípios básicos de estéreo fotométrico foram enunciados em 1980 por Woodham [106]. Ele supôs que a cena era iluminada por uma fonte de luz unidirecional ou estendida com propriedades conhecidas e que o acabamento era uniforme e lambertiano. Com estas suposições, ele demonstrou que é possível determinar a inclinação de um ponto da superfície a partir da variação do campo luminoso, desde que existam no mínimo 3 iluminações unidirecionais cujas direções não são co-planares (a hipótese da invertibilidade da seção 2.7).

Woodham também introduziu os conceitos de gabaritos de iluminação [106, 108] e de tabela de assinatura [106, 107]. Das imagens do gabarito, ele extraía vetores de observação $\mathbf{S}_k = \mathbf{L}(\vec{n}_k)$ associados a um conjunto de normais \vec{n}_k . O cálculo de \vec{n} para um \mathbf{S} qualquer reduz-se então à busca da entrada mais próxima nesta tabela.

Woodham também introduziu a abordagem multi-espectral [108] (seção 2.10.5), graças à qual ele conseguiu aplicar o estéreo fotométrico à razão de 15 quadros por segundo.

Nos anos seguintes, muitos pesquisadores generalizaram o trabalho de Woodham para materiais e condições de iluminação mais gerais ou desconhecidos. Neste capítulo relacionamos os principais trabalhos nesta área.

3.2 Métodos com função de tonalização conhecida

Uma grande classe de métodos propostos para EFIV supõe que a função vetor de tonalização \mathbf{L} é conhecida *a priori*. Em geral, estes métodos supõem as hipóteses simplificadoras do capítulo 2, em particular a do acabamento uniforme. Tipicamente, estes métodos tratam anomalias de iluminação ou acabamento procurando detectar, para cada pixel, as imagens S_i nas quais a função de tonalização é aproximadamente lambertiana e

a iluminação é total, e usam apenas essas imagens para determinar a normal $\vec{n}[p]$.

Vários desses métodos supõem que tanto o acabamento β quanto os fluxos de luz Φ_i são conhecidos, de modo que a função \mathbf{L} pode ser calculada pela fórmula (2.15). Em geral, eles supõem que cada função Φ_i é uma iluminação unidirecional. Dada a simplicidade da inversão da função analítica nestes casos, estes métodos são preferidos para aplicações onde o tempo de processamento é importante.

Coleman e Jai [59] generalizaram a abordagem de Woodham para objetos lustrosos. Seu método detecta e exclui as imagens S_i onde o pixel está afetado por termos lustrosos, e aplica às demais imagens a solução para acabamento lambertiano. Para garantir a hipótese da invertibilidade, eles supõem que a cena é iluminada por quatro fontes de luz unidirecionais com intensidades e direções conhecidas, posicionadas de tal forma que o reflexo lustroso esteja presente em no máximo uma das quatro imagens.

Basrky e Petrou [16] estenderam essa abordagem. Seu método trabalha com imagens coloridas, e utiliza consistência entre os três canais de cor para detectar imagens onde o pixel é afetado por realces lustrosos e sombras (próprias ou projetadas). Como na solução de Coleman e Jai, elas consideram apenas quatro imagens, sendo que no máximo uma está afetada por reflexos ou sombras. Se as imagens não são consistentes com a função de tonalização (lambertiana), o algoritmo examina cada subconjunto de três imagens.

O algoritmo de Yu *et al.* [112] estende esta abordagem para m imagens, tentando identificar o subconjunto máximo de imagens onde as radiâncias do pixel são compatíveis com um acabamento lambertiano excluindo sombras e realces. A identificação deste subconjunto é feita por programação linear inteira. Uma vez obtido este subconjunto, o albedo e a normal são determinados pela inversão algébrica da função de tonalização lambertiana usando mínimos quadrados, como descrito na seção 2.8.1.

Outros métodos desta classe não supõem conhecimento do acabamento e da iluminação, mas supõem que a função vetor de tonalização \mathbf{L} é obtida experimentalmente através de um gabarito de iluminação. Estes trabalhos incluem os de Hertzmann e Seitz [52, 53] e Ackerman e Goesele [3], e serão descritos no capítulo 4.

3.3 Métodos com função de tonalização desconhecida

Outra classe de trabalhos procura estender EFIV a situações onde a função vetor de tonalização \mathbf{L} não é completamente conhecida a priori. Em geral, elas supõem conhecimento apenas do acabamento ou apenas do fluxo luminoso, ou conhecimento parcial de ambos.

3.3.1 Abordagens com iluminação conhecida

Os trabalhos desta classe que exigem conhecimento dos fluxos Φ_i , supõem que a FBDR é arbitrária, ou é uma combinação linear arbitrária de um conjunto fixo de acabamentos — como lambertiano, lustroso (de Phong, Trowbridge-Reitz [101], Torrance-Sparrow [100] ou Ward [104]) ou lóbulos genéricos [96]. Muitos destes métodos supõem que a hipótese do acabamento uniforme não é válida, e tentam recuperar os parâmetros desconhecidos da FBDR separadamente para cada pixel. Essa etapa costuma ser complicada e requer frequentemente otimização não linear desses parâmetros.

Nesta sub-classe se inclui o método de Nayar *et al.* [74]. Eles usam um modelo de FBDR com três componentes (difuso, lustroso e especular) combinados linearmente, e um algoritmo capaz de determinar tanto os coeficientes dessa combinação quanto a inclinação da superfície. Eles recomendam fontes de luz extensas para uma determinação mais robusta do termo especular; uma vez que, com uma fonte pontual de direção \vec{u}_i , o termo é significativo apenas quando a normal $\vec{n}[p]$ é muito próxima da bissetriz \vec{m} de \vec{u}_i e $\vec{v}[p]$. O algoritmo de Nayar *et al.* supõe que as fontes luminosas têm a mesma intensidade e formato e estão dispostas de tal maneira que, em cada pixel, os termos lustrosos e especulares da FBDR estão presentes apenas em poucas imagens. Para determinar a normal \vec{n} , eles usam as imagens onde os termos lambertianos dominam; os coeficientes da FBDR são então determinados por um algoritmo de otimização não-linear. Seu método obtém resultados razoáveis para objetos feitos de material lambertiano, plástico liso ou mesmo metal lixado, utilizando em torno de $m = 12$ fontes luminosas.

Kay *et al.* [61] supõem um modelo de acabamento baseado no modelo de Torrance-Sparrow com grau de polimento fixo no termo lustroso. Eles estendem a abordagem de Nayar *et al.*, eliminando a restrição de fontes luminosas idênticas; porém, eles exigem uma quantidade muito maior de iluminações que as abordagens acima ($m \approx 50$ ou mais), todas com direções próximas à direção de vista $\vec{v}[p]$.

Uma abordagem bastante singular nesta classe é a de Magda *et al.* [69]. Eles utilizam um conjunto de luzes pontuais idênticas com posições conhecidas, relativamente próximas à cena. Eles observaram que é possível determinar diretamente a profundidade do ponto $s[p]$ da cena comparando as radiâncias $S_i[p]$ e $S_j[p]$ obtidas com duas luzes colineares com esse ponto, usando o fato de que a radiância da superfície em $s[p]$ é inversamente proporcional ao quadrado da distância de $s[p]$ à fonte de luz. Como a altura de $s[p]$ é desconhecida, eles recorrem a um método iterativo para identificar os pares de luzes que são colineares com esse ponto. Este método independe da FBDR do pixel, não exige a hipótese de acabamento uniforme e é imune a sombras projetadas. Além disso ele dá diretamente a altura da superfície, e não apenas sua normal. Porém, ele exige um número elevado de fontes luminosas ($m \approx 150$) para garantir a existência de pares colineares e obter precisão aceitável no cálculo da profundidade.

Shen *et al.* [88] supõem que o acabamento combina um termo lambertiano com uma componente lustrosa de Ward, de magnitude desconhecida mas uniforme sobre toda a cena; e que cada iluminação é unidirecional com direção e intensidade conhecidas. Seu método utiliza a técnica de Barsky e Petrou *et al.* [16] para determinar um subconjunto de imagens nos quais há pelo menos 3 iluminações livres de sombras onde o termo lustroso é desprezível e usam essas imagens para determinar a normal. A componente lustrosa é então determinada por otimização não linear. No final, as normais dos demais pixels são então recalculadas levando este termo em conta.

Chen *et al.* [27] propõem um método específico para objetos cuja FBDR contém uma componente especular (ou lustrosa de alto polimento). Em contraste com a maioria dos outros métodos, eles ignoram as componentes difusivas da FBDR e utilizam apenas essa componente especular. Supondo fontes de luz pontuais com direções $\vec{u}_1, \vec{u}_2, \dots, \vec{u}_m$ conhecidas, eles detectam os pontos de maior brilho em cada imagem S_i , e estimam o valor da normal supondo que esse ponto é próximo da bissetriz \vec{m} de \vec{u} e \vec{v} . Eles demonstraram esta abordagem em superfícies com comportamento fotométrico muito complexo como casca de laranja, balas de goma, pele humana e chocolate. A desvantagem deste método é a necessidade de um número elevado ($m \approx 200$) de imagens para obter resultados com precisão aceitável.

Higo [55] não modela a FBDR explicitamente, apenas supõe que a função de tonalização é monotônica no produto escalar $\vec{n} \cdot \vec{u}$ e que cada iluminação é unidirecional com direção conhecida. Esta hipótese determina que a $\mathbf{L}(\vec{n}[p])$ deve ser função crescente do produto escalar $\vec{n} \cdot \vec{u}$. Note que esta hipótese implica que o albedo é uniforme, e exclui materiais lustrosos. Higo então constrói uma função energia $E(\vec{n}, a)$ que mede o grau de violação desta condição e determina a normal $\vec{n}[p]$ do pixel por otimização não-linear. Apesar da hipótese restritiva, o método funciona também para objetos moderadamente lustrosos, se o número de imagens for grande.

Alldrin *et al.* [8] usam um modelo de FBDR semelhante ao de Tagare e Figueiredo [96] (vide seção 3.3.2), ou seja, uma combinação linear de um conjunto fixo de k funções FBDRs pré-definidas. Eles também supõem iluminação por fontes unidirecionais de intensidades e direções conhecidas. Ao contrário de Tagare e Figueiredo, eles não impõem condições sobre o posicionamento das fontes luminosas. Para determinar as normais e os coeficientes lineares das FBDR eles utilizam otimização não-linear [75]. Eles ilustram o método com um modelo com $k = 3$ FBDRs e $m \approx 100$, reconstruindo vasos lustrosos e frutas.

Goldman *et al.* [42] propõem uma variante do método de Alldrin *et al.* para o caso especial em que o acabamento de cada objeto da cena é uma combinação linear de apenas duas dentre as k FBDRs possíveis. Assim, eles testam todos os $\binom{k}{2}$ pares de FBDRs e escolhem o par que oferece a melhor solução. Para cada par, a normal e os coeficientes da combinação de FBDRs são determinados alternadamente por otimização quadrática

não linear. O método não é muito robusto e pode necessitar ajuda do usuário para resolver certas ambiguidades. Eles exemplificaram seu método com $m = 12$ imagens e $k = 3$ FBDRs para objetos como folhas, frutas, artefatos de cerâmica e plástico.

McGunnigle [71] supõe que a FBDR é combinação linear de um termo lambertiano e outro especular, e que a iluminação é fornecida por 4 fontes de luz extensas, especialmente projetadas para que cada campo de iluminação $\Phi_i(\vec{u})$ seja função linear das coordenadas de \vec{u} . Nestas condições, McGunnigle demonstra que, tanto para objetos lambertianos quanto completamente polidos, a função de tonalização $L_i(\vec{n})$ e portanto a radiância $S_i[p]$ depende linearmente de $\vec{n}.x$ e $\vec{n}.y$. A partir desta observação, ele apresenta um algoritmo capaz de determinar as componentes $\vec{n}.x$ e $\vec{n}.y$ aumentadas de um fator arbitrário desconhecido. Sua abordagem, no entanto é incapaz de determinar o albedo da cena; e o fator multiplicativo de $\vec{n}.x$ e $\vec{n}.y$ deve ser estimado por algum método *ad-hoc*. Apesar destes problemas, os resultados apresentados pelo autor são bastante razoáveis, e o tempo de processamento é bem reduzido.

3.3.2 Abordagens com iluminação desconhecida

Outra sub-classe dos métodos para tonalização desconhecida supõe que o acabamento da superfície é conhecido, mas que a forma e direção das fontes de luz não são conhecidas e/ou que a hipótese da iluminação uniforme não é garantida.

Uma limitação inerente da maioria destes métodos é a *ambiguidade de baixo-relevo*: sem saber a direção da fonte luminosa, e na ausência de sombras projetadas, é impossível distinguir concavidades de saliências; (mais precisamente, distinguir entre as normais (n_x, n_y, n_z) e $(-n_x, -n_y, n_z)$). A resolução desta ambiguidade exige alguma informação adicional, para cada pixel.

Tagare e Figueiredo [96] estenderam a prova da invertibilidade de Woodham para superfícies com acabamento mais geral. Eles modelam a FBDR por uma combinação linear fixa e conhecida de k termos (*lóbulos de espalhamento*) que podem ser difusivos, especulares ou constantes. Este modelo inclui os acabamentos descritos pelos modelos lambertiano, de Phong e Torrance-Sparrow. Eles supõem que cada iluminação é fornecida por uma combinação de fonte de luz unidirecional, de direção conhecida e intensidade desconhecida E_1 e de uma iluminação isotrópica (“luz ambiente”) de intensidade desconhecida E_2 ; sendo que os parâmetros E_1 e E_2 são os mesmos para todas as imagens. Eles demonstram que, nestas condições é possível determinar a normal da superfície e as intensidades E_1 e E_2 a partir de três imagens com fontes luminosas não co-planares.

Hayakawa [49] introduziu uma técnica interessante, baseada na *decomposição por valores singulares* (DVS) de uma matriz [81]. Ele considera a matriz $P \in \mathbb{R}^{N \times m}$, onde $P_{ij} = S_j[p_i]$ e p_1, p_2, \dots, p_N são os pixels totalmente iluminados por todas as m luzes.

Hayakawa demonstrou que, se a cena possui acabamento lambertiano, a matriz P tem essencialmente posto 3. Portanto ela pode ser aproximada por um produto da forma UDV , onde U é uma matriz $N \times 3$, D é uma matriz diagonal 3×3 e V é uma matriz $3 \times m$. Para obter esta aproximação, Hayakawa obtém a DVS $U'D'V'$ da matriz P , e toma as submatrizes que correspondem aos três maiores valores singulares de P . Cada linha i de U é a normal $\vec{n}[p_i]$ multiplicada pelo albedo $a[p_i]$, em algum sistema de coordenadas (não necessariamente ortonormal). Este sistema de coordenadas depende de maneira complicada da distribuição espacial das fontes de luz. Para determiná-lo, Hayakawa supôs que a intensidade das fontes de luz era conhecida em no mínimo 6 das m imagens. Com esta informação ele mostrou que é possível extrair da matriz V as direções das fontes luminosas (Hayakawa também afirmou que o sistema de coordenadas pode ser determinado conhecendo-se o albedo em 6 pixels; porém esta afirmação foi desprovada por Yuile [114]). Para usar o método de Hayakawa, é necessário eliminar a linha i e/ou a coluna j de P sempre que $S_i[p_j]$ estiver afetado por sombras próprias e/ou projetadas, ou por realces lustrosos. Para eliminar os pixels sombreados, ele eliminava elementos P_{ij} abaixo de um determinado limiar. A eliminação de pixels com realce infelizmente não pode ser feita *a priori*. Ele demonstrou a eficácia de seu método para vários objetos lambertianos usando com $m = 64$ imagens.

Yuile *et al.* [114, 113] estenderam a abordagem de Hayakawa. Eles supõem uma função de tonalização mais complexa, que pode ser descrita por uma combinação linear desconhecida de k (entre 3 e 7) funções da normal \vec{n} . Para obter as normais eles usam uma aproximação UDV da matriz P que corresponde aos k maiores valores singulares, em vez de três. Esta abordagem permite acabamento parcialmente lambertiano e parcialmente lustroso, mas como no método de Hayakawa, as normais obtidas são expressas em um sistema de coordenadas desconhecido. Yuille *et al.* também propuseram uma metodologia mais robusta para a detecção de sombras (sejam próprias ou projetadas) tratando estes dados como *outliers*. Utilizando 70 iluminações distintas, eles conseguiram recuperar a geometria objetos semilambertianos como um capacete de motociclista e faces humanas.

Basri e Jacobs [17, 18] estenderam a abordagem de Hayakawa de maneira diferente, sem impor restrições sobre a uniformidade das iluminações Φ_i , ou sobre o formato das fontes luminosas. Eles modelam a função de tonalização com harmônicos esféricos [73] de ordem máxima $M = 2$, determinando as normais da superfície por meio de uma decomposição UDV da matriz P com 9 valores singulares principais. Eles obtiveram resultados satisfatórios com 10 a 32 imagens. Este método pode ser aplicado a pixels com sombra própria (com certa perda de precisão), mas não funciona com sombras projetadas e realces lustrosos acentuados.

3.4 Outros trabalhos

Nesta seção discutiremos trabalhos que se destacaram mais pelas aplicações do que pelos algoritmos usados.

Funk *et al.* [38] tiveram a ideia de utilizar o monitor de um computador como fonte de iluminação variável para estéreo fotométrico. Eles utilizaram um monitor LCD e uma câmera digital controlada por computador. Para reconstrução, Funk utilizou o método algébrico da seção 2.8.1, supondo iluminações unidirecionais e acabamento lambertiano. Cada iluminação era produzida por um pequeno quadrado branco com fundo negro, exibido em uma posição diferente da tela. Devido à baixa intensidade das duas fontes de luz, a aquisição de imagens era demorada (em torno de 30 segundos por imagem) e teve que ser feita em uma cabine fechada para eliminar toda luz ambiente.

Esta ideia foi elaborada por Clark [29] em 2008, que eliminou a suposição de fontes unidirecionais. O algoritmo de Clark supõe fontes luminosas retangulares. Cada luz era na verdade proveniente de uma imagem arbitrária exibida no monitor; em uma etapa de pré-processamento, cada imagem era analisada para determinar as dimensões e direções de uma fonte retangular aproximadamente equivalente. Esta abordagem permite o uso de padrões luminosos com grande luminosidade total e captura em tempo relativamente pequeno. Entretanto, o seu método requer o conhecimento da posição do objeto analisado em relação ao monitor. Ele dá bons resultados apenas para objetos pequenos, e não funciona na presença de sombras, nem mesmo próprias.

Pintus *et al.* [80] propuseram a utilização de estéreo fotométrico para analisar imagens obtidas por microscópios eletrônicos de varredura. Embora o processo físico de formação das imagens seja completamente diferente, as fórmulas das intensidades $S_i[p]$ são matematicamente similares às do EFIV. Especificamente, as funções de tonalização são muito similares às de uma superfície lambertiana com iluminação unidirecional. Os autores mencionam, inclusive, a existência na função de tonalização de efeitos análogos aos realces lustrosos e sombras projetadas e próprias. Em seu trabalho, eles apresentam resultados com alto grau de detalhamento na reconstrução de objetos como moedas, chips de silício e cartuchos usando quatro diferentes “iluminações”.

Vogiatzis *et al.* [51] usaram EFIV para capturar o movimento e a geometria tridimensional de objetos deformáveis sem textura (como por exemplo, uma roupa branca). Eles utilizam estéreo fotométrico multi-espectral para capturar as normais da cena e, através de integração, criar uma malha de triângulos que descreve sua geometria. Feito isso, eles utilizam técnicas de fluxo ótico para determinar a deformação da malha nos quadros subsequentes.

Higo *et al.* [54] apresentaram um protótipo de câmera para captura de cenas tridimensionais que combina EFIV com estéreo geométrico, usando várias fotos da cena com

diferentes perspectivas. Sua câmera possui uma lâmpada LED de alta intensidade com posição conhecida relativa à lente da câmera, que é a principal fonte de iluminação em cada imagem. Usando técnicas de estéreo geométrico, eles determinam a posição da câmera, e portanto da fonte de luz em cada imagem. Com estes dados, eles utilizam estéreo fotométrico para calcular as normais da cena, supondo acabamento lambertiano. Nos seus testes, eles usaram até $m \approx 65$ imagens.

Outros artigos que descrevem aplicações de EFIV para problemas reais incluem [94, 86, 72].

Uma aplicação de destaque é a captura da geometria facial [40, 48, 102], que será discutida com maior detalhe no capítulo 9.

3.5 Sumário

Uma conclusão que tiramos desta pesquisa bibliográfica é que ainda não existe uma solução geral satisfatória para o problema de estéreo fotométrico de iluminação variável. Todas as soluções encontradas na literatura impõem restrições significativas sobre a natureza da superfície e/ou sobre das condições de iluminação. A tabela 3.1 resume estas exigências.

Optamos nesta tese pela abordagem de estéreo fotométrico baseado em gabaritos. Acreditamos que esta abordagem permite tratar cenas com uma grande gama de materiais sem exigir um número excessivo de iluminações ou conhecimento da FBDR, que são limitações comuns dos métodos EFIV que não usam gabarito. Dada sua simplicidade, esta abordagem pode ser utilizada para EFIV em tempo real. Sua maior fraqueza, a necessidade de colocar-se um gabarito na cena, nem sempre é proibitiva; pois em muitos casos um conjunto único de imagens de gabaritos pode ser usado para inúmeras cenas (vide capítulo 9).

Tabela 3.1: Dados exigidos por vários métodos de estéreo fotométrico, em ordem cronológica. O parâmetro m é o número de imagens usado pelos autores nos seus testes.

Ano	Método	Φ	β	L	m	EFBG
1980	Woodham [106, 107]	Parcial	Parcial	Total	3	Sim
1990	Nayar <i>et al.</i> [74]	Total	Parcial	-	12	-
1991	Tagare e Figueiredo [96]	Parcial	Total	-	6	-
1992	Coleman e Jai [59]	Total	Total	Total	4	-
1994	Woodham [108]	-	-	Total	3	Sim
1994	Hayakawa [49]	-	Total	-	64	-
1995	Kay <i>et al.</i> [61]	Total	Parcial	-	50	-
1997	Yuille <i>et al.</i> [114, 113]	-	Parcial	-	70	-
2001	Magda <i>et al.</i> [69]	Total	-	-	143	-
2003	Barsky e Petrou [16]	Total	Total	Total	4	-
2003	Hertzmann <i>et al.</i> [52, 53]	-	-	Parcial	3	Sim
2005	Shen <i>et al.</i> [88]	Total	Parcial	Parcial	4	-
2006	Chen <i>et al.</i> [27]	Total	Parcial	-	35	-
2007	Basri <i>et al.</i> [18]	-	Total	-	10	-
2008	Alldrin <i>et al.</i> [8]	Total	Parcial	-	100	-
2008	Schindler [87]	-	Parcial	-	3	-
2010	Higo <i>et al.</i> [55]	Total	Parcial	-	50	-
2010	McGunnigle [71]	Total	Total	-	4	-
2010	Acckerman <i>et al.</i> [3]	-	-	Parcial	20	-
2010	Goldman <i>et al.</i> [42]	Total	Parcial	-	10	-
2010	Yu <i>et al.</i> [112]	Total	Total	Total	15	-

Capítulo 4

Modelagem dos gabaritos

Neste capítulo discutimos a etapa preliminar de qualquer método EFBG, ou seja, a captura, pré-processamento das imagens do gabarito e a representação das funções de tonalização extraídas dessas imagens.

4.1 Trabalhos relacionados

Como observamos, a ideia de usar um gabarito para medir experimentalmente a função vetor de tonalização \mathbf{L} foi introduzida por Woodham em 1980 [106, 108]. Woodham demonstrou com sucesso a aplicação desta técnica [106] reconstruindo objetos de cerâmica iluminados por fontes de luz estendidas. Para conseguir efetuar este cálculo em tempo real [108], ele quantizou as amostras $S_i[p]$ em um número pequeno b de bits. Desta forma ele transforma cada vetor de observação $\mathbf{S}[p]$ do gabarito nos índices de uma entrada de uma matriz de dimensão m com 2^{mb} entradas, na qual está armazenada a normal \vec{n} correspondente. No entanto, o tamanho desta matriz limita esta abordagem a um número pequeno de iluminações.

Hertzmann e Seitz [52, 53] estenderam o EFBG para acabamentos que são uma combinação linear variável de 2 ou mais acabamentos fundamentais $\beta_1, \beta_2, \dots, \beta_k$ utilizando um gabarito separado para cada acabamento fundamental β_i . Com esta técnica, eles demonstram que é possível reconstruir cenas com FBDR complexa tais como objetos pintados com tinta lustrosa e esculturas com diversos acabamentos, a partir de $m \approx 30$ imagens, utilizando um gabarito lustroso e um lambertiano.

Ackerman e Goesele [3] oferecem uma alternativa para o problema da inconveniência de colocar-se de um gabarito na cena, visando usar EFBG em cenas ao ar livre. Eles recorrem a técnicas de estéreo geométrico para determinar a geometria de determinadas partes da cena e utilizam estas partes como gabaritos para estimar a função de tonalização \mathbf{L} . Em geral, esta abordagem não oferece uma cobertura completa do hemisfério \mathbb{H}^2 , então

eles usam interpolação para preencher as lacunas.

4.2 As imagens do gabarito

Cada imagem do gabarito de iluminação deve ser obtida sob as mesmas condições de iluminação da imagem da cena correspondente. Denotaremos por \mathcal{G} o conjunto de pixels cobertos pela projeção dos gabaritos nessas imagens. Dependendo da aplicação é conveniente incluir o gabarito de iluminação como parte da cena; neste caso, \mathcal{G} passa a ser um subconjunto de \mathcal{S} . Veja figura 4.1.



Figura 4.1: Incluindo o gabaritos de iluminação como parte da cena.

Em todo caso, vamos supor que, assim como as imagens da cena, as imagens do gabarito também estão perfeitamente alinhadas e usam codificação linear das intensidades luminosas. Como no capítulo 2, para cada ponto p de \mathcal{G} denotaremos por $s[p]$ o ponto da superfície do gabarito que projeta em p ; por $S_i[p]$ a intensidade da imagem i do gabarito neste ponto sob a iluminação Φ_i ; e por $\vec{n}[p]$ e $\vec{v}[p]$ a normal e a direção de vista em $s[p]$.

4.2.1 Qualidade das imagens

Imagens de gabaritos reais apresentam quase sempre ruído proveniente de diversas fontes. Imperfeições da superfície do gabarito (perceptíveis ou não a olho nu) causam desvios da normal da superfície em relação às normais da esfera ideal, e portanto têm assinaturas de iluminação incorretas para esta última. Ruído de medição, ajustes do nível de preto, algoritmos impróprios de debayerização [46] e de erros de quantização adicionam valores espúrios às assinaturas, especialmente em regiões de luminosidade baixa. Além disso, suposições incorretas sobre o tamanho e posição do gabarito nas imagens resultam em assinaturas espúrias para pixels que na verdade estão fora do gabarito, e associam normais incorretas às demais assinaturas. Finalmente, ao longo da borda do gabarito a amostragem efetuada pelo sensor mistura cores do objeto com cores do plano de fundo, e pode ser afetada por filtros de realce de bordas (ativados automaticamente em muitas câmeras fotográficas). Veja a figura 4.2.

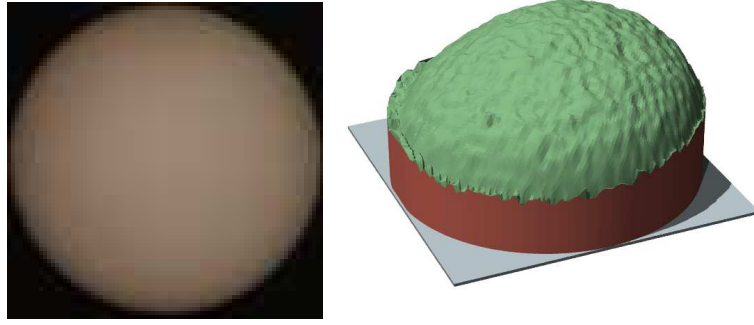


Figura 4.2: Imagem de um gabarito real recortada, e o gráfico tridimensional das intensidades luminosas dos pixels.

Neste capítulo descreveremos algoritmos para reduzir esses erros. A ideia geral é modelar a função de tonalização L_i (seção 2.1) de cada imagem do gabarito por uma aproximação matemática \tilde{L}_i . Idealmente esta aproximação exclui boa parte dos erros de amostragem e/ou defeitos de forma no gabarito. Esta aproximação permite também interpolar ou extrapolar a função L_i para as partes de \mathbb{S}^2 onde as imagens do gabarito não fornecem informação direta sobre a mesma. A lista das funções aproximantes $\tilde{L}_1, \tilde{L}_2, \dots, \tilde{L}_n$ constitui então um *gabarito virtual* \tilde{L} , que pode ser usado no lugar do gabarito real nos algoritmos de EFBG.

No restante deste capítulo vamos omitir o índice i de L_i , pois o problema pode ser tratado independentemente para cada iluminação i . Ou seja, vamos considerar o problema de determinar uma função de tonalização $L(\vec{n}) : \mathbb{S}^2 \rightarrow \mathbb{R}$ genérica a partir da imagem correspondente S do gabarito.

Esta imagem fornece um conjunto de pares (\vec{n}_k, l_k) , $k = 1 \dots N$, onde $\vec{n}_k = \vec{n}[p_k]$ é a normal da superfície do gabarito correspondente a um ponto p_k na projeção \mathcal{G} do mesmo; e l_k é o valor medido de $L(\vec{n}_k)$ — isto é, $S[p_k]/a[p_k]$. Nosso problema é determinar a aproximação $\tilde{L} : \mathbb{S}^2 \rightarrow \mathbb{R}$ de modo que os valores calculados $\tilde{L}(\vec{n}_k)$ sejam o mais próximo possível aos valores medidos l_k , por algum critério.

4.2.2 Critério dos quadrados mínimos

Para comparar o modelo \tilde{L} com os dados (l_k, \vec{n}_k) , podemos utilizar o critério dos *quadrados mínimos* (ou *mínimos quadrados*), que consiste em minimizar o *erro quadrático total* nos pontos de amostragem:

$$Q(\tilde{L}) = \sum_{k=1}^N w_k \left[l_k - \tilde{L}(\vec{n}_k) \right]^2 \quad (4.1)$$

Nesta fórmula, o fator w_k é um *peso* não negativo que indica a confiabilidade da medida de l_k . Em princípio, se o erro de medida em cada amostra l_k for independente dos erros

nas outras amostras, o peso w_k deveria ser o inverso da variância estimada desse erro. Neste trabalho usamos peso $w_k = 1$ para todas as amostras válidas.

4.3 Geometria de projeção do gabarito

4.3.1 Normais amostradas pelo gabarito

Cada imagem i do gabarito fornece dados sobre a função de tonalização $L_i(\vec{n})$ apenas para as normais \vec{n} que correspondem à parte visível do mesmo. Se o tamanho do gabarito é comparável com a distância da câmera ao gabarito, o conjunto de normais amostrado cobre uma calota menor que um hemisfério \mathbb{S}^2 e além disso esse conjunto varia conforme a posição do gabarito na imagem, pois a direção de vista $\vec{v}[p]$ varia com o ponto p .

Entretanto, em situações típicas (incluindo todos os testes da tese) os gabaritos são muito menores que a cena, de modo que a direção $\vec{v}[p]$ pode ser suposta constante sobre todo o gabarito. Especificamente, em todos os testes a distância da câmera ao gabarito foi maior ou igual a 1 metro e o diâmetro do gabarito menor ou igual a 0,04 metros, o que significa que a direção \vec{v} varia menos que 3° de um ponto a outro do gabarito. Esta variação é menor que o erro nas normais calculadas que é devido a outros fatores.

Com esta observação podemos aproximar a projeção de perspectiva restrita ao gabarito por uma projeção paralela oblíqua e supor que a direção de visão $\vec{v}[p]$ é constante no gabarito e igual à direção do centro do gabarito para o centro ótico da câmera. O conjunto das normais amostradas pelo gabarito é então o hemisfério \mathbb{H}^2 de \mathbb{S}^2 centrado na direção \vec{v} . Veja a figura 4.3.

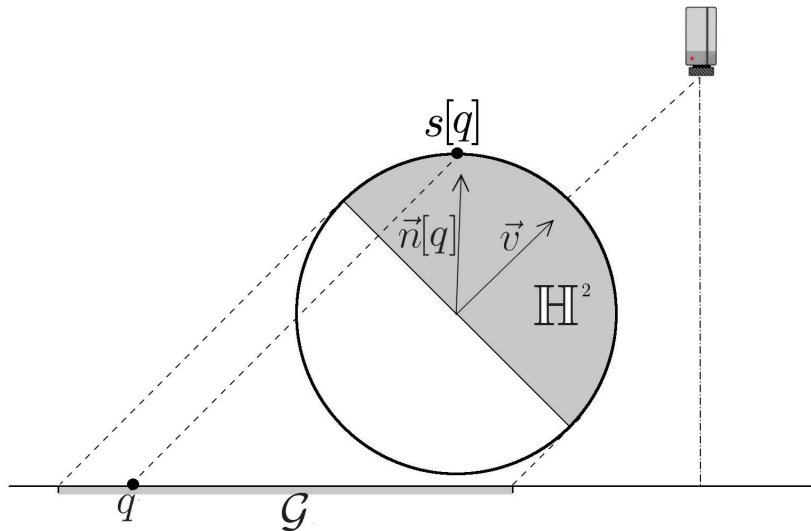


Figura 4.3: O conjunto \mathbb{H}^2 e a aproximação local paralela oblíqua para a projeção em perspectiva com a câmera distante.

4.3.2 Sistema de coordenadas do gabarito

Para cálculos referentes ao gabarito convém usar um sistema de *coordenadas do gabarito*, cujo eixo z é a direção de vista \vec{v} e cujos eixos x e y são direções arbitrárias ortogonais entre si e a \vec{v} .

A conversão entre este sistema de coordenadas e o sistema de coordenadas da câmera 2.1.2 é dada por uma matriz de rotação $M_v = (AR^T)A^T$, onde

$$A = \frac{1}{h^2} \begin{pmatrix} \vec{v}.x & 0 & -\vec{v}.y \\ \vec{v}.y & 0 & \vec{v}.x \\ 0 & h & 0 \end{pmatrix}$$

$$R = \begin{pmatrix} \vec{v}.z & -h & 0 \\ h & \vec{v}.z & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.2)$$

$$h = \sqrt{(\vec{v}.x)^2 + (\vec{v}.y)^2}$$

No caso especial em que $\vec{v}.x = \vec{v}.y = 0$, M_v é a matriz identidade.

4.3.3 Forma da projeção do gabarito

Na aproximação paralela, a projeção de um gabarito esférico é uma elipse cujo eixo maior está alinhado com o centro ótico F da imagem. Os parâmetros desta elipse são o centro c , o raio menor r e o vetor de alongamento \vec{e} . Veja a figura 4.4. Se $\vec{e} = (0, 0)$ a projeção é um círculo de centro c e raio r ; caso contrário, o raio maior da elipse tem a direção de \vec{e} e comprimento $r + |\vec{e}|$.

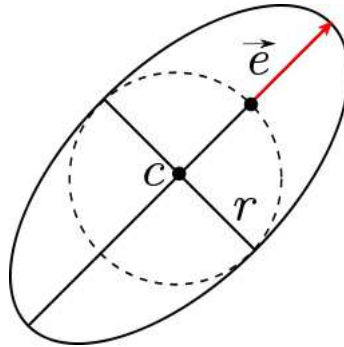


Figura 4.4: Elipse e seus parâmetros c, r, \vec{e} .

Para esta tese, desenvolvemos um programa que determina semi-automaticamente os parâmetros c , r e \vec{e} nas imagens do gabarito. O programa tem uma interface gráfica

que permite ao usuário indicar com o *mouse* os valores aproximados de c , r e \vec{e} . Estes parâmetros são então ajustados iterativamente por um método de otimização multi-escala [92] usando a técnica de gradiente normalizado [115] para obter os contornos do gabarito. Veja a figura 4.5.

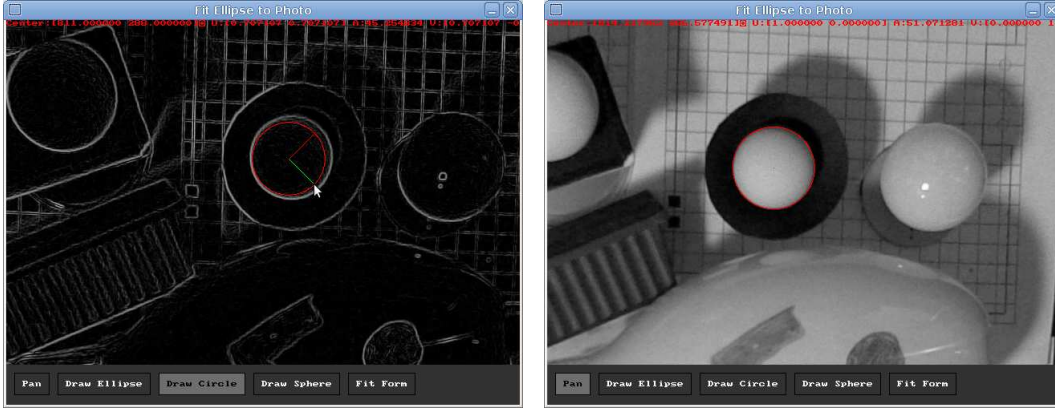


Figura 4.5: Usuário fornecendo estimativa de c , r e \vec{e} na interface gráfica do software (esquerda) e a elipse ajustada pelo programa (direita). Nas duas imagens, as elipses (inicial e ajustada) estão indicadas em vermelho.

A partir de r e \vec{e} calculamos os vetores \vec{a} e \vec{b} que correspondem aos raios menor e maior da elipse :

$$\vec{a} = \begin{cases} \vec{e} \left(1 + \frac{r}{|\vec{e}|}\right) & \text{se } |\vec{e}| \neq (0, 0) \\ (r, 0) & \text{caso contrário} \end{cases} \quad \vec{b} = \begin{cases} R_{90}(\vec{e}) \frac{r}{|\vec{e}|} & \text{se } |\vec{e}| \neq (0, 0) \\ (0, r) & \text{caso contrário} \end{cases} \quad (4.3)$$

onde R_{90} é o operador de rotação por 90° em \mathbb{R}^2 .

4.3.4 Calculando as normais do gabarito

Para calcular a normal $\vec{n}[p]$ em um ponto dado p de \mathcal{G} , primeiro corrigimos a deformação da projeção oblíqua, pelas fórmulas

$$\begin{aligned} \vec{d} &= q - c \\ \vec{v}_a &= \frac{\vec{a}}{|\vec{a}|^2} (\vec{d} \cdot \vec{a}) \\ \vec{v}_b &= \frac{\vec{b}}{|\vec{b}|^2} (\vec{d} \cdot \vec{b}) \\ \vec{h} &= \vec{v}_a + \vec{v}_b \end{aligned} \quad (4.4)$$

O vetor \vec{h} é a projeção ortogonal da normal $\vec{n}[p]$ no plano formado pelos eixos x e y do sistema de coordenadas do gabarito. Portanto, $\vec{n}[p]' = (\vec{h}.x, \vec{h}.y, \sqrt{1 - \vec{h}.x^2 + \vec{h}.y^2})$ nesse sistema local do gabarito. A normal no sistema da cena será então $\vec{n}[p] = \vec{n}[p]'M_v$.

4.3.5 Exclusão dos pixels da borda

Como observamos, as intensidades dos pixels ao longo da borda de \mathcal{G} normalmente são afetadas por mistura com a intensidade do fundo ou por filtros de realce de bordas, o que nos obriga a excluir as amostras referentes a esses pixels. Por exemplo, se \mathcal{G} é uma elipse com $2r$ pixels de largura na parte mais estreita, para excluir todo pixel que cruza a borda basta excluir qualquer normal \vec{n} tal que $\vec{n} \cdot \vec{v} \geq \sqrt{1 - ((r - \sqrt{2})/r)^2}$. Se $r = 50$ pixels, esta condição significa considerar apenas as amostras cujas normais fazem ângulo menor ou igual a $\theta_{\max} = 76^\circ$ com a direção de vista \vec{v} . Veja a figura 4.6.

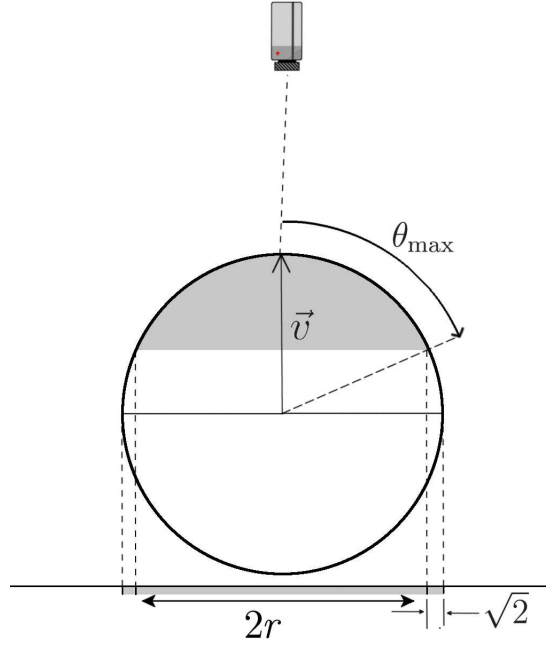


Figura 4.6: O conjunto de normais (em cinza) que podem ser amostradas sem mistura com a cor de fundo.

4.4 Modelos lineares genéricos

Uma estratégia comum para aproximação de uma função arbitrária f é procurar a função aproximadora em algum espaço linear de funções, com mesmo domínio que f . No nosso

caso, essa estratégia significa escolher q funções fixas $\phi_1, \phi_2, \dots, \phi_q$ de \mathbb{S}^2 para \mathbb{R} , a *base de aproximação*; e definir \tilde{L} pela fórmula

$$\tilde{L}(\vec{n}) = \sum_{r=1}^q \alpha_r \phi_r(\vec{n}) \quad (4.5)$$

para todo $\vec{n} \in \mathbb{S}^2$, onde $\alpha_1, \dots, \alpha_q \in \mathbb{R}$ são coeficientes reais a determinar.

Para este tipo de função aproximadora, demonstra-se que os coeficientes $\alpha_1, \dots, \alpha_q$ que minimizam o erro quadrático $Q(\tilde{L})$ são a solução do sistema linear

$$A\alpha = b \quad (4.6)$$

onde α é o vetor coluna dos coeficientes $\alpha_1, \dots, \alpha_q$, A é uma matriz $q \times q$ e b é um vetor coluna de q elementos, dados pelas fórmulas

$$\begin{aligned} A_{ij} &= \sum_{k=1}^N w_k \phi_i(\vec{n}_k) \phi_j(\vec{n}_k) \\ b_i &= \sum_{k=1}^N w_k \phi_i(\vec{n}_k) l_k \end{aligned} \quad (4.7)$$

Dentro desta abordagem geral há muitas variantes possíveis, determinadas pela escolha da base de aproximação ϕ_1, \dots, ϕ_q . Nas nossas pesquisas, utilizamos aproximações lineares com dois tipos de bases: *harmônicos esféricos* e *bases radiais*.

Para estas fórmulas, as direções \vec{n}, \vec{u} são expressas no sistema de coordenadas locais do gabarito onde $\vec{v} = (0, 0, 1)$. As notações $.x, .y$ e $.z$ representam as coordenadas deste sistema local; em particular, $\vec{n}.z = \vec{n} \cdot \vec{v}$.

4.4.1 Harmônicos esféricos

Para funções definidas no círculo \mathbb{S}^1 , a base mais natural (em vários sentidos) é a formada pelos *harmônicos de Fourier* [22].

$$F_k(\theta) = e^{ik\theta} = \cos k\theta + \mathbf{i} \sin k\theta \quad (4.8)$$

onde k (a *frequência* do elemento) é um inteiro, limitado ao um intervalo simétrico $\{-M, \dots, +M\}$, e θ é a posição angular no círculo em radianos. Na esfera \mathbb{S}^2 , a base análoga é a dos *harmônicos esféricos* $Y_{\ell m}(\theta, \varphi)$, onde θ, φ são as coordenadas esféricas (*inclinação* e *azimute*) de um vetor \vec{n} de \mathbb{S}^2 , m é um número natural (a *ordem* ou *grau*) e ℓ um inteiro entre $-m$ e $+m$ (o *índice*). O elemento genérico da base é definido por

$$Y_{\ell m}(\theta, \varphi) = \gamma e^{i\ell\varphi} P_{\ell}^m(\cos \theta) \quad (4.9)$$

onde γ é uma constante de normalização e P é o *polinômio de Legendre* de ordem m e índice l [73]. Note que θ e φ são medidos em relação às coordenadas locais do gabarito, onde a direção de vista \vec{v} tem inclinação zero; e que, portanto, $\cos \theta$ é a coordenada $\vec{n} \cdot z$

Denotamos por \mathcal{H}_M o espaço de funções gerado pelos harmônicos $Y_{\ell m}$ com ordem m entre 0 e M inclusive. Informalmente, M é a “frequência” máxima das funções desse espaço. Verifica-se que o espaço \mathcal{H}_M pode ser obtido por outra base, em termos das coordenadas cartesianas $\vec{n}.x$, $\vec{n}.y$ e $\vec{n}.z$ da normal \vec{n} .

$$\begin{aligned} \vec{n}.x &= \sin \theta \cos \varphi \\ \vec{n}.y &= \sin \theta \sin \varphi \\ \vec{n}.z &= \cos \theta \end{aligned} \quad (4.10)$$

Especificamente, Gomide e Stolfi [43] demonstraram que \mathcal{H}_M coincide com o espaço de todos os polinômios com monômios de grau M ou $M - 1$ nas coordenadas $\vec{n}.x$, $\vec{n}.y$ e $\vec{n}.z$. Em particular \mathcal{H}_0 é o conjunto das funções constantes da esfera e \mathcal{H}_1 consiste das funções que podem ser escritas como $ax + by + cz + d$. Portanto, em vez dos harmônicos $Y_{\ell m}$ podemos usar como base de aproximação os monômios de graus M e $M - 1$, isto é as funções

$$\phi_{ijk}(\vec{n}) = (\vec{n}.x^i)(\vec{n}.y^j)(\vec{n}.z^k) \quad (4.11)$$

para todo $i, j, k \in \mathbb{N}$ tal que $i + j + k = M$ ou $i + j + k = M - 1$. Veja a figura 4.7.

4.4.2 Bases radiais

Uma *função radial* na esfera é uma função de \mathbb{S}^2 para \mathbb{R} cujo valor $F(x)$ depende apenas da distância de x a um *vetor central* $\vec{\mu}$, isto é

$$F(\vec{n}) = \Psi(\text{dist}(\vec{n}, \vec{\mu})/\rho) \quad (4.12)$$

para alguma função Ψ de \mathbb{R} para \mathbb{R} e algum $\rho \in \mathbb{R}$ [23]. A função Ψ é chamada de *função mãe* da função F , e o parâmetro ρ é seu *raio nominal*.

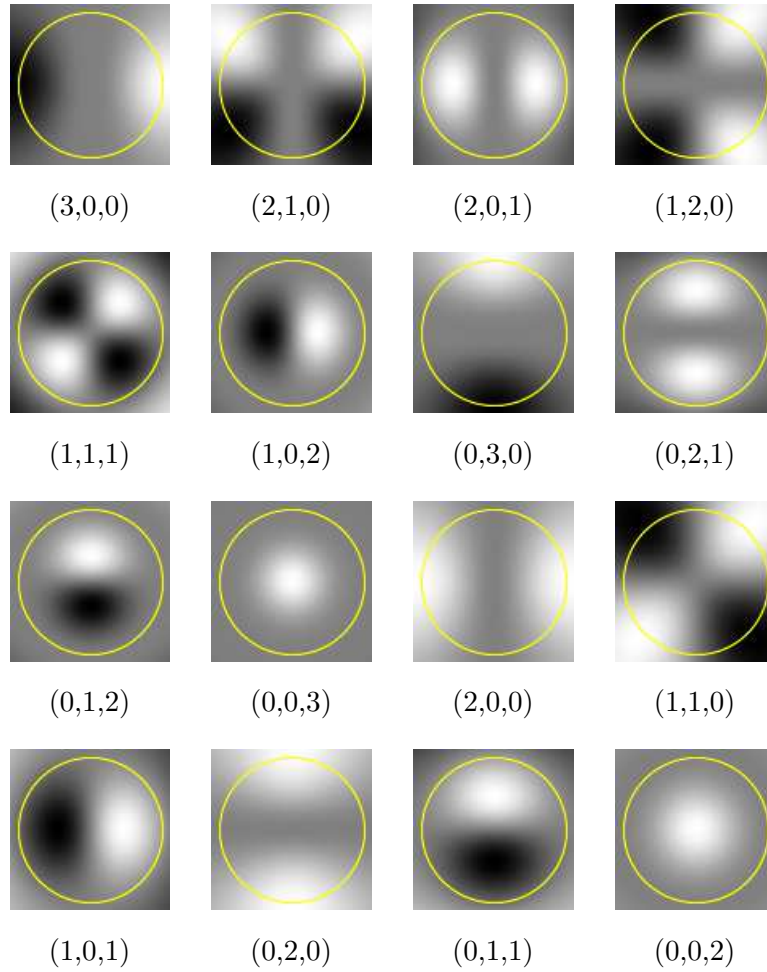


Figura 4.7: Visualização dos elementos ϕ_{ijk} da base de monômios de grau 2 e 3. As imagens são obtidas por projeção estereográfica da esfera \mathbb{S}^2 , com a direção de vista $\vec{v} = (0, 0, 1)$ na origem. O círculo indica o equador de \mathbb{S}^2 (normais perpendiculares a \vec{v}).

Uma *base radial* é uma base de funções ϕ_1, \dots, ϕ_q , onde cada elemento ϕ_k é uma função radial com seu próprio centro μ_k e seu próprio raio ρ_k , mas com função mãe Ψ comum a todos elementos:

$$\phi_r(\vec{n}) = \Psi \left(\frac{\text{dist}(\vec{n}, \vec{\mu}_r)}{\rho_r} \right) \quad (4.13)$$

Uma aproximação usando bases radiais então é uma combinação linear destes elementos ϕ_i , pela fórmula (4.5).

Em geral, deseja-se que o valor de um elemento $\phi(\vec{n})$ seja máximo quando $\vec{n} = \vec{\mu}$, e diminua à medida que \vec{n} se afasta de $\vec{\mu}$. Ou seja, a função mãe Ψ deve ser uma função decrescente de seu argumento. Note que o parâmetro ρ permite variar o “tamanho” da função ϕ .

Uma escolha comum para a distância $\text{dist}(\vec{n}, \vec{\mu})$ na fórmula (4.12) é o ângulo $\alpha = \arccos(\vec{n} \cdot \vec{\mu})$ entre os dois vetores. Entretanto, a fim de evitar o cálculo da função arco-cosseno, no nosso trabalho utilizamos a função distância

$$\text{dist}(\vec{n}, \vec{\mu}) = \sqrt{1 - \vec{n} \cdot \vec{\mu}} \quad (4.14)$$

Observamos que $\vec{n} \cdot \vec{\mu} \approx 1 - \alpha^2/2$ quando \vec{n} é próximo a $\vec{\mu}$, e portanto a fórmula (4.14) vale aproximadamente $\alpha\sqrt{2}$ para α pequeno. Essa fórmula também vale 1 quando $\alpha = 90^\circ$ e $\sqrt{2}$ quando $\alpha = 180^\circ$. Veja a figura 4.8.

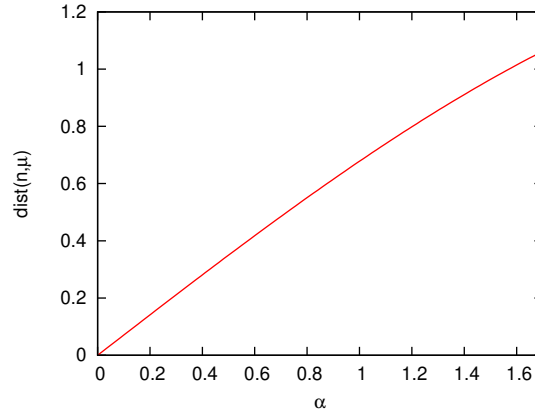


Figura 4.8: Variação da métrica $\text{dist}(\vec{n}, \vec{\mu}_r)$ com o ângulo α entre \vec{n} e $\vec{\mu}_r$.

Para função mãe, escolhemos

$$\Psi(c) = \begin{cases} (1 - c^2)^2 & \text{se } c < 1 \\ 0 & \text{caso contrário} \end{cases} \quad (4.15)$$

Note que Ψ é uma função suave (C_1) do argumento c . Veja a figura 4.9.

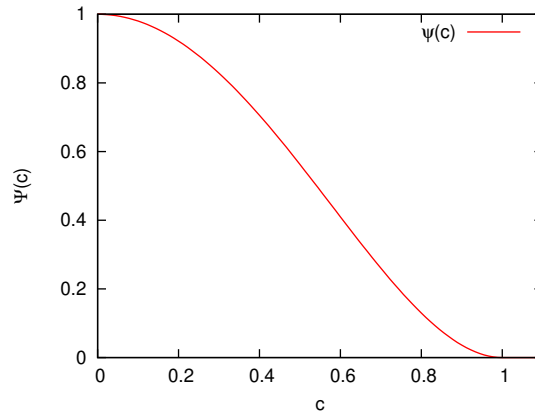


Figura 4.9: A função mãe Ψ da nossa base radial.

Portanto o elemento genérico da base é

$$\phi_r(\vec{n}) = \Psi\left(\frac{\text{dist}(\vec{n}, \vec{\mu}_r)}{\rho_r}\right) = \begin{cases} \left[1 - \frac{(1 - \vec{n} \cdot \vec{\mu}_r)}{\rho_r^2}\right]^2 & \text{se } \vec{n} \cdot \vec{\mu}_r > 1 - \rho_r^2, \\ 0 & \text{caso contrário.} \end{cases} \quad (4.16)$$

Em termos do ângulo α entre \vec{n} e $\vec{\mu}_r$, esta definição equivale a

$$\phi_r(\vec{n}) = \begin{cases} \left[1 - \frac{(1 - \cos \alpha)}{\rho_r^2}\right]^2 & \text{se } \alpha < \alpha_r, \\ 0 & \text{caso contrário.} \end{cases} \quad (4.17)$$

onde $\alpha_r = \arccos(1 - \rho_r^2)$. Veja a figura 4.10.

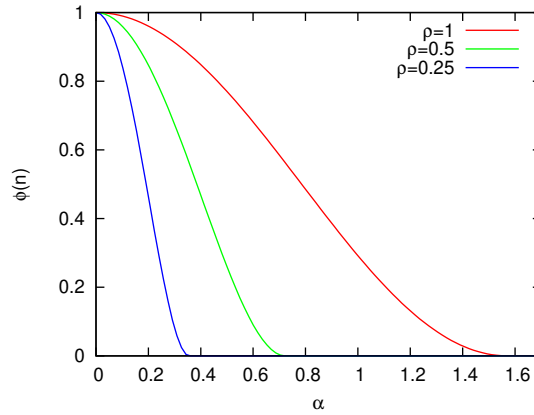


Figura 4.10: Variação do elemento da base radial $\phi_r(\vec{n})$ em função do ângulo α entre \vec{n} e $\vec{\mu}_r$, para diversos valores do raio nominal ρ_r .

Note que, com estas escolhas, o valor de $\phi_r(\vec{n})$ é 1 quando $\vec{n} = \vec{\mu}_r$, e zero quando $\text{dist}(\vec{n}, \vec{\mu}_r) \geq \rho_r$ (isto é, quando $\alpha > \alpha_r$). Ou seja, o suporte de ϕ_r é a calota da esfera \mathbb{S}^2 com centro $\vec{\mu}_r$ e raio angular α_r .

Em nossos experimentos, os centros $\vec{\mu}_1, \dots, \vec{\mu}_n$ dos elementos da base foram escolhidos com distribuição aproximadamente uniforme entre as normais que fazem com a direção de vista \vec{v} um ângulo menor ou igual ao ângulo de amostragem máximo θ_{\max} (seção 4.3.5). Os centros estão sobre t círculos na esfera \mathbb{S}^2 com centro \vec{v} . O círculo de índice i consiste das normais com inclinação $i\theta_{\max}/t$ em relação a \vec{v} , para $i \in \{0, \dots, t-1\}$. Sobre este círculo tomamos n_i centros igualmente espaçados, onde

$$n_i = \left\lceil \pi t \sin\left(\frac{i}{t}\theta_{\max}\right) \right\rceil + 1 \quad (4.18)$$

Em particular, no círculo de índice 0 escolhemos apenas um centro $\vec{\mu}_0 = \vec{v}$. Os raios

$\rho_1, \rho_2, \dots, \rho_n$ dos elementos crescem à medida que $\vec{m}\vec{u}_r$ se afasta de \vec{v} . Especificamente, usamos

$$\rho_r = \frac{\beta \cos \theta_{\max}}{\vec{\mu}_r \cdot \vec{v}} \quad (4.19)$$

onde β é um parâmetro dado entre 0 e 1. Note que $\rho_r \approx \beta \cos \theta_{\max}$ quando $\vec{u} \approx \vec{v}$, e $\rho_r \approx \beta$ quando o ângulo entre \vec{n} e \vec{v} é próximo a θ_{\max} .

Veja a figura 4.11.

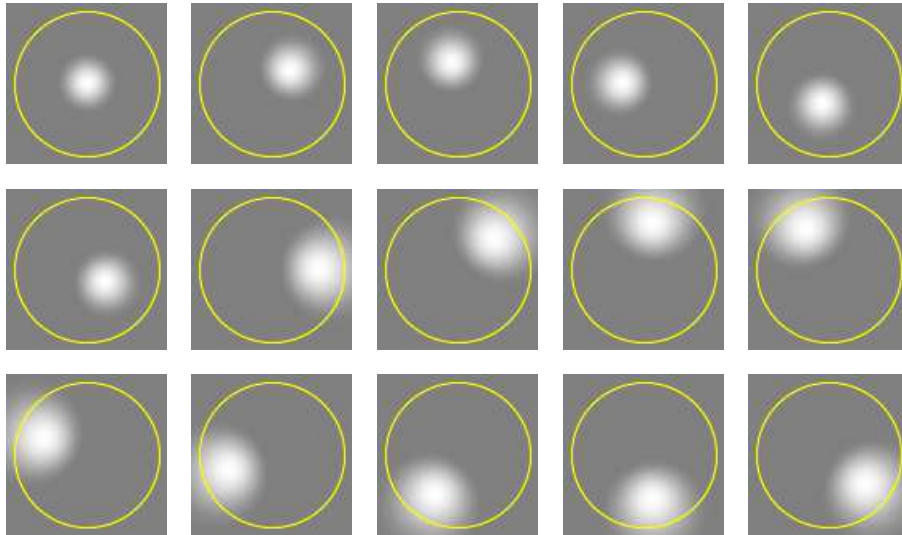


Figura 4.11: Projeção estereográfica da esfera \mathbb{S}^2 mostrando os 15 elementos ϕ_r da base radial, com centros $\vec{\mu}_r$ escolhidos sobre $t = 3$ círculos (com 1, 5 e 9 centros em cada círculo).

4.5 Modelo de fonte de luz compacta única

Os modelos descritos na seção 4.4 são convenientes porque são genéricos, no sentido que podem modelar qualquer função de tonalização suave; e porque são lineares nos parâmetros $\alpha_1, \dots, \alpha_q$, o que permite determinar estes últimos resolvendo o sistema linear (4.6). Por outro lado, esses modelos têm dificuldade para aproximar funções com descontinuidades em lugares arbitrários da esfera \mathbb{S}^2 . Para estas funções pode ser vantajoso utilizar uma aproximação mais adaptada à natureza da função L .

4.5.1 Tonalização lambertiana com iluminação unidirecional

Um exemplo importante de função de tonalização não suave é a função de tonalização de uma superfície com FBDR lambertiana (2.15) com iluminada unidirecional. Como vimos no capítulo 2, nestas condições a função de tonalização L é dada por

$$L(\vec{n}) = E_0 \max\{0, \vec{n} \cdot \vec{u}\} \quad (4.20)$$

onde E_0 é a intensidade da fonte luminosa e \vec{u} é a direção do centro do gabarito à fonte luminosa. Veja a figura 4.12. Esta função tem uma descontinuidade na derivada ao longo do *terminador*, o conjunto de normais \vec{n} perpendiculares à direção da luz \vec{u} . Observe que, devido ao uso da função max, a fórmula (4.20) não é linear no parâmetro \vec{u} ; e que a posição da descontinuidade na esfera \mathbb{S}^2 depende de \vec{u} .

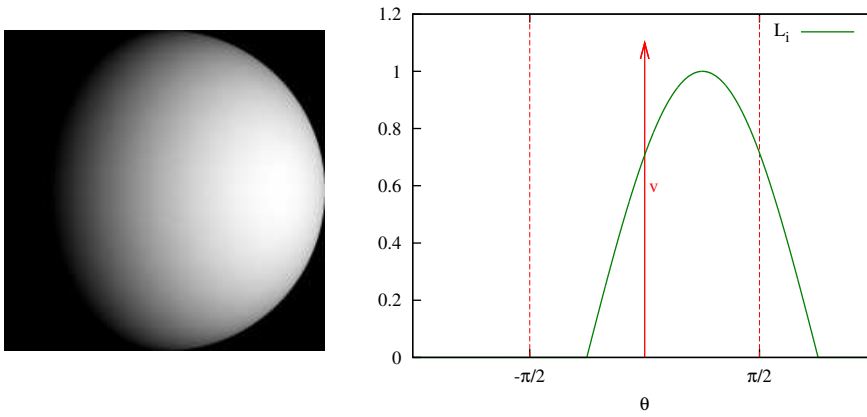


Figura 4.12: Imagem sintética (à esquerda) de um gabarito com superfície lambertiana, com iluminação unidirecional de direção $\vec{u} = (\sin 45^\circ, 0, \cos 45^\circ)$, sendo que a direção da câmera é $\vec{v} = (0, 0, 1)$. O gráfico à direita mostra o valor de $L(\vec{n})$ quando o ponto de amostragem varia ao longo de uma linha horizontal passando pelo centro da imagem, isto é, com $\vec{n} = (\sin \theta, 0, \cos \theta)$ para θ entre $-\pi$ e π . Note que apenas os pontos com θ no intervalo $[-\pi/2, +\pi/2]$ (entre as linhas pontilhadas) são visíveis.

4.5.2 Tonalização lambertiana com luz difusa

A fórmula (4.20) é válida apenas para uma fonte de luz pontual (unidirecional), cujo diâmetro é pequeno quando comparado com a distância da cena. Para fontes múltiplas e/ou com tamanho angular significativo, a função de tonalização é mais complexa. Porém, para qualquer fonte de luz suficientemente distante da cena, ainda que extensa, existe uma fonte de luz unidirecional equivalente, com certa intensidade efetiva E e certa direção média \vec{u} , tal que a fórmula (4.20) vale enquanto o ponto com normal \vec{n} for iluminado por todos os pontos da fonte original. Por exemplo, suponha que o gabarito é iluminado por duas fontes de direção \vec{u}_1, \vec{u}_2 com intensidades E_1, E_2 . Em todo ponto que for iluminado por ambas as fontes, a normal \vec{n} satisfaz $\vec{u}_1 \cdot \vec{n} \geq 0$ e $\vec{u}_2 \cdot \vec{n} \geq 0$. Nesse caso a função max

na fórmula (4.20) pode ser omitida, e temos

$$\begin{aligned} L(\vec{n}) &= E_1(\vec{u}_1 \cdot \vec{n}) + E_2(\vec{u}_2 \cdot \vec{n}) \\ &= (E_1\vec{u}_1 + E_2\vec{u}_2) \cdot \vec{n} \\ &= E(\vec{u} \cdot \vec{n}) \end{aligned} \quad (4.21)$$

onde

$$\begin{aligned} E &= |E_1\vec{u}_1 + E_2\vec{u}_2| \\ \vec{u} &= (E_1\vec{u}_1 + E_2\vec{u}_2)/E \end{aligned} \quad (4.22)$$

Veja a figura 4.13.

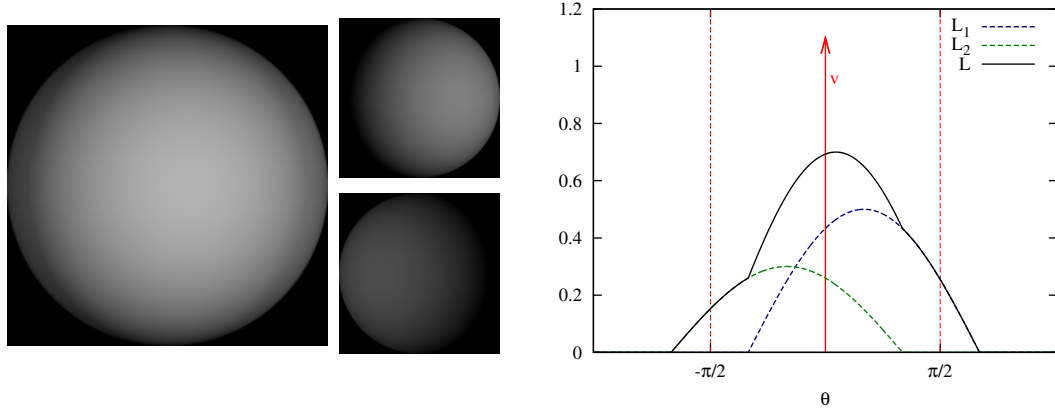


Figura 4.13: À esquerda, imagem sintética de um gabarito com superfície lambertiana iluminado por duas fontes de luz unidirecionais com intensidades respectivamente $E_1 = 0.5$ e $E_2 = 0.3$ e direções $\vec{u}_1 = (\sin 30^\circ, 0.0, \cos 30^\circ)$ (centro,acima) e $\vec{u}_2 = (\sin(-30^\circ), 0.0, \cos(-30^\circ))$ (centro,abaixo). À direita, gráfico de $L(\vec{n})$ (linha cheia) com as mesmas convenções da figura 4.12.

Esta redução pode ser generalizada para qualquer número (inclusive infinito) de fontes unidirecionais. A redução não é válida, entretanto, para qualquer ponto do gabarito que não seja iluminado por alguma das fontes. Por esse motivo, a transição entre a área totalmente iluminada e a região de sombra própria não é mais a transição abrupta característica de uma fonte unidirecional única (de $\vec{n} \cdot \vec{u}$ para 0), mas possui um comportamento mais complexo.

No caso de uma fonte extensa (não unidirecional), essa transição é geralmente suave até primeira ordem (C_1). Especificamente, se a fonte vista do gabarito está contida em um cone com eixo \vec{u} e raio angular ρ , então a região de transição entre a região totalmente

iluminada, onde $L(\vec{n}) = E(\vec{u} \cdot \vec{n})$, e a região da sombra própria, onde $L(\vec{n}) = 0$, é uma faixa com largura 2ρ ao longo do círculo $\vec{n} \cdot \vec{u} = 0$. Veja a figura 4.14.

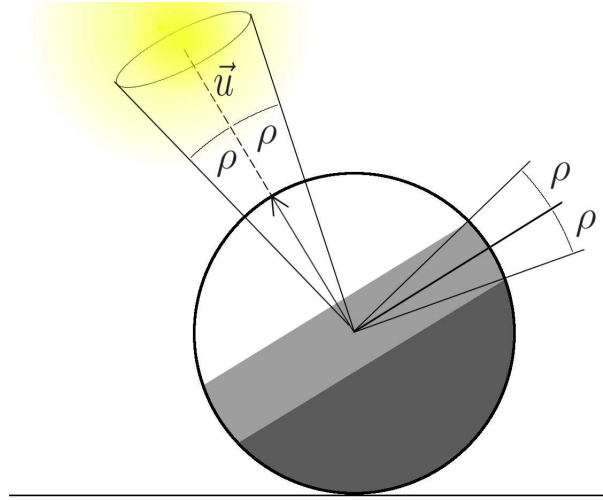


Figura 4.14: Iluminação por fonte luminosa extensa com raio angular ρ e direção \vec{u} . A região branca corresponde à região do gabarito iluminada por toda fonte luminosa, a região em cinza claro à parte parcialmente iluminada pela fonte, e a região cinza escura é a parte do gabarito sem iluminação nenhuma (em sombra própria).

Para modelar esse comportamento, podemos incluir na aproximação \tilde{L} uma *componente lambertiana extensa* $E_L \phi_L(\vec{n})$ onde o elemento ϕ_L é dado pela fórmula

$$\phi_L(\vec{n}) = \begin{cases} \vec{n} \cdot \vec{u} & \vec{n} \cdot \vec{u} \geq s \\ 0 & \vec{n} \cdot \vec{u} < -s \\ \frac{s}{4} \left(1 - \frac{\vec{n} \cdot \vec{u}}{s}\right)^2 & \text{caso contrário} \end{cases} \quad (4.23)$$

onde $s = \sin(\rho)$. Esta fórmula reproduz aproximadamente a função de tonalização para uma fonte que, vista do gabarito, é um disco de raio angular ρ (em radianos) com centro na direção \vec{u} . Veja a figura 4.15. Observe que a fórmula (4.23) reduz-se à tonalização lambertiana com iluminação unidirecional (4.20) quando $\rho = 0$, e que é suave (C_1) quando $\rho > 0$.

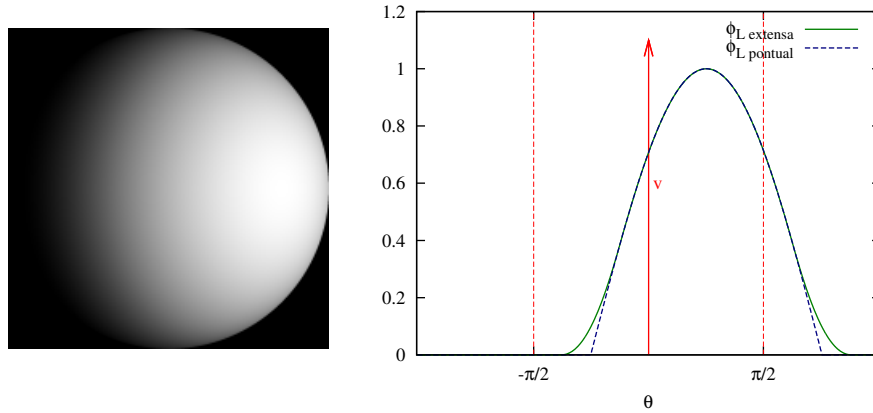


Figura 4.15: À esquerda, imagem sintética de um gabarito com superfície lambertiana iluminado por uma fonte circular com centro na direção $\vec{u} = (\sin 45^\circ, 0, \cos 45^\circ)$ e raio angular $\rho = 23^\circ$. À direita, o gráfico do modelo de $\phi_L(\vec{n})$ para essa fonte circular (linha cheia) e para a fonte unidirecional equivalente (linha tracejada). A transição suave entre a parte iluminada e a sombra própria, que ocorre quando θ está nos intervalos $-45^\circ \pm \rho$ e $45^\circ \pm \rho$, é calculada pela aproximação (4.23). A direção da câmera é $\vec{v} = (0, 0, 1)$ e as linhas pontilhadas delimitam a parte visível.

4.5.3 Modelo para superfície parcialmente lustrosa

Os problemas causados por uma iluminação unidirecional podem ser agravados quando a FBDR do gabarito possui componentes não lambertianos. Isso ocorre, por exemplo, se a superfície tem acabamento parcialmente lustroso. Neste caso a função de tonalização terá valores maiores que os previstos pelo modelo lambertiano, sempre que a normal \vec{n} for próxima à bissetriz \vec{m} das direções \vec{u} e \vec{v} ; e valores um pouco menores para normais \vec{n} distantes de \vec{m} . Para lidar com tais casos, acrescentamos à aproximação \tilde{L} um termo ϕ_P que corresponde ao acabamento lustroso de Phong [78].

$$\phi_P(\vec{n}) = \max\{0, \vec{n} \cdot \vec{u}\}(\max\{0, \vec{n} \cdot \vec{v}\})^K \quad (4.24)$$

definido na seção 2.3.4. Veja a figura 4.16.

Se a fonte de luz é extensa, o realce é mais amplo que o gerado por uma iluminação unidirecional. Nesta tese, vamos supor que o efeito de uma luz extensa pode ser modelado adequadamente pela fórmula (4.24) com um valor de K reduzido. Ou seja, vamos supor que o realce de uma superfície polida sob uma fonte extensa é similar ao de uma superfície menos polida com iluminação unidirecional.

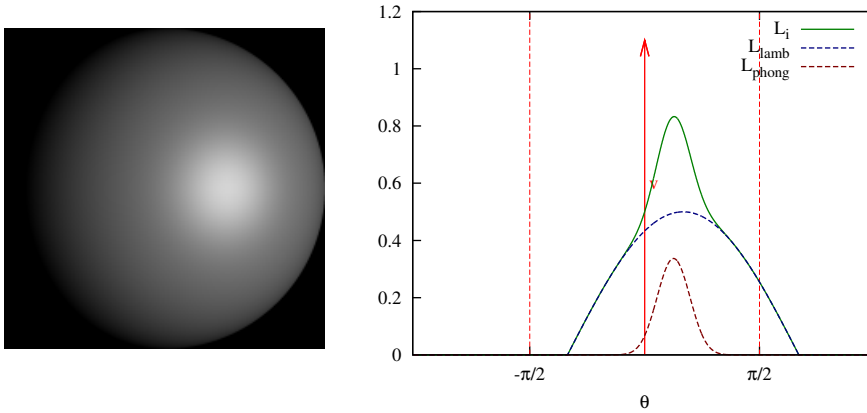


Figura 4.16: À esquerda, imagem sintética de um gabarito com superfície parcialmente lambertiana e parcialmente lustrosa, com iluminação unidirecional de direção $\vec{u} = (\sin 30^\circ, 0, \cos 30^\circ)$. À direita, o gráfico de $L(\vec{n})$, dos componentes lambertiano $E_L \phi_L(\vec{n})$ e lustroso $E_P \phi_P(\vec{n})$ separadamente, para $\vec{n} = (\sin \theta, 0, \cos \theta)$ em função de θ . A componente lustrosa $\phi_P(\vec{n})$ foi calculada pelo modelo de Phong com expoente $K = 10$ adotando-se $E_P = 0.4$ e $E_L = 0.5$.

4.5.4 Iluminação ambiente difusa

Mesmo quando a fonte primária é compacta, o fluxo de luz na cena geralmente inclui iluminação secundária proveniente de luz que é espalhada por paredes, teto e objetos próximos à cena. Como é praxe em computação gráfica, modelamos a contribuição dessa luz acrescentando à função \tilde{L} um termo constante E_0 que corresponde à tonalização de uma superfície lambertiana sob iluminação isotrópica (fluxo $\Phi(\vec{n})$ constante independente de \vec{n}).

Além disso, nos nossos testes a cena tipicamente consiste de um ou mais objetos colocados sobre um plano de fundo. Verifica-se que a luz espalhada por este plano de fundo contribui significativamente para a iluminação do objeto e do gabarito, especialmente nas partes onde a superfície é mais próxima da vertical.

Para levar em conta esta contribuição, podemos acrescentar ao modelo \tilde{L} um termo que corresponde ao fluxo de luz espalhada por um plano de fundo lambertiano de albedo e iluminação uniformes. Veja a figura 4.17.

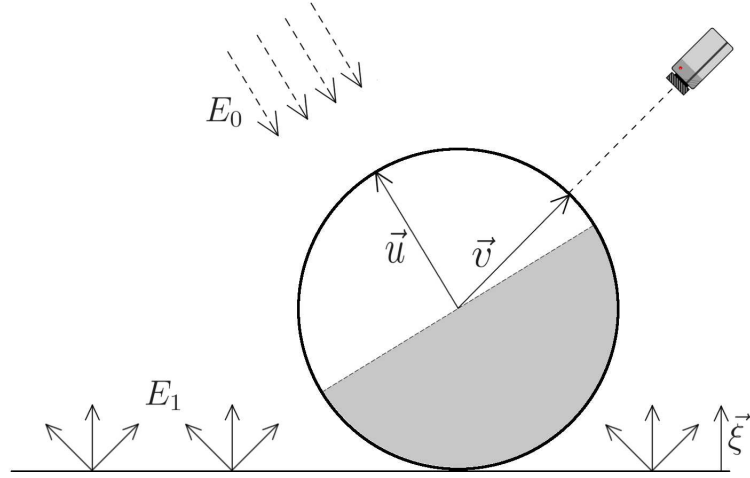


Figura 4.17: Modelo lambertiano com uma iluminação unidirecional e um plano de fundo com emissor difuso.

Ignorando o efeito da sombra projetada do gabarito, este plano de fundo contribui para a função de tonalização \tilde{L} com um termo adicional $E_1(1 - \vec{n} \cdot \vec{\xi})$ onde $\vec{\xi}$ é a normal do plano de fundo. Esta contribuição é máxima quando $\vec{n} = -\vec{\xi}$, e nula quando $\vec{n} = \vec{\xi}$. Combinando este termo com a luz ambiente isotrópica, temos

$$E_0 + E_1(1 + \vec{n} \cdot \vec{\xi}) = (E_0 + E_1) + (E_1\vec{\xi}.x)\vec{n}.x + (E_1\vec{\xi}.y)\vec{n}.y + (E_1\vec{\xi}.z)\vec{n}.z \quad (4.25)$$

Portanto, para representar esta contribuição, podemos incluir na base de aproximação os elementos $\phi_A, \phi_X, \phi_Y, \phi_Z$ definidos por

$$\begin{aligned} \phi_A(\vec{n}) &= 1 \\ \phi_X(\vec{n}) &= \vec{n}.x \\ \phi_Y(\vec{n}) &= \vec{n}.y \\ \phi_Z(\vec{n}) &= \vec{n}.z \end{aligned} \quad (4.26)$$

Os coeficientes destes termos são $E_A = E_0 + E_1$, $E_X = E_1\vec{\xi}.x$, $E_Y = E_1\vec{\xi}.y$ e $E_Z = E_1\vec{\xi}.z$. Portanto, uma vez determinados os coeficientes E_X, E_Y, E_Z , podemos recuperar a direção $\vec{\xi}$ e a intensidade E_1 da luz de fundo pelas fórmulas $E_1 = \sqrt{E_X^2 + E_Y^2 + E_Z^2}$ e $\vec{\xi} = (E_X, E_Y, E_Z)/E_1$. Note que os elementos (4.26) são os monômios de grau 0 e 1 nas coordenadas de \vec{n} , e portanto geram o espaço \mathcal{H}_1 dos harmônicos esféricos de grau ≤ 1 .

4.5.5 Modelo geral para fonte compacta única

Em conclusão, nosso modelo geral para uma única fonte relativamente compacta, incluindo iluminação secundária, é uma combinação linear dos elementos ϕ_L , ϕ_P , ϕ_A , ϕ_X , ϕ_Y e ϕ_Z :

$$\tilde{L} = E_L\phi_L + E_P\phi_P + E_A\phi_A + E_X\phi_X + E_Y\phi_Y + E_Z\phi_Z \quad (4.27)$$

Note que este modelo depende não apenas dos coeficientes E_L , E_P , E_A , E_X , E_Y e E_Z , mas também dos parâmetros \vec{u} , K e ρ embutidos nas funções ϕ_L e ϕ_P .

4.6 Ajustando o modelo de fonte única

Se os parâmetros \vec{u} , K e ρ do modelo (4.27) de fonte única são conhecidos, apenas os coeficientes E_L , E_P , E_A , E_X , E_Y e E_Z precisam ser determinados. Como a fórmula (4.27) é linear em todos estes parâmetros, estes podem ser determinados resolvendo-se um sistema linear, como visto na seção 4.4. Porém, é importante notar que quando a direção \vec{u} da fonte é próxima da direção de vista \vec{v} , a função ϕ_L é praticamente uma combinação linear de ϕ_X , ϕ_Y e ϕ_Z . Portanto neste caso temos que excluir ϕ_X , ϕ_Y e ϕ_Z da aproximação.

Se os parâmetros \vec{u} , K e ρ não são conhecidos, o método da seção 4.4 não pode ser usado pois a fórmula (4.27) não é linear nestes parâmetros. Nesta seção, vamos examinar alguns métodos para resolver este problema.

4.6.1 Determinando \vec{u} com K , ρ conhecidos

Vamos considerar primeiro o caso em que a direção \vec{u} não é conhecida, mas ρ e K são conhecidos. Uma técnica muito popular para determinar a direção \vec{u} é determinar a direção \vec{n}^* onde $L(\vec{n})$ é máxima [27, 55]. Se L é dominada pelo termo lambertiano μ_L , então \vec{n}^* a direção \vec{u} da fonte. Se L é dominada pelo termo lustroso, então \vec{n} é a bissetriz de \vec{n} e \vec{v} . Este método não pode ser usado quando as duas componentes são significativas pois nesse caso \vec{n}^* pode ser qualquer normal entre \vec{u} e \vec{n} .

Neste caso, podemos tentar determinar \vec{u} pelo método dos quadrados mínimos, considerando apenas as amostras (\vec{n}_k, l_k) onde ϕ_L é linear em \vec{u} e ϕ_P é desprezível. Vamos supor que temos uma estimativa aproximada inicial $\vec{\mu}_{\vec{u}}$ para \vec{u} , e um limite superior $\epsilon_{\vec{u}}$ para o desvio angular de \vec{u} em relação a $\vec{\mu}_{\vec{u}}$.

Para excluir amostras da região de sombra própria e iluminação parcial devemos excluir toda amostra tal que $\vec{n}_k \cdot \vec{u} < \sin(\rho)$; ou seja, todo \vec{n}_k tal que $\vec{n}_k \cdot \vec{\mu}_{\vec{u}} < \sin(\rho + \epsilon_{\vec{u}})$. Para este fim, definimos um fator de ponderação w'_k para cada amostra, onde

$$w'_k = \begin{cases} 1 & \text{se } \vec{n}_k \cdot \vec{u} > \sin(\rho + \epsilon_{\vec{u}}) \\ 0 & \text{caso contrário} \end{cases} \quad (4.28)$$

Veja a figura 4.18.

Para excluir o termo $E_P \phi_P$, temos que excluir todas as amostras (l_k, \vec{n}_k) para as quais $\phi_P(\vec{n}_k)$ pode ter valor significativo comparado com $\phi_L(\vec{n}_k)$ ou seja, cuja normal \vec{n}_k faz angulo pequeno com a bissetriz \vec{m} de \vec{u} e \vec{v} . Especificamente, excluimos todas as amostras tais que o fator $(\vec{n} \cdot \vec{m})^K$ da fórmula (4.24) é menor que uma tolerância δ dada. Este critério elimina as normais contidas em um cone com eixo \vec{m} e largura $\sigma = \arccos \delta^{1/K}$. Em termos dos dados conhecidos $\vec{\mu}_{\vec{u}}$ e $\epsilon_{\vec{u}}$, temos que o critério é eliminar toda amostra (l_k, \vec{n}_k) tal que $\vec{n}_k \cdot \vec{\mu}_{\vec{m}} > \cos(\epsilon_{\vec{u}} + \sigma)$ onde $\vec{\mu}_{\vec{m}}$ é a bissetriz de $\vec{\mu}_{\vec{u}}$ e \vec{v} . Para este fim definimos para cada amostra o fator de ponderação

$$w''_k = \begin{cases} 1 & \text{se } \vec{n}_k \cdot \vec{w} < \cos(\sigma + \epsilon_{\vec{u}}) \\ 0 & \text{caso contrário} \end{cases} \quad (4.29)$$

Veja a figura 4.19. O peso w_k de cada amostra (l_k, \vec{n}_k) é então o produto destes dois fatores,

$$w_k = w'_k w''_k \quad (4.30)$$

Veja a figura 4.20.

Se os parâmetros K, ρ não são conhecidos exatamente, podemos usar em seu lugar estimativas conservadoras. Se sabemos que ρ está no intervalo $[\mu_\rho - \epsilon_\rho, \mu_\rho + \epsilon_\rho]$ e K está em $[\mu_K - \epsilon_K, \mu_K + \epsilon_K]$ podemos usar as fórmulas (4.28) e (4.29) com $\rho = \mu_\rho + \epsilon_\rho$ e $K = \mu_K - \epsilon_K$.

Uma vez excluídas as amostras onde ϕ_L não é linear em \vec{u} e ϕ_P é significativo, podemos determinar \vec{u} e os coeficientes E_L, E_A resolvendo um sistema linear, como na seção 4.4 Para tal fim, supomos que os termos ϕ_P, ϕ_X, ϕ_Y e ϕ_Z são desprezíveis, e decomponos o produto escalar $\vec{n} \cdot \vec{u}$ em três termos, obtendo o modelo reduzido

$$\tilde{L}(\vec{n}) = E_L(\vec{n}.x \vec{u}.x + \vec{n}.y \vec{u}.y + \vec{n}.z \vec{u}.z) + E_A \quad (4.31)$$

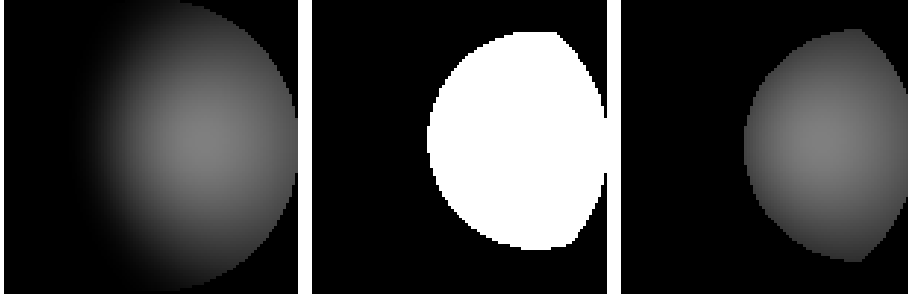


Figura 4.18: À esquerda, imagem de um gabarito com acabamento lambertiano iluminado por uma fonte de luz estendida distante com direção $\vec{u} = (0.627, 0.040, 0.777)$ e raio angular $\rho = 0.3$. No centro, pesos w'_k calculados para a direção luz média $\vec{\mu}_{\vec{u}} = (0.707, 0, 0.707)$ e desvio máximo $\epsilon_{\vec{u}} = 0.15$ radianos $\approx 8^\circ$, excluindo sombra própria e regiões parcialmente iluminadas. À direita, amostras selecionadas por este critério.

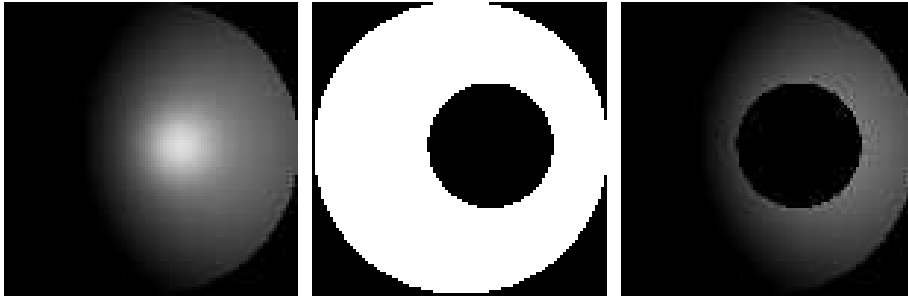


Figura 4.19: À esquerda, imagem de um gabarito parcialmente lambertiano e parcialmente lustroso com componente $K = 10$, com iluminação unidirecional de direção $\vec{u} = (0.627, 0.040, 0.777)$. No centro, os pesos w''_k calculados para a direção média $\vec{\mu}_{\vec{u}} = (0.707, 0, 0.707)$ e desvio máximo $\epsilon_{\vec{u}} = 0.15 \approx 8^\circ$ com tolerância $\sigma = 0.1$. À direita, as amostras selecionadas por este critério.

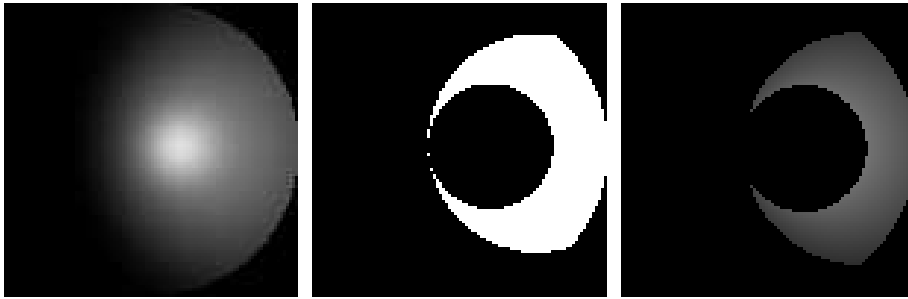


Figura 4.20: À esquerda, imagem de um gabarito com acabamento parcialmente lambertiano e parcialmente lustroso com componente lustrosa $K = 10$, com iluminação unidirecional de direção $\vec{u} = (0.627, 0.04, 0.777)$, raio angular $\rho = 0.3$. Ao centro, pesos w_k calculados pela fórmula (4.30) supondo da estimativa $\vec{\mu}_{\vec{u}} = (0.707, 0, 0.707)$ e incerteza $\epsilon_{\vec{u}} = 0.15 \approx 8^\circ$. À direita, as amostras selecionadas por este critério.

Definimos então os parâmetros auxiliares

$$\begin{aligned}\alpha_1 &= E_L \vec{u}.x \\ \alpha_2 &= E_L \vec{u}.y \\ \alpha_3 &= E_L \vec{u}.z \\ \alpha_4 &= E_A\end{aligned}\tag{4.32}$$

e as funções de base

$$\begin{aligned}\phi_1(\vec{n}) &= \vec{n}.x \\ \phi_2(\vec{n}) &= \vec{n}.y \\ \phi_3(\vec{n}) &= \vec{n}.z \\ \phi_4(\vec{n}) &= 1\end{aligned}\tag{4.33}$$

Desta forma, a aproximação \tilde{L} da fórmula (4.31) passa a ser equivalente à fórmula (4.5), linear nos parâmetros $\alpha_1, \dots, \alpha_4$. Resolvendo o sistema (4.6) para esta base, determinamos os valores de $\alpha_1, \alpha_2, \alpha_3, \alpha_4$. Podemos então inverter as fórmulas (4.32), obtendo os valores de E_L, E_A e uma estimativa para o vetor \vec{u} :

$$\begin{aligned}E_L &= \sqrt{\alpha_1^2 + \alpha_2^2 + \alpha_3^2} \\ \vec{u}.x &= \alpha_1/E_L \\ \vec{u}.y &= \alpha_2/E_L \\ \vec{u}.z &= \alpha_3/E_L \\ E_A &= \alpha_4\end{aligned}\tag{4.34}$$

Observe que o vetor \vec{u} definido por estas fórmulas tem comprimento unitário.

Este método pode atribuir valores negativos ao coeficiente E_A , o que não faz sentido fisicamente. Caso isso aconteça podemos eliminar o termo correspondente ϕ_4 e repetir o cálculo usando a base ϕ_1, ϕ_2, ϕ_3 e as primeira quatro fórmulas (4.34) para determinar o coeficiente E_L e a direção \vec{u} .

Dependendo dos dados $\vec{\mu}_{\vec{u}}, \epsilon_{\vec{u}}, \mu_\rho, \epsilon_\rho, \mu_K, \epsilon_K$ e σ , o conjunto de amostras usáveis pode ser muito pequeno, e portanto a direção $\vec{\mu}_{\vec{u}}$ calculada pode ser pouco confiável. Neste caso, podemos utilizar um método iterativo. Em cada iteração, adotamos uma estimativa da direção \vec{u} e usamos as equações (4.32) a (4.34) para calcular os coeficientes E_L, E_A e uma nova estimativa de \vec{u} . Esta estimativa geralmente tem uma incerteza $\epsilon_{\vec{u}}$ menor que a estimativa inicial $\vec{\mu}_{\vec{u}}$ e portanto o conjunto de amostras válidas será maior. Repetimos estes cálculos até que a alteração nos valores de E_L, E_A e \vec{u} seja suficientemente pequena.

Uma limitação deste modelo é a impossibilidade de separar o efeito da luz de fundo uma vez que nas amostras consideradas o elemento lambertiano ϕ_L é uma combinação linear dos elementos ϕ_X, ϕ_Y e ϕ_Z . Por essa razão, se a luz de fundo for significativa, a normal calculada \vec{u} será distorcida na direção $-\vec{\xi}$.

4.6.2 Determinando \vec{u}, ρ, K e ξ por otimização não linear

O método da seção 4.6.1 é inerentemente impreciso e limitado; em geral sua utilidade é fornecer uma estimativa razoável da direção \vec{u} . A determinação precisa de \vec{u} e dos demais parâmetros não lineares (K e ρ) exige o uso de técnicas de otimização não linear. Para este fim, nós utilizamos o *método do simplex dividido* de Stolfi, Pereira e Saracchini [77]. Este é um algoritmo geral de otimização que tenta encontrar o vetor $\gamma^* \in \mathbb{R}^n$ que maximiza ou minimiza uma função suave $F : \mathbb{R}^n \rightarrow \mathbb{R}$ arbitrária. O ótimo γ^* é procurado dentro de uma esfera com centro $\gamma^{(0)}$ e raio $\epsilon^{(0)}$ dados.

Para aplicar este método ao nosso problema é necessário empacotar os parâmetros não lineares K, ρ e \vec{u} em um vetor de parâmetros escalares $\gamma = \gamma_1, \dots, \gamma_n$. Neste empacotamento, a direção \vec{u} é transformada por projeção estereográfica em dois parâmetros escalares γ_i e γ_{i+1} . O raio ρ da fonte e o polimento K contribuem com apenas um parâmetro cada. A escala de cada parâmetro γ_i é ajustada de modo que uma variação unitária desse parâmetro equivale à incerteza ($\epsilon_K, \epsilon_\rho$ ou $\epsilon_{\vec{u}}$) do parâmetro original correspondente. Se uma dessas incertezas for zero, o parâmetro é considerado fixo, e excluído do vetor γ .

A função objetivo $F(\gamma)$ a ser minimizada é o erro quadrático $Q(\tilde{L})$, onde \tilde{L} é dada pela fórmula (4.27). Para calcular esta fórmula, os valores de K, ρ e \vec{u} são extraídos, do vetor σ , e os coeficientes E_A, E_L, E_P, E_X, E_Y e E_Z são determinados resolvendo-se o sistema linear (4.27).

O vetor inicial $\gamma^{(0)}$ é definido pelas estimativas iniciais $\vec{\mu}_{\vec{u}}, \mu_K$ e μ_ρ , e o raio inicial $\epsilon^{(0)}$ é 1 (correspondente às incertezas $\epsilon_{\vec{u}}, \epsilon_K$ e ϵ_ρ).

4.7 Testes

Nesta seção, descrevemos testes que realizamos para avaliar a robustez e precisão dos métodos de aproximação descritos neste capítulo. Utilizamos quatro conjuntos de testes, dois sintéticos (**slamb** e **spoli**) e dois reais (**rlamb** e **rlatx**).

Nos testes sintéticos, o conjunto de amostras (\vec{n}_k, l_k) foi gerado a partir de aproximadamente 10.000 normais \vec{n}_k distribuídas uniformemente na região visível da esfera \mathbb{S}^2 . Para testes com gabaritos reais, extraímos uma amostra (\vec{n}_k, l_k) no centro de cada pixel da projeção \mathcal{G} do gabarito, excluindo pixels que tocam ou cruzam a borda de \mathcal{G} .

Para cada conjunto de dados, determinamos aproximações \tilde{L} nos três modelos descritos neste capítulo: harmônicos esféricos (seção 4.4.1, equação (4.11)), bases radiais (seção 4.4.2, equação (4.16)) e de luz compacta (seção 4.5.5, equação (4.27)). Para aproximação utilizando harmônicos esféricos escolhemos o grau máximo $M = 10$, resultando em uma base (**HARMONIC**) com 81 elementos. Como base radial utilizamos uma lista

(**RADIAL**) de 118 elementos escolhidos como descrito na seção 4.4.2 com centros sobre $t = 8$ círculos. Para estes dois modelos, os coeficientes foram determinados pelo sistema linear (4.6). Para o modelo de luz compacta (**COMPACT**) usamos o método da seção 4.6.1 para obter a direção inicial de \vec{u} , seguido de otimização não-linear para K, ρ, \vec{u} e ξ .

Em cada teste, a qualidade da aproximação foi analisada calculando o erro $E(\vec{n}) = \tilde{L}(\vec{n}) - L(\vec{n})$ nas normais amostradas $\vec{n}_1, \vec{n}_2, \dots, \vec{n}_N$. Calculamos também o erro máximo (E_{\max}) e a raiz quadrada da média do quadrado do erro (E_{rms}).

4.7.1 Gabarito sintético lambertiano com luz unidirecional

As amostras do conjunto de teste **slamb** foram obtidas de um gabarito sintético modelado por uma esfera com acabamento lambertiano e albedo unitário, com iluminação unidirecional de direção $\vec{u} = (0.500, 0.000, 0.866)$ e intensidade luminosa $E_L = 1$. Veja a figura 4.21.

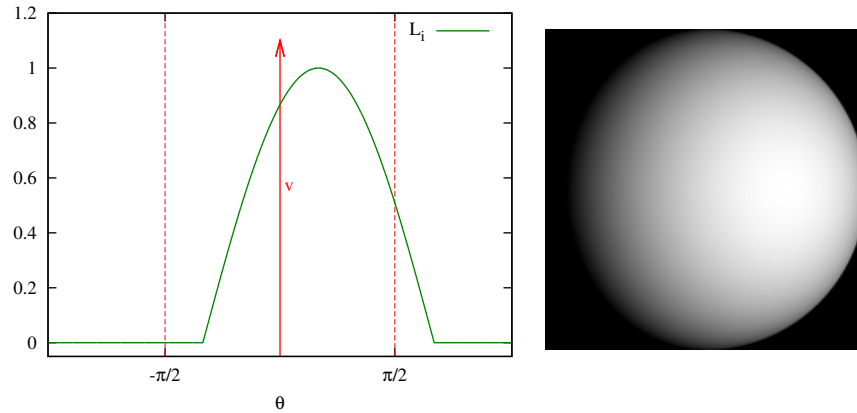


Figura 4.21: Função de tonalização $L(\vec{n})$ para do conjunto **slamb**, e imagem sintética do gabarito.

Para este teste, o modelo de luz compacta foi reduzido à componente ϕ_L , excluindo $\phi_P, \phi_A, \phi_X, \phi_Y$ e ϕ_Z . A direção \vec{u} foi determinada por otimização não linear com estimativa inicial $\vec{\mu}_{\vec{u}} = (0, 0, 1)$ e $\epsilon_{\vec{u}} = 0.1$ (radianos) e $\mu_\rho = \epsilon_\rho = 0$.

Todas as aproximações foram muito próximas ao original. Como esperado, a melhor aproximação foi obtida com o modelo de luz compacta. A otimização não linear deste último conseguiu recuperar os valores exatos de E_L e \vec{u} . O valor médio (no sentido de raiz da média dos quadrados) do erro $E(\vec{n})$ foi 6×10^{-6} . As aproximações com bases radiais e harmônicos esféricos, como esperado, tiveram problemas na região da linha do terminador. Veja as figuras 4.22, 4.23 e a tabela 4.1.

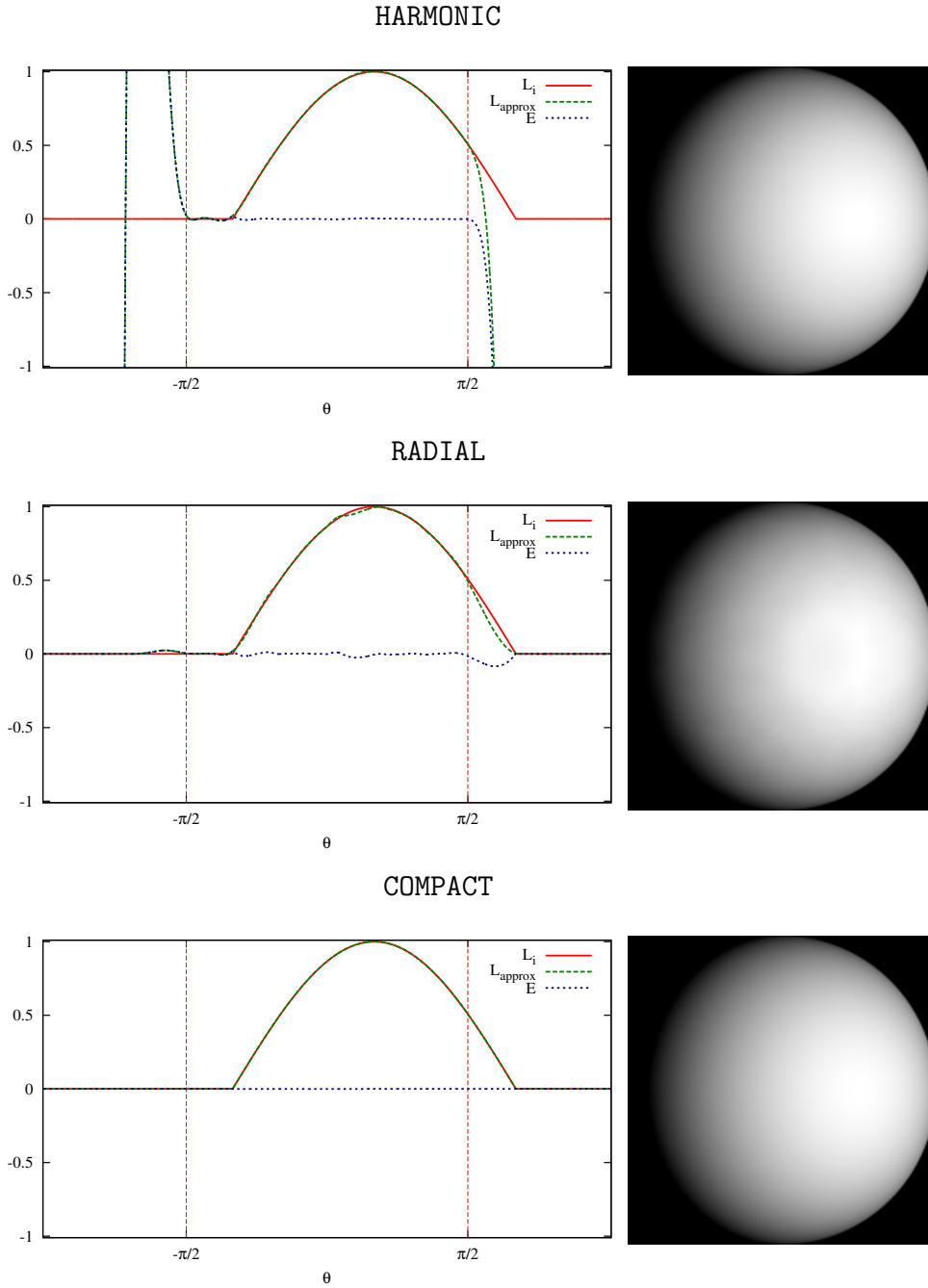


Figura 4.22: Na coluna da esquerda, gráficos da função de tonalização $L(\vec{n})$ (linha cheia), das aproximações $\tilde{L}(\vec{n})$ em cada modelo (linha tracejada) e dos erros $E(\vec{n}) = \tilde{L}(\vec{n}) - L(\vec{n})$ (linha pontilhada), ao longo de uma linha horizontal na imagem, em função do ângulo θ entre a normal \vec{n} e a direção de vista $(1, 0, 0)$. As linhas verticais tracejadas delimitam a região visível do gabarito. Na coluna da direita, imagem sintética do gabarito, calculada usando a aproximação $\tilde{L}(\vec{n})$.

Tabela 4.1: Erro máximo E_{\max} e erro médio E_{rms} (raiz da média dos quadrados) das aproximações.

Modelo	E_{\max}	E_{rms}
HARMONIC	0.028	4.0×10^3
RADIAL	0.027	5.0×10^3
COMPACT	0.000	6.0×10^6

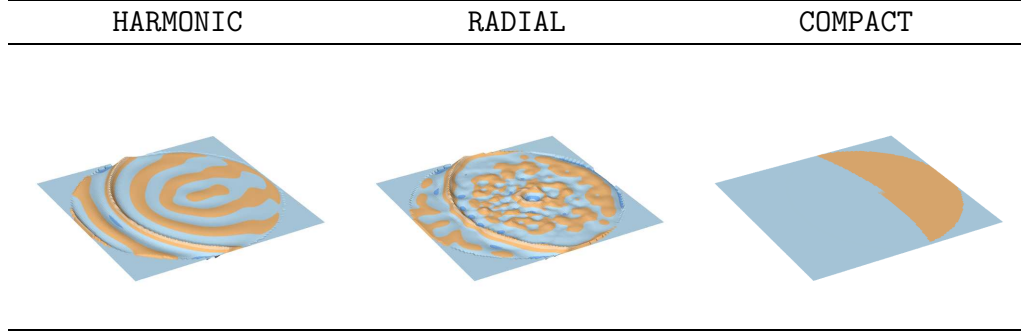


Figura 4.23: Gráficos dos erros de aproximação $E(\vec{n}) = \tilde{L}(\vec{n}) - L(\vec{n})$ para o teste **slamb** usando projeção estereográfica de \mathbb{H}^2 no plano \mathbb{R}^2 . Cada faixa de cor corresponde a uma diferença de 0.03.

4.7.2 Gabarito sintético lustroso com luz compacta

As amostras do conjunto de teste **spoli** foram obtidas de um gabarito sintético com função de tonalização descrita pelo modelo de luz compacta completo (4.27) com coeficientes acabamento parcialmente lustroso e parcialmente polido com lustrosidade $K = 10$. Para iluminação consideramos uma fonte na direção $\vec{u} = (0.500, 0.000, 0.866)$ com raio angular $\rho = 0.2$ e luz de fundo com direção $\xi = (0, 0.173, 0.985)$. Os coeficientes da fórmula (4.27) eram $E_L = 0.5$, $E_P = 0.4$, $\sqrt{E_X^2 + E_Y^2 + E_Z^2} = 0.1$ e $E_A = 0.1$ (correspondentes a $E_0 = 0$ e $E_1 = 0.1$). Veja a figura 4.24.

Para ajuste do modelo compacto com otimização não linear, utilizamos como estimativas iniciais $\vec{\mu}_{\vec{u}} = (0, 0, 1)$, $\mu_\rho = 0.3$, $\mu_K = 7$ e $\mu_\xi = (0, 0, 1)$ com incertezas $\epsilon_{\vec{u}} = 0.1$, $\epsilon_\rho = 0.3$, $\epsilon_K = 2.0$.

Nestes testes obtivemos aproximações adequadas \tilde{L} em todos os três modelos (harmônicos esféricos, bases radiais e luz compacta). Os erros da aproximações com bases de harmônicos foi menor neste teste do que no teste **slamb**, o que era esperado pois a função de tonalização deste teste é mais suave (C_1). O algoritmo de otimização não linear conseguiu recuperar os parâmetros do modelo de luz compacta, especificamente $\vec{u} = (0.494, 0.009, 0.869)$, $\rho = 0.19$, $K = 9.8$ e $\xi = (-0.007, 0.115, 0.993)$ com $E_L = 0.49$, $E_P = 0.4$, $E_A = 0.103$ e $\sqrt{E_X^2 + E_Y^2 + E_Z^2} = 0.103$. Veja a figuras 4.25, 4.26 e a tabela 4.2.

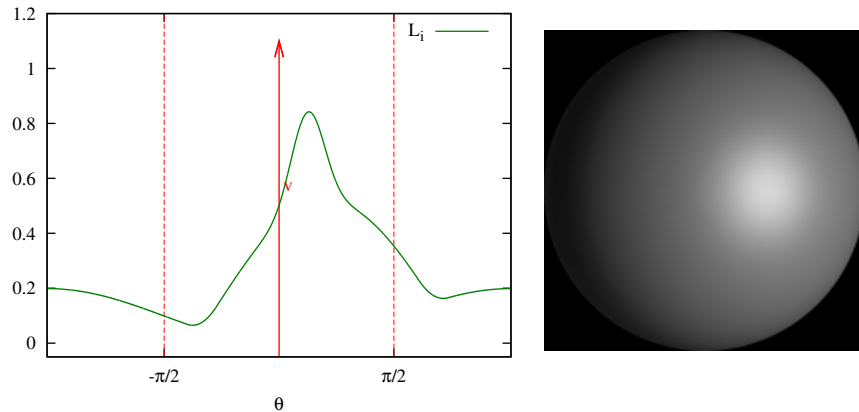


Figura 4.24: À esquerda, a função de tonalização $L(\vec{n})$ do conjunto de teste sintético **spoli**. À direita, imagem sintética do gabarito. Note que a luz de fundo é responsável pela iluminação não nula nas regiões de sombra própria da fonte compacta.

Tabela 4.2: Erro máximo E_{\max} e erro E_{rms} médio (raiz da média dos quadrados) para o gabarito **spoli**

Modelo	E_{\max}	E_{rms}
HARMONIC	0.013	2.0×10^3
RADIAL	0.033	5.0×10^3
COMPACT	0.028	9.0×10^3

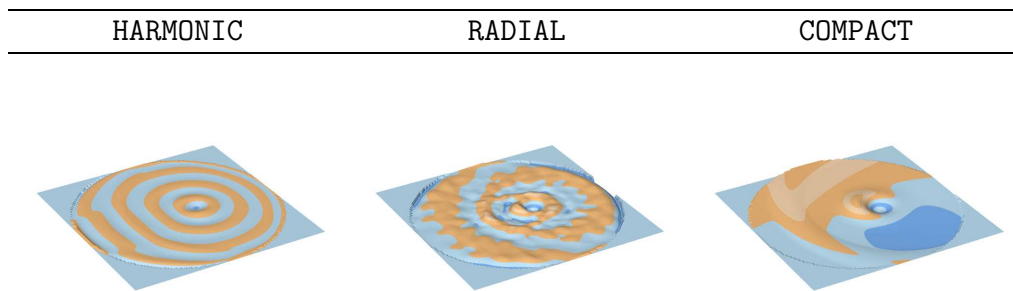


Figura 4.25: Gráficos dos erros $E(\vec{n}) = \tilde{L}(\vec{n}) - L(\vec{n})$ das aproximações para o teste **spoli** em projeção estereográfica de \mathbb{H}^2 em \mathbb{S}^2 . Cada faixa de cor corresponde a uma diferença de 0.03.

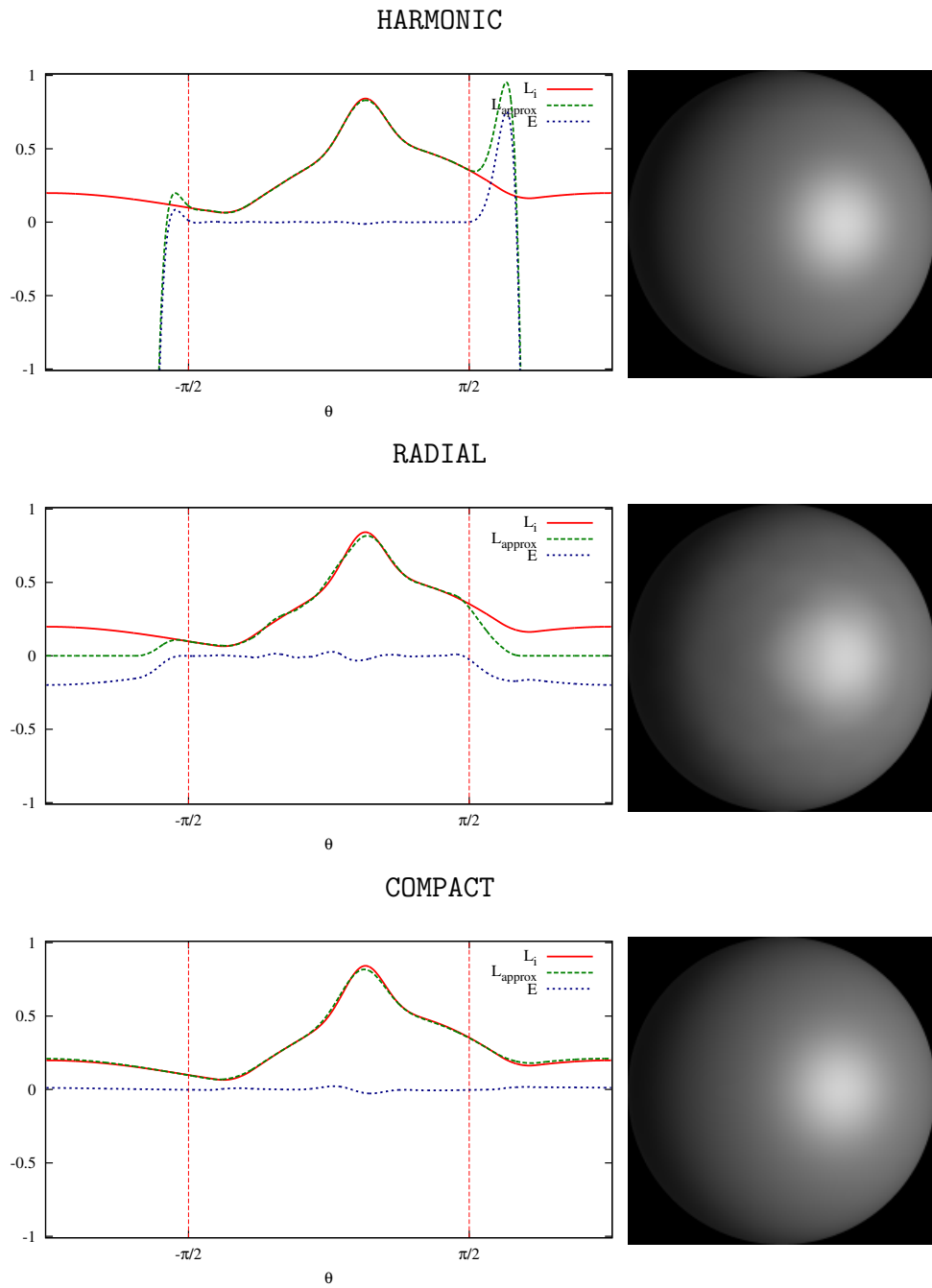


Figura 4.26: Na coluna da esquerda, gráficos da função de tonalização $L(\vec{n})$ (linha cheia) das aproximações $\tilde{L}(\vec{n})$ em cada modelo (linha tracejada) e do erro $E(\vec{n})$ (linha pontilhada), ao longo de uma linha horizontal na imagem, em função do ângulo θ entre a normal \vec{n} e a direção de vista $(1, 0, 0)$. As linhas verticais tracejadas indicam a região visível do gabarito. Na coluna da direita, imagem do gabarito calculado usando a aproximação \tilde{L} .

4.7.3 Esfera pintada

O conjunto de teste **rlamb** obtido de um gabarito real, consistindo de uma bola de tênis de mesa, pintada com tinta *spray* fosca branca. Este gabarito é adequado para cenas com acabamento aproximadamente lambertiano, tais como fragmentos de cerâmica e esculturas de barro. Obtivemos 5 fotos digitais desse gabarito com uma câmera Sony DSC-V50 de 3 Megapixels, posicionada em um tripé de fotógrafo a 119 cm de distância. O flash e o filtro de realce de nitidez da câmera foram desligados, e utilizamos captura temporizada para evitar vibrações indesejadas. A iluminação foi provida por uma lâmpada incandescente halogênica cilíndrica de 500W com um refletor $8\text{cm} \times 16\text{cm}$, que foi posicionada em várias direções e 1 m de distância do gabarito. As direções foram escolhidas aleatoriamente com inclinação θ variando de 0° a 25° . Veja a figura 4.27.



Figura 4.27: Imagens do gabarito **rlamb** para as cinco fontes de luz.

Para a otimização não-linear do modelo de luz compacta utilizamos apenas os elementos ϕ_L e ϕ_A com estimativas iniciais $\vec{\mu}_{\vec{u}} = (0, 0, 1)$, $\mu_\rho = 0.1$ e incertezas $\epsilon_{\vec{u}} = 0.25$, $\epsilon_\rho = 0.2$.

Os resultados das aproximações podem ser vistos nas figuras 4.28, 4.29 e na tabela 4.3. Todos os modelos forneceram aproximações $\tilde{L}(\vec{n})$ adequadas, sendo que a aproximação por harmônicos esféricos foi a mais precisa. Em geral, os erros maiores ocorreram ao longo da borda da projeção \mathcal{G} onde há mistura de pixels.

Tabela 4.3: Erros máximos e médios (raiz das médias dos quadrados) para o gabarito real **rlamb**.

	HARMONIC		RADIAL		COMPACT	
Luz	E_{\max}	E_{rms}	E_{\max}	E_{rms}	E_{\max}	E_{rms}
0	0.054	0.003	0.054	0.003	0.058	0.005
1	0.099	0.003	0.097	0.003	0.100	0.007
2	0.062	0.003	0.062	0.004	0.069	0.006
3	0.095	0.004	0.095	0.004	0.103	0.008
4	0.115	0.004	0.115	0.005	0.119	0.011

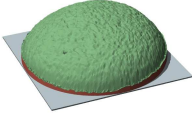
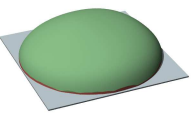
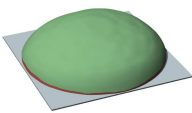
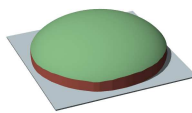
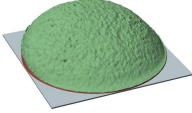
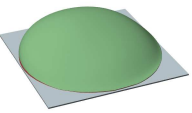
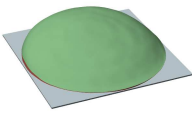
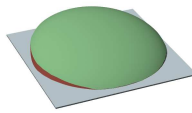
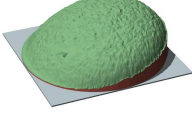
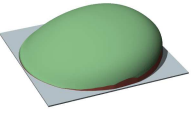
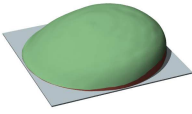
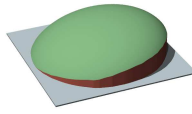
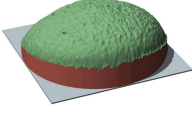
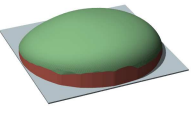
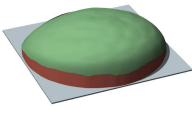
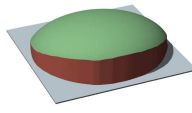
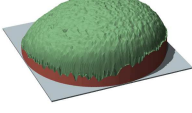
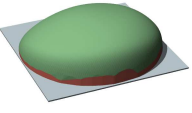
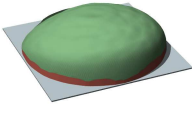
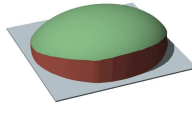
Luz i	Original	HARMONIC	RADIAL	COMPACT
1				
2				
3				
4				
5				

Figura 4.28: Gráficos das amostras l_k originais e das aproximações $\tilde{L}(\vec{n}_k)$ em cada modelo do gabarito real `rlamb`

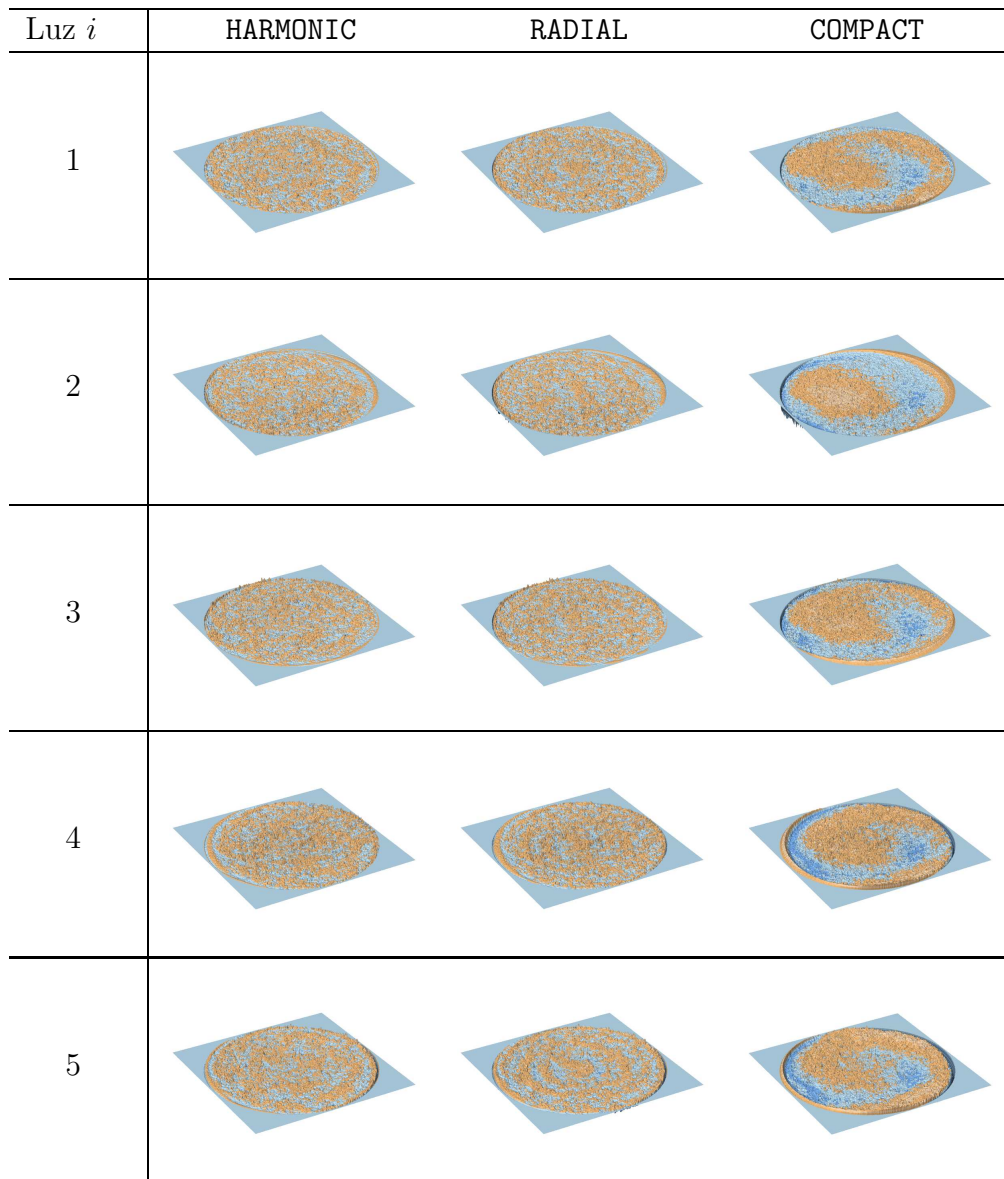


Figura 4.29: Gráficos dos erros de aproximação $E(\vec{n}_k) = \tilde{L}(\vec{n}) - l_k$ para o gabarito `rlamb`. Cada faixa de cor corresponde a uma diferença de 0.03.

4.7.4 Esfera de látex

O conjunto de dados `rlatx` foi obtido de uma bola de tênis de mesa coberta com duas películas de látex (bexigas de festa) esticadas. Este gabarito pretende emular o acabamento de pele humana e outros tecidos biológicos. Obtivemos 5 imagens deste gabarito com uma câmera AVT Pike FC-100 disparada remotamente e ajustada para produzir imagens coloridas de 500×500 pixels com codificação linear. Para estes testes, trabalhamos diretamente com as imagens brutas (formato RAW) debayerizadas, sem correções automáticas da câmera.

A iluminação foi fornecida por 5 luminárias de fibra ótica com haste posicionável, distribuídas em torno do gabarito a aproximadamente 15 cm de distância. Para garantir uma iluminação mais uniforme, foram colocados discos de papel vegetal na frente de cada luminária. A câmera foi posicionada a 30 cm da cena com uma lente de distância focal de 25mm resultando em um campo de visão de aproximadamente 6.5 cm de largura. A projeção de cada gabarito nas imagens é aproximadamente um círculo de 300 pixels de raio. Veja a figura 4.30.



Figura 4.30: Imagens do gabarito `rlatx` sob as 5 iluminações.

Para ajuste do modelo compacto com otimização não linear, utilizamos os elementos ϕ_L, ϕ_P e ϕ_A com estimativas iniciais $\vec{m}u_{\vec{u}} = (0, 0, 1)$, $\mu_\rho = 0.1$, $\mu_K = 35$ e incertezas $\epsilon_{\vec{u}} = 0.25$, $\epsilon_\rho = 0.2$ e $\epsilon_K = 2.0$.

Para este teste, todos os modelos ofereceram aproximações adequadas. As bases harmônicas apresentaram erro significativo na região do realce lustroso. Para modelar essa parte seria necessário usar uma base harmônica com um número muito maior de elementos, mas nesse caso o modelo também reproduziria muitas das imperfeições do gabarito. O modelo de bases radiais foi o que apresentou uma função aproximadora mais fiel. Quando usamos o modelo de luz compacta, o algoritmo de otimização de simplexo dividido teve algumas dificuldades para encontrar a direção correta da luz. No entanto, as aproximações são relativamente boas para todos os métodos. Veja a tabela 4.4 e as figuras 4.31 e 4.32.

Tabela 4.4: Erros máximos e médios (raízes das médias dos quadrados) para o gabarito `rlatx`.

	HARMONIC		RADIAL		COMPACT	
Luz	E_{\max}	E_{rms}	E_{\max}	E_{rms}	E_{\max}	E_{rms}
1	0.158	0.005	0.142	0.005	0.144	0.012
2	0.126	0.003	0.119	0.003	0.120	0.006
3	0.152	0.003	0.147	0.003	0.143	0.006
4	0.042	0.002	0.036	0.002	0.036	0.005
5	0.116	0.003	0.101	0.003	0.101	0.007

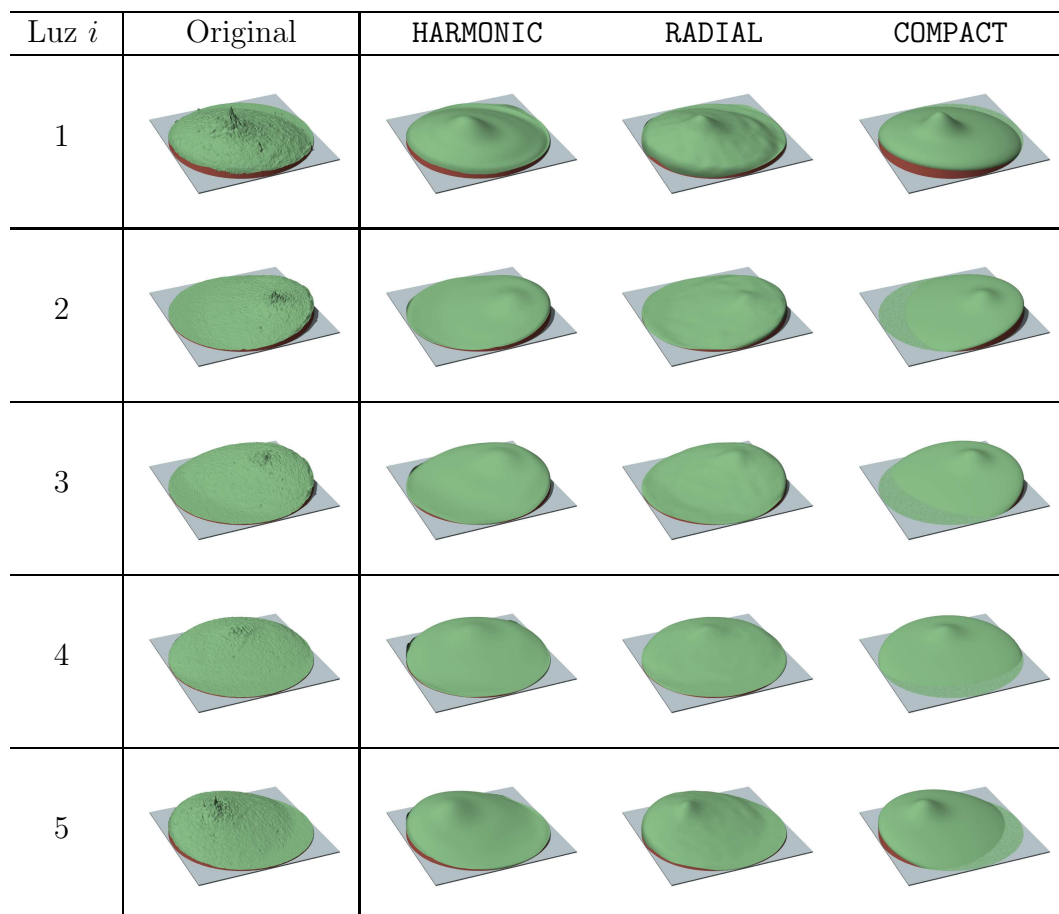


Figura 4.31: Gráficos das amostras l_k originais das aproximações $\tilde{L}(\vec{n}_k)$ para o gabarito real `rlatx`

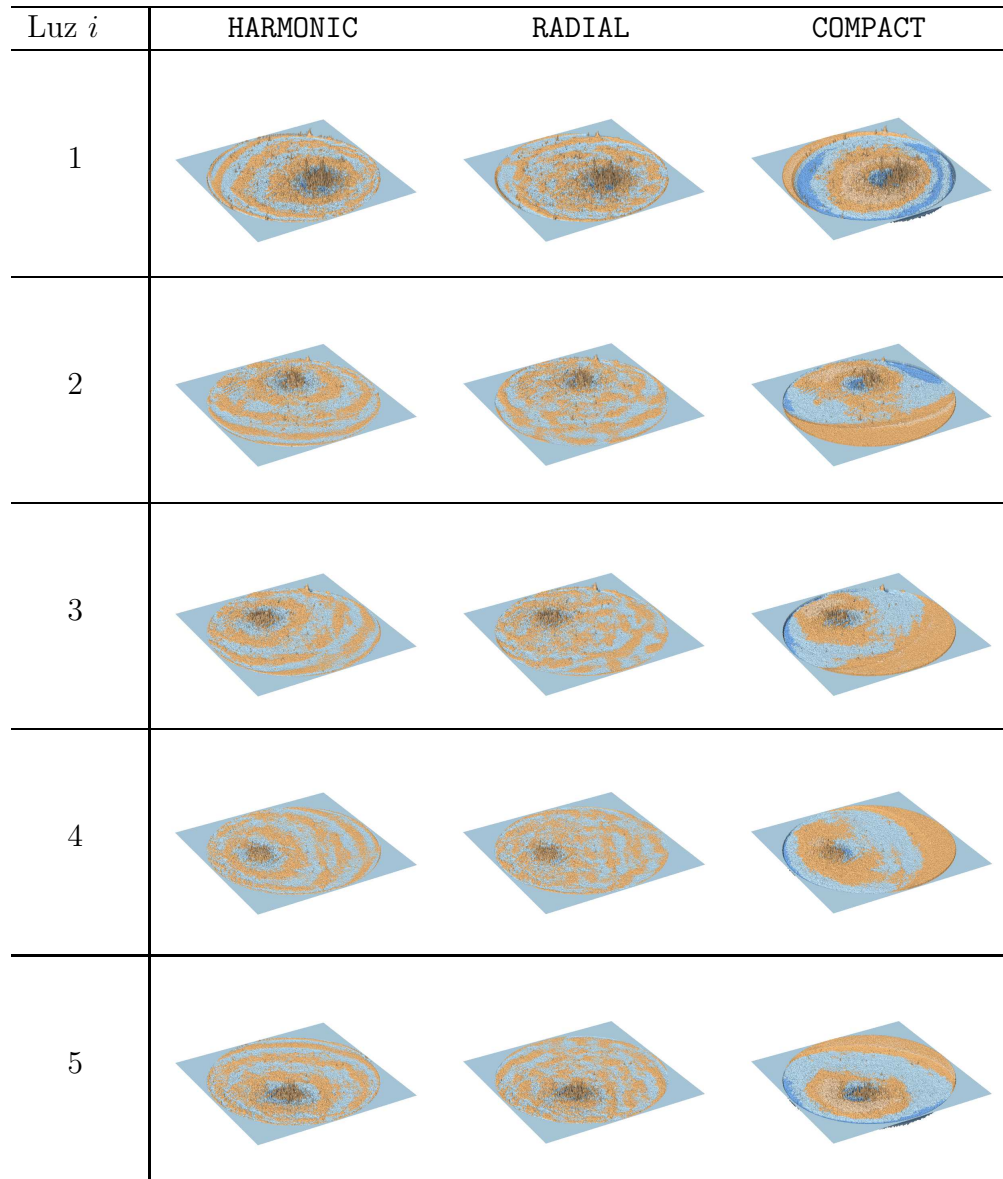


Figura 4.32: Gráficos dos erros de aproximação $E(\vec{n}) = \tilde{L}(\vec{n}) - l_k$ para as fontes de luz do gabarito `rlatx`. Cada faixa de cor corresponde a uma diferença de 0.03.

Capítulo 5

Inversão por busca em tabela

5.1 Tabela de assinaturas

Uma maneira de efetuar a inversão da função assinatura de tonalização \mathbf{l} (seção 2.6.2) é construir uma *tabela de assinaturas*, uma lista \mathbb{T} de pares $t_k = (\vec{n}_k, \mathbf{g}_k) \in \mathbb{S}^2 \times \mathbb{S}^{m-1}$ para $k = 1, 2, \dots, N$; onde os vetores \vec{n}_k formam um conjunto suficientemente denso de normais em \mathbb{H}^2 , e cada \mathbf{g}_k é a assinatura $\mathbf{l}(\vec{n}_k)$. A normal $\vec{n}[p]$ de cada pixel p pode então ser determinada procurando-se nessa tabela a assinatura \mathbf{g}_k mais próxima à assinatura $\mathbf{s}[p]$ do pixel, segundo alguma métrica “dist” do \mathbb{R}^m . Nesta tese, adotamos para métrica “dist”, a distância euclidiana do \mathbb{R}^m .

Com esta abordagem, a determinação da normal em EFBG fica reduzida a um problema de busca do elemento mais próximo em um conjunto de pontos do \mathbb{R}^m . Este método é extremamente geral e não depende de nenhuma hipótese sobre a FBDR e as iluminações, além das hipóteses básicas de uniformidade da iluminação, similaridade de acabamento e invertibilidade. Entretanto, para um resultado razoavelmente preciso é geralmente necessário usar uma tabela \mathbb{T} com dezenas de milhares de entradas. Por essa razão, a busca da entrada mais próxima na tabela \mathbb{T} costuma ser a etapa mais custosa deste método.

O algoritmo “força bruta” para esta busca consiste em calcular $\text{dist}(\mathbf{g}_k, \mathbf{s}[p])$ para toda entrada da tabela, e escolher a que tem distância mínima. Este algoritmo (que denominaremos por BRUTE) calcula N distâncias no espaço \mathbb{R}^m , e portanto o limitante assintótico superior e inferior [30] para seu custo é $\Theta(mN)$. Para imagens grandes, esta busca exaustiva pode levar horas, mesmo para um número pequeno de iluminações.

Na literatura encontram-se algoritmos mais eficientes (vide seção 5.2) que efetuam esta busca em tempo médio $O(m \log N)$. Desenvolvemos uma técnica baseada em uma grade bidimensional [7]. Com esta técnica, o tempo esperado médio de busca por pixel foi reduzido para $O(m)$.

5.2 Trabalhos anteriores

Diversas abordagens foram propostas na literatura para acelerar a busca na tabela \mathbb{T} . Como mencionado no capítulo 3, Woodham [107] propôs a utilização de uma grade m -dimensional, cobrindo o hipercubo $[0, 1]^m$, com 2^b células ao longo de cada eixo. Na etapa de pré-processamento, cada assinatura de observação \mathbf{g}_k do gabarito é quantizada na forma de m inteiros com b bits por coordenada, e esses m inteiros são usados para identificar uma célula da grade onde a normal correspondente \vec{n}_k é armazenada. (Woodham na verdade usou os vetores de observação \mathbf{G}_k e não as assinaturas normalizadas, mas como ele supunha albedos uniformes e iguais para a cena e o gabarito, o resultado era o mesmo). Na etapa de busca, cada assinatura de observação $\mathbf{s}[p]$ da cena é quantizada da mesma forma e usada para localizar a célula contendo a normal desejada. A desvantagem óbvia deste método é o tamanho da grade: 2^{mb} células, ou seja, aproximadamente 250,000 células para $m = 3$ e $b = 6$.

Trabalhos posteriores propuseram a utilização de vários algoritmos gerais para busca do vizinho mais próximo no espaço m -dimensional. Uma solução clássica para este problema é a *árvore de busca k -dimensional* (*k -d tree*) [19], que (como veremos) não é muito eficiente. Hertzman e Seitz [53] propuseram o uso de um algoritmo para busca do vizinho mais próximo aproximado (*approximate nearest neighbor*, ou ANN) [11], enquanto Zhou e Little [116] utilizaram *hashing* sensível à localidade (*locally sensitive hashing*, ou LSH) [58] (Ver seção 5.6).

Em teoria, estes algoritmos reduzem o tempo médio de busca de $O(mN)$ para $O(m \log N)$. Estes métodos possuem, entretanto, uma limitação comum: eles supõem que o conjunto de assinaturas é uma nuvem genérica de pontos espalhada num espaço m -dimensional, e portanto utilizam algoritmos de busca de pontos m -dimensionais genéricos — que são inerentemente caros em termos de espaço e tempo, com perda de eficiência progressiva quando m aumenta: a chamada “maldição da dimensionalidade” [58]. De modo geral, o benefício destes algoritmos só começa a ser percebido quando $N \approx 2^m$, ou seja, eles são ineficazes quando $m \gtrsim 15$. Além disso, alguns destes métodos, como ANN, são inerentemente imprecisos e nem sempre retornam a melhor entrada da tabela.

5.3 Busca rápida com grade de baldes 2D

Supondo que o acabamento β do gabarito seja livre de picos (como os produzidos por reflexão especular) e outras descontinuidades, e que as fontes de luz sejam extensas, as funções de tonalização $L_i(\vec{n})$ dadas pela fórmula (2.15) são contínuas. De fato, se tanto o acabamento quanto o campo de iluminação forem suaves, a função de tonalização será bem suave, com máximos bem esparsos e pouco distintos.

A observação chave para nosso método é que o conjunto das assinaturas \mathbb{G} da tabela é uma cobertura razoavelmente densa e uniforme de $\mathbb{K} = \mathbf{l}(\mathbb{H}^2)$, a imagem do hemisfério de normais visíveis de \mathbb{H} através da função assinatura de tonalização \mathbf{l} . Supondo que \mathbf{l} seja suave, o conjunto \mathbb{K} é uma variedade bidimensional contida na esfera \mathbb{S}^{m-1} de \mathbb{R}^m . Na verdade, o conjunto \mathbb{K} de assinaturas válidas está contido no quadrante positivo de \mathbb{R}^m , cujo diâmetro angular é 90 graus visto da origem de \mathbb{R}^m . Verificamos além disso que o conjunto \mathbb{K} é razoavelmente próximo a um subespaço afim bidimensional (plano) P de \mathbb{R}^m . Veja a figura 5.1. Essa figura ilustra com precisão a distância das assinaturas de \mathbb{G} em relação ao plano P e apoia nossa afirmação de que o conjunto \mathbb{G} é relativamente plano em sua maior parte.

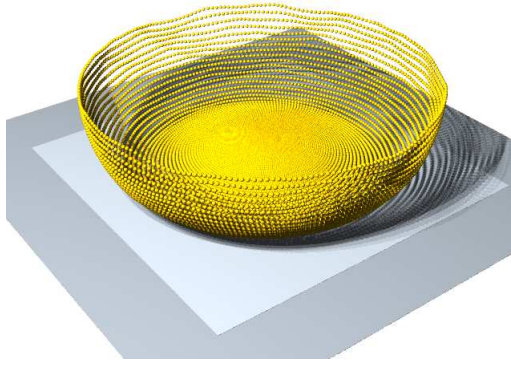


Figura 5.1: Visualização de um conjunto \mathbb{G} de amostras de $\mathbb{K} = \mathbf{l}(\mathbb{H}^2)$ e de sua posição relativa a um plano aproximador P , para a função assinatura de tonalização \mathbf{l} de uma superfície lambertiana com $m = 48$ iluminações unidirecionais. Cada esfera amarela representa uma amostra $\mathbf{g}_k = \mathbf{l}(\vec{n}_k)$ para alguma direção $\vec{n}_k \in \mathbb{H}^2$. A posição horizontal da esfera é a projeção ortogonal da assinatura \mathbf{g}_k no plano P ; a região cinza escura é o quadrilátero $[-1, +1] \times [-1, +1]$ neste plano. A posição vertical é sua distância euclidiana do plano P no espaço \mathbb{R}^m . A mesma escala é usada nos três eixos.

Note que os deslocamentos de cada assinatura em relação a sua projeção *não* são vetores paralelos no espaço \mathbb{R}^m , como pode parecer neste gráfico. No espaço \mathbb{R}^m , o conjunto \mathbb{G} pode estar torcido de maneira complicada.

A busca da normal de uma assinatura de observação $\mathbf{s}[p]$ então pode ser feita projetando-se a mesma em P e examinando-se as entradas da tabela cujas assinaturas têm projeções em P próximas à de $\mathbf{s}[p]$.

5.3.1 A grade 2D

A primeira etapa do nosso método é a escolha do plano P , o *plano de projeção de assinaturas*. Para isso calculamos o centróide \mathbf{b} do conjunto \mathbb{G} de todas as assinaturas de \mathbb{T} , vistas

como um conjunto de pontos de \mathbb{R}^m ; e dois vetores unitários ortogonais $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$ que definem as direções de maior dispersão de \mathbb{G} . Estes vetores são determinados construindo a matriz $M \in \mathbb{R}^{m \times m}$ de momentos das coordenadas das diferenças $\mathbf{g}_k - \mathbf{b}$, e escolhendo os autovetores unitários de M associados aos dois maiores autovalores. O ponto \mathbf{b} e os vetores \mathbf{u}, \mathbf{v} definem o plano P , que é aproximadamente coplanar com \mathbb{G} . A projeção ortogonal em P de uma dada assinatura \mathbf{g} será denotada por $\mathbf{g} \downarrow P$.

Em seguida, nós escolhemos uma grade uniforme de $K \times K$ células quadradas no plano de projeção P . Esta grade é centrada em \mathbf{b} , possui seus lados paralelos aos vetores \mathbf{u} e \mathbf{v} e é larga o suficiente para conter a projeção $\mathbf{g} \downarrow P$ de qualquer assinatura \mathbf{g} em \mathbb{G} com uma pequena margem de segurança ε . Mais especificamente, a grade cobre um quadrado de tamanho $2R$ em P , onde

$$R = \varepsilon + \max \{ |(\mathbf{g} - \mathbf{b}) \cdot \mathbf{u}|, |(\mathbf{g} - \mathbf{b}) \cdot \mathbf{v}| : \mathbf{g} \in \mathbb{G} \} \quad (5.1)$$

Cada célula da grade é portanto um quadrado com lado $\tau = 2R/K$.

Tendo escolhido a grade, nós construímos, para cada célula $C[i, j]$ da mesma, um *balde* $B[i, j]$, que é uma lista ligada de todas as entradas $(\mathbf{g}_k, \vec{n}_k)$ da tabela \mathbb{T} cujas assinaturas \mathbf{g}_k são projetadas naquela célula. Calculamos também a *média do balde* $\mu[i, j]$, definida como o baricentro de todas as assinaturas \mathbf{g}_k na lista $B[i, j]$; e o *raio do balde* $\rho[i, j]$, definido como a distância máxima de $\mu[i, j]$ para qualquer assinatura \mathbf{g}_k nessa lista. Veja a figura 5.2.

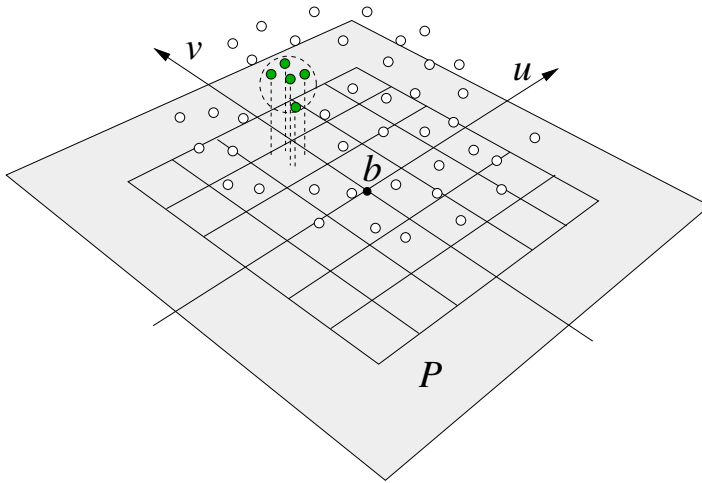


Figura 5.2: Ilustração da grade de baldes 2D, para uma tabela com $m = 3$ iluminações, mostrando algumas assinaturas de observação em \mathbb{G} (pequenos círculos abertos), o plano de projeção de assinaturas P (quadrilátero cinza), o baricentro \mathbf{b} e os eixos principais \mathbf{u}, \mathbf{v} de \mathbb{G} , a grade de células no plano P , uma lista de baldes $B[i, j]$ (pequenos círculos preenchidos), e a esfera delimitadora (círculo grande tracejado) definida pelo centróide $\mu[i, j]$ e raio $\rho[i, j]$.

O formato bidimensional do conjunto \mathbb{K} significa que as assinaturas que caem no mesmo balde $B[i, j]$ são razoavelmente próximas entre si, mesmo quando suas distâncias ao plano P são muito maiores que o tamanho da célula. Verificamos que esta propriedade mantém-se (e acentua-se) quando m aumenta.

5.3.2 O algoritmo de busca

Uma vez construída a grade de baldes, a melhor entrada correspondente a uma assinatura da cena $\mathbf{s}[p]$ pode ser calculada pelo procedimento BGRID descrito abaixo. Seus passos são explicados nas seções 5.3.3 a 5.3.5.

Procedimento 1 (BGrid:) *Dada uma assinatura de observação \mathbf{s} , encontrar a entrada $t_{best} = (t_{best}.g, test.\vec{n})$ na tabela \mathbb{T} com assinatura $t_{best}.g$ mais próxima de \mathbf{s} .*

1. $i \leftarrow \lfloor ((\mathbf{s} - \mathbf{b}) \cdot \mathbf{u} + R)/\tau \rfloor$;
2. $j \leftarrow \lfloor ((\mathbf{s} - \mathbf{b}) \cdot \mathbf{v} + R)/\tau \rfloor$;
3. $dbest \leftarrow +\infty$; $t_{best} \leftarrow \mathbf{null}$;
4. Para cada par (δ_i, δ_j) em Δ , ordenadamente, faça
 5. Se $dbest \leq \tau \Lambda(\delta_i, \delta_j)$, retorne t_{best} .
 6. $(i', j') \leftarrow (i, j) + (\delta_i, \delta_j)$;
 7. Se $0 \leq i' < K$ e $0 \leq j' < K$, então
 8. Se $dbest > \text{dist}(\mathbf{s}, \mu[i', j']) - \rho[i', j']$, então
 9. Para cada t em $B[i', j']$, faça
 10. Defina $d \leftarrow \text{dist}(\mathbf{s}, t.g)$;
 11. Se $d < dbest$ defina $dbest \leftarrow d$ e $t_{best} \leftarrow t$.
 12. Retorne t_{best} .

O procedimento BGRID inicia pelo cálculo dos índices (i, j) da célula que contém a projeção $\mathbf{s} \downarrow P$ da assinatura \mathbf{s} (passos 1–2). Ele então varre a lista $B[i, j]$ em busca da entrada mais próxima de \mathbf{s} (passos 9–11). Se necessário, a busca continua em baldes $B[i', j']$ próximos a $B[i, j]$ em uma ordem apropriada (passos 4–8). Note que alguns dos baldes podem estar vazios, e que a melhor entrada para \mathbf{s} pode não estar em $B[i, j]$ —mesmo quando este balde não é vazio.

5.3.3 Condição de varredura do balde

Os atributos $\mu[i, j]$ e $\rho[i, j]$ do balde permitem descartar rapidamente baldes que não têm chance de conter um melhor candidato para a assinatura \mathbf{s} que o encontrado até o momento. Especificamente, vale a pena examinar um balde $B[i', j']$ apenas se a assinatura

procurada \mathbf{s} for mais próxima à esfera delimitadora do balde do que da melhor assinatura já encontrada (passo 8). Isto é, apenas se

$$\text{dist}(\mathbf{s}, \mu[i', j']) - \rho[i', j'] < \text{dist}(\mathbf{s}, t_{\text{best}}.\mathbf{g}) = d_{\text{best}} \quad (5.2)$$

Nós chamamos esta condição (5.2) de *condição de varredura de balde*.

5.3.4 Ordem de busca de baldes e retorno precoce

A condição de varredura de balde (5.2) frequentemente permite ignorar um balde $B[i', j']$, sem a necessidade de examinar seu conteúdo. Entretanto, se aplicarmos este critério a todos os baldes individualmente, o tempo de execução ainda será excessivo; especificamente, proporcional a K^2 , o total de células na grade — que, como em qualquer algoritmo de *hash*, é idealmente proporcional ao número N de entradas na tabela. Nessas condições, o custo de BGRID seria $\Theta(N)$, diferindo do custo do algoritmo BRUTE apenas por um fator constante.

Para evitar examinar a grade inteira, examinamos os baldes $B[i', j']$ em uma ordem específica, começando pelo balde $B[i, j]$ que contém a projeção $\mathbf{s} \downarrow P$ e prosseguindo com baldes gradualmente mais afastados (passo 4). Um segundo critério (passo 5) nos permite abandonar a busca tão logo detectamos que nenhum dos baldes remanescentes pode conter um melhor candidato do que o já escolhido. Tipicamente, isto acontece após examinar uma pequena fração da grade.

A ordem de busca dos baldes vizinhos é definida por uma lista Δ de pares de inteiros, construída como descrito a seguir. Considere duas assinaturas \mathbf{s}' e \mathbf{s}'' que são projetadas respectivamente nas células $C[i', j']$ e $C[i'', j'']$. É fácil ver que a distância entre as assinaturas no espaço \mathbb{R}^m é maior ou igual à distância entre as células no plano P , isto é

$$\text{dist}(\mathbf{s}', \mathbf{s}'') \geq \text{dist}(C[i', j'], C[i'', j'']) \quad (5.3)$$

Nesta fórmula, $\text{dist}(C[i', j'], C[i'', j''])$ é a distância mínima entre duas células, vistas como subconjuntos de P , que é dada pela fórmula

$$\text{dist}(C[i', j'], C[i'', j'']) = \tau \Lambda(i' - i'', j' - j'') \quad (5.4)$$

onde

$$\Lambda(\delta_i, \delta_j) = \sqrt{(|\delta_i| \ast 1)^2 + (|\delta_j| \ast 1)^2} \quad (5.5)$$

e onde

$$x \ast y = \max \{0, x - y\} \quad (5.6)$$

Note que o valor de $\Lambda(\delta_i, \delta_j)$ é um pouco menor que a norma euclidiana $|(\delta_i, \delta_j)| =$

$\sqrt{\delta_i^2 + \delta_j^2}$ do par de inteiros.

Concluimos que um balde $B[i', j']$ pode ser ignorado se a restrição (5.3) para a distância excluir a possibilidade de que ele possa conter uma entrada melhor que a melhor já encontrada; isto é, se

$$\tau \Lambda(i' - i, j' - j) \geq dbest \quad (5.7)$$

lembrando que $\text{dist}(\mathbf{s}, tbest.\mathbf{g}) = dbest$.

Note que a condição (5.7) é mais fraca que a condição de varredura de balde (5.2). Entretanto, a condição (5.7) depende apenas da melhor entrada $tbest$ já encontrada e a das diferenças $i' - i$ e $j' - j$ entre os índices das células. Portanto, se examinarmos os baldes $B[i', j']$ numa ordem tal que $\Lambda(i' - i, j' - j)$ seja crescente, podemos interromper a busca tão logo a condição (5.7) seja satisfeita (passo 5).

Com este propósito, como parte do pré-processamento da tabela, nós construímos uma lista Δ de todos os pares de inteiros (δ_i, δ_j) tais que $|\delta_i|, |\delta_j| < N$, em ordem crescente de $\Lambda(\delta_i, \delta_j)$. Para cada assinatura \mathbf{s} da cena, definimos a ordem de enumeração dos baldes vizinhos $B[i', j']$ tomando os vetores de deslocamento (δ_i, δ_j) da lista Δ , nessa ordem. Para cada um desses pares, calculamos o par de índices i', j' do novo balde por $(i', j') \leftarrow (i, j) + (\delta_i, \delta_j)$ (passo 6), descartando índices que estiverem fora do intervalo $\{0, \dots, N-1\}$ (passo 7).

Na prática, é mais eficiente trabalhar com o quadrado $\Lambda^2(\delta_i, \delta_j)$ do limitante (5.5), que é um inteiro, em vez de $\Lambda(\delta_i, \delta_j)$. Veja a figura 5.3.

8	5	4	4	4	5	8	45	41	33	27	34	42	46
5	2	1	1	1	2	5	37	21	17	11	18	22	38
4	1	0	0	0	1	4	29	13	05	03	06	14	30
4	1	0	0	0	1	4	25	09	01	00	02	10	26
4	1	0	0	0	1	4	31	15	07	04	08	16	32
5	2	1	1	1	2	5	39	23	19	12	20	24	40
8	5	4	4	4	5	8	47	43	35	28	36	44	48

Figura 5.3: a função quadrática de distância das células $\Lambda^2(\delta_i, \delta_j)$ (esquerda) e a ordem de varredura dos baldes vizinhos ao balde inicial na lista Δ (direita) para uma grade 4×4 .

5.3.5 Análise

O custo médio do algoritmo 1 para um pixel p da cena é aproximadamente $Bb + Dd + O(1)$, onde b é o número médio de baldes examinados (nos passos 5–7), d a média do número de funções dist de distância de assinatura calculadas (nos passos 8 e 10), e B, D são os custos associados a estas duas operações, respectivamente.

No caso extremo quando $K = 1$, teremos $b = 1$ e $d = N + 1$, que é essencialmente uma busca por força bruta em \mathbb{T} . Este também é o pior caso teórico, quando todas as assinaturas são projetadas no mesmo balde. Entretanto, dada a maneira que a projeção é feita, isto acontece apenas quando as assinaturas são muito similares em todo o gabarito. Por sua vez, isso indica que todas as imagens foram obtidas com iluminação quase isotrópica, de maneira que a função \mathbf{l} é independente da normal da superfície. Obviamente tais dados são inadequados para estéreo fotométrico.

No outro extremo, quando K é muito grande, espera-se que a maioria dos baldes contenha apenas zero ou uma entrada, de modo que o parâmetro d será minimizado. Ao mesmo tempo, b aumentará imediatamente por um fator de 10, uma vez que $\Lambda(\delta_i, \delta_j)$ é zero para os primeiros 9 pares (δ_i, δ_j) da lista Δ . Em seguida, b crescerá lentamente em proporção a K^2 , já que o procedimento terá que percorrer um número cada vez maior de baldes vazios antes de encontrar alguma assinatura. O valor ótimo de K que minimiza o tempo de execução depende principalmente da razão B/D , e do formato da nuvem de assinaturas \mathbb{G} no espaço \mathbb{R}^m . Nossos testes indicaram que os efeitos de b crescente e d decrescente tendem a se cancelar de maneira que o tempo de execução é quase independente do tamanho da grade para densidades $\kappa < 0.5$. Em geral, o tempo de busca tem um mínimo aproximado quando a densidade κ está entre $\kappa = 1/4$ e $\kappa = 1/16$, isto é, quando K está entre $2\sqrt{N}$ e $4\sqrt{N}$. Veja as tabelas de 5.1 a 5.4.

5.4 Testes

Para avaliar nosso método, nós realizamos testes com imagens sintéticas e com fotos de uma cena real. Embora imagens sintéticas não sejam suficientes para validar algoritmos EFBG, elas permitem avaliar o desempenho do procedimento de busca para assinaturas da cena ideais, livres de perturbações, erros de alinhamento das imagens, iluminação não-uniforme, ruídos de medida e quantização e variações de acabamento. Por outro lado, testes com fotos de cenas reais são necessários para avaliar o impacto dessas perturbações na precisão e eficiência do algoritmo.

Repetimos cada um destes testes com cinco diferentes tamanhos de grade $K \times K$: 422×422 , 299×299 , 211×211 , 149×149 e 105×105 , correspondendo a razões aproximadas entradas/baldes $\kappa = N/K^2$ de 6%, 12%, 25%, 50% e 100%, respectivamente. Nós também processamos cada imagem com uma grade de tamanho $K = 1$, caso em que o procedimento BGRID é essencialmente equivalente a uma busca por força bruta. Nós aplicamos o algoritmo a cada canal de cor (R , G , e B) separadamente para obter os respectivos albedos $a[p, c]$, e ao canal de luminância $Y = 0.2989 R + 0.5866 G + 0.1145 B$ para obter a normal $\vec{n}[p]$.

A precisão do método foi avaliada comparando as normais calculadas com as normais

corretas, quando estas são conhecidas. Os erros (medidos em radianos) foram resumidos pelo máximo E_{\max} e pela raiz da média dos quadrados E_{rms} . Para fins de ilustração apenas, nós também calculamos um mapa de alturas $z[p]$ a partir do mapa de normais, usando nosso integrador multi-escala descrito no capítulo 7.

O algoritmo BGRID foi implementado na linguagem C e compilado pelo compilador GNU GCC sem otimização. Os testes foram efetuados em um computador com processador Intel Xeon de 2.5Ghz com 8GB de memória.

5.4.1 Testes com imagens sintéticas de cena lambertiana

Para os testes com imagens sintéticas, construímos um modelo geométrico tridimensional de uma cena virtual simples, consistindo de blocos de diversos formatos e cores posicionados sobre uma superfície plana. Veja a figura 5.4.

Para todos os objetos, usamos acabamento lambertiano com cores variadas. Em particular, a base tem textura xadrez, e o objeto localizado no canto inferior esquerdo é uma calota esférica pintada com tons degradê de azul-cinza, de maneira a dar uma falsa impressão de relevo.

Usamos o traçador de raios POV-Ray [111] para gerar um conjunto **sglamb** de $m = 48$ imagens sintéticas desta cena. Em cada imagem, a iluminação foi fornecida por uma fonte unidirecional. As direções das fontes foram distribuídas uniformemente, com inclinação máxima $\theta_{\max} = 25^\circ$ com a vertical. A opção **shadowless** do traçador foi utilizada para suprimir sombras projetadas, mantendo as sombras próprias. Veja a figura 5.5.

Cada fonte de luz foi posicionada de modo que sua distância da cena era aproximadamente 50.000 vezes a largura H da região visível desta última — grande o suficiente para garantir iluminação uniforme em toda a imagem. A distância da câmera virtual à cena foi definida como $500 H$, de maneira a obter projeção quase paralela e ortogonal ao plano da base. As cenas do POV-Ray foram geradas no formato PNG colorido [85] com 16 bits por amostra e codificação linear. Elas foram reduzidas de 630×450 para 420×300 pixels com filtro de Lanczos [32] e convertidas para o formato PPM [50] de 16 bits por amostra.

Para este teste, usamos como gabarito virtual, a função vetor de tonalização \mathbf{L} de uma superfície lambertiana com as iluminações unidirecionais usadas pelo POV-Ray. Para o procedimento BGRID, geramos a tabela \mathbb{T} por amostragem da função de tonalização \mathbf{L} , com $N = 11172$ normais \vec{n}_k cobrindo todo o hemisfério \mathbb{H}^2 .

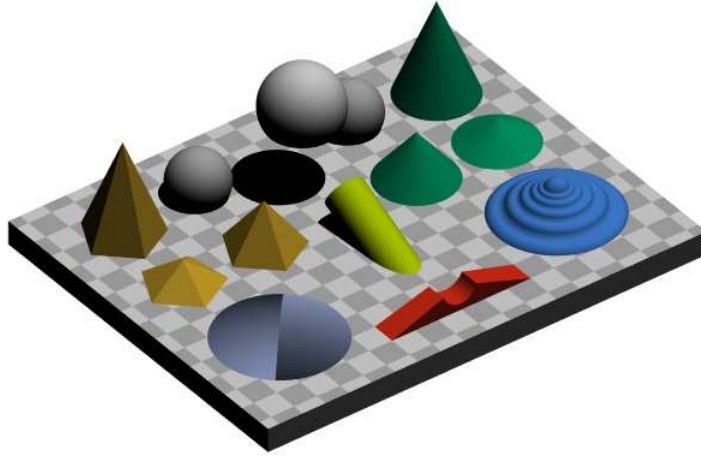


Figura 5.4: Visão em perspectiva da cena virtual usada para produzir os conjuntos **sglamb** e **sglstr**.

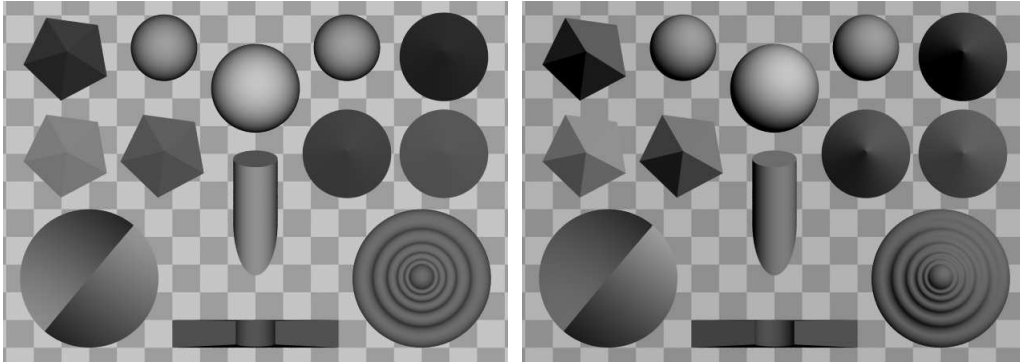


Figura 5.5: Duas das 48 imagens de entrada sintéticas do conjunto **sglamb**, para fontes de luz com inclinação $\theta = 5^\circ$ (esquerda) e $\theta = 14^\circ$ (direita).

A figura 5.6 ilustra a posição das assinaturas $\mathbf{g}_k = \mathbf{L}(\vec{n}_k)$ da tabela em relação ao plano de projeção P . Pode-se observar que a maioria das assinaturas é bem próxima de P ; as assinaturas \mathbf{g}_k mais distantes de P correspondem a normais \vec{n}_k mais afastadas da vertical. No gráfico da esquerda (para $m = 6$) são visíveis as descontinuidades pronunciadas de ordem 1 (“quinas”) correspondentes aos terminadores das luzes; enquanto que estas descontinuidades são quase imperceptíveis no gráfico da direita (para $m = 48$).

A figura 5.7 mostra as médias $\mu[i, j]$ e os raios $\rho[i, j]$ da grade de $K \times K = 75 \times 75$ baldes construída a partir desta tabela (com razão entradas/balde $\kappa \approx 1.98$). Assim como na figura 5.6, ressaltamos que os vetores $\mu[i, j] - c[i, j]$ não são necessariamente paralelos em \mathbb{R}^m , como parecem na figura.

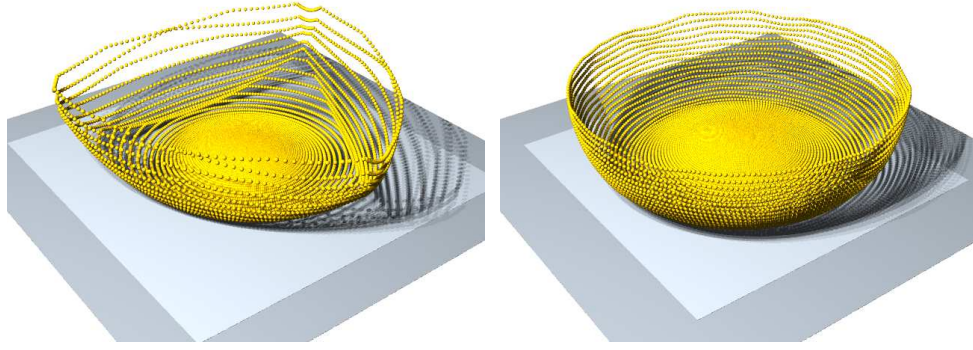


Figura 5.6: Posição das assinaturas amostradas \mathbf{g}_k do gabarito relativas ao plano P , com iluminações unidirecionais com inclinação máxima $\theta_{\max} = 25^\circ$, para $m = 6$ (esquerda) e $m = 48$ (direita) imagens do conjunto de teste **sglamb**. A região cinza escura é o quadrado $[-1, +1] \times [-1, +1]$ no plano de projeção P , e a região cinza claro é a região $[-R, +R] \times [-R, +R]$ coberta pela grade. Cada esfera representa uma assinatura $\mathbf{g}_k = \mathbf{l}(\vec{n}_k)$. A posição horizontal de cada esfera é a projeção ortogonal $\mathbf{g}_k \downarrow P$ da assinatura. A distância da esfera ao plano é a distância euclidiana $|\mathbf{g}_k - \mathbf{g}_k \downarrow P|$ no espaço \mathbb{R}^m . A mesma escala é usada nos três eixos da figura.

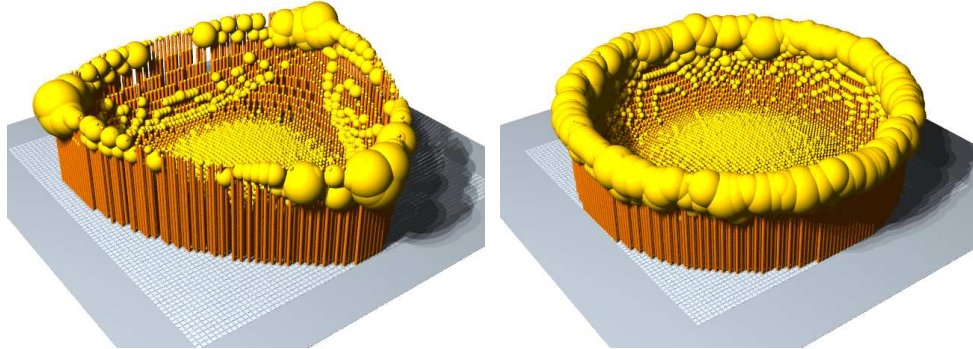


Figura 5.7: Visualização da grade de baldes para o conjunto de testes **sglamb**, com $K \times K = 75 \times 75$ células para $m = 6$ (esquerda) e $m = 48$ (direita) imagens. A região cinza mais clara é o quadrado $[-R, +R] \times [-R, +R]$ de P coberto pelas células da grade. A altura de cada palito é a distância do centroide $\mu[i, j]$ das entradas de cada balde para o centro $c[i, j]$ das células no plano P . O raio da esfera é o raio $\rho[i, j]$ do balde. A mesma escala é usada para todas estas posições e dimensões.

A figura 5.8 mostra o número de entradas em cada balde $B[i, j]$. Nesta figura, o tamanho da grade é $K \times K = 211 \times 211$, correspondendo a uma razão entradas/baldes $\kappa = 0.25$. Note que as assinaturas de observação ficaram distribuídas quase uniformemente sobre uma porção substancial da grade.

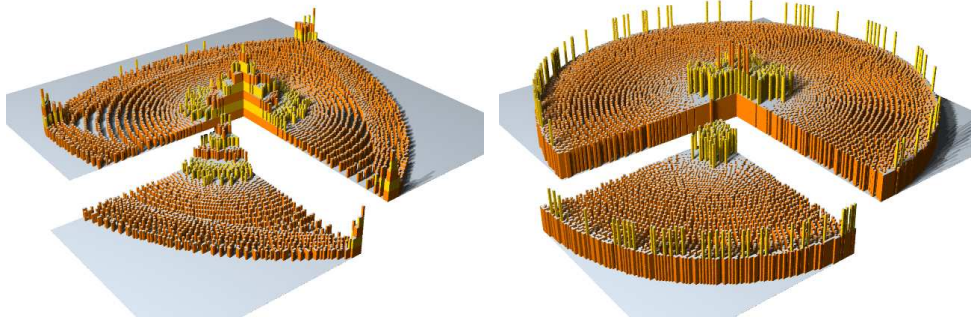


Figura 5.8: Número de entradas em cada balde da grade para o gabarito do conjunto de testes **sglamb** com $K = 211$, usando $m = 6$ (esquerda) e $m = 48$ (direita) luzes de inclinação máxima $\theta_{\max} = 25^\circ$. O quadrilátero cinza mostra região $[-R, +R] \times [-R, +R]$ do plano P coberta pela grade. Cada listra colorida nas barras representa uma entrada a mais no balde. Os baldes mais cheios possuem 8 entradas (esquerda) e 3 entradas (direita).

A figura 5.9 mostra o mapa de normais \vec{n} calculado pelo procedimento BGRID para $m = 24$ imagens de entrada, e a figura 5.10 mostra o mapa de cor intrínseca calculado a partir das mesmas imagens. Resultados visualmente idênticos foram obtidos para todos os valores de $m \geq 6$.

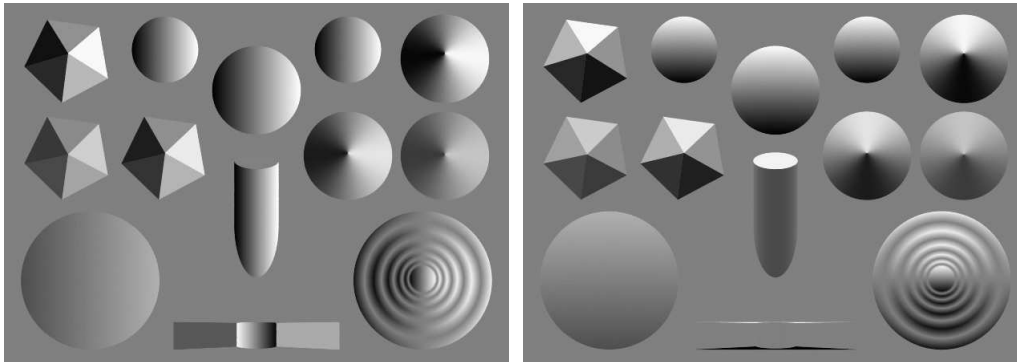


Figura 5.9: As componentes $\vec{n}.x$ e $\vec{n}.y$ do mapa de normais calculado pelo procedimento BGRID a partir do canal Y de $m = 24$ imagens do conjunto **sglamb**.

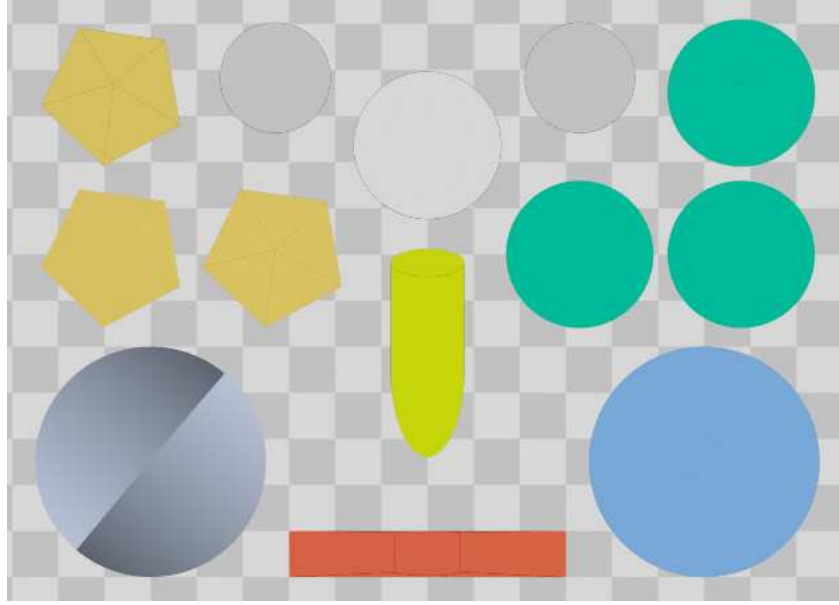


Figura 5.10: O mapa de cor intrínseca calculado a partir de $m = 24$ imagens sintéticas coloridas do conjunto **sglamb**, pelo procedimento BGRID aplicado aos canais R, G e B separadamente.

A figura 5.11 mostra o erro no mapa de normais, isto é, a diferença entre os vetores normais $\vec{n}[p]$ calculados pelo procedimento BGRID e os vetores normais corretos (determinados a partir do modelo tridimensional da cena). Atribuímos estes erros a diversos fatores: mistura de cores nos pixels que contêm descontinuidades e silhuetas de objetos, amostragem inadequada do objeto nas regiões onde a superfície é quase vertical, e erro de quantização nas imagens de entrada.

A figura 5.12 mostra uma visualização do mapa de alturas obtido pela integração do mapa de normais e imageado obliquamente com POV-Ray. A distorção visível dos cones e das pirâmides nesta imagem não é causada por erro nas normais, mas por esse efeito inevitável de paralaxe, decorrente da distância finita da câmera à cena.

A tabela 5.1 mostra o desempenho médio do algoritmo BGRID para processamento de um único pixel, (uma única busca na tabela). As colunas são: o número m de imagens de entrada, o tamanho K da grade, a taxa de ocupação da grade $\kappa = N/K^2$, o tempo de busca t em milissegundos, o número b de baldes $B[i', j']$ examinados e o número d de entradas da tabela comparadas (i.e., o número de cálculos de $\text{dist}(\mathbf{s}, \mathbf{g})$ efetuados). Os valores foram obtidos calculando-se as médias desses parâmetros sobre todos os pixels de \mathcal{S} . Quando $K > 1$, o valor do parâmetro d inclui também o cálculo dos produtos escalares $(\mathbf{s} - \mathbf{b}) \cdot \vec{u}$ e $(\mathbf{s} - \mathbf{b}) \cdot \vec{v}$ dos passos 1 e 2, cada qual considerado equivalente a um cálculo da distância dist .

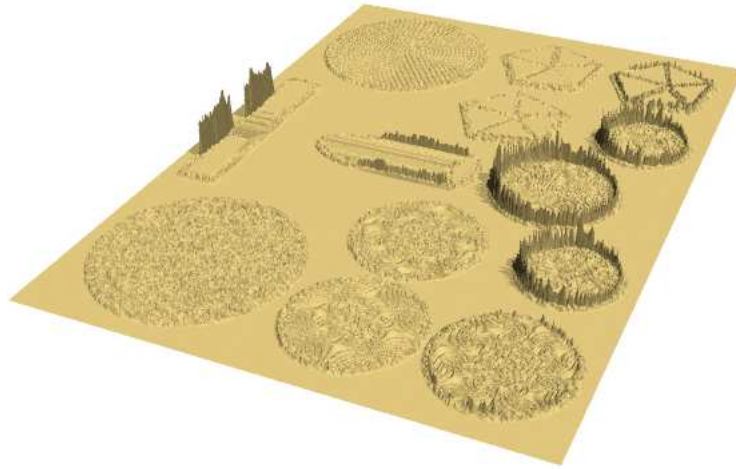


Figura 5.11: Gráfico dos erros das normais calculadas da figura 5.9. O erro máximo E_{\max} corresponde a um ângulo de 0.16 radianos (aproximadamente 9°) e o erro médio E_{rms} a 0.009 radianos (0.52°).

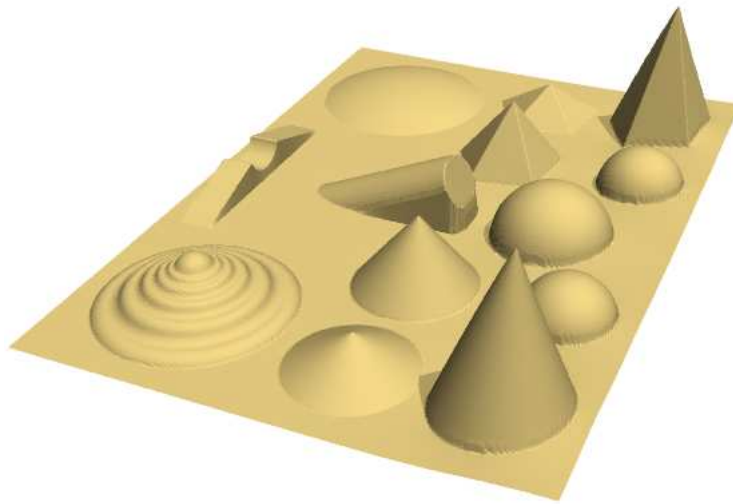


Figura 5.12: Visualização do mapa de alturas obtido pela integração do mapa de normais da figura 5.9.

Observe que a quantidade b de baldes examinados para de diminuir quando a taxa de ocupação κ ultrapassa 0.25, apesar de que o número d de distâncias calculadas continua diminuindo. Esse fenômeno é devido ao fato de que os baldes $B[i, j]$ não vazios começam a tornar-se cada vez mais esparsos.

Tabela 5.1: Custos médios por pixel do procedimento BGRID para imagens sintéticas do conjunto **sglamb** com diversos valores de m e K . As entradas com $K = 1$ correspondem à busca por força bruta.

m	K	κ	t	d	b
6	1	—	1499.7	11172.0	1.0
6	105	1.01	5.9	27.9	10.2
6	149	0.50	4.7	18.4	10.3
6	211	0.25	4.0	14.2	10.6
6	299	0.12	3.7	11.3	11.3
6	422	0.06	3.7	10.1	12.6
12	1	—	2120.5	11172.0	1.0
12	105	1.01	6.6	18.0	10.4
12	149	0.50	6.0	14.0	10.8
12	211	0.25	5.2	10.3	11.5
12	299	0.12	4.8	8.5	13.1
12	422	0.06	4.5	5.3	16.4
24	1	—	3745.2	11172.0	1.0
24	105	1.01	9.9	15.4	10.4
24	149	0.50	9.4	13.7	10.9
24	211	0.25	8.3	10.0	11.7
24	299	0.12	7.8	7.6	13.5
24	422	0.06	7.1	5.0	17.0
48	1	—	7110.6	11172.0	1.0
48	105	1.01	17.1	15.0	10.5
48	149	0.50	15.8	12.5	11.0
48	211	0.25	14.2	9.8	11.9
48	299	0.12	12.9	6.8	14.0
48	422	0.06	11.7	4.4	18.0

5.4.2 Testes com imagens sintéticas de cena lustrosa

Em outra série de testes, examinamos o comportamento do algoritmo BGRID para superfícies com acabamento lustroso. Para tanto, utilizamos a mesma cena virtual da seção 5.4.1 com albedo uniforme e acabamento parcialmente lambertiano e parcialmente lustroso de Phong com polimento $K = 10$. As iluminações unidirecionais (com $\rho_i = 0$) são as mesmas do conjunto **sglamb**, mas com uma luz de fundo com direção $\vec{\xi} = (0, 0.173, 0.985)$. Como gabarito virtual \mathbf{L} , usamos o modelo COMPACT (4.27), com os coeficientes $E_L = 0.5$, $E_P = 0.4$, $\sqrt{E_X^2 + E_Y^2 + E_Z^2} = 0.1$ e $E_A = 0.1$ (correspondentes a $E_0 = 0$ e $E_1 = 0.1$). Devido à dificuldade de reproduzir esta tonalização com o POV-Ray, aplicamos a função \mathbf{L} ao mapa de normais dessa cena, obtendo um conjunto **sglstr** de

$m = 48$ imagens sintéticas com sombras próprias, mas não projetadas. A tabela \mathbb{T} foi gerada amostrando a função vetor de tonalização \mathbf{L} . Veja a figura 5.13.

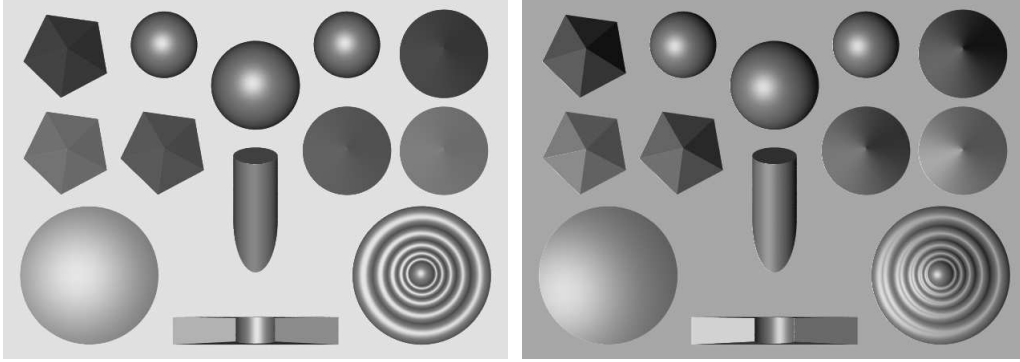


Figura 5.13: Duas das 48 imagens de entrada sintéticas do conjunto **sglstr**, com fontes de luz de inclinação $\theta = 5^\circ$ (esquerda) e $\theta = 20^\circ$ (direita).

Analogamente aos testes da seção 5.4.1, geramos a tabela \mathbb{T} por amostragem da função de tonalização \mathbf{L} , em $N = 11172$ normais \vec{n}_k cobrindo o hemisfério \mathbb{H}^2 . A figura 5.14 mostra a posição das assinaturas da tabela em relação ao plano de projeção P . Observe que o conjunto \mathbb{G} dessas assinaturas ainda é relativamente plano; entretanto sua curvatura é consideravelmente maior nas bordas, que correspondem a normais com inclinação próxima de 90° e em um anel menor que corresponde as normais com inclinação em torno de 35° . No primeiro caso, a curvatura é devida aos termos de luz ambiente e de fundo que são mais intensos para essas normais. No segundo caso, a dobra ocorre para normais onde o termo lustroso e lambertiano mais interferem entre si. Nota-se inclusive que assinaturas de normais bem distintas são projetadas sobre o mesmo ponto (u, v) de P . Observe também que a região de P coberta pelas projeções é menor do que na figura 5.6 devido à presença do termo constante E_A na função \mathbf{L} .

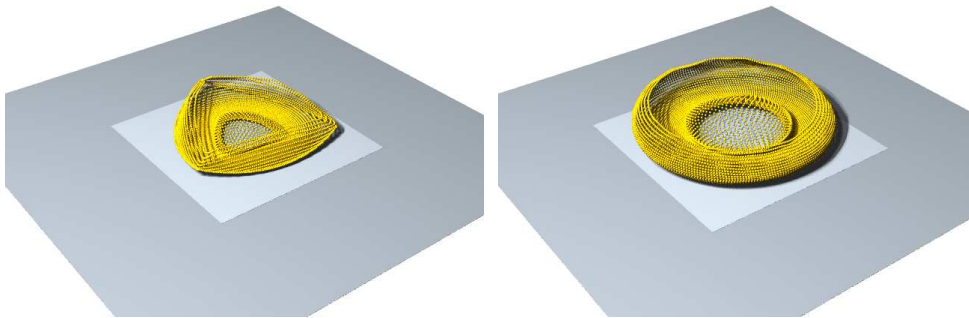


Figura 5.14: Posição das assinaturas da tabela em relação ao plano P , com $m = 6$ luzes (esquerda) e $m = 48$ (direita) para o conjunto de teste **sglstr**.

A figura 5.15 mostra as médias $\mu[i, j]$ e raios $\rho[i, j]$ dos baldes da grade de $K \times K = 75 \times 75$ células ($\kappa \approx 1.98$) construída a partir desta tabela. A figura 5.16 mostra o número de entradas em cada balde $B[i, j]$. Observe que os baldes na periferia tem raios muito maiores, refletindo a “dobra” do conjunto \mathbb{G} . Ressaltamos porém que, apesar de terem a mesma projeção, as assinaturas de normais contidas nesses baldes são bem distintas no espaço \mathbb{R}^m . Portanto, essas colisões na tabela afetam o custo do algoritmo, mas não o resultado, uma vez que este é sempre a entrada da tabela com assinatura mais próxima a \mathbf{s} .

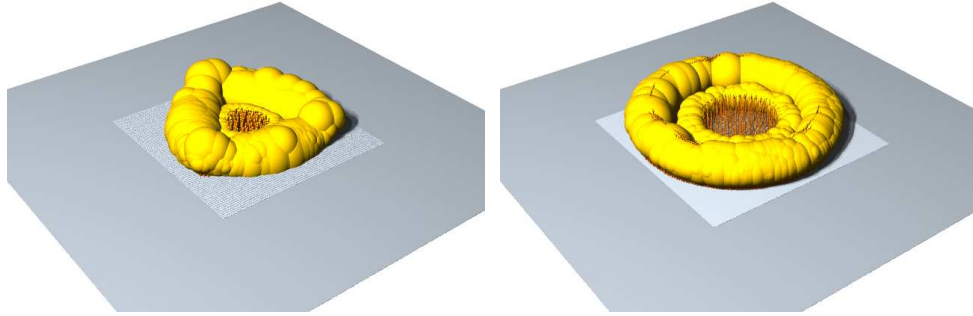


Figura 5.15: Visualização das médias $\mu[i, j]$ e raios $\rho[i, j]$ dos baldes da grade com $K \times K = 75 \times 75$ células, para o teste com imagens sintéticas **sglstr**, com inclinação máxima $\theta_{\max} = 25^\circ$ da fonte luminosa, $m = 6$ (esquerda) e $m = 48$ (direita) imagens.

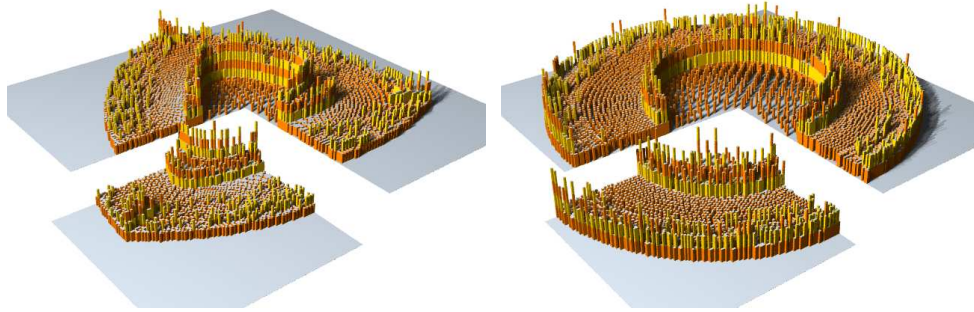


Figura 5.16: Número de entradas em cada balde da grade de $K \times K = 211 \times 211$ células, para o conjunto de dados **sglstr**, usando $m = 6$ (esquerda) e $m = 48$ (direita) imagens. Os baldes mais cheios possuem 8 entradas (esquerda) e 4 entradas (direita).

A figura 5.17 mostra o mapa de normais \vec{n} calculado pelo procedimento BGRID para $m = 24$ imagens de entrada. Resultados visualmente idênticos foram obtidos para todos os testes com $m \geq 6$ de imagens deste conjunto. O mapa de alturas obtido pela integração destas normais também é visualmente idêntico ao calculado para o conjunto **sglamb**.

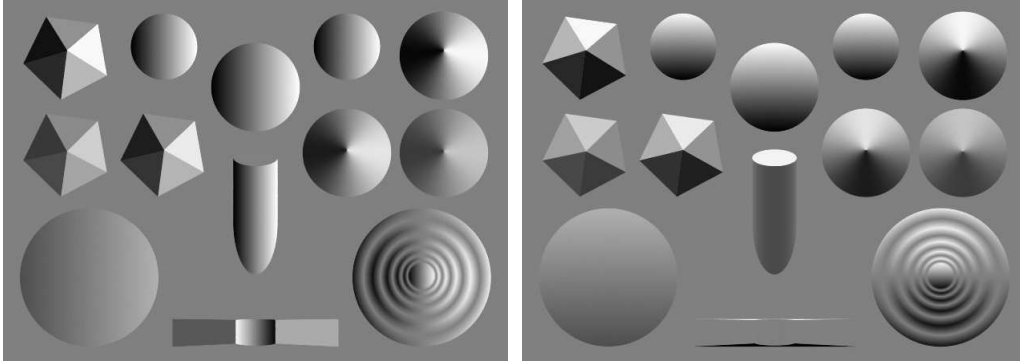


Figura 5.17: As componentes $\vec{n}.x$ e $\vec{n}.y$ do mapa de normais calculado por BGRID a partir do canal Y de $m = 24$ imagens do conjunto **sglstr**.

A figura 5.18 mostra o erro no mapa de normais, isto é, a diferença entre os vetores normais $\vec{n}[p]$ calculados por BGRID e os vetores normais corretos (determinados a partir do modelo tridimensional da cena). Observamos que o erro no cálculo das normais é um pouco menor que no teste **sglamb**. Isto se deve possivelmente ao fato que as imagens S_i da cena foram calculadas aplicando a função de tonalização \mathbf{L} ao mapa de normais corretos, evitando a mistura de assinaturas ao longo da silhueta dos objetos.

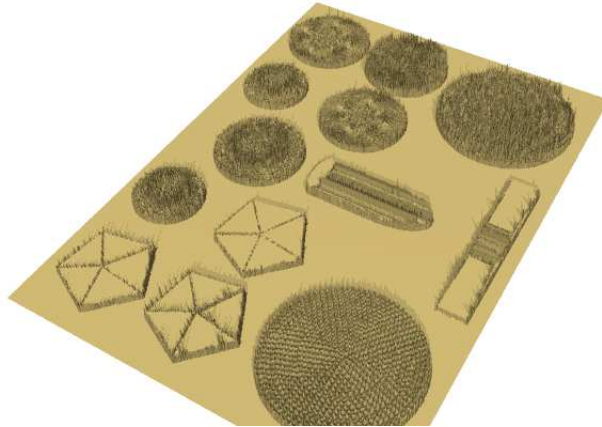


Figura 5.18: Gráfico dos erros das normais calculadas da figura 5.17. O erro máximo E_{\max} corresponde a um ângulo de 0.04 radianos (aproximadamente 2°) e o erro médio E_{rms} a 0.007 radianos (0.43°).

A tabela 5.2 mostra o desempenho médio do procedimento BGRID para um pixel, com os mesmos parâmetros e convenções da tabela 5.1. Pode-se ver que o desempenho do algoritmo é bem próximo ao observado com o conjunto **sglamb**, porém a quantidade de baldes examinados b é significativamente maior quando $\kappa > 0.25$. O benefício esperado para valores mais elevados de K é sobrepujado por este efeito.

Tabela 5.2: Custos médios por pixel do procedimento BGRID para imagens sintéticas com diversos valores de m e K . As entradas com $K = 1$ correspondem à busca por força bruta.

m	K	κ	t	d	b
6	1	—	1084.91	11172.00	1.00
6	105	1.01	3.36	19.34	10.51
6	149	0.50	2.80	11.97	11.39
6	211	0.25	2.80	8.21	13.14
6	299	0.12	2.80	6.03	16.56
6	422	0.06	2.80	4.80	23.61
12	1	—	1759.83	11172.00	1.00
12	105	1.01	5.04	15.95	11.80
12	149	0.50	4.48	10.68	13.80
12	211	0.25	3.92	7.61	17.99
12	299	0.12	3.92	6.01	26.06
12	422	0.06	5.04	5.24	42.01
24	1	—	3517.42	11172.00	1.00
24	105	1.01	8.40	15.65	11.92
24	149	0.50	6.72	10.61	14.17
24	211	0.25	6.16	7.68	18.50
24	299	0.12	6.72	6.08	26.93
24	422	0.06	7.28	5.33	43.86
48	1	—	6907.14	11172.00	1.00
48	105	1.01	14.00	15.45	12.30
48	149	0.50	12.32	10.57	14.90
48	211	0.25	10.64	7.74	19.96
48	299	0.12	10.64	6.25	29.81
48	422	0.06	11.20	5.53	49.22

5.4.3 Testes com fotos de uma cena real

Para testar o algoritmo BGRID em condições mais realistas, utilizamos um conjunto `rglamb` de fotos de uma cena real, uma escultura de gesso pintada com tinta guache. Veja a foto 5.19. Como gabaritos, usamos quatro bolas de tênis de mesa, pintadas com tinta esmalte *spray* de acabamento fosco e cor branca. Cada gabarito foi colocado em um cone raso de papel preto, para protegê-lo de luz espalhada pela cena. Incluímos na cena também dois gabaritos de calibração de tons escala de cinza (discos com diversos tons de cinza, abaixo e acima da cena) para identificar a linearidade da codificação das amostras nas fotos.

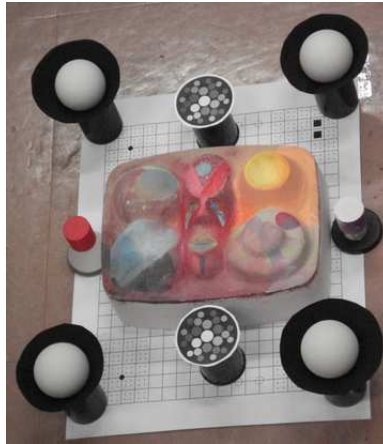


Figura 5.19: Foto em perspectiva da cena usada nos testes realistas do conjunto **rglob**. Observe os quatro gabaritos de iluminação colocados dentro dos cones protetores (nos cantos) e as duas escalas de tons de cinza (acima e abaixo da escultura).

Nós obtivemos 48 fotos digitais da cena com uma câmera Sony DSC-V50 de 3 megapixels, posicionada em um tripé de fotógrafo 119 cm acima da folha de papel e apontada para baixo. Veja a figura 5.20. O flash e o filtro de realce de nitidez da câmera foram desligados, e utilizamos o temporizador da câmera para reduzir a vibração durante a captura. A iluminação foi fornecida por uma lâmpada halogênica cilíndrica de 500W com um refletor $8\text{cm} \times 16\text{cm}$, que foi posicionada aproximadamente dois metros de distância da cena, em diversas direções com inclinação θ variando de 0° a 25° graus. A captura foi realizada à noite, com a iluminação da sala desligada, mas sem nenhuma precaução especial para evitar iluminação secundária devida à luz refletida pelas paredes e mobília.



Figura 5.20: Três das 48 fotos da cena **rglob**.

As fotos foram gravadas com a qualidade ‘fine’ (2816×2122 pixels) no formato JPEG. Corrigimos a distorção radial, alinhamos as imagens com precisão de ± 0.25 pixels, recortamos a região de interesse (834×1158 pixels) contendo a escultura e a reduzimos para 278×386 pixels com filtro Lanczos, a fim de reduzir o ruído da câmera e artefatos da compressão JPEG.

O campo de visão relativamente largo (cerca de 30 cm de largura a 119 cm de distância, ou 14°) resultou em distorção de paralaxe nas bordas da imagem. Particularmente, as projeções dos gabaritos ficaram distintamente elípticas, e a projeção do ponto com normal $(0,0,1)$ ficou perceptivelmente deslocada do centro da elipse. Estas distorções foram levadas em conta na construção da tabela de assinaturas, como descrito na seção 4.2.

Para este conjunto de testes, utilizamos apenas o gabarito localizado no canto inferior direito da figura 5.19. Observamos que as funções de tonalização L_i extraídas deste gabarito foram afetadas pelo ruído da câmera e por pequenos defeitos na pintura. Para eliminar estes defeitos, calculamos uma aproximação \tilde{L}_i pela base RADIAL para cada iluminação como descrito na seção 4.7.3. Veja a figura 5.21.

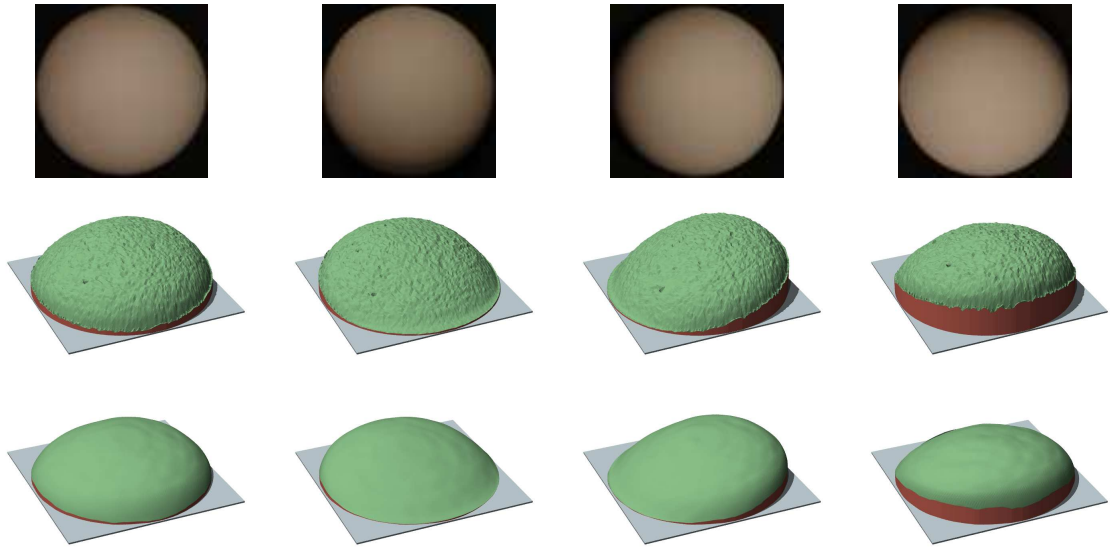


Figura 5.21: No alto, imagens do gabarito utilizado para o conjunto de teste **rglamb**, (localizado no canto inferior direito da figura 5.19), sob as mesmas iluminações das imagens da figura 5.20. No centro, gráficos tridimensionais das amostras $S_i[p]$ brutas extraídas dessas imagens. Embaixo, gráficos das funções de tonalização \tilde{L}_i modeladas pela base RADIAL e ajustadas a estas amostras.

Para o algoritmo BGRID, geramos duas tabelas \mathbb{T} correspondentes a $N = 11172$ normais \vec{n}_k cobrindo o hemisfério \mathbb{H}^2 . Na primeira tabela “bruta” \mathbb{T}' , as assinaturas \mathbf{g}_k foram obtidas por interpolação bilinear das intensidades $S_i[p]$ das fotos. Na tabela “limpa” \mathbb{T}'' , as assinaturas \mathbf{g}_k foram obtidas amostrando o gabarito virtual $\tilde{\mathbf{L}}$ descrito acima. Em ambos os casos, foi necessário eliminar entradas ao longo das bordas de \mathcal{G} de ambas as tabelas que tinham ou $|\mathbf{G}_k|$ muito baixo, e portanto assinaturas \mathbf{g}_k muito incertas, ou (na tabela \mathbb{T}'') valores de $S_i[p]$ negativos. Restaram com isso $N = 10967$ amostras na tabela \mathbb{T}' e $N = 9837$ na tabela \mathbb{T}'' .

As figuras 5.22 e 5.23 mostram o conjunto \mathbb{G}' das assinaturas da tabela bruta \mathbb{T}' . Como nos testes com imagens sintéticas, observamos que o conjunto é razoavelmente planar,

apesar da presença de ruído. A figura 5.23 mostra as medidas $\mu[i, j]$ e raio $\rho[i, j]$ da grade de baldes construída com estas assinaturas. A planaridade do conjunto \mathbb{K} é confirmada pelo raio relativamente pequeno das esferas delimitadoras. A figura 5.24 ilustra o número de entradas em cada balde $B[i, j]$ para uma grade $K \times K = 211 \times 211$ (taxa de ocupação $\kappa = 25\%$).

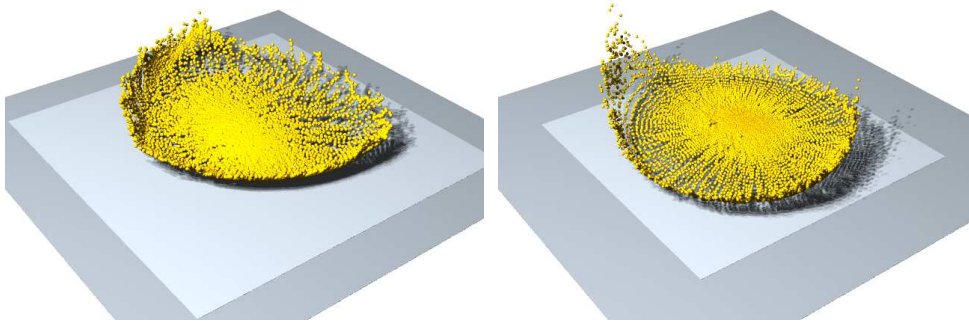


Figura 5.22: Posições das assinaturas brutas \mathbb{G}' do conjunto **rglamb** relativas ao plano P , com $m = 6$ luzes (esquerda) e $m = 48$ (direita) com as mesmas convenções gráficas da figura 5.6.

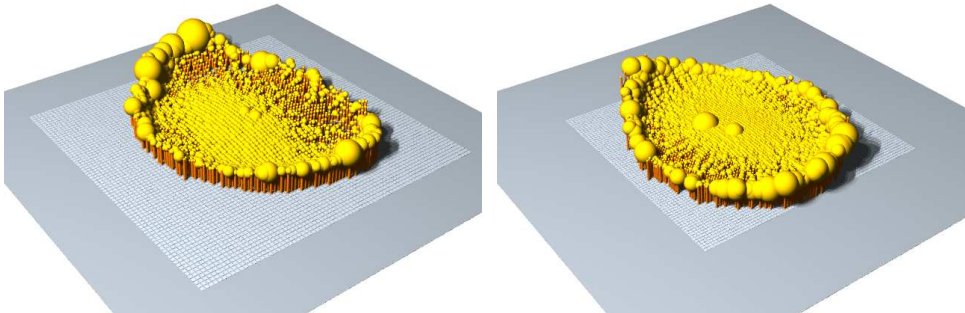


Figura 5.23: Médias $\mu[i, j]$ e raios $\rho[i, j]$ dos baldes de uma grade com $K \times K = 75 \times 75$ células, construída a partir da tabela bruta \mathbb{T}' do conjunto **rglamb**, com as mesmas convenções gráficas da figura 5.7, para $m = 6$ luzes (esquerda) e $m = 48$ (direita).

As figuras 5.25 e 5.26 mostram respectivamente, o formato do conjunto \mathbb{G}'' , as médias e raios dos baldes da grade da tabela \mathbb{T}'' derivada do gabarito virtual ajustado aos dados brutos. A figura 5.27 mostra o número de entradas em cada balde $B[i, j]$ desta grade.

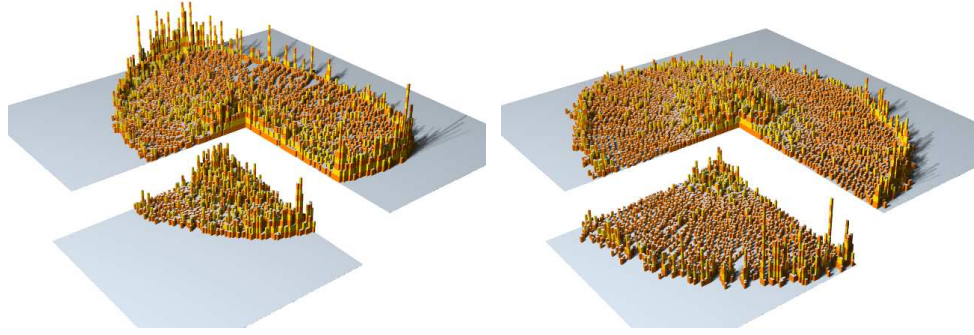


Figura 5.24: Número de entradas em cada balde $B[i, j]$ na grade de $K \times K = 211 \times 211$ células construída com a tabela de assinaturas brutas \mathbb{T}' do conjunto `rglamb`, com $m = 6$ luzes (esquerda) e $m = 48$ (direita). As maiores listas contém 13 entradas (esquerda) e 9 entradas (direita).

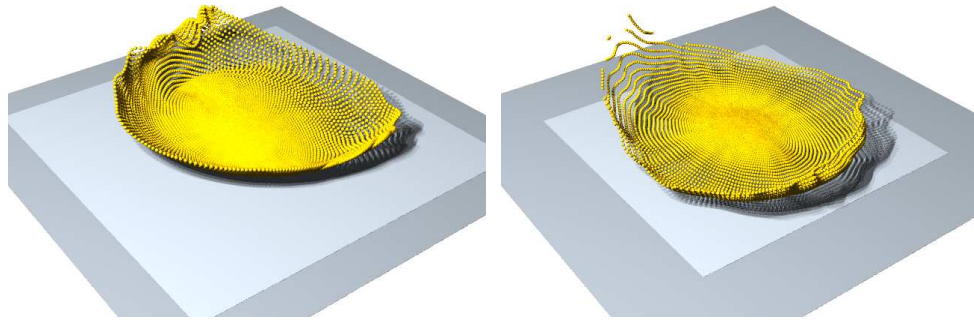


Figura 5.25: Posição relativa ao plano P das assinaturas da tabela “limpa” \mathbb{T}'' do conjunto `rglamb`, com $m = 6$ luzes (esquerda) e $m = 48$ (direita).

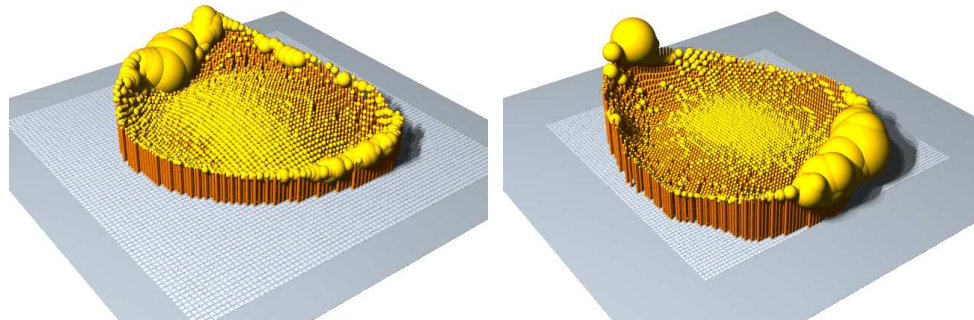


Figura 5.26: Médias $\mu[i, j]$ e raios $\rho[i, j]$ dos baldes da grade de $K \times K = 75 \times 75$ células construída a partir da tabela limpa \mathbb{T}'' do conjunto `rglamb`, com $m = 6$ luzes (esquerda) e $m = 48$ (direita).

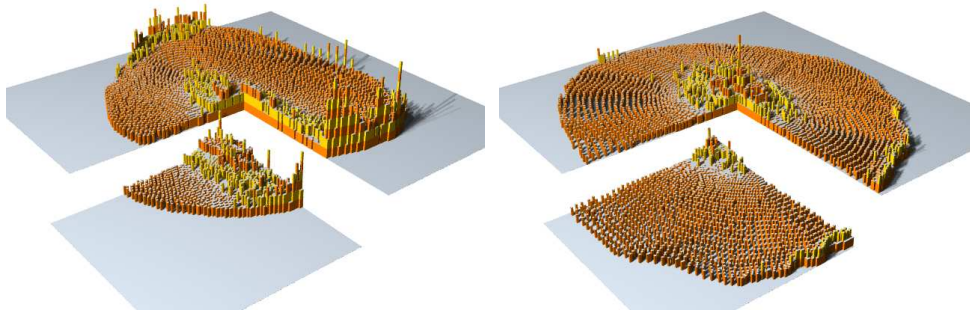


Figura 5.27: Número de entradas em cada balde $B[i, j]$ da grade de $K \times K = 211 \times 211$ células construída a partir da tabela limpa \mathbb{T}'' do conjunto **rglamb**, com $m = 6$ luzes (esquerda) e $m = 48$ (direita). As maiores listas contém 9 entradas (esquerda) e 4 entradas (direita).

A figura 5.28 mostra o mapa de normais da cena calculado pelo procedimento BGRID a partir do canal de luminância Y com $m = 24$ imagens de entrada usando a tabela de assinaturas limpa \mathbb{T}'' . Os mapas obtidos com $m = 6$ até $m = 48$ luzes, com qualquer das duas tabelas são indistinguíveis. A figura 5.29 mostra o mapa de cor intrínseca obtido aplicando BGRID separadamente para cada canal R, G e B . A figura 5.30 é a uma vista oblíqua do mapa de alturas obtido pela integração do mapa de normais da figura 5.28.

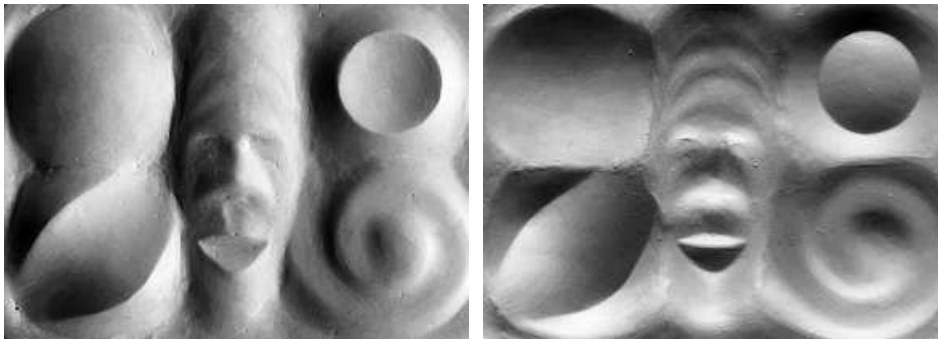


Figura 5.28: As componentes $\vec{n}.x$ e $\vec{n}.y$ do mapa de normais calculado a partir do canal de luminosidade Y de $m = 24$ fotos do conjunto **rglamb** usando a tabela limpa \mathbb{T}'' .

As tabelas 5.3 e 5.4 mostram o desempenho médio por pixel do procedimento BGRID, executado no canal Y das imagens da cena real usando a tabela bruta \mathbb{T}' e limpa \mathbb{T}'' , respectivamente, para diversos valores de K e m .

Observe que o número b de baldes examinados geralmente decresce e número d de distâncias calculadas aumenta com a taxa de ocupação κ ; e que o menor tempo de execução é obtido com taxa $\kappa = 0.25$, para a maioria dos valores de m .



Figura 5.29: O mapa de cor intrínseca calculado a partir dos canais R,G e B de $m = 24$ fotos do conjunto `rglamb` com a tabela limpa \mathbb{T}'' .

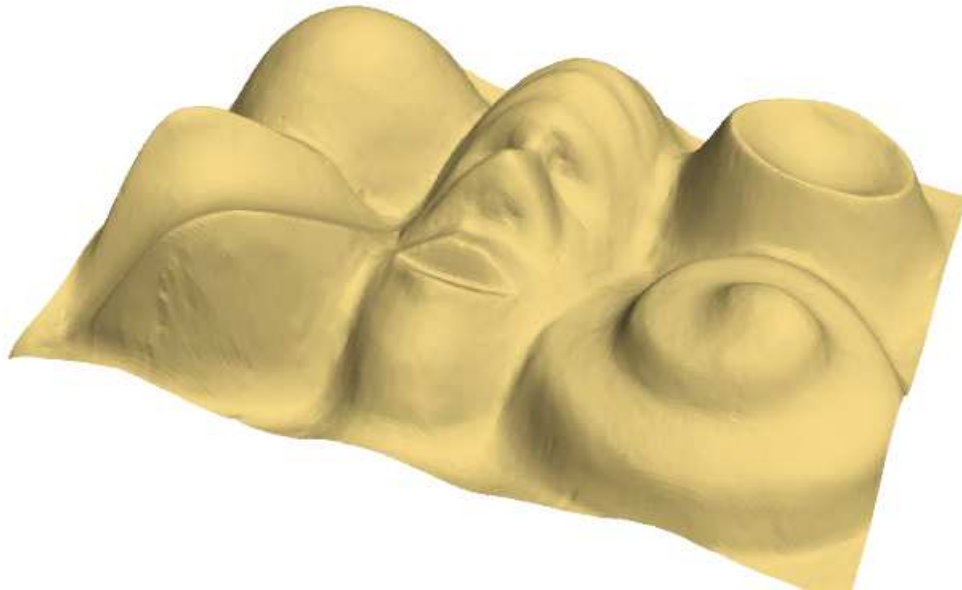


Figura 5.30: Visualização do mapa de alturas obtido pela integração do mapa de normais da figura 5.28.

Tabela 5.3: Custos médios por pixel do procedimento BGRID para diversos valores de m e K com a tabela de assinaturas brutas \mathbb{T}' nas imagens do conjunto `rglamb`. Entradas com $K = 1$ correspondem a busca por força bruta.

m	K	κ	t	d	b
6	1	—	1458.2	10967.0	1.0
6	105	0.99	7.5	37.5	12.1
6	149	0.49	6.5	28.6	15.8
6	211	0.25	6.3	24.0	22.9
6	299	0.12	6.6	20.7	35.0
6	422	0.06	7.1	18.2	56.1
12	1	—	2137.1	10967.0	1.0
12	105	0.99	12.1	44.6	16.5
12	149	0.49	11.5	37.3	24.6
12	211	0.25	11.0	31.9	37.6
12	299	0.12	11.2	27.2	60.5
12	422	0.06	12.5	23.8	102.1
24	1	—	3920.5	10967.0	1.0
24	105	0.99	28.9	73.8	25.7
24	149	0.49	27.1	64.8	39.6
24	211	0.25	26.5	58.4	63.8
24	299	0.12	27.3	53.2	108.7
24	422	0.06	30.1	48.5	191.4
48	1	—	6972.5	10967.0	1.0
48	105	0.99	74.5	117.7	28.4
48	149	0.49	66.4	102.3	44.0
48	211	0.25	62.2	90.4	71.8
48	299	0.12	59.8	80.6	123.2
48	422	0.06	60.6	72.6	218.5

Tabela 5.4: Custos médios por pixel do procedimento BGRID para diversos valores de m e K com a tabela de assinaturas \mathbb{T}'' , no conjunto de imagens `rglamb`. Entradas com $K = 1$ correspondem a busca por força bruta.

m	K	κ	t	d	b
6	1	—	1038.28	10749.00	1.00
6	105	0.97	5.11	33.48	13.07
6	149	0.48	4.78	27.40	18.06
6	211	0.24	5.11	25.88	26.61
6	299	0.12	5.77	24.47	41.27
6	422	0.06	6.93	21.46	67.16
12	1	—	1581.75	10227.00	1.00
12	105	0.93	8.25	34.69	17.63
12	149	0.46	8.08	31.64	26.55
12	211	0.23	8.74	30.00	40.60
12	299	0.11	9.57	26.86	66.07
12	422	0.06	11.38	23.29	112.22
24	1	—	2925.09	9977.00	1.00
24	105	0.90	18.97	53.84	26.15
24	149	0.45	18.81	50.51	40.00
24	211	0.22	20.79	52.25	64.84
24	299	0.11	22.77	50.69	110.46
24	422	0.06	26.07	45.91	194.61
48	1	—	5744.10	9837.00	1.00
48	105	0.89	44.71	75.99	29.28
48	149	0.44	41.41	67.25	45.32
48	211	0.22	44.05	69.33	74.19
48	299	0.11	48.34	71.90	127.87
48	422	0.06	51.97	67.12	226.94

5.5 Comparação com algoritmo k -d tree

Em outra série de testes, comparamos o desempenho do procedimento BGRID com a busca por força bruta (BRUTE) e com o algoritmo de ponto mais próximo baseado na estrutura de dados de árvore k -dimensional (k -d tree) [19]. O algoritmo de busca por força bruta (BRUTE) é o mesmo código de BGRID executado com tamanho de grade $K = 1$.

Nossa implementação do algoritmo de busca com árvore (KD TREE) foi escrita na linguagem C. Para tornar a comparação com BGRID mais significativa, definimos o parâmetro d para este algoritmo como sendo $1/m$ vezes o número de nós internos da árvore examinados na busca. A justificativa para este critério é que em cada nó da

árvore, o algoritmo KDTREE faz um teste de uma coordenada de assinatura, enquanto que cada comparação do algoritmo BGRID (a função dist) leva em conta m coordenadas.

Para estes testes utilizamos as mesmas imagens e tabelas de assinaturas dos testes da seção 5.4 Os resultados estão na tabela 5.5.

Tabela 5.5: Comparação do tempo médio de execução t (milissegundos) e quantidade média d de distâncias calculadas para os três métodos considerados para os vários conjuntos de imagens, tabelas de assinaturas e número de luzes

m .

	t			d		
m	BGRID	KDTREE	BRUTE	BGRID	KDTREE	BRUTE
conjunto sglamb						
6	4.0	59.8	1499.7	14.2	179.5	11172.0
12	5.2	77.6	2120.5	10.3	160.9	11172.0
24	8.3	114.8	3745.2	10.0	155.5	11172.0
48	14.2	208.0	7110.6	9.8	169.0	11172.0
conjunto sglstr						
6	2.80	36.41	1084.91	8.21	158.04	11172.00
12	3.92	54.33	1759.83	7.61	145.82	11172.00
24	6.16	86.26	3517.42	7.68	139.37	11172.00
48	10.64	152.91	6907.14	7.74	139.95	11172.00
conjunto rglamb com tabela bruta T'						
6	6.3	66.3	1458.2	24.0	201.9	10967.0
12	11.0	101.5	2137.1	31.9	216.8	10967.0
24	26.5	217.6	3920.5	58.4	300.9	10967.0
48	62.2	468.3	6972.5	90.4	387.3	10967.0
conjunto rglamb com tabela limpa T''						
6	5.11	62.04	1038.28	25.88	260.33	10749.00
12	8.74	67.98	1581.75	30.00	185.08	10227.00
24	20.79	156.57	2925.09	52.25	256.61	9977.00
48	44.05	337.07	5744.10	69.33	311.36	9837.00

Observamos que todos estes métodos têm a mesma precisão, retornando sempre o mesmo valor que a busca por força bruta. O método BGRID foi consistentemente 10 vezes mais rápido que KDTREE e de 100 a 400 vezes mais rápido que BRUTE.

Observamos que o custo de BGRID e KDTREE é 2 a 4 vezes maior nos testes com fotos reais do que nos testes com imagens sintéticas. Este custo adicional é devido principalmente ao ruído presente nas assinaturas da cena, deixando-as mais distantes do conjunto $\mathbb{K} = \mathcal{I}(\mathbb{H}^2)$ de assinaturas ideais. Os dois algoritmos precisam então examinar um número maior de entradas da tabela até ter certeza de que a melhor entrada foi encontrada.

Observamos também que os custos de BGRID e KDTree são um pouco maiores com a tabela de assinaturas brutas \mathbb{T}' do que com a tabela limpa \mathbb{T}'' obtida do gabarito virtual. Esta diferença é devida ao maior espalhamento das entradas da tabela em relação à superfície \mathbb{K} ideal, o que aumenta o tempo de busca pela mesma razão acima.

5.6 Comparação com ANN e LSH

Outros métodos de busca de ponto mais próximo que foram propostos para a inversão da tabela de assinaturas são o *locally sensitive hashing* (LSH) utilizado por Zhong e Little [116], e o *approximate nearest neighbours* (ANN) utilizado por Hertzman e Seitz [53]. De acordo com os autores, o primeiro é uma melhora do segundo.

O algoritmo LSH projeta os pontos dados (no nosso caso, as assinaturas da tabela \mathbb{T}) em um certo número q de grades unidimensionais definidas por q vetores no espaço \mathbb{R}^m . Cada projeção seleciona uma lista de candidatos fornecida pela união de um conjunto de baldes consecutivos da grade correspondente. Estas q listas são combinadas por intersecção e o melhor candidato é escolhido nesta lista combinada. O algoritmo escolhe automaticamente o número de grades q e os q vetores de projeção em função da dimensão m do espaço e dos pontos dados. Ele, no entanto, não garante que o resultado seja a entrada da tabela com assinatura mais próxima à procurada.

Não conseguimos obter a implementação do método LSH tal como aplicado a estéreo fotométrico por Zhong e Little. Tentamos então utilizar a implementação original do método LSH de Andoni [9]. Entretanto, esse programa retorna uma lista aproximada de todos os pontos dentro de um raio r fornecido como dado. A escolha deste parâmetro é um problema não-trivial: se r for muito pequeno, o algoritmo retorna quase sempre uma lista vazia, se for muito grande, ele pode retornar metade das entradas de \mathbb{T} ou mais.

Impossibilitados de usar o algoritmo de Zhong e Little nos nossos dados de teste, tivemos que nos contentar com uma comparação indireta de desempenho. Zhong e Little informam que sua implementação oferece uma busca até 50 vezes mais rápida que a busca por força bruta para um teste específico (a cena `bottle` do autor, com $N = 10000$ entradas na tabela e $m = 24$ luzes). Em comparação, no nosso teste com a cena real `rglamb`, o algoritmo BGRID foi em torno de 140 mais rápido que BRUTE, sob condições similares. Embora não seja válido comparar tempos absolutos de execução em máquinas distintas, as acelerações relativas em relação à busca por força bruta indicam que nosso método é no mínimo duas vezes mais eficiente que a busca baseada em LSH.

De qualquer maneira, deve-se notar que o método LSH retorna uma normal aproximada (de acordo com seu artigo, com erros até 8 graus); enquanto BGRID sempre retorna a entrada da tabela com assinatura mais próxima à procurada.

5.7 Efeito do ruído na imagem

A presença de ruído nas imagens de entrada S_i pode afetar (i) o tempo de busca, (ii) a precisão das normais recuperadas, e (iii) a precisão do mapa de alturas integrado. Para analisar estes efeitos, repetimos dois dos testes da seção 5.4 após contaminar as imagens por quantidades variáveis de ruído. Especificamente, utilizamos as mesmas 24 imagens dos testes **sglamb** e **rglamb** (tabela bruta \mathbb{T}'), exceto que cada amostra $S_i[p]$ da cena (mas não do gabarito) foi misturada com 2%, 5%, 10% e 20% de ruído aleatório distribuído uniformemente em $[0, 1]$. Nestes testes usamos tamanho de grade 211×211 , que ofereceu o melhor desempenho nos testes anteriores.

5.7.1 Efeito no custo da busca.

Observamos que a presença de ruído nas imagens S_i aumenta o custo de busca, pelas razões descritas na seção 5.4.3. Os resultados estão na tabela 5.6.

Tabela 5.6: Número médio d de distâncias de assinaturas calculadas pelos métodos BGRID, KDTree e BRUTE para $m = 24$ imagens, com diversos níveis η de ruído.

η	d					
	Imagens sintéticas (sglamb)			Fotos reais (rglamb , \mathbb{T}')		
	BGRID	KDTREE	BRUTE	BGRID	KDTREE	BRUTE
0%	10.0	155.5	11171.0	58.4	300.9	10966.0
2%	22.9	203.4	11171.0	82.1	351.3	10966.0
5%	63.6	279.0	11171.0	190.7	521.1	10966.0
10%	177.0	390.0	11171.0	517.5	847.3	10966.0
20%	558.9	612.3	11171.0	1467.9	1446.2	10966.0

Observamos que níveis elevados de ruído afetam muito mais o desempenho de BGRID do que o de KDTree. Este efeito é esperado pois o ruído aumenta o número de bal-des examinados pelo BGRID. Nestes testes, observamos que uma boa parte do tempo de busca (tanto para BGRID quanto para KDTree) é devido a um pequeno número de pixels que possuem assinaturas $\mathbf{s}[p]$ extremamente inconsistentes, e que portanto exigem um número de comparações muito grande na busca. Veja a figura 5.31. Uma vez que a melhor entrada da tabela para estes *pixels* será quase que certamente a entrada incorreta, poderíamos melhorar o desempenho do algoritmo interrompendo a busca após um determinado número de distâncias. Nos testes deste capítulo não usamos esta otimização, e sempre retornamos a melhor entrada, mesmo para estes pixels.

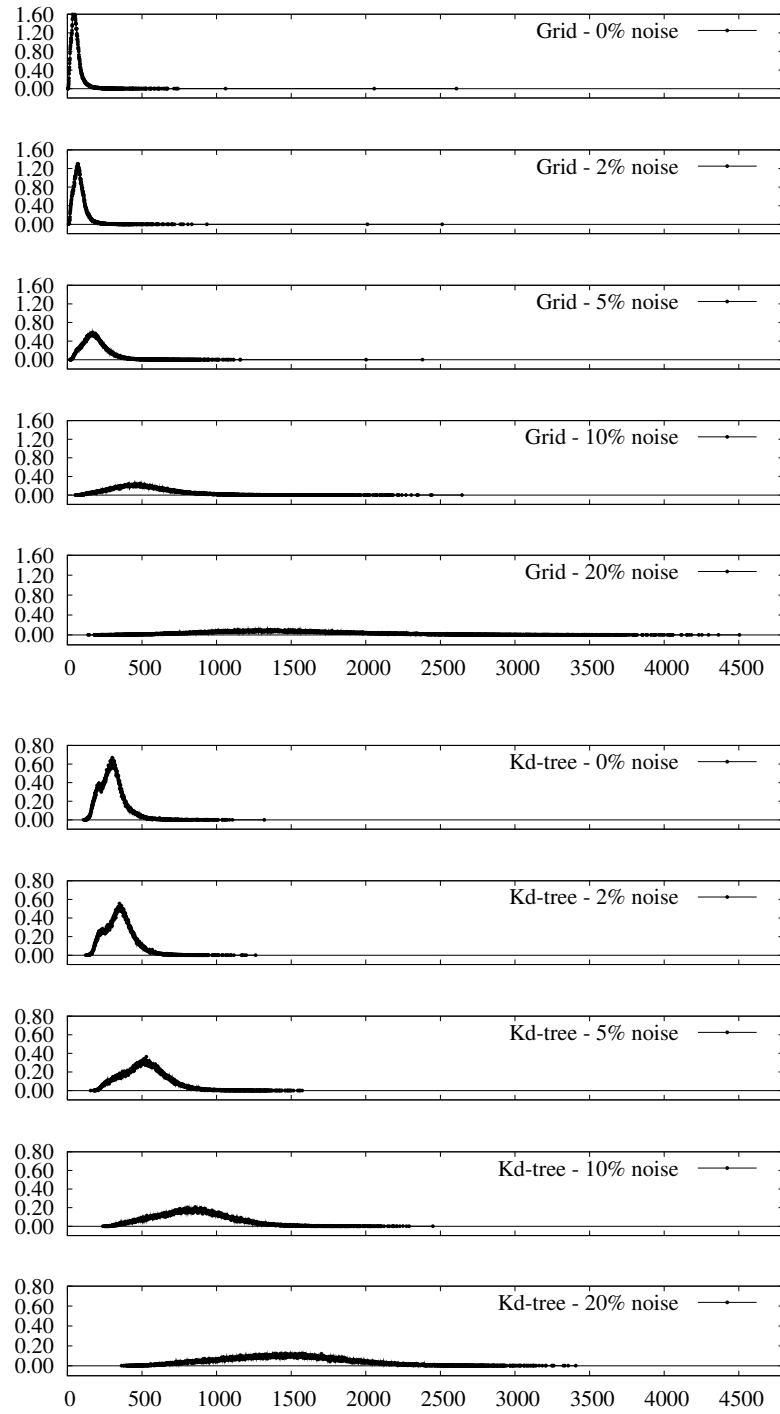


Figura 5.31: Histogramas da porcentagem dos pixels do domínio \mathcal{S} (eixo vertical) que exigem um dado d número de cálculos da função de distância (eixo horizontal), para os algoritmos BGRID (no alto) e KD-tree (em baixo), para o conjunto `rlamb` de fotos da cena real com diversos valores de ruído e tabela de assinaturas brutas.

5.7.2 Efeito na precisão do resultado

Como seria de esperar, a adição de ruído também acarreta erros nas normais calculadas. Com 2%, 5%, 10% e 20% de ruído adicionado, o erro E_{rms} nas normais calculadas foi de 0.014, 0.030, 0.080 e 0.170 radianos (0.8° , 2.1° , 5.1° , e 10.2°), respectivamente. Veja a figura 5.32.

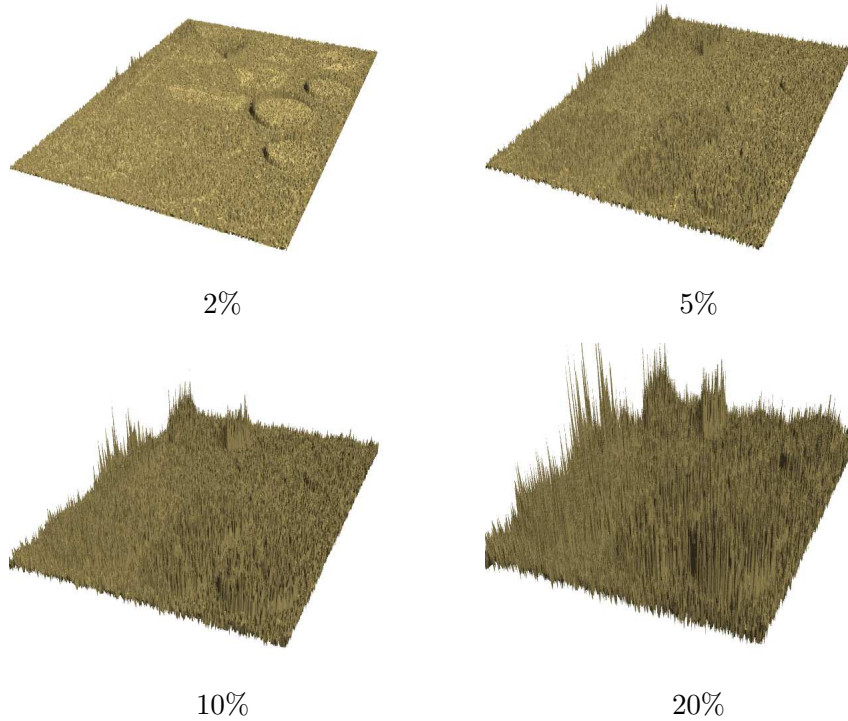


Figura 5.32: Erro nas normais calculadas da cena **sglamb** com ruído gaussiano adicionado misturado nas amostras S_i para $m = 24$ iluminações.

Observe que a precisão das normais calculadas não é uma propriedade do algoritmo BGRID (nem do KD TREE), mas sim do critério adotado para comparações de assinaturas (distância euclidiana no espaço \mathbb{R}^m) e a da densidade de amostragem da tabela \mathbb{T} . Portanto, qualquer outro algoritmo exato terá a mesma precisão que BGRID e KD TREE para qualquer nível de ruído, e qualquer algoritmo aproximado só pode ser menos preciso do que eles.

Para concluir, mostramos na figura 5.33 o resultado da integração do mapa de normais obtido com 24 imagens do conjunto **sglamb** com 10% de ruído adicionado. Observe que o resultado ainda é bastante adequado, pois (como veremos no capítulo 7) a integração é pouco sensível a erros aleatórios no mapa de normais.

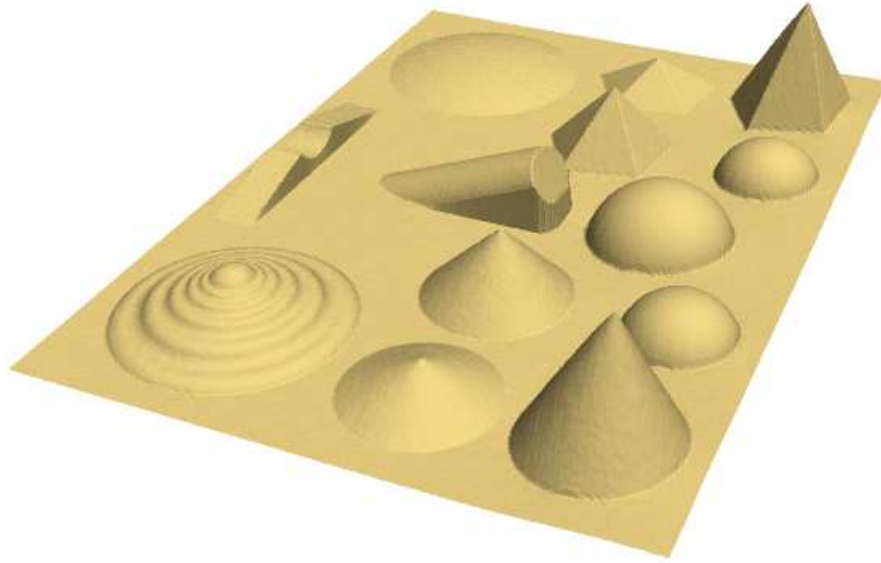


Figura 5.33: Vista oblíqua do mapa de alturas obtido pela integração do mapa de normais calculado por BGRID a partir de $m = 24$ imagens do conjunto `sglamb` com $\eta = 10\%$ de ruído.

Capítulo 6

Sombras projetadas e realces lustrosos

É muito difícil garantir que a iluminação seja uniforme sobre toda a cena, mesmo em cenas simples e com posicionamento cuidadoso dos objetos e fontes de iluminação. Fontes de luz reais geralmente emitem intensidade de luz diferente em cada direção. Além disso, se a fonte de luz está a uma distância finita, tanto a intensidade quanto a direção da luz variam com o ponto p da imagem e com a altura (desconhecida) do ponto $s[p]$. Finalmente, se a superfície não é convexa, é muito difícil evitar sombras próprias e iluminação secundária.

É difícil também garantir que o acabamento da cena seja uniforme e igual ao do gabarito; especialmente nas componentes lustrosas, cuja intensidade e grau de polimento são facilmente alterados por desgaste, sujeira, etc. Aliás, em materiais lustrosos típicos como plásticos, tintas e pedras polidas, a magnitude da componente lustrosa da FBDR (que é devida à reflexão na superfície do objeto) geralmente varia independentemente da magnitude da componente lambertiana (que é devida à absorção e ao espalhamento por partículas logo abaixo da superfície). Nesse caso, o acabamento (e não apenas o albedo) varia também. Ou seja, EFBG com gabarito único só deveria ser aplicada, em teoria, se todas as componentes da FBDR, incluindo a lambertiana e a lustrosa tiverem a mesma magnitude em toda a cena.

Essas *anomalias de iluminação* e *anomalias de acabamento* podem alterar o valor da radiância $S_i[p]$, deixando-a maior ou menor que a prevista pela equação (2.14). Quando estas anomalias ocorrem em diferentes graus em imagens diferentes, os vetores de observação $\mathbf{S}[p]$ e $\mathbf{L}(\vec{n}[p])$ deixam de ser colineares, e os métodos de cálculo de normais descritos nos capítulos 2 e 5 não funcionam. Neste capítulo vamos analisar a natureza e as consequências dessas anomalias e descrever nossa solução para esse problema.

6.1 Anomalia de radiância e pontos canônicos

Seja L_i a função de tonalização da imagem i . Para cada ponto p da cena definimos a *anomalia de radiância* $\kappa_i[p]$ de p como sendo a diferença entre a radiância observada e a prevista pela equação (2.14), ou seja

$$\kappa_i[p] = S_i[p] - a[p]L_i(\vec{n}[p]) \quad (6.1)$$

Em particular, se o ponto p está localizado em uma região de sombra *projetada* na imagem S_i , a anomalia κ_i geralmente será negativa. Note que sombras *próprias* afetam tanto a cena quanto a função de tonalização L_i , e portanto não contribuem para a anomalia κ_i . Por outro lado, se a radiância S_i foi intensificada devido à iluminação secundária, a anomalia κ_i será positiva.

A anomalia κ_i pode ser diferente de zero também quando o acabamento da superfície no ponto $s[p]$ for diferente do acabamento do gabarito. Por exemplo, se o gabarito for lambertiano e o acabamento da cena tiver uma componente lustrosa, todo ponto onde esta componente é significativa na imagem S_i terá geralmente κ_i positivo, enquanto que os demais pontos da imagem terão κ_i negativo. Ressaltamos, porém, que componentes não lambertianas da FBDR não contribuem para a anomalia κ_i se elas estiverem presentes tanto no gabarito quanto na cena, na mesma proporção em relação às demais componentes.

Dizemos que um ponto p de \mathcal{S} é *canônico* na imagem S_i (ou que a iluminação i é *canônica* para p) quando o valor de κ_i é devido apenas a erros de medida e quantização da câmera; ou seja, quando o ponto $s[p]$ tem o mesmo acabamento que o gabarito, e a iluminação desse ponto na imagem S_i é a mesma do gabarito. Caso contrário, dizemos que a iluminação i é *anômala* para p , e p é *anômalo* na imagem i .

Verifica-se que a anomalia de radiância $\kappa_i[p]$ tem uma distribuição de probabilidade bastante complexa que não pode ser adequadamente aproximada por uma única distribuição gaussiana. Para pixels canônicos, a anomalia tem distribuição aproximadamente gaussiana com média 0 e desvio padrão pequeno, devido aos erros de medida e quantização; enquanto a anomalia dos pixels afetados por sombras projetadas ou por diferenças de acabamento (*outliers*) tem uma distribuição muito mais ampla e assimétrica. Por esta razão, métodos de EFIV que procuram minimizar o erro médio quadrático entre o vetor de observação previsto a $\mathbf{L}(\vec{n})$ e o vetor observado $\mathbf{S}[p]$ não dão resultados satisfatórios na presença de anomalias, pois eles implicitamente supõem que as anomalias κ_i têm distribuição gaussiana.

6.2 A hipótese de dados suficientes

O estéreo fotométrico ainda é possível para um pixel p , mesmo na presença de anomalias de iluminação ou de acabamento, se de alguma forma conseguirmos identificar um subconjunto K de $\{1 \dots m\}$ com pelo menos três iluminações nas quais p é canônico e que são suficientes para satisfazer a hipótese de invertibilidade. Esta condição será denominada *hipótese de dados suficientes* para o pixel p . Note que esta condição *não* exclui imagens S_i onde p está em sombra própria, e que o subconjunto K pode ser diferente para cada ponto p de S .

Entretanto, não é fácil determinar *a priori* tal conjunto. Estritamente falando, é *impossível* determinar com total confiança se um ponto p de S é canônico analisando-se apenas seus valores $S_1[p], \dots, S_m[p]$, sem levar em conta as funções de tonalização L_1, L_2, \dots, L_m . Por outro lado, qualquer análise que utilize também as funções de tonalização L_1, L_2, \dots, L_m para esta decisão deve necessariamente identificar a normal $\vec{n}[p]$. Ou seja, a determinação das imagens onde p é canônico deve ser uma consequência, e não uma etapa preliminar, da resolução do problema de estéreo fotométrico para esse pixel.

6.3 Soluções anteriores

Na literatura encontramos várias propostas de heurísticas que tentam determinar um subconjunto K de imagens que satisfaça a hipótese dos dados suficientes. A maioria destas heurísticas procura apenas eliminar imagens onde o pixel está afetado por sombras (projetadas ou próprias), e supõe que anomalias positivas (realces e iluminação secundária) serão tratadas ou eliminadas no decorrer do cálculo da normal.

6.3.1 Descarte de imagens anômalas

A heurística mais simples consiste em incluir no conjunto K apenas imagens i cuja radiância $S_i[p]$ está dentro de um dado intervalo $[S_{\min}, S_{\max}]$. Esta metodologia é utilizada (com $S_{\max} = \infty$) principalmente para eliminar sombras próprias ou projetadas. Muitos trabalhos [106, 108, 88, 27, 38, 55] optam por esse tratamento. Uma variante desta técnica é utilizar um subconjunto K de tamanho fixo k que corresponde aos k maiores valores de $S_i[p]$ [74, 96, 61]. Embora esta heurística seja razoável (supondo que as fontes de iluminação possuam mesma intensidade), há sempre o risco de se incluir no conjunto K imagens onde o pixel está afetado por realces, e/ou excluir imagens onde p é canônico.

Outra abordagem usada por alguns autores é calcular uma normal preliminar $\tilde{\vec{n}}$ usando todas as imagens S_i , e então calcular uma normal definitiva, após excluir toda iluminação

i para qual a radiância observada $S_i[p]$ é demasiadamente incompatível com radiância $aL_i(\vec{n})$ prevista para essa normal.

Esta abordagem foi introduzida por Coleman e Jain [59], especificamente para detecção de anomalias devidas a acabamento lustroso. Utilizando $m = 4$ fontes luminosas, eles enumeram todos subconjuntos de três imagens possíveis, determinam a normal para cada subconjunto supondo acabamento lambertiano, calculam a média $\tilde{\vec{n}}$ dessas normais e comparam cada $S_i[p]$ com o valor esperado $aL_i(\tilde{\vec{n}})$. A iluminação cuja anomalia κ_i for maior que um dado limiar é considerada não canônica e excluída do cálculo final. Barski e Petrou [16] estenderam o algoritmo de Coleman e Jain para detectar também sombras. Elas utilizam imagens coloridas, e levam em conta as anomalias nos três canais de cor para determinar as iluminações não canônicas.

Tanto Coleman e Jain quanto Basri e Petrou usam apenas $m = 4$ luzes, e portanto permitem no máximo uma iluminação anômala em cada pixel. Shen *et. al* [88], generalizaram o algoritmo de Basri e Petrou para m imagens, e acrescentaram um algoritmo para determinar a magnitude da componente lustrosa do acabamento da cena, supondo que esta é a mesma para todo pixel p . Uma variante deste tipo de abordagem foi adotada por Atkinson *et. al* [48]. Yu *et. al* [112] reformulam este problema (para superfícies com acabamento lambertiano para luzes unidirecionais e intensidades conhecidas) como um problema de programação linear inteira.

Hayakawa [49] e Georghiades *et al.* [40] procuram determinar *a priori* um conjunto de pixels $P \subseteq \mathcal{S}$ e, simultaneamente, um conjunto de imagens $K \subseteq \{1 \dots m\}$ tais que p em P é canônico em toda imagem i de K . Para este fim, eles usam o critério de limiar de intensidade para excluir regiões onde ocorrem sombras e/ou reflexos lustrosos.

6.3.2 Outros métodos

No método de grade m -dimensional de Woodham [106, 108], os pixels anômalos podem ser detectados porque suas assinaturas de observação $\mathbf{s}[p]$ são mapeadas para células da grade distintas das células que contêm as assinaturas do gabarito. Uma vez que Woodham usava apenas $m = 3$ luzes nos seus testes, este critério lhe permitia apenas detectar pixels anômalos, mas não determinar a normal correta.

Alguns métodos de estéreo fotométrico que usam decomposição em valores singulares para determinar a normal, como o de Yuille [113] e de Basri e Jacobs [17], afirmam que esse problema pode ser contornado utilizando-se uma quantidade muito grande de iluminações: Yuille, por exemplo usa $m = 70$ ou mais. Eles esperam que, nestas condições, a DVS seja dominada pelo componente lambertiano da FBDR. Yuille *et. al* [113] utilizam otimização não-linear para detectar sombras (próprias e projetadas) nas imagens após obter uma estimativa inicial de $\vec{n}[p]$ por este método.

Chandraker *et. al* [25] supõem que sombras projetadas têm bordas nítidas, e utilizam a técnica de segmentação por corte em grafos [44] para detectar essas bordas e assim inferir os pixels afetados por sombras projetadas. Este método não funciona com fontes de luz extensas, que produzem transições graduais (penumbras) entre regiões iluminadas e sombras. Nesse caso, “estar na sombra projetada” não é um predicado binário (verdadeiro ou falso), e a transição suave pode não ser detectada por métodos de segmentação de imagem. A presença de iluminação secundária é outra complicação para esta abordagem

Vale observar que o método de estéreo fotométrico de Magda [69], baseado na variação da radiância em função da distância da fonte, é imune a efeitos de sombras projetadas e anomalias de acabamento. Porém, a precisão deste método é bastante limitada, e ele exige muito mais iluminações do que outros métodos considerados nesta tese.

6.4 Nossa abordagem: inferência bayesiana

Para melhor entender o efeito de anomalias de iluminação e acabamento nos vetores de observação, considere as figuras 6.1, 6.2 e 6.3. Estas figuras referem-se a uma instância do problema de estéreo fotométrico com $m = 9$ iluminações. Cada gráfico (à esquerda) refere-se a um pixel p da cena e a uma normal arbitrária \vec{g} . As cruzes vermelhas nas outras duas imagens mostram o pixel p da cena (no centro), e o pixel q do gabarito com essa normal \vec{g} (à direita). Cada marca “o” no gráfico representa um par de intensidades $b_i = (L_i(\vec{g}), S_i[p]) = (S_i[q], S_i[p])$ para $i \in \{1, \dots, m\}$.

A figura 6.1 mostra uma situação ideal, onde $\vec{g} = \vec{n}[q] = \vec{n}[p]$ e o pixel p é canônico em todas as imagens S_i ; e portanto, o vetor de observação $\mathbf{S}[p]$ é múltiplo do vetor $\mathbf{L}(\vec{g})$. Observe que os m pares b_i formam uma linha reta com a origem, exceto por pequenos desvios devidos à quantização e ruído. A inclinação da reta é o albedo $a[p]$.

A figura 6.2 é típica de um caso onde a normal $\vec{n}[p]$ no ponto p difere da normal considerada $\vec{g} = \vec{n}[q]$. Neste caso, não existe um conjunto significativo de pares b_i que sejam colineares com a origem; ou seja, não é possível determinar um albedo $a[p]$ tal que $\mathbf{S}[p] = a[p]\mathbf{L}(\vec{g})$.

Finalmente, a figura 6.3 mostra a situação onde a normal considerada $\vec{g} = \vec{n}[q]$ é a normal correta do ponto p , mas p é canônico em apenas quatro das nove imagens. Os quatro pares b_i correspondentes a estas imagens continuam formando uma linha reta com a origem (sendo que 3 deles inclusive coincidem), enquanto que os demais caem abaixo da linha (devido a sombras projetadas) ou acima dela (devido a realces ou iluminação secundária).



Figura 6.1: À esquerda, gráfico dos pares $b_i = (L_i(\vec{g}), S_i[p])$ para uma situação ideal, quando o ponto p da cena (centro) é canônico em todas as $m = 9$ imagens e a normal considerada \vec{g} é a normal correta $\vec{n}[p]$. As fotos mostram o ponto p na cena (no centro) e o ponto q do gabarito com $\vec{n}[q] = \vec{g}$ (à direita).



Figura 6.2: Pares $b_i = (L_i(\vec{g}), S_i[p])$ quando p é canônico em todas as $m = 9$ imagens, mas sua normal $\vec{n}[p]$ não é a normal considerada $\vec{g} = \vec{n}[q]$.

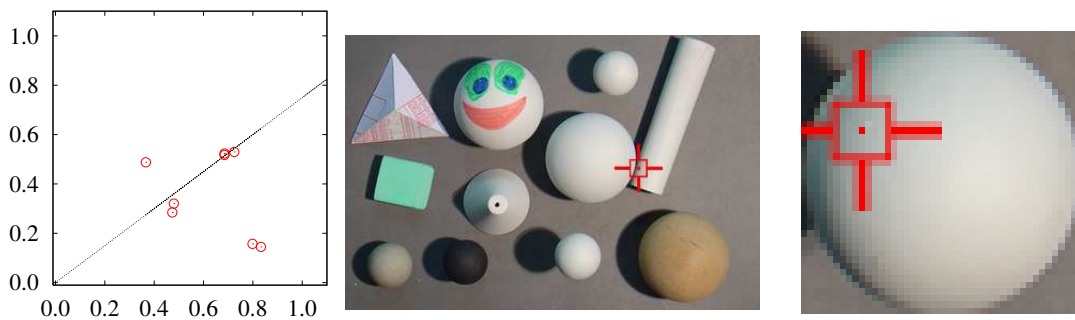


Figura 6.3: Pares $b_i = (L_i(\vec{g}), S_i[p])$ quando a normal considerada $\vec{g} = \vec{n}[q]$ é a normal correta $\vec{n}[p]$ do ponto p , mas este ponto está afetado por sombras projetadas e/ou anomalias de acabamento em cinco das $m = 9$ imagens. Note que três dos quatro pares b_i praticamente coincidem no gráfico.

A chave para o problema das anomalias está em distinguir as situações das figuras 6.1 e 6.3 da situação da figura 6.2.

Observe que o alinhamento de apenas dois pares $b_i = (L_i(\vec{g}), S_i[p]), b_j = (L_j(\vec{g}), S_j[p])$ com a origem não é significativo. Variando a normal \vec{g} sobre \mathbb{H}^2 , podemos fazer com que as abscissas $L_i(\vec{g})$ e $L_j(\vec{g})$ de b_i e b_j variem independentemente uma da outra. Portanto, espera-se que exista uma infinidade unidimensional de normais \vec{g} em \mathbb{H}^2 para os quais os pares b_i e b_j estão alinhados com a origem — mesmo quando p não é canônico em S_i e/ou S_j .

Por outro lado, a ocorrência de três pares b_i, b_j, b_k alinhados com a origem é uma situação que normalmente ocorre apenas para um conjunto finito e pequeno de normais \vec{g} , pois exige que *duas* condições sejam satisfeitas pela normal \vec{g} . Se vale a hipótese dos dados suficientes para o subconjunto $\{i, j, k\}$, este alinhamento só ocorre quando \vec{g} é a normal correta $\vec{n}[p]$. Portanto, o alinhamento de três ou mais pontos b_i com a origem (a menos de erros de medida) é uma evidência de que $\vec{n}[p]$ é a normal considerada \vec{g} ; e a evidência fica tanto mais forte quanto maior for o número de pares b_i alinhados.

Por outro lado, se a hipótese de dados suficientes não vale para $\{i, j, k\}$, o alinhamento triplo pode ocorrer para uma ou mais normais incorretas \vec{g} ; mas outros conjuntos $\{i', j', k'\}$ que incluem iluminações anômalas para p dificilmente terão alinhamentos triplos espúrios para as *mesmas* normais \vec{g} . Ou seja, a probabilidade de ocorrerem $t \geq 4$ pares b_i alinhados quando \vec{g} é incorreta é muito pequena, e diminui rapidamente quando t aumenta.

6.4.1 Análise bayesiana

Este critério para discriminar a normal correta entre as incorretas pode ser formalizado pelo princípio da *inferência bayesiana*.

Para simplificar a notação, no restante deste capítulo omitiremos o índice $[p]$ das notações $\mathbf{S}[p], \mathbf{s}[p], S_i[p], \vec{n}[p]$ e $a[p]$. Seja $\Pr(a, \vec{n}|\mathbf{S})$, a probabilidade do albedo do pixel p ser a e de sua normal ser \vec{n} , dado que seu vetor de observação é \mathbf{S} . Pelo teorema de Bayes, temos que

$$\Pr(a, \vec{n}|\mathbf{S}) = \frac{\Pr(\mathbf{S}|a, \vec{n}) \Pr(a, \vec{n})}{\Pr(\mathbf{S})} \quad (6.2)$$

Nesta fórmula, $\Pr(\mathbf{S}|a, \vec{n})$ é a probabilidade de que o vetor de observação do pixel p seja \mathbf{S} , supondo que seu albedo é a e sua normal é \vec{n} ; $\Pr(a, \vec{n})$ é a distribuição de probabilidade *a priori* para o albedo e a normal do pixel; e $\Pr(\mathbf{S})$ é a probabilidade do vetor \mathbf{S} ocorrer,

$$\Pr(\mathbf{S}) = \int_{\mathbb{H}^2} \int \Pr(\mathbf{S}|c, \vec{g}) \Pr(c, \vec{g}) \, dc \, d\vec{g} \quad (6.3)$$

A distribuição de probabilidade da normal é obtida então integrando a fórmula (6.2) sobre

todo o intervalo de variação do albedo, ou seja

$$\Pr(\vec{n}|\mathbf{S}) = \int_0^1 \Pr(a, \vec{n}|\mathbf{S}) da = \frac{1}{\Pr(\mathbf{S})} \int_0^1 \Pr(\mathbf{S}|a, \vec{n}) \Pr(a, \vec{n}) da \quad (6.4)$$

Podemos então reformular o problema do estéreo fotométrico como o problema de encontrar, dado o vetor de observação \mathbf{S} de um pixel p , os valores de a e \vec{n} que maximizam a probabilidade $\Pr(a, \vec{n}|\mathbf{S})$, dada pela fórmula (6.2); ou, alternativamente, a direção \vec{n} que maximiza $\Pr(\vec{n}|\mathbf{S})$, dada pela fórmula (6.4).

6.4.2 Simplificação da fórmula bayesiana

Nesta tese vamos supor que a e \vec{n} são variáveis aleatórias independentes, que a distribuição *a priori* do albedo a é uniforme em $[0, 1]$, e que a da normal \vec{n} é uniforme no hemisfério \mathbb{H}^2 ; ou seja $\Pr(a, \vec{n}) = 1/(2\pi)$.

Note que para maximizar a fórmula (6.2) ou (6.4) não é necessário calcular o denominador $\Pr(\mathbf{S})$, pois ele não depende dos parâmetros procurados a e \vec{n} .

Além disso, em vez de maximizar a probabilidade, podemos maximizar o logaritmo da mesma. Ou seja, supondo $\Pr(a, \vec{n})$ uniforme, procuramos pelo par a, \vec{n} que maximiza a função

$$\Gamma(a, \vec{n}|\mathbf{S}) = \log(\Pr(\mathbf{S}|a, \vec{n})) \quad (6.5)$$

Além de simplificar as fórmulas, esta transformação reduz o risco de *underflow* ou *overflow* nos cálculos. Alternativamente, usando a fórmula (6.4), podemos determinar a normal \vec{n} que maximiza a função

$$\Delta(\vec{n}|\mathbf{S}) = \log \left(\int_0^1 \Pr(\mathbf{S}|a, \vec{n}) da \right) \quad (6.6)$$

Vamos supor também que as anomalias κ_i das componentes S_i de \mathbf{S} são variáveis aleatórias independentes entre si, e portanto

$$\Pr(\mathbf{S}|a, \vec{n}) = \prod_{i=1}^m \Pr(S_i|a, \vec{n}) \quad (6.7)$$

Portanto a função Γ pode ser escrita como

$$\Gamma(a, \vec{n}|\mathbf{S}) = \sum_{i=1}^m \log \Pr(S_i|a, \vec{n}) \quad (6.8)$$

e a função Δ como

$$\Delta(\vec{n}|\mathbf{S}) = \log \left(\int_0^1 \exp \left[\sum_{i=1}^m \log \Pr(S_i|a, \vec{n}) \right] da \right) \quad (6.9)$$

A integral desta fórmula pode ser calculada por qualquer método de integração numérica. Na nossa implementação utilizamos a fórmula de integração por trapézios [81] com $\lceil \sqrt{m}/\sigma \rceil$ passos.

6.4.3 Modelo probabilístico das anomalias

Para a distribuição condicional $\Pr(S_i|a, \vec{n})$ da intensidade do pixel, supomos que, em cada imagem S_i , o pixel p pode estar em um de três casos disjuntos: ele é canônico, ele está afetado por anomalia negativa (sombra projetada) ou ele está afetado por anomalia positiva (iluminação indireta e/ou realces não incluídos na função de tonalização L).

No primeiro caso supomos que $S_i = \mu_i + N_i$ onde μ_i é a intensidade esperada $aL(\vec{n})$ e N_i é um erro de medida com distribuição gaussiana de média zero e desvio padrão σ conhecido; ou seja,

$$\Pr_C(S_i|a, \vec{n}) = \gamma(S_i - aL_i(\vec{n}), \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-r_i^2/2} \quad (6.10)$$

onde $r_i = (S_i - aL_i(\vec{n}))/\sigma$.

Por outro lado, quando p está em sombra projetada na imagem i supomos que o valor de S_i é um valor arbitrário menor que $\eta_i = aL_i(\vec{n}) + 2\sigma$, acrescido de um erro de medida com distribuição gaussiana de média 0 e desvio padrão σ . Ou seja, a distribuição de S_i é um degrau com transição suave descendente (sigmóide):

$$\Pr_S(S_i|a, \vec{n}) = \left(\frac{1}{\eta_i} \right) \left(\frac{1 - \text{erf}(r_i)}{2} \right) \quad (6.11)$$

onde $\text{erf}(z) = \int_0^z \gamma(x, 1)dx$.

Acrescentamos o termo $+2\sigma$ na forma de η_i para evitar problemas numéricos no caso de $aL_i(\vec{n})$ ser próximo a zero. Fisicamente, isto equivale a supor que mesmo os pixels sombreados podem ter uma anomalia ligeiramente positiva.

Analogamente, na hipótese de que p está afetado por anomalia positiva, adotamos uma distribuição de probabilidade em degrau ascendente suave

$$\Pr_H(S_i|a, \vec{n}) = \left(\frac{1}{\lambda_i} \right) \left(\frac{1 + \text{erf}(r_i)}{2} \right) \quad (6.12)$$

onde $\lambda_i = 1 - aL_i(\vec{n}) + 2\sigma$

Supondo que a probabilidade de p ter anomalia negativa seja ω_S , e a de ter anomalia positiva seja ω_H , podemos então escrever

$$\Pr(S_i|a, \vec{n}) = \omega_S \Pr_S(S_i|a, \vec{n}) + \omega_H \Pr_H(S_i|a, \vec{n}) + (1 - \omega_S - \omega_H) \Pr_C(S_i|a, \vec{n}) \quad (6.13)$$

A figura 6.4 mostra a forma típica desta distribuição. A figura 6.5 mostra o logaritmo da fórmula (6.13). Note que ele poderia ser aproximado por uma fórmula mais eficiente, por exemplo, uma função quadrática por partes. Nesta tese, entretanto, não recorremos a essa otimização.

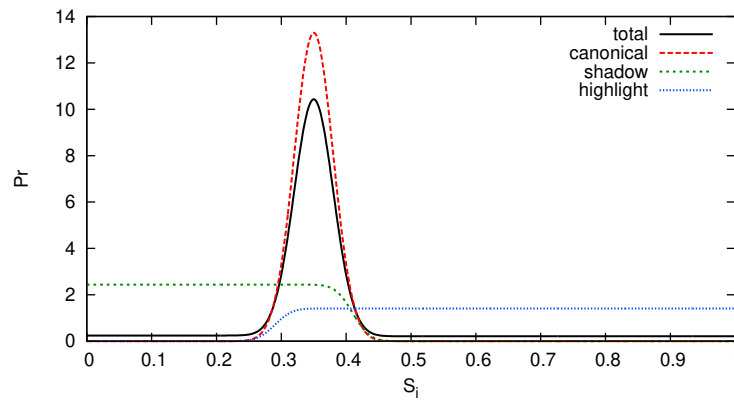


Figura 6.4: Gráfico típico da distribuição $\Pr(S_i|a, \vec{n})$ adotada para as intensidade S_i (linha cheia), para $\sigma = 0.03$, $L_i(\vec{n}) = 0.35$, $\omega_S = 0.1$ e $\omega_H = 0.15$, mostrando as probabilidades específicas $\Pr_C(S_i|a, \vec{n})$, $\Pr_S(S_i|a, \vec{n})$ e $\Pr_H(S_i|a, \vec{n})$ (linhas pontilhadas e tracejadas).

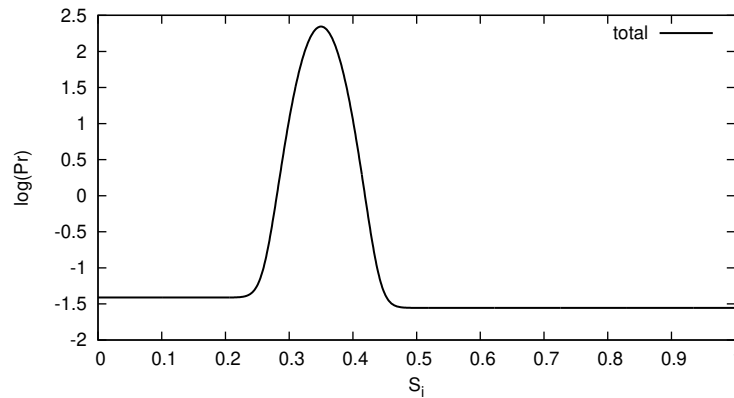


Figura 6.5: Gráfico típico da distribuição $\log(\Pr(S_i|a, \vec{n}))$ adotada com as mesmas condições da figura 6.4.

6.5 Maximização da probabilidade

Mesmo com todas estas simplificações, as funções Γ e Δ são extremamente não lineares. Além disso, quando há anomalias em uma ou mais imagens, elas podem ter vários mínimos locais. Encontrar o seu máximo não é uma operação trivial. As figuras 6.6 e 6.7 ilustram as projeções estereográficas em \mathbb{H}^2 das funções Γ e Δ .

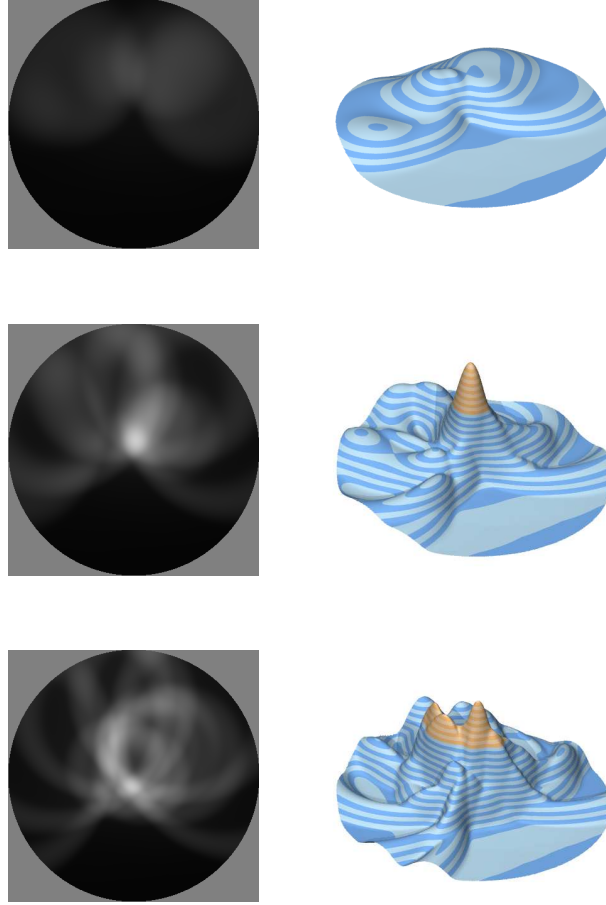


Figura 6.6: Gráficos bidimensionais (esquerda) e tridimensionais (direita) da função $\Gamma(a, \vec{n}|\mathbf{S})$ em projeção estereográfica do hemisfério \mathbb{H}^2 para os valores de albedo $a = 0.50$ (topo), $a = 0.66$ (centro) e $a = 0.80$ (embaixo). Estes gráficos supõem função de tonalização \mathbf{L} de uma superfície com acabamento lambertiano sob $m = 12$ iluminações unidirecionais distintas. O vetor de observação \mathbf{S} é o de um pixel com essa função de tonalização, normal $\vec{n}[p] = (0, 0, 1)$ e albedo relativo $a = 0.66$ com anomalias em quatro das m iluminações. Valores negativos estão indicados com tons azulados e valores positivos com tons avermelhados. Observe que o valor máximo da função corresponde ao par (a, \vec{n}) correto (gráficos do centro).

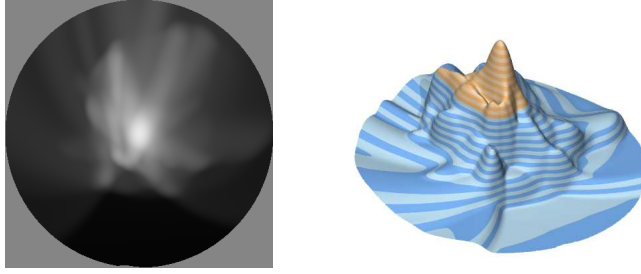


Figura 6.7: Gráficos bidimensionais (esquerda) e tridimensionais (direita) da função $\Delta(\vec{n}|\mathbf{S})$ em projeção estereográfica do hemisfério \mathbb{H}^2 . Estes gráficos supõem a mesma função de tonalização \mathbf{L} , iluminações e vetor de observação \mathbf{S} da figura 6.6. O valor máximo da função corresponde à direção $\vec{n}[p]$ correta.

6.5.1 Enumeração exaustiva

Um algoritmo simples mas de alto custo para maximizar $\Gamma(a, \vec{n}|\mathbf{S})$ é enumerar um conjunto suficientemente amplo de pares (a_j, \vec{n}_r) combinando um conjunto de normais \vec{n}_r que cobre todo \mathbb{H}^2 e um conjunto de valores de albedo a_j que cobre todo intervalo $[0, 1]$. Para cada um desses pares (a_j, \vec{n}_r) , avaliamos $\Gamma(a_j, \vec{n}_r|\mathbf{S})$, e escolhemos o par que fornece o maior valor de Γ . Em princípio o custo desta abordagem é proporcional a Mm onde M é o numero de pares (a_j, \vec{n}_r) testados. Com M suficientemente grande esta abordagem pode ser bastante precisa e robusta, mas tem custo excessivo. Outra abordagem similar é calcular o valor de $\Delta(\vec{n}_r|\mathbf{S})$ para todas as normais \vec{n}_r e escolher o maior valor. Note que cada cálculo de $\Delta(\vec{n}, \mathbf{S})$ essencialmente exige o cálculo de $\Pr(\mathbf{S}|a, \vec{n})$ para um conjunto denso de valores de albedo a , e portanto esta alternativa *não* é mais rápida que a maximização de Γ .

6.5.2 Enumeração exaustiva de subconjuntos

Outra maneira de determinar os valores corretos de a e \vec{n} é enumerar todos os subconjuntos K_1, K_2, \dots, K_M de $\{1, \dots, m\}$ com três imagens, e efetuar a inversão da função \mathbf{l} correspondente a cada subconjunto, supondo p canônico nessas três imagens. Obtemos assim para cada conjunto K_r uma normal tentativa \vec{n}_r e um albedo a_r . Para cada um destes pares (a_r, \vec{n}_r) , calculamos a função $\Gamma(a_r, \vec{n}_r|\mathbf{S})$, e escolhemos o par a_r, \vec{n}_r com maior Γ . Analogamente, pode-se utilizar a função $\Delta(\vec{n}_r|\mathbf{S})$, que leva em conta apenas as normais \vec{n}_r de cada conjunto.

O problema dessa abordagem é que o número de M conjuntos a testar é $\binom{m}{3} \approx m^3/6$, o que torna este método extremamente lento quando se tem mais de uma dezena de iluminações distintas.

6.5.3 Consenso com amostragem aleatória

Uma elaboração da estratégia da seção 6.5.2 é a abordagem de *consenso com amostragem aleatória* [35] (*random sampling consensus* - RANSAC). Nesta abordagem, em vez de todos os subconjuntos de 3 elementos, usa-se apenas uma amostra aleatória de R subconjuntos K_1, K_2, \dots, K_R de $\{1, \dots, m\}$, todos com um número fixo $k \geq 3$ de elementos. Como na seção 6.5.2, para cada conjunto K_r é calculada uma normal \vec{n}_r e um albedo a_r , pelos métodos dos capítulos 2 e 5. Dentre todas as soluções (a_r, \vec{n}_r) assim obtidas, escolhemos o par (a_j, \vec{n}_r) que tem maior $\Gamma(a, \vec{n}|\mathbf{S})$, ou a normal \vec{n}_r com maior $\Delta(\vec{n}, \mathbf{S})$. A vantagem deste método é que R pode ser bem menor que $\binom{m}{k}$. Note que se R não é muito grande, para cada conjunto K_r podemos construir uma estrutura de grade (capítulo 5) que acelera bastante o cálculo da normal \vec{n}_r e albedo a_r . Além disso, se $k \geq 4$, este método é também mais robusto que o método da seção 6.5.2 em face de erros de medida.

Porém este método pode falhar se, por azar, nenhum dos conjuntos K_r escolhidos satisfizer a hipótese de dados suficientes. Para reduzir este risco, o valor de R não pode ser muito pequeno. A probabilidade de que um conjunto K_r com k imagens só tenha imagens canônicas para o pixel p é $t = (1 - m_0/m)^k$. Para garantir que o método RANSAC falhe com probabilidade menor que ϵ , é necessário usar R tal que $(1 - t)^R < \epsilon$, ou seja

$$R \geq \log(\epsilon) / \log(1 - t) = \log_{1-t} \epsilon \quad (6.14)$$

Por exemplo, se temos $m = 30$ imagens, das quais no máximo $m_0 = 10$ são anômalas para cada pixel, utilizando subconjuntos de tamanho $k = 4$, precisaremos de $R \geq 33$ subconjuntos para que a probabilidade de falha ϵ seja menor que 0.1%.

6.5.4 Refinamento da solução

Tanto o método de enumeração exaustiva de subconjuntos (seção 6.5.2) e método RANSAC (seção 6.5.3) são muito sensíveis ao ruído nas imagens, pois utilizam apenas k imagens para calcular cada normal \vec{n}_r , ignorando a informação contida nas imagens remanescentes. Dependendo do nível σ de ruído de medida, pode ser que nenhuma das normais \vec{n}_r calculadas esteja próxima à normal correta \vec{n} .

Para reduzir este último problema, podemos usar o método acima apenas para escolher uma normal provisória $\tilde{\vec{n}}$. Feito isso, repetimos o cálculo da normal (por qualquer das técnicas dos capítulos anteriores) usando o subconjunto K de todas as imagens em que p parece ser canônico para essa normal provisória (isto é, em que $S_i - aL_i(\tilde{\vec{n}})$ é menor que um certo limiar, digamos 3σ).

6.6 Testes

Para validar a eficácia e viabilidade da abordagem bayesiana, executamos alguns testes usando o método RANSAC da seção 6.5.3 com subconjuntos K_1, \dots, K_R de tamanho $k = 4$ (os mesmos subconjuntos para cada pixel). Para calcular a solução (a_r, \vec{n}_r) de cada subconjunto K_r usamos o método BGRID, construindo uma grade de baldes específica para esse subconjunto. A normal foi selecionada pelo critério de maior Δ , sem o refinamento da seção 6.5.4. Usamos dois tipos de dados de teste: imagens sintéticas e fotos digitais de uma cena real. Para todos os testes, supusemos $\sigma = 0.003$, $\omega_S = 0.1$ e $\omega_H = 0.01$.

6.6.1 Testes com cena sintética lambertiana

Para a abordagem bayesiana em cenas com sombras projetadas, criamos o conjunto de dados de teste sintético **shlamb** que consiste de $m = 11$ imagens produzidas pelo software POV-Ray [111] a partir da mesma cena virtual usada no teste **sglamb** da seção 5.4.1. Duas das imagens foram renderizadas com iluminação quase vertical, e portanto quase livre de sombras projetadas. Nas outras 9 imagens, usamos fontes de luz com inclinações θ entre 25° e 45° , que produziram sombras projetadas em regiões extensas da cena. Veja a figura 6.8.

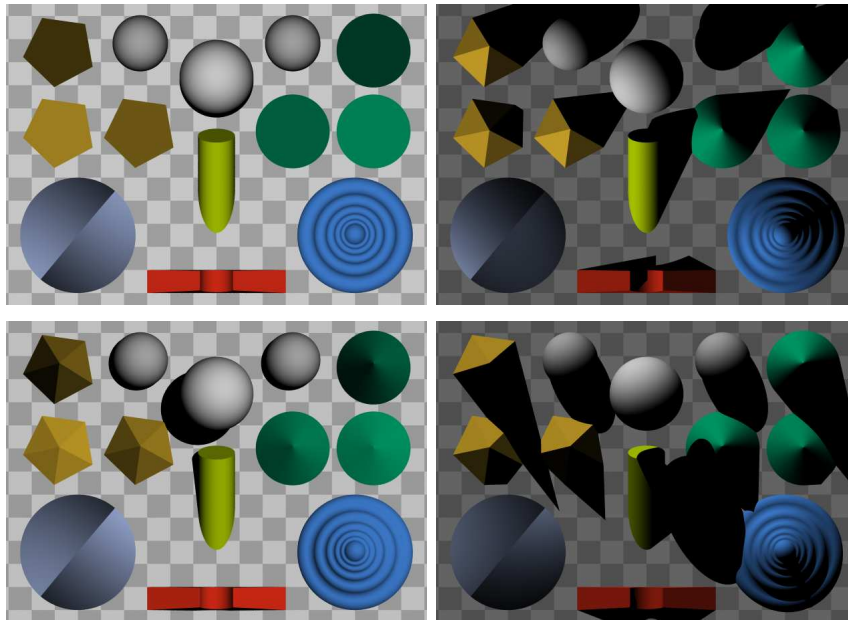


Figura 6.8: Quatro das $m = 11$ imagens sintéticas da cena **shlamb**. Observe que a esfera perto do centro da imagem está “flutuando” bem acima da base.

Todos os objetos da cena foram modelados com acabamento lambertiano. A distância das fontes de luz à cena era aproximadamente 100 vezes a largura da parte visível da cena, e a câmera virtual foi posicionada longe o suficiente para se obter uma projeção essencialmente paralela com $\vec{v}[p] = (0, 0, 1)$ em todo pixel.

Para calcular o albedo e as normais usamos o método RANSAC com $R = 20$ subconjuntos K_r . Para este teste adotamos um gabarito virtual \mathbf{L} de acabamento lambertiano com as iluminações unidirecionais usadas no POV-Ray. As tabelas de assinaturas foram obtidas amostrando o gabarito virtual \mathbf{L} em $N = 11172$ normais distribuídas sobre o hemisfério \mathbb{H}^2 (as mesmas normais usadas nos testes da seção 5.4.1). Como na seção 5.4.1, usamos o canal de luminância Y para calcular as normais $\vec{n}[p]$, e os canais R,G,B para calcular os albedos $a[p, c]$. A figura 6.9 mostra os mapas resultantes de normal e cor própria. Observamos que os maiores erros ocorrem nos pixels próximos à esfera situada logo acima do centro da imagem: eles estão na sombra projetada em quase todas as imagens, e a hipótese de dados suficientes não era satisfeita por nenhum dos conjuntos K_r considerados. No restante da cena, os erros são muito reduzidos. Veja a figura 6.10.

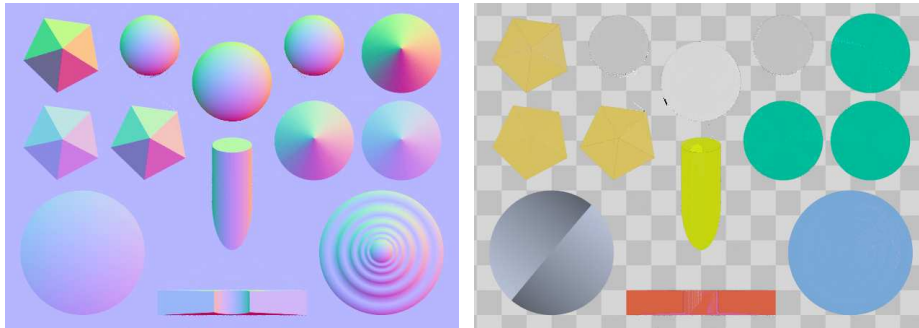


Figura 6.9: Mapas de normais e de cor intrínseca calculados pelo método RANSAC com $k = 4$ e $R = 20$ para o conjunto de $m = 11$ imagens sintéticas do teste **shlamb**.

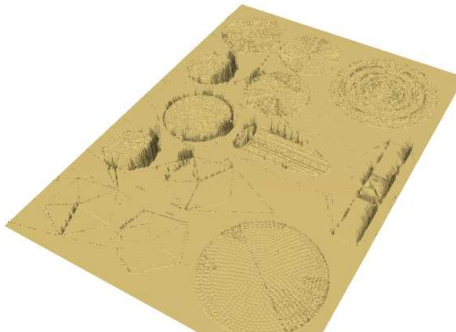


Figura 6.10: Gráfico dos erros das normais calculadas para o conjunto de teste **shlamb**. O erro máximo E_{\max} corresponde a um ângulo de 1.51 radianos (aproximadamente 87°) e o erro médio E_{rms} a 0.04 radianos (2.5°).

A figura 6.11 mostra os mapas de anomalias κ_i para o teste **shlamb**. Observe que há apenas anomalias negativas (sombras projetadas), exceto nos pontos onde o método falhou.

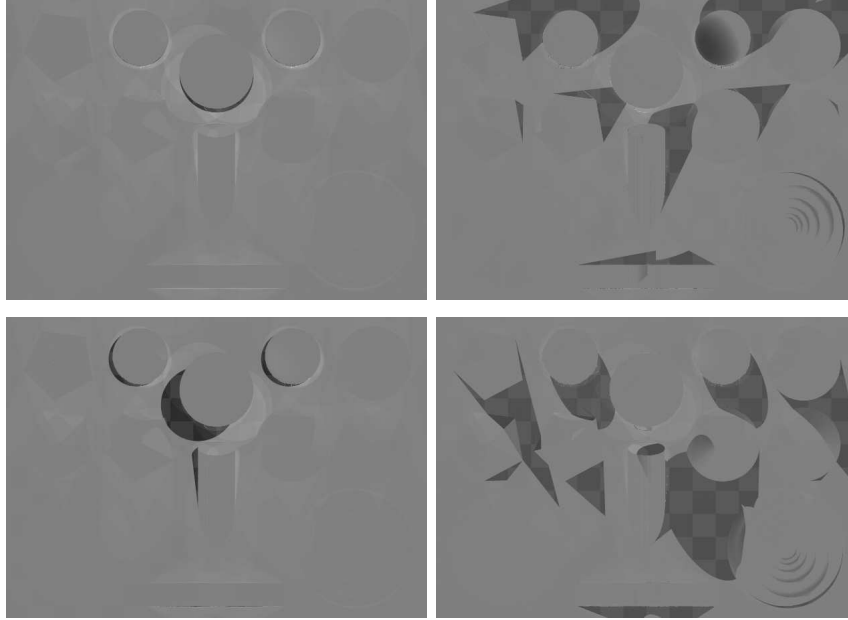


Figura 6.11: Mapas de anomalia κ_i do canal Y calculados para as quatro imagens S_i da figura 6.8 do conjunto de testes **shlamb**

6.6.2 Testes com cena sintética lustrosa

Para testar a abordagem bayesiana em cenas com acabamento lustroso, produzimos o conjunto de dados de testes sintéticos **shlstr**. Ele consiste de $m = 13$ imagens produzidas pelo POV-Ray, a partir da mesma cena virtual usada no teste **shlamb** da seção 6.6.1. Neste teste, entretanto, aplicamos a toda a cena o mesmo acabamento, parcialmente lustroso e parcialmente especular e o mesmo albedo, obtidos especificando os parâmetros do traçador **pigment** = $\langle 0.8, 0.8, 0.8 \rangle$, **difuse** = 0.5, **specular** = 0.4 e **roughness** = 0.05. (O acabamento **specular** do traçador é similar, mas não idêntico ao de Phong.)

Para obter o gabarito, extraímos 9050 pares (\vec{n}_r, \vec{l}_r) da esfera central da cena (que está livre de sombras em todas as imagens) e aproximamos a função vetor de tonalização pela base **RADIAL** da seção 4.7. Veja a figura 6.12.

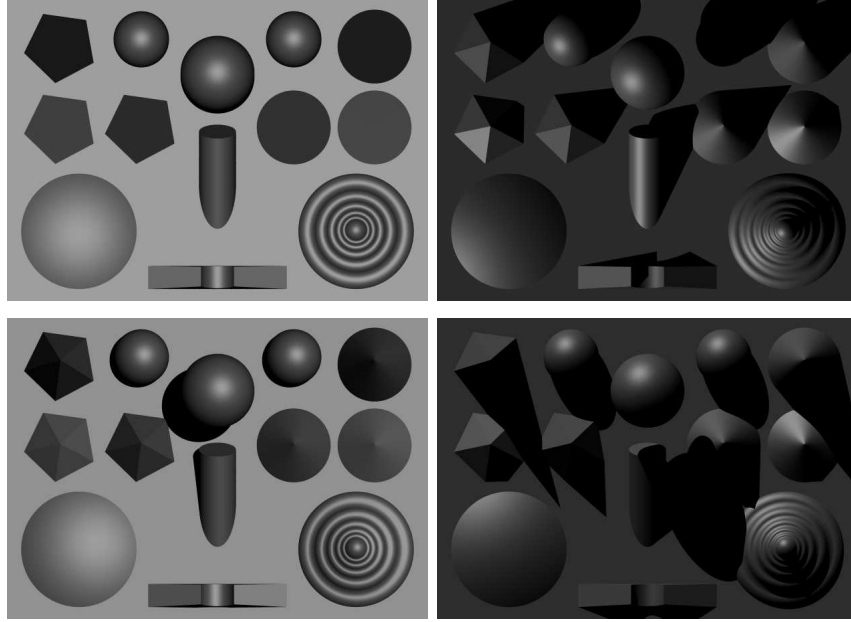


Figura 6.12: Seis das $m = 13$ imagens sintéticas do conjunto de teste **sh1str**.

Para calcular o albedo e as normais usamos o método RANSAC com $R = 20$ subconjuntos de tamanho $k = 4$. As tabelas de assinaturas foram obtidas por amostragem da função de tonalização \tilde{L} em $N = 10550$ normais distribuídas sobre o hemisfério \mathbb{H}^2 . Os mapas resultantes de normal e de albedo estão apresentados na figuras 6.13, e o erro das normais calculadas na figura 6.14.

Observamos que foi necessário aumentar um pouco o número de iluminações m em relação ao teste **sh1amb** (de 11 para 13) para obter resultados aceitáveis em quase todos os pixels. Observamos também um erro sistemático de $\approx 2^\circ$ nas normais, além dos erros esperados, possivelmente devido a uma estimativa incorreta da direção de vista $\vec{v}[p]$ no centro do gabarito durante a amostragem do mesmo para construção do gabarito virtual \tilde{L} . A figura 6.15 mostra os mapas de anomalias κ_i . Observe que os realces lustrosos não geram anomalias positivas, uma vez que eles também ocorrem no gabarito.

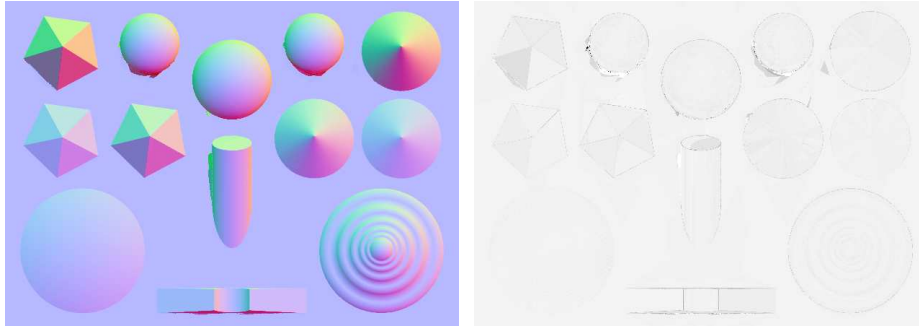


Figura 6.13: Mapas de normais e de cor intrínseca calculado pelo método RANSAC para o conjunto de $m = 19$ imagens sintéticas do teste **sh1str**

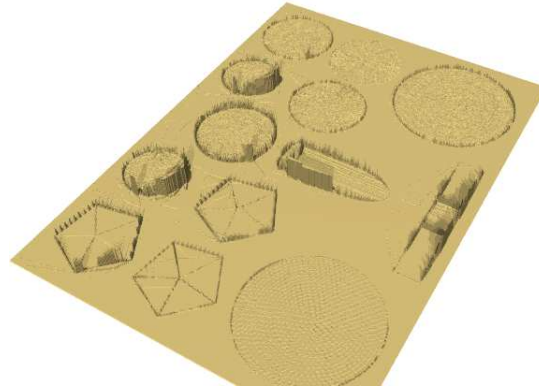


Figura 6.14: Gráfico dos erros das normais calculadas do conjunto **shlstr**. O erro máximo E_{\max} corresponde a um ângulo de 1.40 radianos (aproximadamente 80°) e o erro médio E_{rms} a 0.08 radianos (4°), incluindo o erro sistemático de aproximadamente 2° (vide texto).

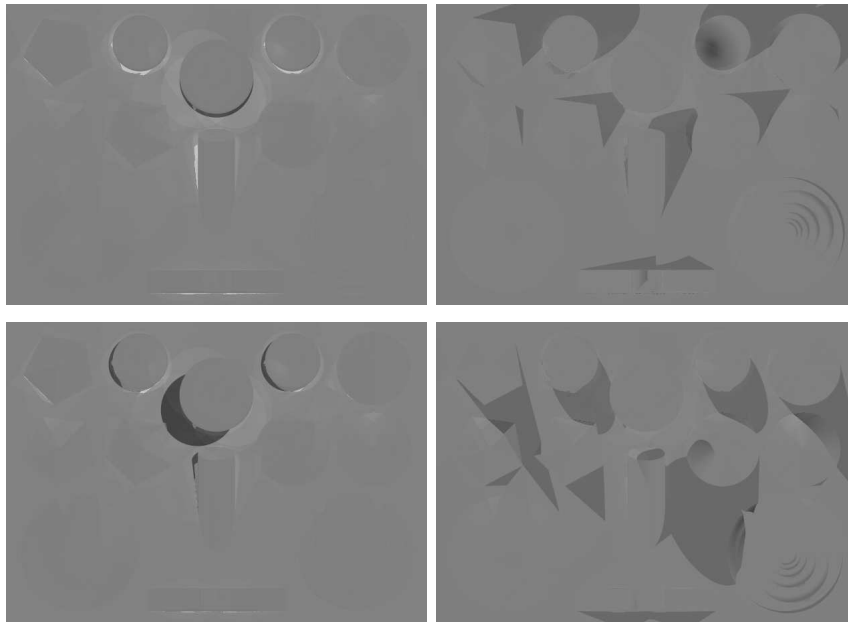


Figura 6.15: Mapas de anomalia κ_i do canal Y calculados para as seis imagens S_i da figura 6.12, do conjunto de teste **shlstr**.

6.6.3 Teste com cena real lambertiana

Para um teste mais realista da abordagem bayesiana, criamos o conjunto de teste **rs lamb** que consiste em $m = 17$ fotos digitais da mesma escultura de gesso e gabaritos utilizados na cena **rlamb** da seção 5.4.3, em disposição similar. As fotos foram obtidas com a mesma câmera de 3 megapixels (Sony CyberShot DSC-V1). Usamos também o mesmo protocolo

e a mesma luminária posicionada em várias direções com azimutes entre 0° e 270° e inclinações de θ entre 25° e 45° . Estas inclinações, maiores que as usadas no teste **rlamb**, deram origem a sombras projetadas sobre algumas partes da cena em várias das imagens. Tomamos precauções para que a hipótese de dados suficientes pudesse ser satisfeita para toda a cena, embora com subconjuntos K de imagens diferentes para cada pixel. As fotos foram alinhadas manualmente por translação com precisão de ± 0.35 pixels, cortadas de 1536×2048 para 1000×672 pixels, e reduzidas para 225×163 pixels (25%). Veja a figura 6.16.

Para este teste adotamos uma função de tonalização \tilde{L} obtida a partir das imagens do mesmo gabarito real usado no conjunto **rlamb**, aproximado pelo modelo **RADIAL**. Para calcular as normais, usamos o método RANSAC, com $R = 20$ subconjuntos de $k = 4$ elementos. As tabelas de assinaturas foram construídas amostrando o gabarito virtual \tilde{L} em $N = 10967$ normais no hemisfério \mathbb{H}^2 . Usamos o canal de luminosidade Y para calcular a normal, e os canais R,G,B separadamente para calcular os albedos $a[p, c]$. A figura 6.17 mostra os mapas de normais e de cor intrínseca obtidos desta forma.



Figura 6.16: Seis das $m = 17$ fotos do teste **rsmb**. Note a presença de sombras projetadas em várias destas imagens.

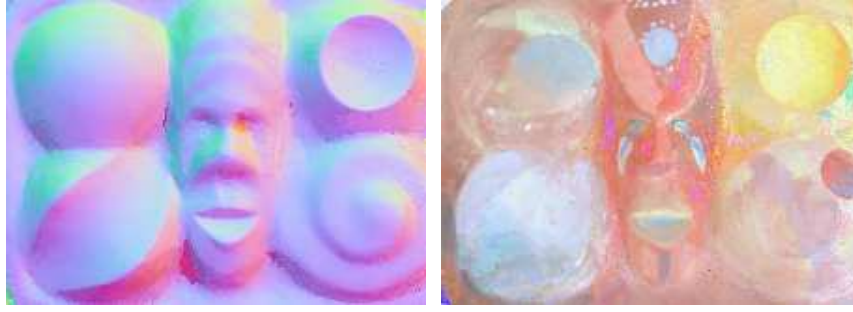


Figura 6.17: Mapas de normais e cor intrínseca calculados pelo método RANSAC com $R = 20$ e $k = 4$ para as $m = 17$ fotos do conjunto **rslamb**.

Observamos uma ligeira anomalia positiva no interior da “cratera” no canto superior direito das fotos. Aparentemente esta anomalia não é devida à realce lustroso ou luz indireta, mas ao fato (constatado após o experimento) de que a tinta laranja usada nessa parte da cena era ligeiramente fluorescente, ou seja, tinha albedo ligeiramente maior que 1. Como nossos algoritmos limitam o albedo ao intervalo $[0, 1]$, eles interpretaram esse excesso de radiância como anomalia.

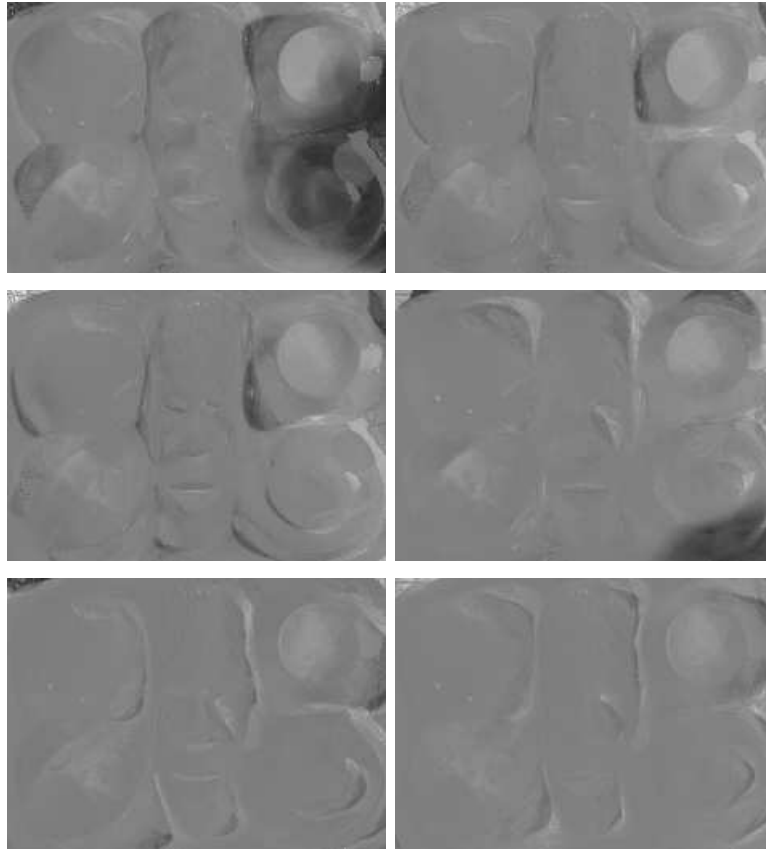


Figura 6.18: Mapas de anomalia κ_i do canal Y calculados para as seis imagens S_i da figura 6.16, do conjunto de teste **rslamb**.

6.6.4 Teste com cena real lustrosa

Para um teste realista de eficácia da abordagem bayesiana na presença de anomalias de acabamento, criamos o conjunto de dados de teste `rslatx` consistindo de $m = 5$ fotos de uma cena real com acabamento parcialmente lustroso; especificamente, uma bola de tênis de mesa envolvida em duas camadas de látex (bexigas de festa) com um “S” de arame inserido entre as duas camadas. Parte desta cena foi pintada com canetas marcadoras permanentes vermelha e preta; essas partes ficaram com acabamento mais lustroso que o resto da superfície.

Para capturar a função de tonalização, usamos o mesmo gabarito do teste `rlatx` da seção 4.7.4 (uma bola de tênis de mesa coberta com duas camadas de látex, sem pintura). Tiramos 5 fotos da cena, e separadamente, do gabarito, com as mesmas 5 fontes de luz do teste `rlatx` e com a mesma câmera e protocolo (câmera AVT Pike FC-100, ajustada para produzir imagens coloridas de 500×500 pixels com codificação linear). Veja a figura 6.19. Observe que, dado o número reduzido de fotos, a hipótese de dados suficientes foi violada para várias normais com inclinação próxima a 90° .

A função de tonalização $\tilde{\mathbf{L}}$ foi obtida aproximando as imagens do gabarito real com a base `RADIAL`. Para calcular as normais e o albedo, usamos o método RANSAC com $R = 10$ e $k = 3$ elementos (ou seja, todos os subconjuntos de $\{1 \dots 5\}$ de tamanho 3). As tabelas de assinaturas foram obtidas amostrando o gabarito virtual $\tilde{\mathbf{L}}$ em $N = 11115$ normais no hemisfério \mathbb{H}^2 . A figura 6.20 mostra os mapas de normais e de cor intrínseca resultantes.

A figura 6.21 mostra os mapas de anomalia κ_i . Note as manchas claras nas regiões onde a cena tem realces mais intensos que o gabarito.



Figura 6.19: As $m = 5$ fotos do teste real `rslatx`. Note os realces lustrosos intensos em vários pontos do objeto.

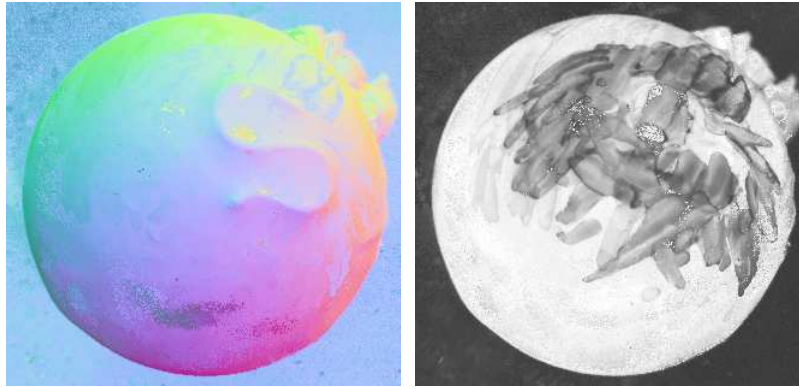


Figura 6.20: Mapas de normais e de cor intrínseca calculados pelo método RANSAC com $R = 10$ e $k = 3$ a partir de $m = 5$ imagens do teste **rslatx**.



Figura 6.21: Mapas de anomalia κ_i calculados para as imagens da figura 6.19.

Parte II

Integração do mapa de normais

Capítulo 7

Integração multi-escala de mapas de gradiente

Métodos de estéreo fotométrico não oferecem diretamente o valor do mapa de alturas da cena. Como observado nos capítulos anteriores, os mesmos retornam apenas o mapa de normais da superfície observada. Entretanto, é possível converter esse resultado em um *mapa de gradiente* (derivada da altura em relação às coordenadas x e y da imagem), e então obter o valor das alturas por *integração* desse mapa de gradientes.

Neste capítulo, descreveremos algoritmos para efetuar tal integração. Em particular, descreveremos uma abordagem desenvolvida por nós que supera os melhores algoritmos conhecidos em escalabilidade e robustez.

7.1 Definição do Problema

Abstratamente, o problema de *integração do gradiente* consiste em determinar uma função real desconhecida Z de duas variáveis x e y , definida em uma região D de \mathbb{R}^2 , dado seu gradiente $\nabla Z = (\partial Z/\partial x, \partial Z/\partial y)$ nessa região. Isto é, desejamos calcular Z tal que

$$\frac{\partial Z}{\partial x}(x, y) = F(x, y) \quad \frac{\partial Z}{\partial y}(x, y) = G(x, y) \quad (7.1)$$

para cada ponto (x, y) dentro da região D ; onde F e G são duas funções reais conhecidas definidas em D . Este problema possui uma solução diferenciável se e somente se

$$\frac{\partial F}{\partial y}(x, y) - \frac{\partial G}{\partial x}(x, y) = 0 \quad (7.2)$$

para todo (x, y) em D . O lado esquerdo da fórmula (7.2) é denominado *rotacional* do campo vetorial (F, G) ; por essa razão a condição (7.2) é frequentemente chamada de

condição de rotacional zero. Satisfeita essa condição, a solução Z pode ser obtida de diversas maneiras. Para um domínio D retangular com canto inferior em $(0,0)$, por exemplo, ela pode ser descrita pela fórmula

$$Z(x, y) = C + \int_0^y G(0, v) dv + \int_0^x F(u, y) du \quad (7.3)$$

onde C é uma constante arbitrária. Note que o grau de liberdade representado por C é uma característica inerente do problema original e não uma limitação do método.

7.1.1 Integração de gradientes obtidos por estéreo fotométrico

Na prática, existem pelo menos três complicações com esta abordagem, que quase sempre ocorrem nos dados obtidos por estéreo fotométrico. Em primeiro lugar, as funções de gradiente F e G são em geral discretizadas, isto é, fornecidas apenas em certos pontos de amostragem $p[u, v]$, formando uma grade ortogonal regular. Cada amostra $f[u, v]$ e $g[u, v]$ é uma média ponderada das derivadas $F[u, v]$ e $G[u, v]$ respectivamente, na vizinhança de $p[u, v]$. Note que métodos de estéreo fotométrico tipicamente estimam essas duas derivadas nos mesmos pontos, isto é no centro de cada pixel das imagens da cena.

Em segundo lugar, os dados estão normalmente contaminados com “ruídos” desconhecidos decorrentes de erros inevitáveis de medição, quantização e representação. Em certos pontos, a magnitude esperada dessas perturbações pode ser tão grande que o valor do gradiente é completamente desconhecido. No caso de estéreo fotométrico em particular, isto acontece em todos os pixels onde a cena é muito escura ou está afetada por sombras ou reflexos, em todas as imagens S_i .

Terceiro, a função Z quase sempre tem *degraus* (descontinuidades) da altura $Z(x, y)$ nas bordas das projeções dos objetos discretos e de algumas partes protuberantes da cena. Métodos de aquisição de gradiente, incluindo os métodos de estéreo fotométrico vistos na parte I desta tese, não conseguem obter o valor correto do gradiente nos pixels que cruzam estes degraus. Em geral, os dados $f[u, v]$ e/ou $g[u, v]$ fornecidos para estes pixels são valores arbitrários que não dão nenhuma informação sobre a altura real do degrau. Veja a figura 7.1. Isto acontece porque o cálculo de $f[u, v]$ e $g[u, v]$ por estes métodos é um processo extremamente não-linear, especialmente quando a normal é quase perpendicular ao ponto de vista. As amostras $f[u, v]$ ou $g[u, v]$ serão médias locais adequadas de F e G apenas quando a cena tiver inclinação constante em toda área de amostragem do pixel.

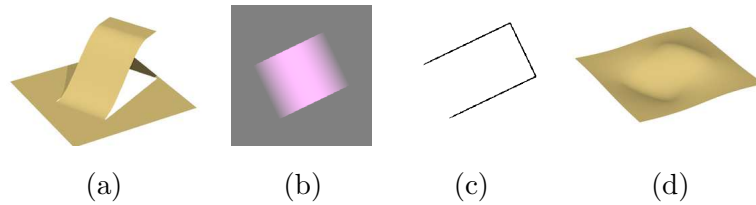


Figura 7.1: Um mapa de alturas com degraus (a), seu mapa de gradiente (b) obtido por estéreo fotométrico, e uma máscara binária (c) mostrando a localização dos degraus. A imagem (d) mostra o mapa de alturas calculado pelo método de Frankot-Chellapa [37], que não leva em conta esses degraus.

Finalmente, a cena pode conter regiões onde as funções de altura e gradiente são indefinidas; por exemplo, onde a superfície é extremamente porosa, coberta por estruturas finas como cabelos, ou transparente.

7.1.2 Mapas de peso

A figura 7.1 mostra que a existência e a posição de degraus (ou de outros erros nos dados) não podem ser deduzidas a partir do mapa de gradientes apenas. Nesse exemplo, o mapa não fornece nenhuma indicação de qual extremidade da rampa está conectada ao plano a partir dos valores armazenados no mesmo.

Para contornar esta limitação, algoritmos robustos de integração de gradientes precisam de um dado adicional: um *mapa de pesos* w , que especifica a confiabilidade de cada amostra do gradiente. Este mapa pode ser uma simples máscara binária, com valor $w[u, v] = 0$ quando os valores de $f[u, v]$ e/ou $g[u, v]$ são desconhecidos ou inconfiáveis, e 1 caso contrário. Veja a figura 7.1 (c). Valores fracionários entre 0 e 1 podem ser usados para indicar graus intermediários de confiabilidade. Idealmente, o valor de $w[u, v]$ deve ser inversamente proporcional ao valor estimado da variância do ruído de medição que afeta as amostras $f[u, v]$ e $g[u, v]$. Se todas as amostras são igualmente confiáveis, pode-se atribuir 1 a todos os pesos.

É possível obter tal mapa de pesos de diversas maneiras, seja por meio de informação externa (por exemplo, por edição pelo usuário, câmeras *time-of-flight*, estéreo geométrico, análise de sombras, etc.) ou, em certos casos, por algoritmos de detecção aplicados ao mapa de gradiente [5, 6, 25, 13, 65, 82]. O mapa de pesos também pode ser um subproduto do algoritmo de estéreo fotométrico que produziu o mapa de gradientes.

7.1.3 Calculando gradientes a partir das normais

Muitos softwares de estéreo fotométrico devolvem um mapa de vetores normais \vec{n} (vide capítulo 2). Estritamente falando, a transformação do mapa \vec{n} em um mapa de gradiente

não é possível, pois a projeção de perspectiva cônica introduz uma distorção que depende da altura desconhecida $Z = s[p].z$.

Portanto para integração é necessário supor que a projeção é cilíndrica, isto é, que a câmera está situada a uma distância tão grande da cena que a imagem é praticamente uma projeção ortogonal desta última. Nesse caso, podemos identificar as coordenadas (x, y, z) da cena com as do sistema de coordenadas da imagem, isto é, com as coordenadas (x, y) do centro do pixel na imagem, e a altura $Z(x, y)$ da cena com a distância do ponto $s[p]$ ao plano de projeção, e podemos calcular o valor do gradiente a partir da normal $\vec{n}[p]$ pelas fórmulas

$$f[x, y] \leftarrow -\frac{\vec{n}.x}{\vec{n}.z} \quad g[x, y] \leftarrow -\frac{\vec{n}.y}{\vec{n}.z} \quad (7.4)$$

7.2 Trabalhos relacionados

A maioria dos algoritmos de integração de mapas de gradiente podem ser classificados em 4 grupos principais: *integração por caminhos*, *filtragem via Fourier*, *resolução direta* e *métodos iterativos locais*.

7.2.1 Métodos de integração por caminhos

Os métodos de integração por caminho atribuem uma altura arbitrária a um pixel de referência p , para então calcular a altura de qualquer outro pixel q por uma integral de linha do gradiente ao longo de um caminho de p até q . Este grupo inclui a simples integração linha-a-linha, que é a versão discreta da equação (7.3) [109]. Métodos mais sofisticados, utilizam caminhos mais complexos a fim de evitar dados ausentes ou de baixa qualidade, como os de Fraile e Hancock [36] e Agrawal [5]. Métodos de integração por caminhos são em geral muito rápidos, uma vez que eles efetuam apenas $O(N)$ operações para uma imagem com N pixels. Eles são, porém, muito sensíveis a ruído nos mapas de gradiente. Se a altura de dois pixels adjacentes p', p'' é calculada por caminhos distintos, a integração do componente de ruído do gradiente ao longo de cada caminho irá resultar em uma diferença de alturas espúria entre eles. Veja a figura 7.2.

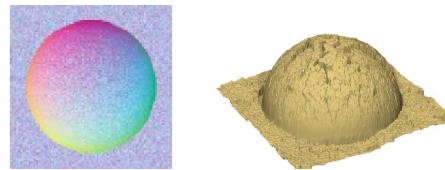


Figura 7.2: Reconstrução de um mapa de gradientes com ruído pelo método de integração de caminhos de Fraile and Hancock [36]

Este problema pode ser atenuado, mas não resolvido, calculando-se a média das integrais obtidas por diversos caminhos distintos entre os dois pixels [84]. Robles-Kelly e Hancock propuseram um método deste tipo onde o gradiente é integrado ao longo de uma linha reta entre cada par de pixels do mapa para obter uma estimativa da diferença de suas alturas. A média destas diferenças é usada para obter a altura de cada pixel. Embora esta abordagem evite a formação de degraus espúrios, seu custo é proibitivo (proporcional a $N^{2.5}$ para uma imagem com N pixels) e seus resultados ainda assim são inferiores aos resultados obtidos por outros métodos.

7.2.2 Métodos de filtragem via Fourier

Outra classe de métodos baseia-se na observação de que integrar uma função corresponde a dividir cada componente de sua transformada de Fourier por 2π vezes a sua frequência. Esta abordagem foi proposta inicialmente por Frankot e Chellapa [37]. No domínio da frequência, é também fácil eliminar-se o componente de rotacional do gradiente, bem como aplicar diversos filtros de suavização [105].

Usando algoritmos para transformada rápida de Fourier (FFT ou DCT), estes métodos de integração calculam o mapa de alturas para N pixels utilizando apenas espaço $O(N)$ e $O(N \log N)$ operações aritméticas. Entretanto, esta abordagem não permite o uso de mapas de peso, uma vez que a FFT sempre atribui o mesmo peso a todas as amostras do mapa de gradiente. Em consequência, estes métodos achatam qualquer degrau que for invisível no mapa de gradientes, deformando as regiões adjacentes. Veja figura 7.1(d).

Além disso, a integração por Fourier supõe que a função Z é periódica, implicando que a integral ao longo de cada linha de F e cada coluna de G deve ser zero. É possível impor esta condição espelhando e invertendo os mapas de gradiente, o que quadruplica a área total a ser integrada. Note que implementações populares do algoritmo de Frankot-Chellapa ignoram este fato, e portanto falham se os valores de Z ao longo de cada lado do retângulo D diferem dos valores no lado oposto.

7.2.3 Métodos de Poisson com resolução direta

O problema de integração de gradiente pode ser reduzido para um sistema de N equações lineares onde as incógnitas são as N alturas, e cada equação relaciona um valor de altura com os seus vizinhos utilizando as derivadas nas suas vizinhanças (veja seção 7.3). Este sistema pode ser resolvido por um método direto (não iterativo), como Fatoração LU ou de Cholesky.

Caso as equações sejam não-lineares, elas devem ser linearizadas e o processo deve ser iterado; como é feito, por exemplo, no método de Newton-Raphson. Estas variantes normalmente possuem convergência de segunda ordem, exigindo poucas iterações. Esta

abordagem, em particular, é utilizada por vários métodos descritos por Agrawal [6, 82] baseados nas equações de Poisson, incluindo Estimadores M, Minimização de Energia, Superfície- α e Transformadas Afins.

Como veremos na seção 7.3, as equações locais a serem resolvidas podem ser determinadas de diversas maneiras, incluindo minimização de energia [6], a solução via mínimos quadrados de um sistema sobredeterminado [57], ou por média das estimativas das alturas obtidas pelas alturas adjacentes e gradientes [90]. Entretanto, todos estes critérios locais produzem alguma versão discreta (e possivelmente não-linear) da equação de Poisson $\nabla^2 Z(x, y) = H(x, y)$. Isto é, cada equação determina a identidade entre duas estimativas numéricas do laplaciano do mapa de alturas: uma ($\nabla^2 Z$) calculada das alturas desconhecidas, e outra (H) calculada a partir do mapa de gradientes dado. Este último tipicamente é uma estimativa numérica do divergente $\nabla \cdot (F, G)$ do vetor gradiente.

Uma vez que cada equação envolve apenas um número limitado de incógnitas, o sistema todo requer apenas espaço $O(N)$. Ao contrário dos métodos de integração por caminhos, a abordagem pela equação de Poisson utiliza quase toda informação presente no mapa de gradientes, descartando automaticamente o componente rotacional; e não gera degraus espúrios na presença de ruído.

Se todos os pesos forem iguais, a solução deste sistema é matematicamente idêntica à obtida por filtragem de Fourier. Neste caso, a transformada rápida de Fourier pode ser utilizada para resolver eficientemente a equação de Poisson, como demonstrado por Georgiades *et al.* [39]. Porém a principal vantagem desta abordagem, enfatizada por Agrawal *et al.* em 2006 [6], é que cada equação do sistema linear pode ser individualmente ajustada para ignorar dados inconfiáveis, conforme especificado pelo mapa de pesos. Neste caso a transformada rápida não pode ser usada; a resolução do sistema de Poisson fica mais lenta do que a filtragem por Fourier, e utiliza substancialmente mais memória. Embora a matriz do sistema seja bem esparsa, seus fatores de Gauss ou Cholesky são substancialmente densos, de maneira que o espaço utilizado cresce mais do que linearmente em função do número N de pixels. De acordo com nossos testes, o método de Estimadores-M de Agrawal, por exemplo, precisa de $5N$ elementos não-nulos para a matriz do sistema, mas em torno de $5N^{1.15}$ em seu fator de Cholesky, mesmo com ordenação ótima de colunas e linhas e algoritmos eficientes para matrizes esparsas. Para $N = 1024 \times 1024 = 2^{20}$, isto significa em torno de $5 \times 2^{23} \approx 8\,000\,000$ elementos não-nulos, ou seja, 128 MB de memória. O número de operações cresce proporcionalmente a $N^{1.5}$.

7.2.4 Métodos iterativos locais

O sistema de equações de Poisson também pode ser resolvido iterativamente, como no algoritmo de Gauss-Seidel [31]. Isto é, partindo-se de uma estimativa inicial das alturas,

cada equação do sistema é usada para recalcular o valor de uma altura, supondo que as alturas dos vizinhos são fixas— até todos os valores das alturas estabilizarem. Esta abordagem é, aliás, a mais antiga, pois foi inicialmente descrita por Horn e Brooks em 1989 [79, 57].

Métodos iterativos locais podem utilizar as mesmas equações de Poisson usadas pelos métodos de resolução direta, incluindo fórmulas ajustadas localmente que levam mapas de peso em consideração. Adicionalmente, eles podem lidar com equações moderadamente não-lineares sem a necessidade da linearização (como a do método de Newton-Raphson) ou de loops externos adicionais.

A vantagem dos métodos iterativos sobre os de solução direta é que eles exigem menos espaço; seu consumo de memória cresce proporcionalmente a N em vez de $N^{1.15}$. Sua principal desvantagem é o tempo excessivo de processamento necessário. Embora cada iteração exija apenas $O(N)$ operações, o número de iterações necessário para reduzir o erro inicial E abaixo de uma tolerância ε é normalmente proporcional a $\log(E/\varepsilon)$ vezes o quadrado do diâmetro da imagem, ou seja a $N \log(E/\varepsilon)$; o que dá um tempo total de processamento proporcional a $N^2 \log(E/\varepsilon)$.

Esta ineficiência pode ser explicada levando-se em conta o efeito da iteração de Gauss-Seidel em cada componente de Fourier ϕ do mapa de erro δ corrente (a diferença entre o mapa de altura Z atual e o correto). Cada iteração essencialmente reduz a amplitude de ϕ por um fator $\alpha = \cos(2\pi/\omega)$, onde ω é o comprimento de onda de ϕ . Em uma malha quadrada com N pixels, o componente de menor frequência tem $\omega = \sqrt{2N}$, de maneira que $\alpha = \cos(2\pi/\sqrt{2N}) \approx 1 - \pi^2/N$. Portanto, o número de iterações necessário para reduzir sua amplitude inicial de E para ε é proporcional a $\log(E/\varepsilon)/\log(1 - \pi^2/N) \approx N/\pi^2 \log(E/\varepsilon)$. Por outro lado, qualquer componente do mapa de erro δ cujo comprimento de onda é de apenas alguns pixels será reduzido à insignificância em $O(1)$ iterações. Na verdade, a correção da estimativa inicial propaga-se através do mapa de alturas por um processo de onda de difusão, matematicamente similar ao espalhamento de calor em uma placa sólida.

Uma maneira de acelerar a convergência destes métodos é utilizando técnicas *multi-escala*, como sugerido por Terzopoulos [98, 97] em 1986. Esta é a abordagem que descreveremos na seção 7.5.

7.2.5 Outros métodos

Um método introduzido por Ng *et al.* [76] é especializado para mapas de gradientes esparsos com grandes regiões de dados inválidos. Ele reduz o problema de integração para um problema de ajuste de função utilizando determinadas bases de funções. Esta abordagem pode lidar com a distribuição irregular de amostras e segundo o autor, preenche as regiões

onde dados são ausentes de maneira mais “natural” do que métodos baseados nas equações de Poisson. Entretanto, este método requer a resolução de sistemas de equações lineares consideravelmente maiores ($3N \times 3N$) que o sistema de Poisson, sendo portanto muito mais custoso em termos de tempo e espaço.

Chan, Wang e Wang [28] propuseram um método que combina integração por caminhos (sem mapas de peso) com a abordagem multi-escala. Seu algoritmo resolve uma sequência de $N \times N$ sistemas de equações de Poisson. O sistema do nível k relaciona cada altura $z[u, v]$ com alturas $z[u \pm 2^k, v \pm 2^k]$. Integrais de linha (de alto custo computacional) são utilizadas para calcular o lado direito de cada equação. Este método, no entanto, não é verdadeiramente multi-escala, uma vez que o problema não é reduzido e expandido em escalas sucessivas. O uso de passos longos melhora substancialmente o tempo de convergência, mas a velocidade e precisão apresentados pelos autores é ainda assim inferior aos dos métodos de filtragem por Fourier.

7.3 Equações ponderadas de Poisson

Nosso integrador usa uma variante ponderada das equações discretas de Poisson, e as resolve por uma versão multi-escala do método iterativo de Gauss-Seidel (ou Gauss-Jacobi). Nesta seção, nós primeiro formulamos a versão contínua do método de Poisson para gradientes exatos; em seguida descreveremos uma versão discreta que leva os pesos em conta.

7.3.1 Forma contínua das equações de Poisson

Conceitualmente, na abordagem de Poisson, nós derivamos ambos lados das equações (7.1) e as adicionamos para obter uma única equação funcional

$$\mathbf{L}(Z)(x, y) = \mathbf{H}(F, G)(x, y) \quad (7.5)$$

onde

$$\mathbf{L}(Z) = \frac{\partial^2 Z}{\partial x^2} + \frac{\partial^2 Z}{\partial y^2} \quad \mathbf{H}(F, G) = \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} \quad (7.6)$$

A equação (7.5) afirma que o laplaciano $\mathbf{L}(Z)$ das alturas Z calculado a partir da própria função Z deve ser igual ao laplaciano $\mathbf{H}(F, G)$ calculado a partir dos gradientes dados F, G [97, 79].

7.3.2 Discretizando as alturas

Para discretizar a equação (7.5), a função altura Z é representada por um *mapa de alturas*, uma matriz de amostras de altura $z[u, v]$ posicionadas em uma grade de *pontos de amostragem de alturas* $q[u, v]$. Os operadores diferenciais na equação (7.5) são substituídos por operadores de diferenças finitas aplicados nas matrizes z , f , e g . A equação diferencial (7.5) torna-se então um sistema de equações lineares, cujas incógnitas são os elementos da matriz z .

Para obter estimativas comparáveis das derivadas em ambos lados da equação (7.5), nós supomos que os pontos de amostragem das alturas $q[u, v]$ estão deslocados em relação aos pontos de amostragem dos gradientes $p[u, v]$ por meio pixel em cada direção. Especificamente, nós supomos que $p[u, v]$ é o ponto $(u + 1/2, v + 1/2)$ de \mathbb{R}^2 , enquanto $q[u, v]$ é o ponto (u, v) . Esta convenção é ilustrada na figura 7.3.

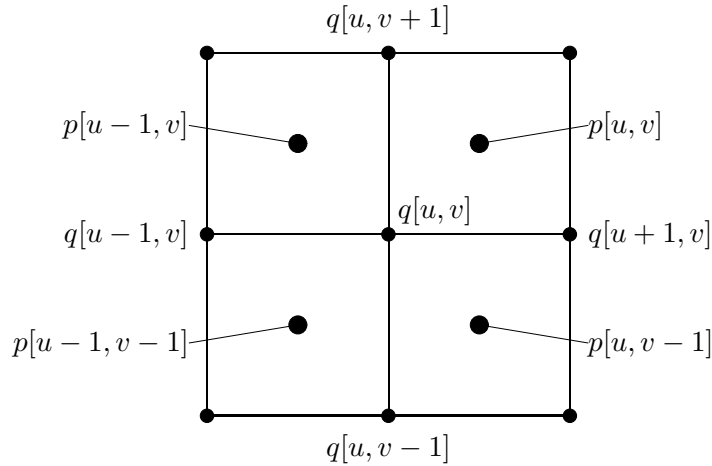


Figura 7.3: Pontos de amostragem de gradientes e alturas adjacentes ao ponto $q[u, v] = (u, v)$.

Portanto, se os mapas de derivadas f, g possuem n_x colunas e n_y linhas, podemos supor que o domínio D do problema é o retângulo $[0, n_x] \times [0, n_y]$ de \mathbb{R}^2 . Cada pixel $[u, v]$ dos mapas de derivadas pode ser identificado com o quadrado de tamanho unitário centrado em $p[u, v]$, com $q[u, v]$ e $q[u + 1, v + 1]$ em cantos opostos. Note que o mapa de altura z , por outro lado, possui $n_x + 1$ colunas e $n_y + 1$ linhas, e podemos supor que cada pixel $z[u, v]$ é um quadrado unitário centrado no ponto $q[u, v]$ que é um vértice dos mapas de gradiente.

Na prática, os mapas de derivadas $f[u, v]$ e $g[u, v]$ são quase sempre as médias das derivadas $\partial Z / \partial x$ e $\partial Z / \partial y$ na vizinhança do ponto $p[u, v]$, calculadas por um *kernel de amostragem de gradiente*, que deve ser simétrico em relação a $p[u, v]$ e sobrepor parcialmente os kernels adjacentes. De maneira similar, a altura calculada $z[u, v]$ será uma

estimativa para a média da altura Z em torno do ponto $q[u, v]$, obtida por outro *kernel de amostragem de altura*. A relação entre estes dois *kernels* está fora do escopo desta tese.

Nós supomos que o mapa de pesos é dado na forma de uma matriz w de valores não-negativos, com mesmas dimensões que f e g e, como estes, referentes aos pontos $p[u, v]$. Convencionamos também que $w[u, v]$ é zero sempre que o ponto $p[u, v]$ está fora do domínio D . Como veremos mais adiante, apenas as magnitudes *relativas* dos pesos são significativas, isto é, os resultados não serão afetados se multiplicarmos todos os pesos por um fator de escala positivo.

Para tornar as formulas mais legíveis, usaremos a notação $z_{\circ\circ}$ para uma amostra qualquer de altura $z[u, v]$, e as seguintes notações para os seus quatro vizinhos:

$$\begin{aligned} z_{-\circ} &= z[u-1, v] & z_{\circ-} &= z[u, v-1] \\ z_{+\circ} &= z[u+1, v] & z_{\circ+} &= z[u, v+1] \end{aligned}$$

Além disso, nós usaremos os seguintes símbolos para os valores das derivadas interpoladas pelo nosso algoritmo, a partir dos mapas f e g , nos pontos médios das arestas entre $q[u, v]$ e seus quatro vizinhos:

$$\begin{aligned} f_{-\circ} &\approx \frac{\partial Z}{\partial x}(u - \frac{1}{2}, v) & g_{\circ-} &\approx \frac{\partial Z}{\partial y}(u, v - \frac{1}{2}) \\ f_{+\circ} &\approx \frac{\partial Z}{\partial x}(u + \frac{1}{2}, v) & g_{\circ+} &\approx \frac{\partial Z}{\partial y}(u, v + \frac{1}{2}) \end{aligned}$$

Nós também utilizaremos $w_{-\circ}$, $w_{+\circ}$, $w_{\circ-}$, e $w_{\circ+}$ para denotar *pesos de confiabilidade de arestas*, atribuídos pelo nosso algoritmo para as inclinações interpoladas $f_{-\circ}$, $f_{+\circ}$, $g_{\circ-}$ e $g_{\circ+}$, respectivamente. Veja a figura 7.4. As fórmulas de interpolação para estas derivadas e pesos são descritas na seção 7.4.

7.3.3 Equações discretas para pesos uniformes

Nas suas versões discretas, os operadores \mathbf{L} e \mathbf{H} das equações 7.5 são substituídos por estimadores de diferenças finitas \mathcal{L} e \mathcal{H} (denominados frequentemente de “kernels de Poisson” na literatura).

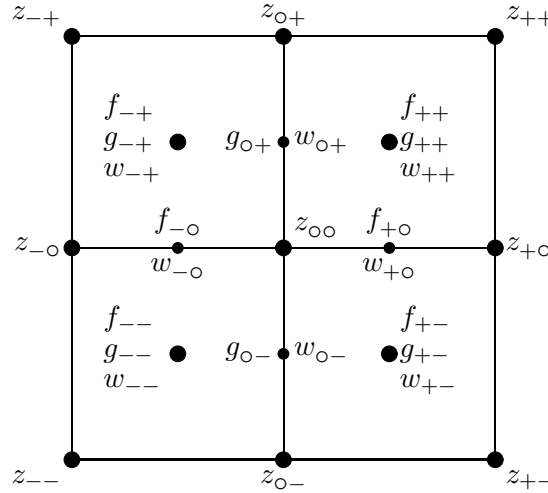


Figura 7.4: Notações para valores de altura, gradiente e peso interpolados em torno do ponto $q[u, v]$.

Se todos os pesos são 1, podemos usar os operadores discretos $\tilde{\mathcal{L}}(z)$ e $\tilde{\mathcal{H}}(f, g)$ para estimar $\mathbf{L}(Z)$ e $\mathbf{H}(F, G)$, onde

$$\begin{aligned} \tilde{\mathcal{L}}(z)[u, v] = & + (z_{o+} - z_{oo}) - (z_{oo} - z_{o-}) \\ & + (z_{+o} - z_{oo}) - (z_{oo} - z_{-o}) \end{aligned} \quad (7.7)$$

$$\tilde{\mathcal{H}}(f, g)[u, v] = f_{+o} - f_{-o} + g_{o+} - g_{o-} \quad (7.8)$$

A versão discreta da equação (7.5) é então o conjunto de equações lineares

$$\tilde{\mathcal{L}}(z)[u, v] = \tilde{\mathcal{H}}(f, g)[u, v] \quad (7.9)$$

para todo $[u, v]$ no domínio de z .

7.3.4 Equações axiais para pesos gerais

Note que o primeiro termo $(z_{o+} - z_{oo})$ na fórmula (7.7) é outra estimativa para a derivada $\partial Z / \partial x$ no ponto médio $(u + \frac{1}{2}, v)$ da aresta horizontal da grade entre $q[u, v]$ e $q[u + 1, v]$, calculada a partir do mapa de alturas z . Esta pode ser comparada com o primeiro termo f_{+o} da equação (7.8) calculada por interpolação do mapa f dado. Similarmente, os outros termos da fórmula (7.7) são as estimativas das derivadas numéricas das alturas, que podem ser comparados com os termos f_{-o} , g_{o-} , e g_{o+} da fórmula (7.8). Inspirados nesta observação, nós dividimos cada equação (7.9) em quatro equações separadas:

$$\begin{aligned}
z_{+o} - z_{oo} &= +f_{+o} & z_{o+} - z_{oo} &= +g_{o+} \\
z_{-o} - z_{oo} &= -f_{-o} & z_{o-} - z_{oo} &= -g_{o-}
\end{aligned} \tag{7.10}$$

Geralmente, o sistema (7.10) obtido desta forma é sobredeterminado, uma vez que possui aproximadamente $2N$ equações (após eliminar equações repetidas) com aproximadamente N incógnitas. Nós então procuramos uma solução aproximada desse sistema pelo critério dos mínimos quadrados; isto é, um mapa z que minimize a soma ponderada dos quadrados das diferenças entre os dois lados da equação. Nesta soma nós utilizamos para cada equação o peso (w_{-o} , w_{+o} , w_{o-}) ou w_{o+} do dado interpolado da aresta. Desta forma, a solução de mínimos quadrados das equações (7.10) reduz-se a solução de outro sistema de N equações lineares com N incógnitas. Cada equação deste sistema afirma que cada valor de altura $z[u, v]$ é igual a média ponderada das alturas de seus quatro vizinhos, acrescidos das derivadas da altura interpoladas no meio das arestas correspondentes:

$$\begin{aligned}
z_{oo} = & + \frac{w_{-o}}{W_{oo}}(z_{-o} + f_{-o}) + \frac{w_{+o}}{W_{oo}}(z_{+o} - f_{+o}) \\
& + \frac{w_{o-}}{W_{oo}}(z_{o-} + g_{o-}) + \frac{w_{o+}}{W_{oo}}(z_{o+} - g_{o+})
\end{aligned} \tag{7.11}$$

Nesta fórmula, W_{oo} é o *peso total do vértice*,

$$W_{oo} = W[u, v] = w_{o-} + w_{o+} + w_{-o} + w_{+o} \tag{7.12}$$

Note que apenas os valores relativos dos pesos são significativos. Rearranjando a equação (7.11) para separar termos conhecidos e desconhecidos, obtemos

$$-\mathcal{L}(z)[u, v] = -\mathcal{H}(f, g)[u, v] \tag{7.13}$$

onde

$$\begin{aligned}
-\mathcal{L}(z)[u, v] = & + z_{oo} - \frac{w_{o-}}{W_{oo}}z_{o-} - \frac{w_{o+}}{W_{oo}}z_{o+} \\
& - \frac{w_{-o}}{W_{oo}}z_{-o} - \frac{w_{+o}}{W_{oo}}z_{+o}
\end{aligned} \tag{7.14}$$

e

$$\begin{aligned}
-\mathcal{H}(f, g)[u, v] = & - \frac{w_{o-}}{W_{oo}}g_{o-} + \frac{w_{o+}}{w_{oo}}g_{o+} \\
& - \frac{w_{-o}}{W_{oo}}f_{-o} + \frac{w_{+o}}{w_{oo}}f_{+o}
\end{aligned} \tag{7.15}$$

Note que quando uma aresta tem peso zero, o termo correspondente é eliminado das equações (7.11), (7.14), e (7.15). Ao longo da margem inferior ($v = 0$), por exemplo, a

equação (7.13) irá relacionar $z[u, v]$ com os seus três vizinhos z_{o+} , z_{+o} e z_{-o} , ignorando o vizinho não existente z_{o-} . Denominamos a fórmula (7.13) de *equação axial de Poisson* para $z[u, v]$.

O peso total $W[u, v]$ pode ser interpretado como a confiabilidade do valor calculado de $z[u, v]$, supondo que os valores adjacentes estejam corretos. Valores de $z[u, v]$ que possuem $W[u, v] = 0$ são portanto espúrios e devem ser desconsiderados em cálculos posteriores.

7.3.5 Equações diagonais para pesos gerais

Como veremos na seção (7.4), quando os pesos dados $w_{++}, w_{+-}, w_{-+}, w_{--}$ incluem dois elementos diagonalmente opostos com peso zero, todo os quatro pesos $w_{o+}, w_{o-}, w_{+o}, w_{-o}$ definidos serão zero. Neste caso, substituímos a equação axial (7.13) por uma *equação diagonal de Poisson*, que relaciona z_{oo} a os seus quatro vizinhos diagonais,

$$\begin{aligned} z_{--} &= z[u-1, v-1] & z_{+-} &= z[u+1, v-1] \\ z_{-+} &= z[u-1, v+1] & z_{++} &= z[u+1, v+1] \end{aligned}$$

Especificamente, nós usamos a equação (7.13) com os operadores \mathcal{L}, \mathcal{H} redefinidos como se segue:

$$\begin{aligned} -\mathcal{L}(z)[u, v] &= z_{oo} - \frac{w_{--}}{W_{oo}} z_{--} - \frac{w_{-+}}{W_{oo}} z_{-+} \\ &\quad - \frac{w_{+-}}{W_{oo}} z_{+-} - \frac{w_{++}}{W_{oo}} z_{++} \end{aligned} \tag{7.16}$$

e

$$\begin{aligned} -\mathcal{H}(f, g)[u, v] &= \\ &+ \frac{w_{--}}{W_{oo}} (+f_{--} + g_{--}) + \frac{w_{-+}}{W_{oo}} (+f_{-+} - g_{-+}) \\ &+ \frac{w_{+-}}{W_{oo}} (-f_{+-} + g_{-+}) + \frac{w_{++}}{W_{oo}} (-f_{++} - g_{++}) \end{aligned} \tag{7.17}$$

e com o peso total do vértice W_{oo} redefinido por

$$W_{oo} = W[u, v] = w_{--} + w_{-+} + w_{+-} + w_{++} \tag{7.18}$$

7.3.6 Alturas desconexas

A fórmula (7.18) também pode ter resultado zero. Especialmente, isso acontece se e somente se $w_{++} = w_{-+} = w_{+-} = w_{--} = 0$ — ou seja, quando os quatro valores dados de gradiente em torno do ponto $q[u, v]$ tem confiabilidade nula. Neste caso, nós dizemos que a altura $z[u, v]$ é *desconexa*. Nós tratamos este caso especial atribuindo valor zero a $z[u, v]$

e removendo sua equação correspondente do sistema. Note que os valores de elementos desconexos $z[u, v]$ não afetam nenhum elemento vizinho.

7.3.7 Montando o sistema

Após substituir equações axiais de peso nulo por equações diagonais, e excluir as alturas desconexas, existe uma correspondência um-para-um entre as equações discretas de Poisson (7.13) ou (7.18) e as demais alturas $z[u, v]$; de modo que o número de equações é igual ao número de incógnitas. Nós juntamos estas equações em um sistema linear

$$\mathbf{M}\mathbf{z} = \mathbf{b} \quad (7.19)$$

onde \mathbf{z} é o mapa de alturas a determinar armazenado em um vetor com $N \leq (n_x+1)(n_y+1)$ elementos; \mathbf{M} é a matriz $N \times N$ de coeficientes do lado esquerdo da equação (7.14); e \mathbf{b} é o vetor de comprimento N contendo o lado direito de (7.15).

Note que \mathbf{M} depende dos pesos $w[u, v]$ mas não dos valores dos gradientes f e g ; enquanto que \mathbf{b} depende de w , f e g . Observe também que qualquer mapa de alturas constante ($z[u, v] = C$ para todo u, v e algum C) satisfaz o sistema homogêneo $\mathbf{M}\mathbf{z} = 0$. Consequentemente, a solução do sistema (7.19) tem um grau de indeterminação, que é por uma constante aditiva arbitrária.

Após a eliminação de alturas desconexas, pode ocorrer que o mapa de alturas fique particionado em duas ou mais regiões, tais que nenhuma equação relaciona valores de uma região aos de qualquer outra. Neste caso, o sistema (7.19) divide-se em dois ou mais sistemas independentes cada qual com sua constante aditiva arbitrária — isto é, com deslocamentos indeterminados entre as alturas dessas regiões.

7.4 Estimando as derivadas de aresta

Para calcular o lado direito da equação (7.15) nós precisamos de estimativas para as derivadas no ponto médio de cada aresta da grade. Por exemplo, precisamos da estimativa g_{o+} para a derivada $\partial Z / \partial y$ no ponto médio da aresta $r = (u, v + \frac{1}{2})$ entre $q[u, v]$ e $q[u, v+1]$. Para obter esta estimativa, nós usamos os quatro valores $g_a = g[u-2, v]$, $g_b = g[u-1, v]$, $g_c = g[u, v]$ e $g_d = g[u+1, v]$, que são as derivadas amostradas em torno dos pontos $(u - \frac{3}{2}, v + \frac{1}{2})$, $(u - \frac{1}{2}, v + \frac{1}{2})$, $(u + \frac{1}{2}, v + \frac{1}{2})$ e $(u + \frac{3}{2}, v + \frac{1}{2})$, respectivamente. Note que as distâncias (com sinal) destes pontos para r são $-\frac{3}{2}$, $-\frac{1}{2}$, $+\frac{1}{2}$ e $+\frac{3}{2}$ respectivamente. Denotaremos por w_a , w_b , w_c e w_d os pesos de confiabilidade destes valores. Veja a figura 7.5.

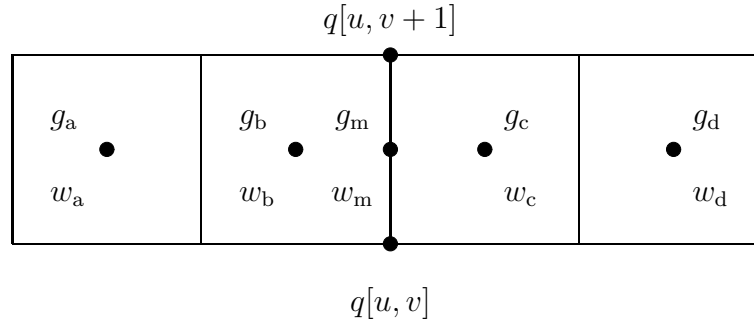


Figura 7.5: Interpolando a derivada $\partial Z/\partial y$ no ponto $(u, v + \frac{1}{2})$.

Considerando pares consecutivos destes quatro valores, por interpolação linear ou extra-polação, nós obtemos três estimativas para a derivada $\partial Z/\partial y$ em r :

$$\begin{aligned} g_- &= (3g_b - g_a)/2 \\ g_o &= (g_b + g_c)/2 \\ g_+ &= (3g_c - g_d)/2 \end{aligned} \quad (7.20)$$

Dada a interpretação de $w[u, v]$ como a recíproca da variância de $g[u, v]$ (veja seção 7.1.2), os pesos destas estimativas serão

$$\begin{aligned} w_- &= 4/(9/w_b + 1/w_a) \\ w_o &= 4/(1/w_b + 1/w_c) \\ w_+ &= 4/(9/w_c + 1/w_d) \end{aligned} \quad (7.21)$$

Nós então tomamos a média ponderada g_m destas três estimativas, e atribuímos a ela o peso w_m apropriado:

$$g_m = \frac{w_- g_- + w_o g_o + w_+ g_+}{w_- + w_o + w_+} \quad (7.22)$$

$$w_m = w_- + w_o + w_+$$

Como dito antes, todas as amostras que se situam fora do domínio de g recebem peso zero, de maneira que seu valor será ignorado neste cálculo. Note que quaisquer das estimativas g_- , g_o ou g_+ que dependa de uma amostra de peso zero terá peso zero também e portanto não contribuirá para o resultado final.

Não seria seguro ampliar a lista (7.20) com interpolações de dados não consecutivos, como $(g_a + 3g_c)/4$, porque se as amostras intermediárias (g_b no caso) tiverem peso baixo ou zero, pode haver um degrau de altura desconhecida neste pixel e nesse caso o valor calculado seria inválido, apesar de usar apenas dados válidos.

Caso todos os pesos w_- , w_0 , e w_+ sejam zero, o peso final w_m será zero pela fórmula (7.22). Neste caso, nós definimos arbitrariamente $g_m = g_0$.

Nós denotaremos o cálculo (7.20)–(7.22) por

$$(g_m, w_m) \leftarrow \text{INTERPOLATE}(g_a, w_a, g_b, w_b, g_c, w_c, g_d, w_d) \quad (7.23)$$

Na notação da fórmula (7.15), w_m é w_{0+} e g_m é g_{0+} .

Para obter f_{+0} , o valor estimado de $\partial Z / \partial x$ no ponto médio $s = (u + \frac{1}{2}, v)$ de uma aresta horizontal, nós utilizamos essa mesma função INTERPOLATE nas amostras verticalmente adjacentes à aresta, isto é $f_a = f[u, v - 2]$, $f_b = f[u, v - 1]$, $f_c = f[u, v]$ e $f_d = f[u, v + 1]$, com seus respectivos pesos. O mesmo algoritmo é utilizado para estimar g_{0-} e f_{-0} .

7.5 Solução multi-escala

Como observamos na seção 7.2, em aplicações típicas o sistema de Poisson (7.19) é grande demais para ser resolvido eficientemente por métodos de solução direta. Para uma imagem 1024×1024 , por exemplo, ele tem aproximadamente um milhão de equações e incógnitas. Portanto, nós utilizamos o método iterativo de Gauss-Seidel.

Embora a matriz \mathbf{M} seja grande, ela é muito esparsa, com no máximo 5 termos não nulos em cada linha. Portanto, ela requer apenas $O(N)$ espaço na memória, e o produto $\mathbf{M}\mathbf{z}$ pode ser calculado em tempo $O(N)$. Ao construir o sistema (7.19), utilizamos a mesma ordenação para as alturas em \mathbf{z} e para as linhas de \mathbf{M} , de maneira que as diagonais de \mathbf{M} sempre têm valor 1, enquanto os elementos não-diagonais são sempre negativos com valor total -1 . Estas características garantem a convergência do método iterativo.

Por outro lado, como discutido na seção 7.2, a convergência do método iterativo é muito lenta. Para acelerá-la nós usamos a abordagem multi-escala proposta por Terzopoulos [98]. Isto é, para obter uma estimativa inicial da solução, nós reduzimos o mapa de gradientes f, g pela metade de sua largura e altura original, calculamos a partir do dado em escala reduzida um mapa de alturas z , e expandimos o mesmo para a escala anterior. O problema reduzido é resolvido recursivamente, de maneira similar.

Em outras palavras, nós construímos uma pirâmide de mapas de derivadas horizontais $f^{(0)}, f^{(1)}, \dots, f^{(m)}$, onde $f^{(0)} = f$ e cada mapa $f^{(k+1)}$ é uma cópia reduzida do mapa anterior $f^{(k)}$. Da mesma forma construímos uma pirâmide $g^{(k)}$ de derivadas verticais obtidas a partir de g . Interrompemos a redução em um nível m tal que $f^{(m)}$ e $g^{(m)}$ são pequenos o suficiente para serem integrados eficientemente (quer, pelo método de Gauss-Seidel, quer por eliminação ou fatoração), resultando em um mapa de alturas $z^{(m)}$ de mesma escala. Nós integramos então cada par de mapas $f^{(k)}$ e $g^{(k)}$, em ordem decrescente de k , com um método iterativo utilizando a solução $z^{(k+1)}$ para produzir a estimativa

inicial, e obtendo uma solução mais detalhada $z^{(k)}$. O resultado final do processamento é o mapa $z^{(0)}$. Veja a figura 7.6.

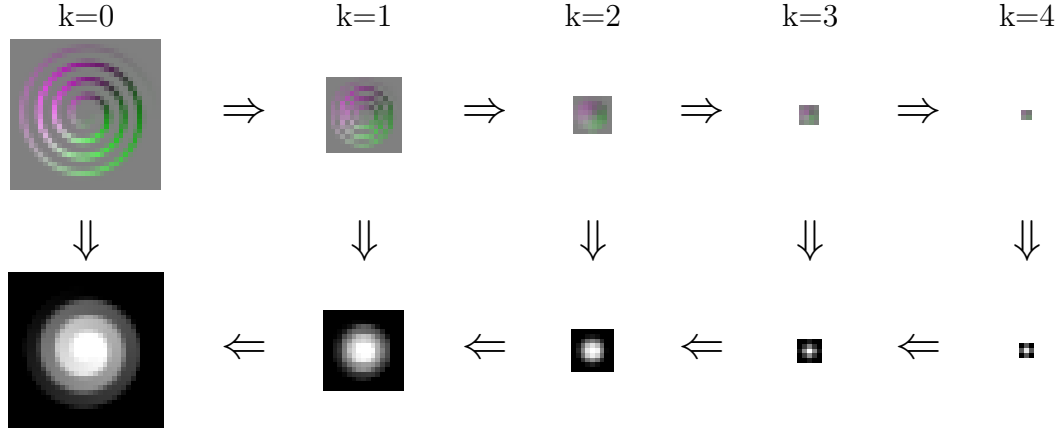


Figura 7.6: Diagrama da integração multi-escala com $m = 4$.

Com passos de redução e expansão adequados, a estimativa inicial calculada recursivamente será próxima da solução correta, exceto por regiões com detalhes de alta frequência. O método iterativo deve então convergir rapidamente, uma vez que ele necessita apenas ajustar detalhes na escala de um ou dois pixels.

Terzopoulos [98] usou subamostragem simples para redução, e interpolação bilinear para expansão. Embora este método seja adequado para superfícies contínuas e suaves, ele não dá bons resultados na presença de variações bruscas de derivadas. Para fazer funcionar a abordagem multi-escala nestes casos, nós utilizamos fórmulas mais elaboradas de expansão e redução dos mapas de derivadas, que levam o mapa de pesos em conta, descritas nas seções 7.5.2 e 7.5.3.

7.5.1 O algoritmo

A parte central do nosso algoritmo é o procedimento recursivo `MSMAPINTEGRATE`, cujo pseudo-código é dado na figura 7.7. Como discutido na seção 7.3.2, os mapas f, g, w devem possuir as mesmas dimensões, isto é n_x colunas por n_y linhas, enquanto o mapa de alturas calculado z terá $n_x + 1$ colunas e $n_y + 1$ linhas. O parâmetro κ de `MSMAPINTEGRATE` é o número máximo de iterações de Gauss-Seidel permitidas, e ϵ é um critério numérico de convergência, ambos válidos para escala 0 da pirâmide. As constantes λ e n_{small} são parâmetros internos.

Procedimento **MSMAPINTEGRATE**($f, g, w, \kappa, \epsilon$)

1. Se $f.nx < n_{\text{small}}$ e $f.ny < n_{\text{small}}$ então
2. $\mathbf{z} \leftarrow (0, 0, \dots, 0)$;
3. senão
4. $f' \leftarrow \text{REDUCEGRADIENTMAP}(f, w)$;
5. $g' \leftarrow \text{REDUCEGRADIENTMAP}(g, w)$;
6. $w' \leftarrow \text{REDUCEWEIGHTMAP}(w)$;
7. $\mathbf{z}' \leftarrow \text{MSMAPINTEGRATE}(f', g', w', \lambda\kappa, \epsilon/\lambda)$;
8. $\mathbf{z} \leftarrow \text{EXPANDHEIGHTMAP}(\mathbf{z}')$;
9. $\mathbf{M}, \mathbf{b} \leftarrow \text{BUILDSYSTEM}(f, g, w)$;
10. $\mathbf{z} \leftarrow \text{SOLVESYSTEM}(\mathbf{M}, \mathbf{b}, \mathbf{z}, \kappa, \epsilon)$;
11. Retorna \mathbf{z} .

Figura 7.7: Procedimento principal do integrador.

O procedimento **BUILDSYSTEM** constrói a matriz do sistema \mathbf{M} e o vetor do lado direito \mathbf{b} como descrito na seção 7.3. A chamada do procedimento **SOLVESYSTEM**($\mathbf{M}, \mathbf{b}, \mathbf{z}, \epsilon$) é o algoritmo de Gauss-Seidel, que calcula

$$\mathbf{z}_i \leftarrow \mathbf{b}_i - \sum_{j \neq i} \mathbf{M}_{ij} \mathbf{z}_j \quad \text{para } i = 0, \dots, N \quad (7.24)$$

até no máximo κ vezes, parando quando todas as alturas \mathbf{z}_i tem diferença menor ou igual a ϵ com seus valores na iteração anterior. Os procedimentos **REDUCEGRADIENTMAP**, **REDUCEWEIGHTMAP** e **EXPANDHEIGHTMAP** são explicados nas seções 7.5.2 e 7.5.3 abaixo.

7.5.2 Reduzindo os mapas de gradiente e pesos

A cada nível de recursão, os procedimentos *ReduceGradientMap* e *ReduceWeightMap* são utilizados para calcular os mapas em escala reduzida f' , g' e w' . Observe que o operador de integração é linear quando as entradas do mesmo são as derivadas f, g , mas não é linear quando as entradas são as normais de superfície \vec{n} . Portanto, ao reduzir o problema para uma escala menor, é importante trabalhar com os valores dos gradientes, em vez das normais.

Por hipótese, cada amostra $f'[u, v]$ do mapa reduzido de derivada horizontal deve ser o valor médio de $\partial Z / \partial x$ sobre o pixel do mapa reduzido f' com índices $[u, v]$; que corresponde a um bloco de 2×2 pixels do mapa f original, centrado no ponto $(2u + 1, 2v + 1)$. Seria natural portanto definir $f'[u, v]$ como a média *não ponderada* das amostras originais $f_{--} = f[2u, 2v]$, $f_{+-} = f[2u + 1, 2v]$, $f_{-+} = f[2u, 2v + 1]$ e $f_{++} = f[2u + 1, 2v + 1]$.

A variância do erro de medida dessa média é $1/4$ da média da variância dos erros nas quatro parcelas. Portanto, nesse caso o peso $w'[u, v]$ da amostra reduzida deverá ser

$$w'[u, v] = \frac{16}{\frac{1}{w_{++}} + \frac{1}{w_{+-}} + \frac{1}{w_{-+}} + \frac{1}{w_{--}}} \quad (7.25)$$

onde $w_{--} = w[2u, 2v]$, $w_{+-} = w[2u+1, 2v]$, $w_{-+} = w[2u, 2v+1]$ e $w_{++} = w[2u+1, 2v+1]$.

Entretanto, a fórmula (7.25) atribui valor nulo (ou muito baixo) ao peso $w'[u, v]$ se qualquer um dos pesos $w_{--}, w_{-+}, w_{+-}, w_{++}$ for nulo (ou muito baixo). Esta característica faz com que lacunas isoladas no mapa f se perpetuem no mapa f' com perda da informações conhecidas nos outros três valores.

Em vista disso, nós optamos por uma fórmula mais robusta, baseada na observação de que a média de duas amostras diagonalmente opostas no bloco é uma estimativa aceitável para as derivadas no centro do bloco. Especificamente, nós calculamos as estimativas

$$f'_a = (f_{--} + f_{++})/2 \quad f'_b = (f_{-+} + f_{+-})/2 \quad (7.26)$$

com os respectivos pesos

$$w'_a = \frac{4}{\frac{1}{w_{--}} + \frac{1}{w_{++}}} \quad w'_b = \frac{4}{\frac{1}{w_{-+}} + \frac{1}{w_{+-}}} \quad (7.27)$$

Nós tomamos então a média ponderada das duas estimativas, e atribuímos a ela o peso adequado:

$$f'[u, v] = \frac{w'_a f'_a + w'_b f'_b}{w'_a + w'_b} \quad (7.28)$$

$$w'[u, v] = w'_a + w'_b \quad (7.29)$$

Nestas fórmulas, como sempre, amostras situadas fora do domínio do mapa f recebem um valor arbitrário com peso zero. Usamos as mesmas fórmulas para reduzir o mapa de derivadas verticais g . Note que, com as fórmulas (7.27) e (7.29), o peso $w'[u, v]$ será zero apenas se ambas as diagonais do bloco de 2×2 pixels incluírem no mínimo um amostra com peso zero.

7.5.3 Expandindo o mapa de alturas

O procedimento EXPANDHEIGHTMAP calcula uma estimativa z para o mapa de alturas em uma escala k , dado o mapa de alturas z' calculadas para a escala $k+1$. Sejam $z'_a = z'[u-1, v]$, $z'_b = z'[u, v]$, $z'_c = z'[u+1, v]$, $z'_d = z'[u+2, v]$, e sejam W'_a, W'_b, W'_c, W'_d

os respectivos pesos calculados como explicado nas seções 7.3.4 e 7.3.5. Nós utilizamos a função INTERPOLATE definida na seção 7.4 para estimar a altura z_x no ponto médio da aresta entre $q[u, v]$ e $q[u + 1, v]$ e seu peso estimado W_x

$$(z_x, W_x) = \text{INTERPOLATE}(z'_a, W'_a, z'_b, W'_b, z'_c, W'_c, z'_d, W'_d)$$

Veja a figura 7.8.

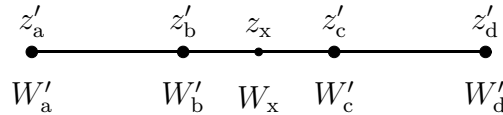


Figura 7.8: Interpolando a altura z_x no ponto médio de uma aresta da grade, e seu peso W_x .

Da mesma maneira nós calculamos uma estimativa z_y da altura no ponto médio da aresta entre $q[u, v]$ e $q[u, v + 1]$, e o respectivo peso W_y . Nós definimos então

$$\begin{aligned} z[2u, 2v] &= 2z'[u, v] \\ z[2u + 1, 2v] &= 2z_x \\ z[2u, 2v + 1] &= 2z_y \\ z[2u + 1, 2v + 1] &= 2 \frac{W_x z_x + W_y z_y}{W_x + W_y} \end{aligned} \tag{7.30}$$

para todo par u, v , sempre que o lado esquerdo de cada equação esteja definido. O fator de 2 nestas fórmulas (7.30) leva em conta o fato de que o espaçamento dos pixels dobra da escala $k + 1$ para k . Como caso especial, quando ambos W_x e W_y são zero, nós definimos a altura $z[2u + 1, 2v + 1]$ como sendo a soma $z_x + z_y$.

7.5.4 Ordenando as equações

No método de Gauss-Seidel, a ordem das variáveis tem um impacto substancial na velocidade da convergência. Idealmente, variáveis cujas variáveis vizinhas estão mais próximas dos valores corretos devem ser recalculadas primeiro, para que os cálculos seguintes possam usar valores já corrigidos.

Em geral, espera-se que pontos $q[u, v]$ que possuem peso total de vértice $W[u, v]$ maior sejam mais próximos do valor correto do que os de menor peso. Portanto, para ordenar as equações, nós criamos um grafo dirigido, cujos vértices são as incógnitas $z[u', v']$, com uma aresta partindo de $z[u, v]$ para $z[u', v']$ quando ambos ocorrem na mesma equação e o peso do vértice $W[u, v]$ é maior que $W[u', v']$. A ordem das variáveis e equações é então definida por ordenação topológica [30] dos vértices deste grafo.

7.5.5 Termo linear indeterminado

Ao contrário das equações (7.1), a equação de Poisson contínua (7.5) possui solução para *quaisquer* funções F, G . Na verdade, a equação (7.5) possui infinitas soluções, pois sua versão homogênea $\mathbf{L}(Z) = 0$ é satisfeita por qualquer função afim da posição $Z(x, y) = Ax + By + C$. Quando qualquer destas soluções triviais é somada a uma solução específica da equação (7.5), obtemos uma solução distinta para as equações (7.5).

A constante aditiva C é uma indeterminação intrínseca do problema original, mas a parte linear $Ax + By$ é um artefato do método usado para obter a equação (7.5). Em uma solução com uma escala só, o termo $Ax + By$ fica determinado em parte pelas equações de vértices ao longo da fronteira do domínio efetivo (vértices nas bordas do domínio D bem como os vértices adjacentes a amostras de gradiente com peso zero). Nestes pontos, os laplacianos discretos $\mathcal{L}(z)$ e $\mathcal{H}(f, g)$ ficam sensíveis às derivadas de primeira ordem do mapa z , e portanto ao termo $Ax + By$.

Note que a parcela linear $Ax + By$ é preservada pela expansão dos mapas de altura. Portanto, se a iteração de Gauss-Seidel no nível m for interrompida prematuramente, qualquer erro na parcela $Ax + By$ que estiver presente no chute inicial afetará o resultado. Este tipo de erro é comum em mapas de altura integrados por métodos iterativos de uma escala só, onde o cálculo quase sempre tem que ser interrompido bem antes da convergência do método. Nestes casos, pode-se corrigir a parte mais grosseira deste termo comparando o valor médio (sobre D) do gradiente ∇Z estimado a partir de f e g com o valor estimado a partir do mapa de alturas calculado z , e subtraindo-se o termo linear apropriado $Ax + By$ de z do último para igualar os dois valores. Porém, esta solução não corrigirá erros devido ao término prematuro das iterações.

7.5.6 Análise

Para entender a razão da eficiência deste algoritmo, é necessário considerar sua atuação no domínio da frequência, ou seja, pela análise de Fourier [22]. Quando os mapas de gradiente são reduzidos, suas componentes de Fourier de frequência alta (menores que 2 pixels) são perdidas, enquanto que as componentes de baixa frequência terão seus comprimentos de onda divididos ao meio. Por hipótese de indução, portanto, a solução $z^{(k+1)}$ calculada recursivamente para o problema reduzido, após ser expandida para a escala original, estará correta nas frequências mais baixas; o erro estará apenas nas frequências mais altas, que foram suprimidos na redução. A correção destes erros requer um pequeno número de iterações pelo método de Gauss-Seidel, ou seja, neste algoritmo cada componente de Fourier do mapa de alturas será calculada na escala onde seu comprimento de onda é de apenas alguns pixels.

De qualquer forma, o algoritmo limita o número de iterações máximo em cada escala

k para $\kappa\lambda^k$. Portanto o tempo gasto em iterações de Gauss-Seidel em cada escala será proporcional à $N\kappa(\lambda/4)^k$. O tempo total para resolver o sistema para todas as escalas será $1 + \lambda/4 + \lambda^2/4^2 + \dots + \lambda^m/4^m$ vezes o trabalho da escala 0. Contanto que $\lambda < 4$, esta soma será menor que $1/(1 - \lambda/4)$, portanto o custo será $O(N)$. O custo de construir o sistema linear em cada escala k , incluindo a ordenação topológica, é proporcional a $N/4^k$; e portanto esta parte do custo será $O(N)$ também.

7.6 Testes

Para avaliar a eficiência e precisão do nosso algoritmo, testamos o mesmo com quatro conjuntos de dados artificiais e um conjunto de dados obtido de uma cena real. Veja a tabela 7.1 e a figura 7.9.

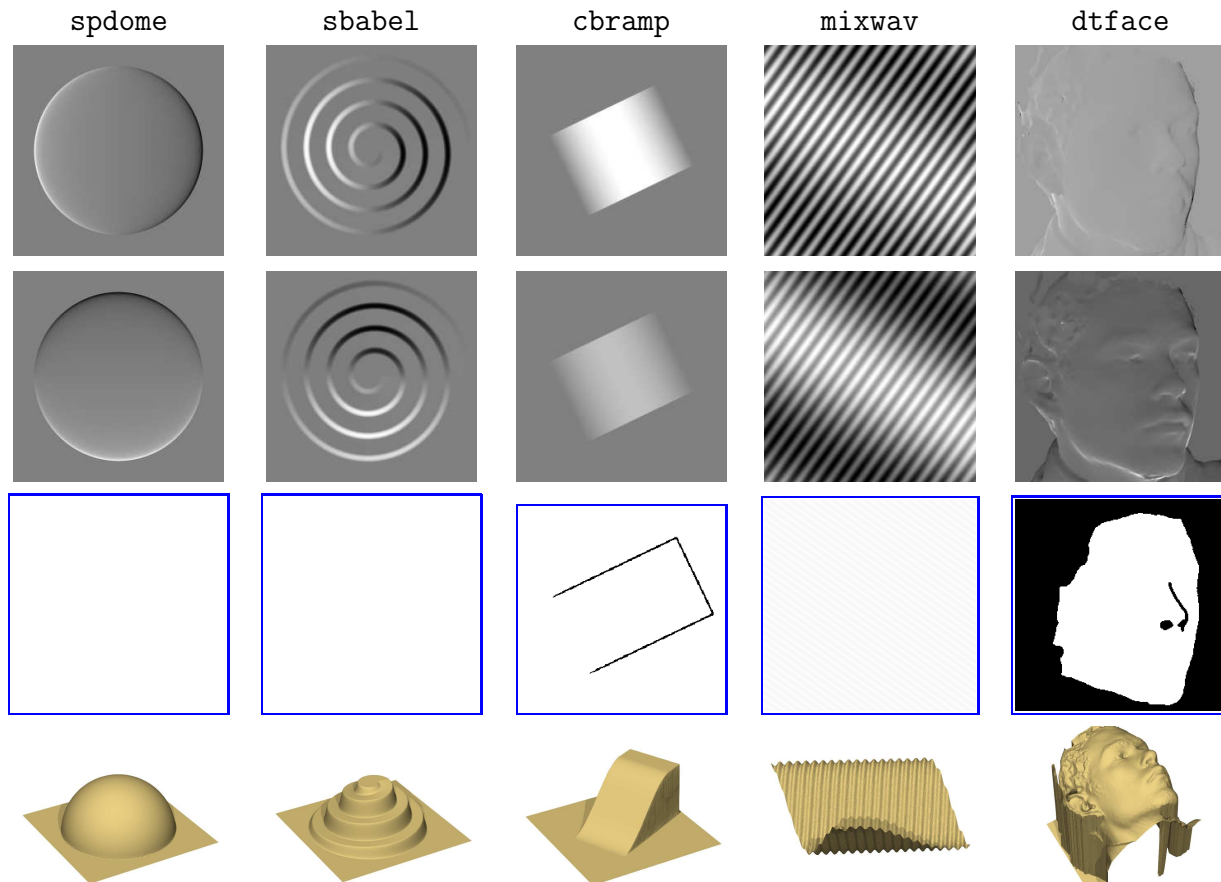


Figura 7.9: Os conjuntos de testes, para o algoritmo de integração multi-escala, mostrando os mapas de gradiente f (topo), g (segunda fileira), os mapas de peso w (terceira fileira) e mapa de alturas correto Z (embaixo).

Tabela 7.1: Conjuntos de testes.

Conjunto	Fonte	Descrição
spdome	Sint.	Domo esférico
sbabel	Sint.	Torre cônica com rampa em espiral
cbramp	Sint.	Rampa suave com degraus
mixwav	Sint.	Ondas de alta e baixa frequência
dtface	Real	Face obtida por scanner tridimensional

Conjuntos de dados artificiais: Cada um destes conjuntos de dados foi obtido a partir de uma função de $Z(x, y)$ definida por uma fórmula algébrica. A função Z do conjunto **spdome** é um uma calota esférica, um pouco menor que um hemisfério posicionado sobre um plano; ela é contínua e suave C_∞ exceto por uma descontinuidade de gradiente (mas não altura) ao longo da borda da calota. O mapa do conjunto **sbabel** é uma rampa em espiral conectando um plano de base a uma plataforma horizontal elevada no centro. Os degraus entre as voltas da rampa são bastante inclinados, mas não verticais. Esta função é contínua e suave (C_1) em todo lugar, exceto nas extremidades da rampa, onde há uma descontinuidade de gradiente (mas não de altura). A função Z do conjunto **cbramp** consiste de uma rampa de onde a altura é definida por um polinômio de grau 3 conectando o chão plano com uma plataforma elevada. Esta função é contínua e suave (C_1) exceto pelos degraus verticais em três dos lados da rampa. Finalmente, a função Z do conjunto **mixwav** é a soma de duas ondas senoidais com frequências espaciais e direções bem distintas.

Para estas cenas virtuais, cada amostra $f[u, v]$ ou $g[u, v]$ é obtida por sub-amostragem da correspondente derivada analítica de $Z(x, y)$ em uma grade 11×11 de pontos cobrindo um quadrado de tamanho 2×2 centrado no ponto $p[u, v]$, e calculando-se a média destas subamostras com uma ponderação bidimensional de Hann [21]. Este procedimento de amostragem tenta emular o processo de aquisição de gradiente por estéreo fotométrico, no sentido que é insensível a degraus na função Z , e está sujeito à ruído de amostragem em qualquer ponto onde o gradiente ∇Z muda abruptamente. Atribuímos peso 1 a todos os pixels de todos estes casos de teste; exceto para o conjunto **cbramp**, onde os pixels ao longo dos degraus receberam peso zero.

Na geração de dados de teste para integração, muitos autores usam a abordagem trivial de amostrar primeiro a função Z , e então calcular os mapas f e g por diferenças finitas do mapa z . Esta abordagem *não* é realista, uma vez que tais mapas de gradiente estariam corretos em todos os pontos — inclusive em regiões com degraus, e portanto trivialmente integráveis pelos métodos de integração por linhas, com erro zero.

O mapa de referência $z^*[u, v]$ foi obtido de maneira semelhante, calculando-se a média de $Z(x, y)$ em uma grade de 5×5 pontos de subamostragem no quadrado 1×1 centrado

em $q[u, v]$.

Conjunto de teste real: Este conjunto de teste, que denominamos **dtface**, foi obtido a partir da face de uma pessoa. A captura foi feita por um dispositivo de aquisição de superfícies da marca 3dMD [2], que combina técnicas de luz estruturada e estéreo geométrico. O resultado da captura foi uma malha de 84590 triângulos. Nós utilizamos o programa traçador de raios POV-RAY [24] para extrair o mapa de gradiente e o mapa de alturas de referência z^* desta malha. O mapa z^* foi obtido imageando a malha com projeção ortogonal após aplicar à mesma uma textura dependente da altura Z . O mapa de gradiente foi obtido renderizando a cena com as texturas **slope x**, **slope y** e **slope z**. Nas três imagens assim obtidas, a intensidade de cada pixel era o ângulo θ entre a normal no pixel e os vetores $(1, 0, 0)$, $(0, 1, 0)$ e $(0, 0, 1)$ respectivamente, codificado com o número real $v = 1 - \theta/\pi$. Estas três imagens foram convertidas nos mapas de derivadas f e g por um programa separado. O mapa de pesos binário w foi criado manualmente com um editor de imagens, excluindo as regiões de dados inconfiáveis como cabelos, plano de fundo, queixo, orelhas, etc.

7.6.1 Algoritmos de integração

Nesta avaliação, nós comparamos nosso integrador com os melhores métodos disponíveis representando as principais abordagens relacionadas na tabela 7.2.

Tabela 7.2: Métodos de integração testados.

Sigla	Descrição	Tipo	Aceita w
PI	Integração Combinatória de Superfície [36]	Integração por caminhos	Sim
FC	Frankot-Chellappa [37, 64]	Integral por Fourier	Não
UP	Poisson sem pesos [6, 4]	Poisson por DCT	Não
AT	Difusão com Transformação Afim [6, 4]	Poisson sol. direta	Sim
ME	Estimadores-M [6, 4]	Poisson sol. direta	Sim
MS	Nosso método multi-escala	Poisson iter. multi-escala	Sim

Para representar os algoritmos de integração por caminhos (PI) nós escolhemos o método Integração Combinatória de Superfície (*Combinatorial Surface Integration*) de Fraile e Hancock [36]. Utilizamos sua implementação em Matlab [83], com árvore de caminhos calculada pelo método *Norm*. Nós modificamos o código para aceitar o mapa de peso w como dado (em vez de tentar estimá-lo pelo rotacional) e para excluir da árvore quaisquer arestas que usem amostras de gradiente com peso zero.

Para representar a abordagem de integração por Fourier, nós escolhemos o algoritmo de Frankot e Chellappa (FC) [37] implementado em Matlab por Peter Kovesi [64]. Este código

utiliza funções do Matlab para calcular a FFT, e não aceita mapas de peso. Não utilizamos a reformulação de Wei [105] deste algoritmo, uma vez que ela consiste meramente em uma modificação dos filtros de frequência para suavizar os dados.

Os métodos baseados na fórmula de Poisson são representados pelo algoritmo Poisson sem pesos (UP) de Agrawal *et al.* [6] e suas variantes Estimadores-M (ME) e Difusão com Transformação Afim (AT) que aceitam um mapa de pesos. Para estes três algoritmos, usamos as implementações em Matlab dos autores [4]. O algoritmo UP utiliza a transformada cosseno discreta (DCT) para resolver o sistema linear de Poisson, enquanto AT e ME resolvem o sistema por meio de funções de biblioteca do Matlab. O estimador Laplaciano utilizado por AT depende de um número variável de valores de z (por volta de 6 em nossos testes), enquanto ME e UP utilizam um estimador de 5 valores, como o nosso. As versões originais de AT e ME tentam identificar *outliers* e dados com ruído no mapa de gradiente por um método de dedução de pesos iterativo; nós modificamos estas implementações de maneira a utilizar nosso mapa w como mapa de pesos inicial para esta iteração.

O método MS é nosso integrador multi-escala iterativo. Em todos os testes, definimos o número máximo de iterações de Gauss-Seidel κ como 50, e o limiar de convergência ϵ (na escala 0) como 0.0005, com parâmetros internos $\lambda = 2$ e $n_{\text{small}} = 2$.

Nós não testamos o algoritmo de Ng [76], uma vez que, segundo dados do autor, seus custos de memória e tempo são muito mais altos que os dos outros métodos testados, e sua vantagem aparenta ser principalmente a capacidade de preencher lacunas entre os dados de maneira mais “natural”. Nós tentamos testar também os algoritmos Superfície- α e Minimização de Energia de Agrawal [6], que também usam solução direta do sistema de Poisson com pesos; mas não conseguimos modificá-los para aceitar nossos mapas de peso. Entretanto, verificamos que para conjuntos de teste sem peso, estes métodos produzem resultados similares aos dos outros métodos testados, mas são consideravelmente mais lentos.

7.6.2 Robustez e precisão

Para avaliar a robustez e precisão do nosso método, nós comparamos a saída de cada um dos algoritmos da tabela 7.2 para todos os conjuntos de teste com o correspondente mapa de alturas de referência z^* . Os resultados estão apresentados na tabela 7.3 e nas figuras 7.10 e 7.11.

Mapas de gradiente obtidos de objetos reais geralmente estão contaminados por níveis significativos de ruído. Uma vez que este ruído pode afetar a precisão e o tempo de convergência de integração, nós também repetimos todos os testes misturando aos mapas de gradiente 30% de ruído gaussiano com média zero e variância unitária. Os resultados

destes testes estão apresentados na tabela 7.4 e nas figuras 7.12, 7.13.

Avaliação: Uma pequena dificuldade que encontramos nas comparações decorre do fato de nosso método produzir mapas de altura com uma linha e uma coluna a mais do que os mapas de gradiente, como discutido na seção 7.3.2. Todos os outros métodos geram sempre mapas de altura com mesmo tamanho que os mapas de gradiente. A fim de fazer uma comparação justa, tivemos que gerar mapas de referência Z com o tamanho adequado para cada integrador.

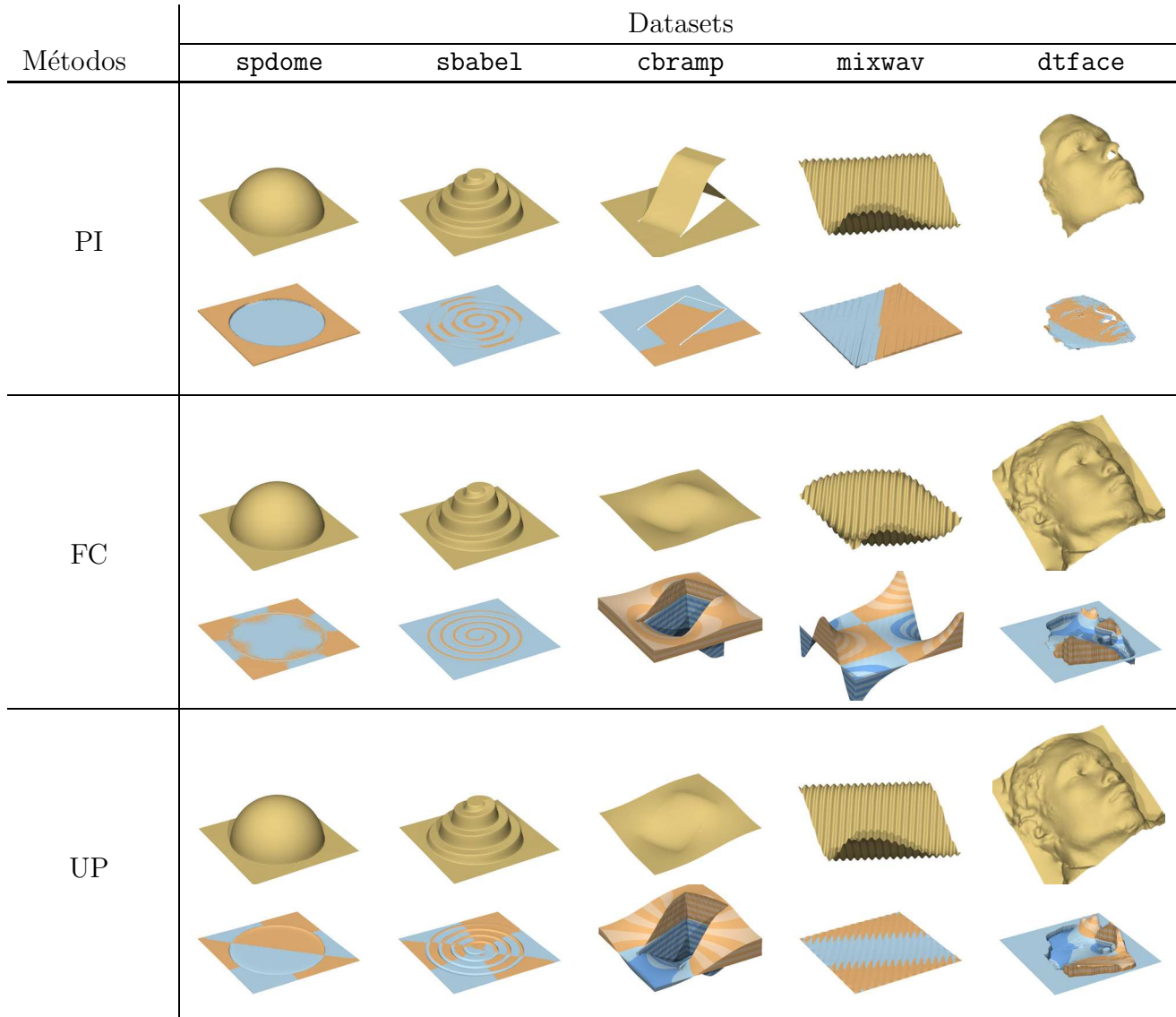


Figura 7.10: Mapas de altura calculados e mapas de erro para os métodos PI, FC e UP e os conjuntos de teste. Azul e laranja indicam que a altura calculada está respectivamente abaixo ou acima da altura correta.

Em cada caso, nós comparamos a raiz do erro médio quadrático e entre dois mapas de altura integrados e o erro relativo e/R , onde R^2 é o valor médio quadrático dos dois mapas de altura. Os valores de e e R são calculados após deslocar ambos mapas em Z de maneira que suas alturas tenham média zero, a fim de cancelar as constantes arbitrárias de integração presentes em cada mapa. No cálculo de e e R , cada parcela correspondendo a um ponto $q[u, v]$ é ponderada pelo peso total W_{oo} do desse ponto, definido pela formula (7.12) ou (7.18) para a escala 0.

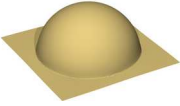
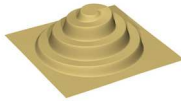

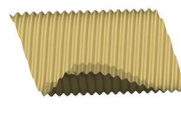

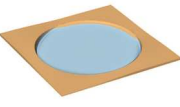
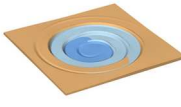
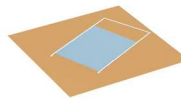
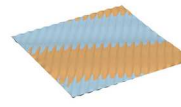

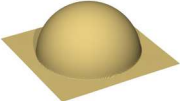
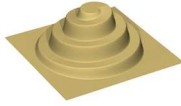

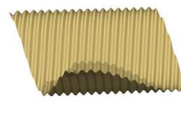

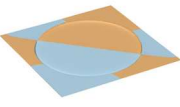
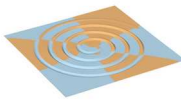
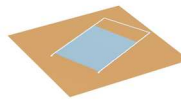
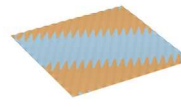

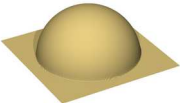
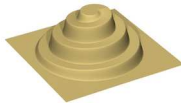

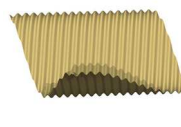

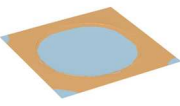
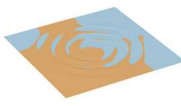
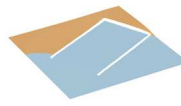
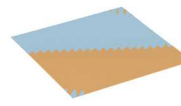
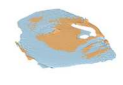
Métodos	Datasets				
	spdome	sbabel	cbramp	mixwav	dtface
AT					
					
ME					
					
MS					
					

Figura 7.11: Mapas de altura calculados e mapas de erro para os métodos AT, ME e MS e conjuntos de teste.

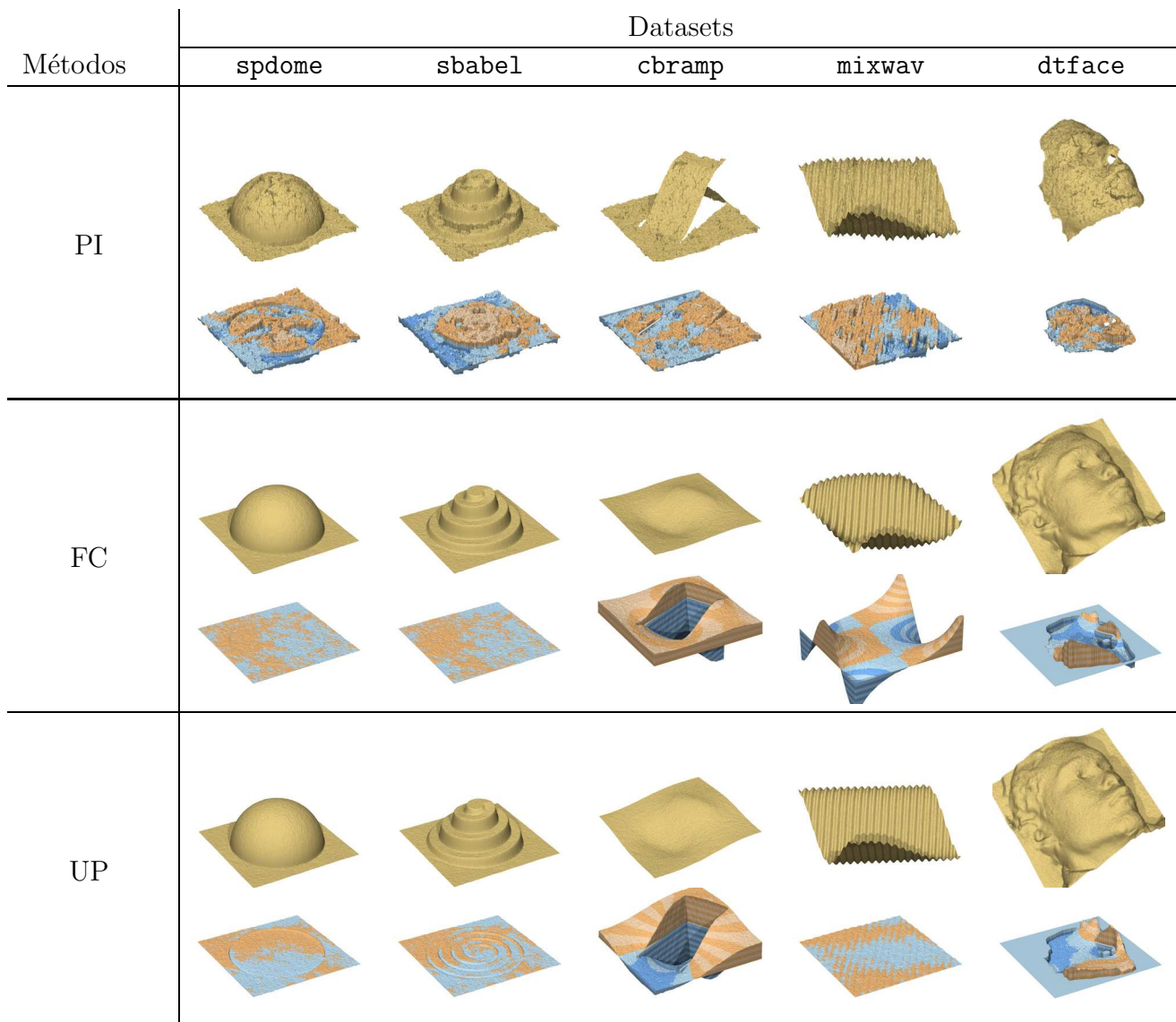


Figura 7.12: Mapas de altura calculados e mapas de erro obtidos pelos métodos PI, FC e UP nos conjuntos de teste com 30% de ruído Gaussiano misturado.

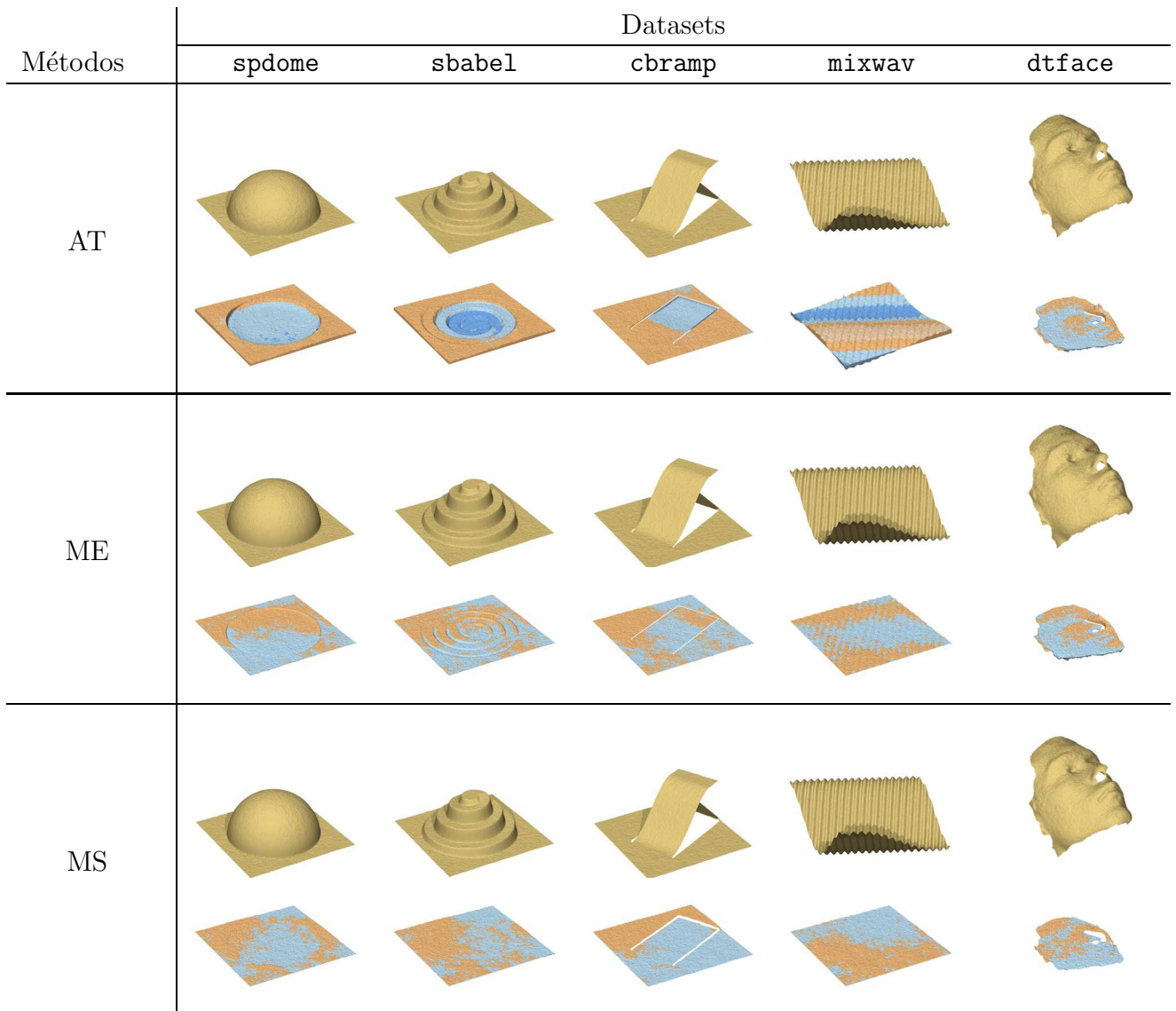


Figura 7.13: Mapas de altura calculados e mapas de erro obtidos pelos métodos AT, ME e MS nos conjuntos de teste com 30% de ruído Gaussiano misturado.

Tabela 7.3: Erro RMS relativo de cada método para mapas de gradiente sem ruído.

	spdome		sbabel		cbramp		mixwav		dtface	
Meth.	e	e/R	e	e/R	e	e/R	e	e/R	e	e/R
PI	1.90	5.5%	0.06	0.2%	0.02	0.1%	2.78	7.2%	0.27	1.5%
FC	0.08	0.2%	0.05	0.2%	30.25	120.4%	15.08	42.9%	7.91	48.8%
UP	0.58	1.6%	0.57	2.1%	29.84	107.3%	0.46	1.2%	5.43	32.2%
AT	1.82	5.2%	3.19	12.4%	0.14	0.4%	0.89	2.3%	0.70	4.0%
ME	0.67	1.9%	0.57	2.1%	0.14	0.4%	0.46	1.2%	0.72	4.1%
MS	0.19	0.5%	0.15	0.6%	0.89	2.6%	0.36	0.9%	0.41	2.3%

Tabela 7.4: Erro RMS relativo de cada método para mapas de gradiente com ruído.

	spdome		sbabel		cbramp		mixwav		dtface	
Meth.	e	e/R	e	e/R	e	e/R	e	e/R	e	e/R
PI	3.84	10.8%	5.70	19.3%	3.10	8.9%	4.25	10.8%	3.64	19.8%
FC	0.30	0.9%	0.30	1.1%	30.26	120.5%	15.09	42.9%	7.96	49.1%
UP	0.64	1.8%	0.65	2.4%	29.83	107.4%	0.51	1.3%	5.34	31.8%
AT	3.30	9.8%	3.94	15.5%	1.53	4.5%	4.75	13.0%	0.85	4.9%
ME	0.71	2.0%	0.65	2.4%	0.43	1.2%	0.51	1.3%	0.77	4.4%
MS	0.40	1.1%	0.41	1.5%	0.82	2.4%	0.44	1.1%	0.65	3.7%

Discussão: Como evidenciado pelas tabelas 7.3 e 7.4, os únicos métodos que ofereceram resultados confiáveis para todos os conjuntos de teste foram os métodos AT e ME de Agrawal, e nosso método multi-escala MS. A sensibilidade do método PI ao ruído é evidente na tabela 7.4. Como esperado, os métodos que não aceitam pesos (FC e UP) falharam completamente nos conjuntos de teste com degraus e dados inválidos no mapa de gradiente. O método FC falhou também no conjunto `mixwav` apesar de não haver degraus internos no mapa de alturas, o método criou um degrau nas bordas do domínio D ao adotar topologia toroidal para o mesmo.

Observamos que o processo de subamostragem usado para obter o mapa de gradientes tende a suavizar detalhes de alta frequência, especialmente em regiões de alta curvatura (por exemplo ao longo da borda da calota em `spdome` ou entre as voltas da rampa em `sbabel`). Portanto, por melhor que seja o algoritmo, é inevitável que haja pequenas diferenças entre o mapa integrado z e o mapa de referência z^* .

7.6.3 Tempo e memória

Para comparar a eficiência de nosso método, nós medimos o tempo e memória necessários para processar dois mapas de gradiente `spdome` e `dtface`, amostrados em vários tamanhos de mapas, desde 64×64 até 512×512 pixels.

Comparamos nosso método com os dois integradores de Agrawal *et al.* [4], *Difusão com Transformação Afim* (AT) e o algoritmo de construção e resolução de equações de *Poisson com pesos* (PC) que é o passo mais interno dos algoritmos Estimadores-M, Minimização de Energia e Superfície- α . Estes são os únicos métodos descritos na literatura que aceitam mapas de peso (portanto, resolvendo o mesmo problema que o nosso) e são rápidos o suficiente para uso prático. Nós removemos o loop externo destes três métodos uma vez que estamos avaliando apenas a integração do mapa de gradientes, não a geração de mapa de pesos.

Todos os testes foram efetuados em um Dell XPS 1340, com processador Intel Core Duo P8600 2.4 GHz e 4GB de memória disponível. Utilizamos versões de 64 bits de todo software envolvido. Os tempos absolutos não são diretamente comparáveis uma vez que os integradores de Agrawal foram executados no Matlab 2008b no Windows Vista, enquanto o nosso foi compilado no GNU GCC e executado no Mandriva Linux 2010. Para tornar a comparação significativa, analisamos a variação do uso de tempo e memória em função dos tamanhos dos mapas de entrada.

Tempo: Nas avaliações do tempo de processamento, medimos apenas a construção e solução do sistema linear, excluindo *overhead* de entrada e saída. Para a medição de memória, contamos apenas com o espaço utilizado pelo sistema linear e pelo espaço utilizado em sua resolução. No caso de nosso algoritmo, incluímos nestes custos também o tempo e a memória utilizados para preparar a pirâmide de gradientes e pesos. As figuras 7.14 e 7.15 e na tabela 7.5 apresentam os resultados.

Os gráficos da figura 7.14 mostram que os tempos de execução crescem de maneira bem diferente para os três algoritmos: como $\Theta(N)$ para o nosso algoritmo multi-escala e aparentemente como $\Theta(N^{1.5})$ para os outros métodos que utilizam resolução direta de Poisson.

Memória: Para avaliação de consumo de memória, observamos que os métodos diretos precisam armazenar a matriz \mathbf{M} e também o fator triangular de Gauss \mathbf{U} (ou fator de Cholesky \mathbf{R}). Note que o fator de Gauss menor \mathbf{L} não precisa ser armazenado. Para estes métodos, nós contamos os elementos não nulos $N_{\mathbf{M}}$ em \mathbf{M} e $N_{\mathbf{U}}$ em \mathbf{U} , e estimamos o uso de memória conservadoramente como $12N_{\mathbf{M}} + 16N_{\mathbf{U}}$ bytes supondo uma representação genérica de matriz esparsa para \mathbf{U} . Nós observamos que $N_{\mathbf{M}}$ não excede $5N$ para PC e $7N$ para AT, mas $N_{\mathbf{U}}$ é muito maior e escala como $O(N^{1.15})$ (linhas pontilhadas). O uso de memória da implementação MatLab é muito maior que esta estimativa: tanto assim

que PC e AT não conseguiram alocar memória na máquina utilizada para testes dado um mapa 1024×1024 .

Nosso integrador multi-escala também utiliza menos memória que os métodos diretos, veja a figura 7.15. Seu uso de memória é dominado pela matriz de Poisson M que possui em sua maior parte $5N$ elementos não nulos e é armazenada em uma estrutura de dado esparsa que utiliza $60N$ bytes. Os mapas de gradiente em escala reduzida utilizam $5N$ bytes adicionais de espaço.

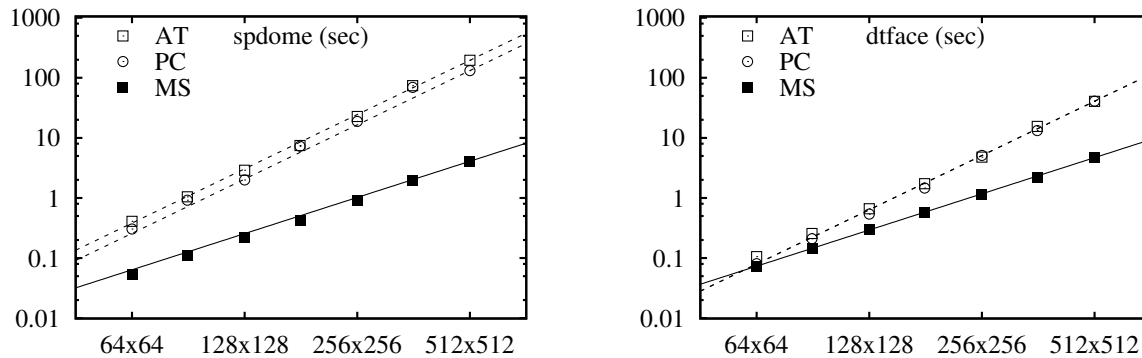


Figura 7.14: Gráficos log-log do tempo de execução de dois métodos de resolução direta (PC,AT) e nosso método multi-escala(MS), em segundos, em dois conjuntos representativos de teste e em diversas resoluções. As linhas retas são funções ajustadas $O(N)$ (linha cheia) e $O(N^{1.5})$ (linha tracejada).

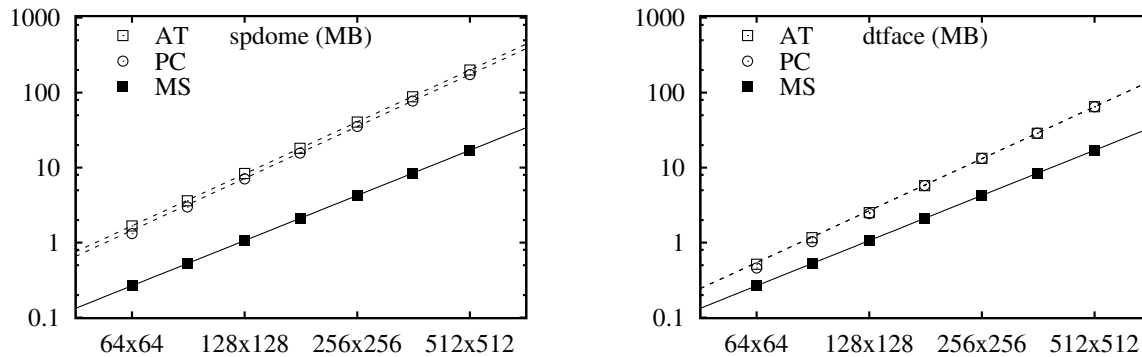


Figura 7.15: Gráficos log-log de uso de memória para a matriz de sistema M e seu fator U (se aplicável), em MBytes. As linhas retas são ajustadas as funções $O(N)$ (solida) e $O(N^{1.15})$ (pontilhada).

Note que os custos de tempo e memória são muito menores para o conjunto `dtface` do que `spdome` para os métodos PC e AT. Isto porque estes algoritmos excluem do sistema de Poisson os valores de altura sem dado de gradiente ($w[u, v] = 0$) e atribuem zero a eles diretamente.

Tabela 7.5: Custos de tempo e memória para os métodos MS, PC, e AT em dois conjuntos de teste.

spdome						
N	tempo			memória		
	AT	PC	MS	AT	PC	MS
$64 \times 64 = 4096$	0.41	0.30	0.05	1.7	1.3	0.3
$90 \times 90 = 8100$	1.05	0.92	0.11	3.6	3.0	0.5
$128 \times 128 = 16384$	2.89	1.99	0.22	8.4	7.1	1.1
$180 \times 180 = 32400$	7.46	7.27	0.43	18.1	15.6	2.1
$256 \times 256 = 65536$	22.67	18.82	0.92	40.5	35.5	4.3
$360 \times 360 = 129600$	74.03	69.21	1.94	88.1	77.2	8.4
$512 \times 512 = 262144$	195.44	131.84	4.08	200.2	174.1	17.0
dtface						
$64 \times 64 = 4096$	0.11	0.08	0.07	0.5	0.5	0.3
$90 \times 90 = 8100$	0.26	0.21	0.14	1.2	1.0	0.5
$128 \times 128 = 16384$	0.66	0.54	0.29	2.5	2.4	1.1
$180 \times 180 = 32400$	1.73	1.46	0.57	5.7	5.8	2.1
$256 \times 256 = 65536$	4.79	5.10	1.13	13.3	13.3	4.3
$360 \times 360 = 129600$	15.51	13.24	2.21	28.7	28.8	8.4
$512 \times 512 = 262144$	40.60	40.84	4.69	64.8	64.7	17.0

7.7 Calculando pesos a partir dos gradientes

Vários dos métodos de integração com mapa de pesos encontrados na literatura tentam calcular os pesos a partir do mapa de gradientes. Alguns desses métodos supõem que os erros de medida nestes últimos podem ser detectados e estimados pela violação da equação (7.1). Eles então atribuem a cada amostra um peso $w[u, v]$ que é alguma função decrescente do rotacional na vizinhança do ponto $p[u, v]$, estimado por alguma versão discreta da equação (7.2). Nesta classe estão os algoritmos de Agrawal *et al.* [5, 6].

Outros métodos atribuem os pesos iterativamente, analisando a consistência entre o mapa de alturas integrado e o mapa de gradiente dado. Um exemplo desta abordagem foi descrito por Durouet *et al.* [33] em 2009. Outro exemplo desta classe é o método de Reddy *et al.* [82]. Eles observaram que os integradores baseados em Poisson ou Fourier implicitamente aproximam os mapas f, g de gradiente dados pelos mapas f', g' com rotacional nulo mais próximos a f, g na métrica de erro ℓ_2 (raiz da soma dos quadrados). Uma vez que estimativas ℓ_2 são notoriamente sensíveis a *outliers*, eles propõem determinar f', g' que seja mais próximo a f, g pela métrica ℓ_1 (soma dos valores absolutos). Entre-

tanto, observamos que a métrica ℓ_1 de um vetor v coincide com uma versão ponderada da métrica ℓ_2 , cujos pesos são escolhidos adequadamente (dependentes do vetor v). Ou seja, o integrador de Reddy *et al.* é um método de integração por Poisson que usa um mapa de pesos calculado de maneira específica.

A maioria destes métodos de estimação do mapa de pesos podem ser utilizadas com nosso integrador multi-escala. Entretanto, como já ressaltamos, o mapa de gradientes em geral não contém informação suficiente para determinar a localização de degraus e outros dados não confiáveis. É possível que existam degraus de altura elevada mesmo em regiões onde os mapas de gradiente dados possuem rotacional nulo. De modo geral, para obter resultados confiáveis o mapa de pesos deve ser obtido por outros meios, como por exemplo análise de sombras e realces lustrosos sob diferentes condições de iluminação [65, 25], polarimetria [13], indicações obtidas por meio de estéreo geométrico [47], ou conhecimento *a-priori* da geometria da cena. Consideramos este um problema separado, fora do escopo desta tese.

7.8 Limitações do método multi-escala

A abordagem multi-escala não é válida em situações como a da figura 7.16, quando o domínio efetivo (regiões com peso não-zero) inclui um corredor longo e estreito. Denominaremos mapas de gradiente com estas características como *mapas fracamente conexos*.

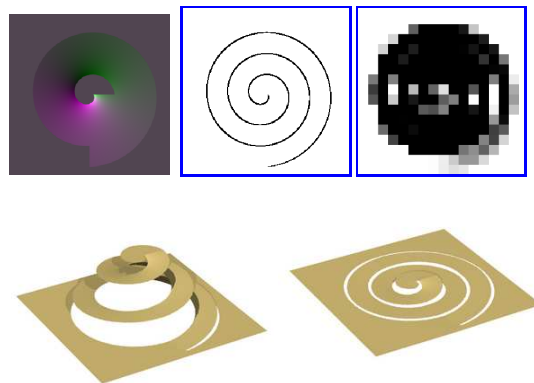


Figura 7.16: Caso patológico para o algoritmo MSMapIntegrate. No alto: mapas $f^{(0)}$ e $w^{(0)}$ (256×256), e o mapa de pesos reduzido $w^{(4)}$ (16×16).

Embaixo: o mapa de alturas z obtido por solução direta do sistema de Poisson (correto) e pelo MSMapIntegrate com 200 iterações por nível (incorreto).

Note que o procedimento REDUCEWEIGHTMAP descrito no algoritmo 7.7 precisa garantir que qualquer pixel que é atravessado por um degrau receberá peso zero. Portanto, se o corredor possui t pixels de largura e T pixels de comprimento, ele ficará interrompido e/ou obliterado na escala $k \approx \log_2 t$. Isto significa que o mapa $z^{(k)}$ calculado nessa escala

terá valores incorretos ao longo de todo o corredor no nível $k - 1$. O algoritmo de Gauss-Seidel terá então que calcular as alturas ao longo do corredor a partir de um chute inicial incorreto e para isso ele necessitará de $\Omega((T/t)^2)$ iterações. Isto é $\Omega(N^2)$ no pior caso possível.

Este problema ocorre também quando o mapa de pesos possui elementos de peso zero espalhados com espaçamento médio t em uma região com diâmetro T . Para estes casos patológicos, a solução direta do sistema pode ser muito mais rápida que o método iterativo. No capítulo 8 apresentaremos um algoritmo melhorado que resolve este problema.

Capítulo 8

Integração de mapas de gradiente por malhas não regulares

Apresentamos neste capítulo outro método iterativo multi-escala para integração de mapas de normais, como o algoritmo MSMAPINTEGRATE, que admite mapa de pesos mas é eficiente mesmo para mapas fracamente conexos — resolvendo portanto o problema descrito na seção 7.8.

8.1 Malha de diferenças ponderadas

Para este algoritmo, utilizamos uma representação de *malha de diferenças ponderadas* para o mapa de gradiente. Ela consiste de um grafo abstrato dirigido G com vértices (nós) $\mathcal{V}G$ e arestas orientadas (arcos) $\mathcal{E}G$, onde cada vértice v é associado a uma altura desconhecida $z[v]$, e cada aresta dirigida e é associada a dois parâmetros numéricos: a *diferença* $d[e]$ e o *peso* $w[e]$.

O parâmetro $d[e]$ é uma estimativa da diferença $z[v] - z[u]$ entre as alturas do vértice destino $v = \text{DST}(e)$ e do vértice de origem $u = \text{ORG}(e)$. Esta estimativa é derivada de gradientes de superfície entre os pontos de amostragem de altura (veja seção 8.2). O peso da aresta $w[e]$ é um número positivo que expressa a confiabilidade da estimativa. Mais especificamente, nós supomos que a diferença da aresta inclui um erro de medida aleatório gaussiano, com valor esperado zero e variância proporcional a $1/w[e]$. Excluímos arestas com pesos zero, pois elas não fornecem nenhuma informação sobre as alturas.

Por definição, para cada aresta dirigida e na malha ponderada, a aresta dirigida oposta $\text{SYM}(e)$ também está presente na malha, com $d[\text{SYM}(e)] = -d[e]$ e $w[\text{SYM}(e)] = w[e]$. Porém, quando desenhamos a malha, desenhamos apenas uma dessas duas arestas. Veja a figura 8.1.

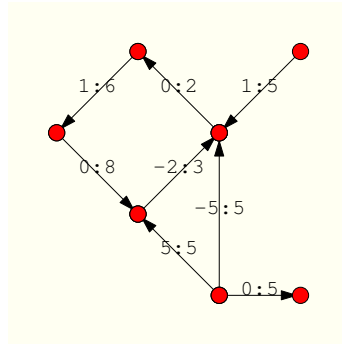


Figura 8.1: Uma pequena malha de diferenças ponderadas. Os rótulos das arestas são os pares $d[e] : w[e]$. Note que o desenho mostra apenas uma aresta de cada par $e, \text{SYM}(e)$.

Dizemos que a malha é *simples* se ela não tem arestas paralelas (duas ou mais arestas com mesma origem e mesmo destino). Em uma malha simples, podemos identificar cada aresta e com o par ordenado (u, v) de seus vértices de origem e destino. Neste caso podemos denotar $d[e]$ também por $d[u, v]$ e $w[e]$ por $w[u, v]$.

8.1.1 O problema de integração da malha

Uma malha ponderada G pode ser interpretada como um sistema de *equações de aresta*, as equações lineares

$$z[\text{DST}(e)] - z[\text{ORG}(e)] = d[e] \quad (8.1)$$

para cada aresta dirigida e . O problema de *integração* da malha G é calcular o valor mais provável da altura $z[v]$ para cada vértice v , supondo dados os parâmetros $d[e], w[e]$ para cada aresta do grafo e .

É evidente que cada componente conexa de G pode ser tratada como uma instância separada deste problema. Portanto, de agora em diante vamos supor que G é um grafo conexo. Além disso, uma vez que as equações (8.1) dependem apenas das diferenças entre as alturas, a solução para uma malha conexa possui um (e apenas um) grau de liberdade: uma constante aditiva correspondente à constante de integração C da seção 7.5.5.

Um conjunto de alturas z , é dito *livre de tensão* se satisfaz todas as equações (8.1) exatamente. Este é o caso se e somente se a soma (com sinal) das diferenças das arestas ao longo de qualquer ciclo dirigido for zero. Esta propriedade é o equivalente da condição de integrabilidade, ou rotacional zero do capítulo 7. Em particular, se a malha tem apenas um caminho simples entre quaisquer vértices (ou seja, é uma árvore) ela sempre possui uma solução livre de tensão.

Se G tem ciclos, entretanto, o sistema de equações (8.1) é sobredeterminado. Neste caso, é quase impossível que ele tenha uma solução livre de tensão; ou seja, é quase

sempre impossível satisfazer todas as equações ao mesmo tempo. Dada a suposição da existência de erros de medida gaussianos independentes nas arestas, a análise bayesiana diz que o conjunto mais provável de alturas z é a solução por mínimos quadrados para o sistema (8.1); isto é, as alturas z que minimizam a *função discrepância quadrática*

$$Q(z) = \sum_{e \in \mathcal{E}G} w[e](z[\text{DST}(e)] - z[\text{ORG}(e)] - d[e])^2 \quad (8.2)$$

A seguinte analogia pode ajudar a compreender o problema. Cada vértice v é modelado como uma placa horizontal sem peso que é livre para deslizar verticalmente mas não pode mover-se horizontalmente ou girar. A altura $z[v]$ é a posição vertical desta placa. Cada aresta $e = (u, v)$ é modelada como uma mola vertical ideal com coeficiente de rigidez $w[e]$, conectada às placas u e v de tal maneira que esta exerce sobre estas duas placas uma força vertical com magnitude $w[e](z[v] - z[u] - d[e])$; isto é, a mola tenta forçar que a distância $z[v] - z[u]$ seja igual à $d[e]$. O vetor de alturas $z[v]$ que minimiza $Q(z)$ corresponde à situação de equilíbrio mecânico, onde a força total sobre cada placa é zero. Isto porque a energia total potencial das molas é $\frac{1}{2}Q(z)$, e o sistema estará em repouso (sem forças atuando em cada vértice) quando a energia potencial for mínima. Em particular a solução z é livre de tensão, se a energia potencial $Q(z)$ é zero; isto é, se cada mola tiver seu comprimento de repouso

8.1.2 Integração por caminhos

Se a malha admite uma solução livre de tensão, $z[v]$, ela pode ser calculada escolhendo-se uma árvore geradora T para G , associando-se uma altura arbitrária para algum vértice v_0 , e então utilizando-se a equação (8.1) para calcular as alturas dos outros vértices em ordem crescente de distância de v_0 ao longo de T . Note que os pesos das arestas são irrelevantes neste caso. Este algoritmo é a versão para malhas dos algoritmos de integração por caminhos descritos na seção 7.2.1.

Se aplicarmos o algoritmo de integração por caminhos nesse caso, obteremos uma solução aproximada que depende da escolha do vértice inicial e da árvore T .

8.1.3 Equação de equilíbrio dos vértices

Suponha que a malha G é simples, e denotemos por $G[u]$ o conjunto dos vizinhos do vértice u em G . A função Q é minimizada quando cada vértice u está em equilíbrio. Isto é, se e somente se nós tivermos

$$\sum_{v \in G[u]} w[u, v](z[v] - z[u] - d[u, v]) = 0 \quad (8.3)$$

Podemos reescrever a equação (8.3) como

$$z[u] = \frac{1}{w_t[u]} \sum_{v \in G[u]} w[u, v](z[v] - d[u, v]) \quad (8.4)$$

onde

$$w_t[u] = \sum_{v \in G[u]} w[u, v] \quad (8.5)$$

Em outras palavras, o equilíbrio ocorre quando $z[u]$ é a média ponderada de $z[v] - d[u, v]$, sobre todos os vizinhos v de u , sendo que o termo de cada vizinho v é ponderado pelo peso $w[u, v]$. A solução z é livre de tensão se todos os termos $z[v] - d[u, v]$ tem o mesmo valor. A equação (8.4) pode ser escrita também como

$$z[u] - \sum_{v \in G[u]} \lambda[u, v] z[v] = \sum_{v \in G[u]} \lambda[u, v] d[u, v] \quad (8.6)$$

onde $\lambda[u, v]$ é $w[u, v]/w_t[u]$, o *peso relativo* de v entre os vizinhos de u .

Outra analogia para este problema matemático é um circuito elétrico onde cada vértice v é um nó condutor, a variável $z[v]$ é o potencial elétrico do nó (em volts), e cada aresta (u, v) é uma bateria com voltagem $d[u, v]$ com resistência interna $1/w[u, v]$ (em ohms) conectada aos nós u e v . A corrente (u, v) (em amperes) que chega em u pela aresta $[u, v]$ é portanto $-w[u, v](z[v] - z[u] - d[u, v])$. O funcional $Q(z)$ é a energia extra dissipada pelos resistores quando o potencial elétrico de cada nó v é z . A equação (8.3) é então a lei de Kirchoff, que é verdadeira quando o circuito está em equilíbrio elétrico e a corrente total entrando e saindo de cada circuito é zero.

8.1.4 Formulação matricial

Para expressar o problema em forma de matriz, sejam

- v_1, v_2, \dots, v_n , os vértices de G , em ordem arbitrária;
- e_1, e_2, \dots, e_m uma lista de arestas dirigidas de G , em ordem arbitrária, que inclui exatamente uma versão dirigida de cada aresta não dirigida;
- A matriz \mathbf{A} de incidência $n \times m$ tal que \mathbf{A}_{ij} vale $+1$ se o vértice v_i é o destino da aresta e_j , vale -1 se v_i é a origem de e_j , e vale 0 caso contrário;
- \mathbf{W} uma matriz diagonal $m \times m$ tal que $\mathbf{W}_{jj} = w[e_j]$;
- \mathbf{d} um vetor de m elementos tal que $\mathbf{d}_j = d[e_j]$;
- \mathbf{z} um vetor coluna de n elementos tal que $\mathbf{z}_k = z[v_k]$;

para todo $i, j \in \{1, \dots, m\}$ e $k \in \{1, \dots, n\}$. O funcional Q pode então ser expresso como o produto matricial

$$Q(z) = (z^T A - d)W(z^T A - d)^T \quad (8.7)$$

onde T denota o operador de transposição. O vetor z que minimiza Q pode ser calculado derivando esta fórmula em relação a cada z_k e igualando o resultado a zero. Na forma matricial, estas equações são

$$Mz = b \quad (8.8)$$

onde

$$M = AWA^T \quad (8.9)$$

e

$$b = AWd^T \quad (8.10)$$

A solução do sistema (8.8) pode então ser obtida pelo método de Gauss-Seidel ou resolução direta.

8.2 Conversão de grade para malha

A malha ponderada generaliza propriamente o conceito de mapa de gradiente ponderado, no sentido de que todo mapa pode ser convertido para uma malha essencialmente equivalente.

Especificamente, sejam dados os mapas de derivadas $f[r, s], g[r, s]$, com n_x colunas e n_y linhas, contendo amostras das derivadas $\partial Z/\partial X$ e $\partial Z/\partial Y$, respectivamente, em uma grade regular de pontos de amostragem (r, s) como no capítulo 7. Suponha também que é dado um mapa $w[r, s]$ de mesmo tamanho, com pesos não negativos que expressam a confiabilidade destes valores de gradiente. Mais precisamente, suponha que $1/w[r, s]$ é a variância dos erros de medida (supostos gaussianos de média zero) que contaminam $f[r, s]$ e $g[r, s]$. Particularmente, $w[r, s] = 0$ significa que as amostras $f[r, s]$ e $g[r, s]$ são completamente desconhecidas.

Como no capítulo 7, nós interpretamos cada pixel de índices (r, s) como um quadrado unitário cujo centro $\rho[r, s] = (r + 1/2, s + 1/2)$ é o ponto de amostragem das derivadas $f[r, s]$ e $g[r, s]$. Por definição, os vértices da malha equivalente G são os cantos destes pixels, que conceitualmente formam uma grade com $n_x + 1$ colunas e $n_y + 1$ linhas. As arestas da malha correspondem aos lados dos quadrados. A diferença $d[e]$ de cada aresta é o valor de média de derivada de altura ao longo da aresta; ou seja, $\partial Z/\partial X$ se e é

horizontal ou $\partial Z/\partial Y$ se e é vertical. Este valor é interpolado no ponto médio da aresta a partir das quatro amostras de $f[r, s]$ ou $g[r, s]$ que circundam a aresta e (duas de cada lado) pela fórmula INTERPOLATE descrita na seção 7.4. O peso $w[e]$ dessa aresta também é calculado como descrito nessa seção.

8.3 Integração multi-escala de malha

Nosso algoritmo de integração em malhas utiliza uma abordagem multi-escala semelhante à do algoritmo descrito no capítulo 7. Em princípio, o algoritmo poderia ser aplicado a qualquer malha de diferença ponderada; entretanto sua eficiência só foi comprovada para malhas planares (como são as malhas obtidas pelo método da seção 8.2).

O cerne do nosso algoritmo é uma rotina de *dizimação* que remove uma certa fração dos vértices da malha de entrada G , produzindo uma malha menor G' . Os vértices de G' são um subconjunto de $\mathcal{V}G$, e as arestas de G' são definidas de maneira a resumir as informações de peso e diferença contidas nas arestas de G . O problema de integração é então resolvido recursivamente para a malha G' , gerando uma aproximação z' para seus vértices. Por interpolação destas alturas obtemos um chute inicial z para a malha original G . As alturas z são então ajustadas pelo método iterativo de Gauss-Seidel. A recursão é interrompida quando o grafo G fica reduzido a um único vértice v , ao qual podemos atribuir altura z nula.

Em outras palavras, construímos uma pirâmide $G^{(0)}, G^{(1)}, \dots, G^{(m)}$ de malhas, onde $G^{(0)}$ é a malha de entrada G , $G^{(m)}$ é um único vértice v , e cada malha $G^{(k+1)}$ é obtida por dizimação da malha anterior $G^{(k)}$. Calculamos então as soluções $z^{(m)}, z^{(m-1)}, \dots, z^{(0)}$, nesta ordem, onde $z^{(m)}[v]$ é zero, e cada $z^{(k)}$ é obtido a partir de $z^{(k+1)}$ por interpolação e iterações de Gauss-Seidel. A saída do algoritmo é então o mapa de alturas $z^{(0)}$.

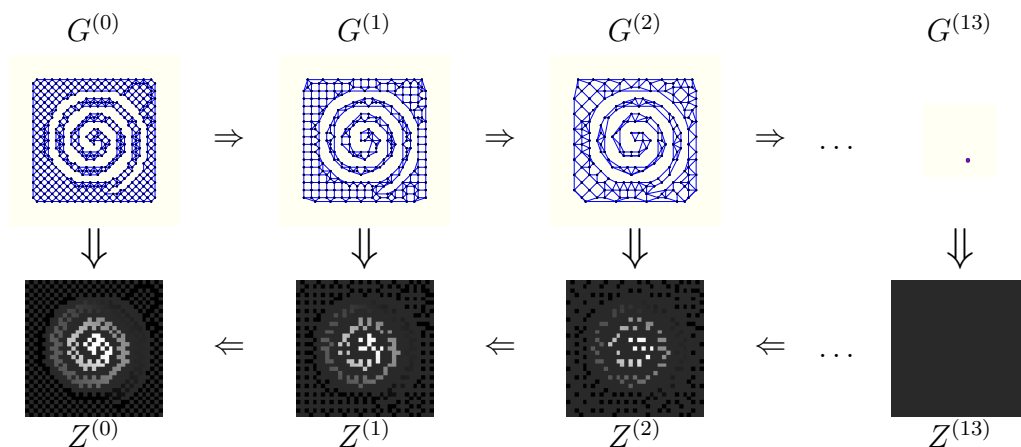


Figura 8.2: Procedimento de integração multi-escala por malhas.

Formalmente, o algoritmo é o procedimento recursivo `MSMESHINTEGRATE` cujo pseudocódigo é dado na figura 8.3. Ele recebe como entradas a malha de diferenças ponderadas G , o número máximo de iterações q e uma tolerância ε , e retorna um vetor de alturas z para $\mathcal{V}G$.

Procedimento `MSMESHINTEGRATE`(G, q, ε)

1. Se $\#\mathcal{V}G = 1$ então
 2. Seja v o único vértice em $\mathcal{V}G$; faça $z[v] \leftarrow 0$;
 3. senão
 4. $G' \leftarrow \text{DECIMATE}(G)$;
 5. $\beta \leftarrow \#\mathcal{V}G' / \#\mathcal{V}G$;
 6. $z' \leftarrow \text{MSMESHINTEGRATE}(G', q / \sqrt{\beta}, \varepsilon \sqrt{\beta},)$;
 7. $z \leftarrow \text{INTERPOLATE}(z', G)$;
 8. $z \leftarrow \text{SOLVESYSTEM}(z, G, q, \varepsilon)$;
 9. Retorna z .
-

Figura 8.3: Procedimento principal do integrador de multi-escala para malhas de diferenças ponderadas

8.3.1 Dizimação de malha

O procedimento `DECIMATE`, chamado na etapa 4, recebe uma malha simples G planar e conexa, e retorna uma malha menor G' , também simples, planar e conexa. Como o procedimento `DECIMATE` sempre devolve uma malha conexa, a conectividade e planaridade da malha original é preservada em todos os níveis da pirâmide multi-escala. O algoritmo `MSMESHINTEGRATE` é portanto imune a problemas causados por perda prematura de conectividade, e funciona mesmo na presença de corredores estreitos.

O procedimento `DECIMATE` primeiro particiona $\mathcal{V}G$ em um conjunto R de vértices a serem removidos, e um conjunto K de vértices a serem mantidos. O conjunto R é um subconjunto maximal de $\mathcal{V}G$ de vértices independentes (não vizinhos) com grau 6 no máximo. Para isso, o procedimento utiliza um atributo *mark* para cada vértice, que pode possuir 3 possíveis estados: `REMOVE`, `KEEP` e `BLANK`. Inicialmente todos os vértices são marcados como `BLANK`. Para cada grau k , de 1 a 6, o procedimento varre sequencialmente todos os vértices que estão com estado `BLANK` em ordem arbitrária. Quando um vértice de grau k é encontrado, este é marcado com `REMOVE` e todos os seus vizinhos são marcados com `KEEP`. No final, o conjunto R consiste em todos os vértices que estão marcados como `REMOVE`, e K de todos os marcados `KEEP` ou `BLANK`.

Após definir os conjuntos K e R , os vértices em R são removidos de G . Toda vez que um vértice u é removido, as arestas incidentes a u são removidas também. Se u possui

grau 1, nenhum passo adicional é necessário. Se u possui grau maior ou igual a 2, novas arestas são adicionadas a G' , conectando os vizinhos de u . Note que todos estes vizinhos estão em K e portanto serão sempre vértices de G' . As extremidades, pesos e diferenças das novas arestas são escolhidos de tal forma que a solução $z'[v]$ para a malha G' seja o mais próxima possível da solução $z[v]$ para cada vértice $v \in K$.

Mais precisamente, seja k o grau de u em G ; sejam e_0, e_1, \dots, e_{k-1} as arestas incidentes a u , saindo deste vértice, em ordem anti-horária; e sejam v_0, v_1, \dots, v_{k-1} os vértices de destino correspondentes. Seja w_i o peso da aresta e_i e d_i a sua diferença. É fácil mostrar que a solução z' para G' seria um subconjunto de uma solução z para G se, para cada par i, j , adicionássemos uma aresta $e'_{i,j}$ de v_i para v_j com o seguintes atributos:

$$d'_{ij} = d_j - d_i \quad w'_{ij} = \frac{w_i w_j}{w_t} \quad (8.11)$$

onde w_t é a soma dos pesos w_0, w_1, \dots, w_{k-1} . A esta operação — remoção de u , remoção de todos os arcos e_i incidentes a u , e adição de todas os arcos e'_{ij} entre vizinhos de u — damos o nome de *troca estrela-clique*. Em particular, se o vértice u tem grau $k = 2$, a troca adicionará apenas um par de novas arestas opostas e'_{01} e e'_{10} . Se o grau k é 3, serão criados apenas os arcos e_{01} , e'_{12} , e'_{02} e seus opostos. Em ambos casos, a planaridade da malha é preservada.

Entretanto se o grau k é maior ou igual a 4, a adição de todas as $k(k-1)$ arestas $e'_{i,j}$ tornaria G' não planar e interferiria severamente na eficiência do algoritmo. Portanto, quando $k \geq 4$ adicionamos apenas as arestas $e'_{i,i+1}$ que conectam os vértices sucessivos v_i e v_{i+1} , para $i \in \{0, 1, \dots, k-1\}$ em um ciclo, assim como suas arestas opostas (onde todos os índices são calculados módulo k). Chamamos esta operação de *troca estrela-ciclo*. As diferenças $d'_{i,i+1}$ destas arestas são dadas pela fórmula (8.11), isto é:

$$d'_{i,i+1} = d_{i+1} - d_i \quad (8.12)$$

Os pesos $w'_{i,i+1}$, por outro lado são dados por fórmulas distintas, de acordo com o grau k , listadas na tabela 8.1.

Tabela 8.1: Fórmulas para o peso w'_{01} da nova aresta $e'_{01} = (v_0, v_1)$ criada pela troca estrela-ciclo, para cada grau k . As mesmas fórmulas são válidas para qualquer outra aresta $e'_{i,i+1}$, exceto pelos índices que são incrementados por i módulo k .

k	w'_{01}
2	$w_0 w_1 / w_t$
3	$0.5(w_0 w_1 + w_1 w_2) / w_t$
4	$(w_0 w_1 + 0.5(w_0 w_2 + w_1 w_3)) / w_t$
5	$(w_0 w_1 + 1.1690(w_2 w_4 + w_0 w_2 + w_1 w_4)) / w_t$
6	$(w_0 w_1 + 2w_5 w_2 + 1.5(w_5 w_1 + w_0 w_2)) / w_t$

Ao contrário da troca estrela-clique, a troca estrela-ciclo não garante que as alturas determinadas pela malha G' sejam exatamente iguais às determinadas pela malha G . Na verdade, não é possível obter a equivalência exata adicionando apenas um subconjunto das arestas e'_{ij} do clique, se os pesos e as diferenças forem calculados apenas por fórmulas locais (i.e. dependentes apenas das arestas e_i). Para escolher valores de d_i e w_i que mantêm os valores da solução z , seria necessário analisar a malha G inteira e resolver o problema da integração.

Entretanto, nossos experimentos mostraram que adicionando apenas as $2k$ arestas do ciclo, com os pesos da tabela 8.1, a solução z' da malha G' tem os componentes de baixa frequência corretos da solução z de G . Isto implica que o erro é altamente localizado, e pode ser removido por poucas iterações de Gauss-Seidel.

A troca estrela-ciclo pode criar arestas paralelas. Estas podem ser tanto novas arestas adicionadas por trocas estrela-ciclo distintas, quanto arestas de G que possuem ambas extremidades em K e não foram removidas. Portanto, após realizar todas as trocas estrela-ciclo, o procedimento DECIMATE substitui todas os grupos de arestas com mesma origem e mesmo destino por uma única aresta equivalente. Em particular, se as arestas e' e e'' possuem a mesma origem e destino, elas são substituídas por uma única aresta e com mesmas extremidades, com os atributos

$$w[e] = w[e'] + w[e''] \quad d[e] = \frac{w[e']d[e'] + w[e'']d[e'']}{w[e'] + w[e'']} \quad (8.13)$$

e este processo é repetido até eliminar todas as arestas paralelas. É fácil perceber que este processo não modifica a solução z definida pelas equações (8.4).

8.3.2 Interpolação

Uma vez que a solução z' foi obtida a partir da malha reduzida G' (passo 6), ela é convertida em uma estimativa inicial de z para a malha completa G , pelo procedimento

INTERPOLATE (passo 7). Inicialmente, para cada vértice v em K (que é compartilhado pelas duas malhas), nós fazemos a atribuição $z[v] \leftarrow z'[v]$. Então, para cada vértice u em R (que existe apenas em G), nós calculamos $z[u]$ pela equação de equilíbrio de vértices (8.4). Note que cada vizinho $v \in G[u]$ pertence à K , e portanto sua altura $z[v]$ já está definida neste ponto.

8.3.3 Ajuste iterativo

A estimativa inicial z calculada por INTERPOLATE é utilizada como chute inicial para o procedimento SOLVESYSTEM (passo 8) utilizando o algoritmo de Gauss-Seidel. Cada iteração do mesmo examina cada vértice $u \in \mathcal{V}G$ e utiliza a equação (8.4) para re-calcular sua altura $z[u]$ a partir das alturas atuais $z[v]$ de seus vizinhos. O procedimento termina após um número específico κ de iterações; ou então quando a variação da altura $z[u]$ de uma iteração para outra for menor que uma tolerância especificada ε , para todo vértice u , se isto ocorrer antes.

Note que, para cada nível de recursão, o limite de iterações κ é aumentado por um fator $1/\sqrt{\beta}$, e a tolerância é reduzida por $\sqrt{\beta}$ (passo 6); onde β é o fator de redução do tamanho da malha obtido por DECIMATE (passo 4).

8.4 Análise do algoritmo

8.4.1 Corretude

A troca estrela-ciclo e colapso de arestas paralelas preservam planaridade e conectividade do grafo, de maneira que em todas as chamadas recursivas de MSMESHINTEGRATE a malha G satisfaz estas condições. Além disso, como as equações de vértice (8.4) são dominadas pela diagonal, a aplicação do algoritmo na escala 0 com valores adequados de κ e ε converge para a solução única $z = z^{(0)}$ dessas equações, independente da estimativa inicial obtida da malha dizimada $G^{(1)}$.

8.4.2 Tempo e espaço

A eficiência do algoritmo depende do fator de redução β obtido pela rotina DECIMATE e do número de interações de Gauss-Seidel necessários em cada nível. Sejam $N = \#\mathcal{V}G$, $N_k = \#\mathcal{V}G^{(k)}$, $M = \#\mathcal{E}G$, $M_k = \#\mathcal{E}G^{(k)}$. Seja $\hat{\beta}$ o maior valor de β observado durante a redução. Se $\hat{\beta} < 1$ então a escala máxima m será $\log_{1/\hat{\beta}} N = O(\log N)$ e o total N_{tot} de vértices de todas as malhas será no máximo $N/(1 - \hat{\beta}) = O(N)$.

A planaridade da malha G garante um limite superior para $\hat{\beta}$ menor que 1. Como G é um grafo planar simples, sabe-se que $M \leq 6N$, que $M_k \leq 6N_k$ e que G possui no mínimo $N/7$ vértices de grau menor ou igual a 6. As mesmas conclusões valem para $G^{(k)}$, $N^{(k)}$, $M^{(k)}$; A partir destes fatos obtém-se um limite teórico para o fator $\hat{\beta} \leq 41/42 \approx 0.976$ [62]. Portanto M é no máximo $20 \log_2 N$ e N_{tot} é no máximo $42N$. Na prática, entretanto, o fator de redução β é aproximadamente 0.6 na maior parte dos casos, o que dá $M \approx 1.4 \log_2 N$ e $N_{\text{tot}} \approx 2.5N$.

A quantidade de memória exigida pelo algoritmo é dominada pela estrutura de dados que representa cada malha $G^{(k)}$. Uma representação simples que é suficiente para nossos propósitos consiste de uma *tabela de arestas* com $2M_k$ entradas, cada uma contendo o destino, o peso e a diferença de uma aresta direcionada $G^{(k)}$, ordenadas pelo vértice de origem; e uma *tabela de índices* com N_k entradas, que armazena, para cada vértice v , o índice da primeira aresta na tabela de arestas com origem v . O espaço total de armazenamento é, portanto, no máximo $N_k + 2 \times 3M_k \leq 19N_k$ palavras para cada malha $G^{(k)}$, e no máximo $19N_{\text{tot}} \approx 47.5N$ palavras para todas as malhas da pirâmide.

A condição de planaridade também garante que o algoritmo de dizimação termina em tempo $O(N_k + M_k) = O(N_k)$ para cada nível k . Portanto toda pirâmide é construída em tempo $O(N_{\text{tot}}) = O(N)$. O tempo exigido para uma iteração do método de Gauss-Seidel no nível k é $\Theta(N_k + 2M_k) = \Theta(N_k)$, e o número máximo de iterações nesse nível $\kappa^{(k)}$ é $\kappa/\hat{\beta}^{k/2}$. O tempo gasto em um nível k é $(N\hat{\beta}^k)(\kappa/\hat{\beta}^{k/2}) = N\hat{\beta}^{k/2}$ perfazendo o tempo total de $(N/(1 - \sqrt{\hat{\beta}})) = O(N)$.

8.4.3 Velocidade de convergência

Assim como no algoritmo do capítulo 7, este algoritmo explora o fato de que a iteração de Gauss-Seidel converge rapidamente se o erro consiste principalmente de erros de alta frequência espacial. Quando a malha é dizimada, os componentes de alta frequência dos dados são perdidos, enquanto os componentes de baixa frequência tem seu comprimento de onda reduzido pela metade. Portanto a solução recursivamente calculada $z^{(k+1)}$ do problema, após ser expandida para escala anterior $z^{(k)}$ será correta principalmente em componentes de frequência baixa; apenas detalhes pequenos estarão faltando. Estes detalhes serão recuperados após um número pequeno de iterações de Gauss-Seidel, largamente independente de $N^{(k)}$. O processo recursivo é rápido porque cada componente espectral do mapa de altura é calculado na escala onde sua frequência espacial é pequena.

Infelizmente esta explicação intuitiva não é fácil de formalizar, muito menos demonstrar teoricamente, pois necessita de uma formulação de “frequência” em termos de grafos, que está além do escopo desta tese. Porém, os testes experimentais mostram que a convergência é obtida após poucas iterações, mesmo com instâncias que fazem outros métodos

multi-escala falharem.

8.5 Testes

Nesta seção, nós comparamos o custo e a precisão do nosso integrador multi-escala para malhas irregulares (MG) com outros métodos publicados. Consideramos apenas os algoritmos baseados na formulação de Poisson, uma vez que estes são os únicos que fornecem resultados precisos e podem lidar com descontinuidades no mapa de gradiente. Isto é, utilizamos o algoritmo Estimadores-M (ME) e Difusão com Transformação Afim (AT) de Agrawal [6] para algoritmos de solução direta e o algoritmo multi-escala (MS) descrito no capítulo 7. Veja a tabela 8.2.

Tabela 8.2: Métodos de integração testados.

Sigla	Descrição	Tipo	Utiliza w
AT	Difusão com Transformação Afim [6, 4]	Poisson (sol. direta)	Sim
ME	Estimadores-M [6, 4]	Poisson (sol. direta)	Sim
MS	Multi-escala com pesos	Poisson multi-escala iter.	Sim
MG	Multi-escala com malha ponderada	Poisson multi-escala iter.	Sim

Os algoritmos de Agrawal, implementados pelo autor em Matlab, foram modificados como descrito na seção 7.6. Nossos algoritmos, foram implementados em C, compilados e testados na plataforma Linux. O número máximo κ de iterações de Gauss-Seidel na escala 0 foi fixado em 50 para MS e 20 para MG.

8.5.1 Conjunto de testes

Utilizamos 6 conjuntos de dados de teste. Cinco deles (`spdome`, `cbabel`, `cbramp`, `mixwav` e `cpiece`) foram obtidos amostrando funções matemáticas, de maneira similar ao descrito na seção 7.6. O conjunto de teste real `dtbust` foi obtido a partir de um modelo geométrico tridimensional de um busto humano, capturado pelo mesmo *scanner* tridimensional usado para o conjunto `dtface` da seção 7.6.

Os conjuntos `spdome`, `cbramp` e `mixwave` são análogos aos utilizados na seção 7.6. O conjunto `cbabel` é uma rampa espiral com degraus verticais entre as voltas das rampas onde supomos que não há informação de gradiente. O conjunto `dtbust` é um mapa de gradiente de um busto humano calculado pela metodologia descrita na seção 7.6; ele tem regiões extensas sem dados e um corredor estreito ligando a cabeça ao torso. Finalmente, `cpiece` consiste de três regiões planas — uma plataforma circular, um fosso em anel, e uma região exterior conectadas por rampas estreitas. Os mapas de peso de todos

estes conjuntos de teste foram criados manualmente com um editor de imagens. Veja a figura 8.4.

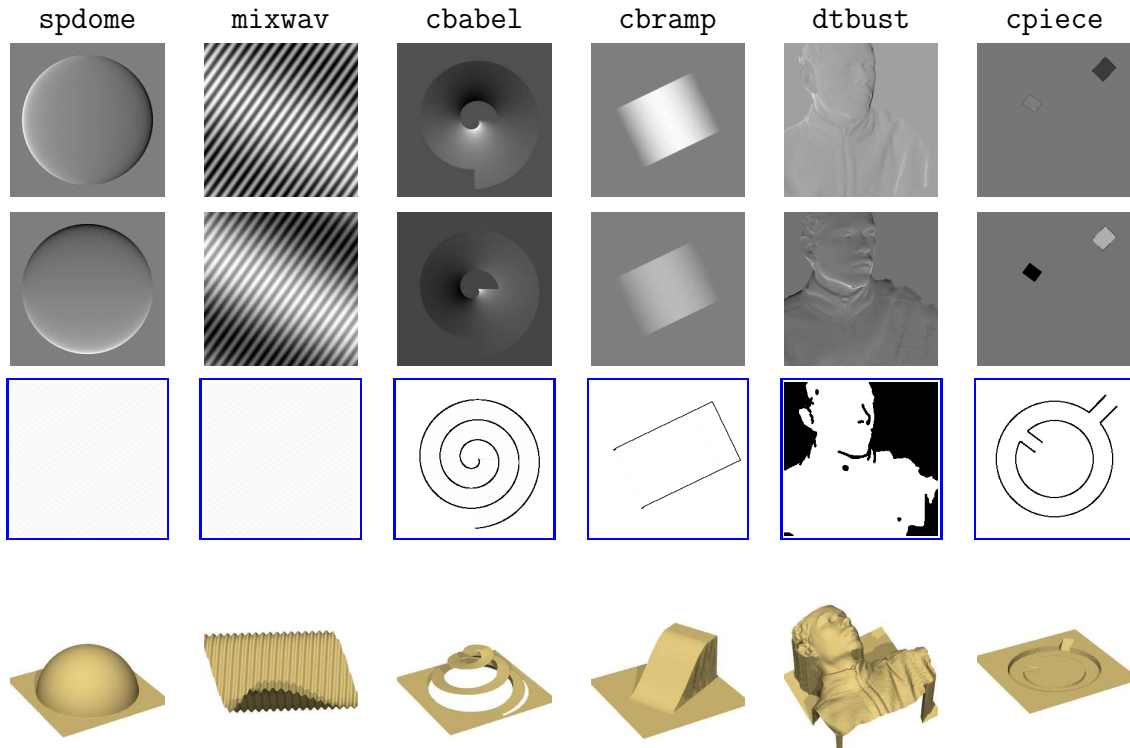


Figura 8.4: Os conjuntos de testes utilizados, mostrando os mapas de gradiente f (topo), g (segunda fileira), os mapas de peso w (terceira fileira) e mapa de alturas correto Z (embaixo).

Como na seção 7.6, fizemos também outra série de testes onde cada elemento de f e g foi misturado com 30% de ruído aleatório (gaussiano) e independente.

8.5.2 Precisão e robustez

Ao comparar a precisão e robustez dos algoritmos, utilizamos a mesma metodologia da seção 7.6.2. Os mapas de altura calculados pelos quatro algoritmos e os respectivos mapas de erro podem ser vistos nas figuras 8.5, 8.6 (para mapas de gradiente sem ruído) e 8.7, 8.8 (para testes com ruído gaussiano adicionado). O erro médio absoluto e e o erro relativo e/R definidos como na seção 7.6, estão resumidos nas tabelas 8.3, 8.4, 8.5 e 8.6.

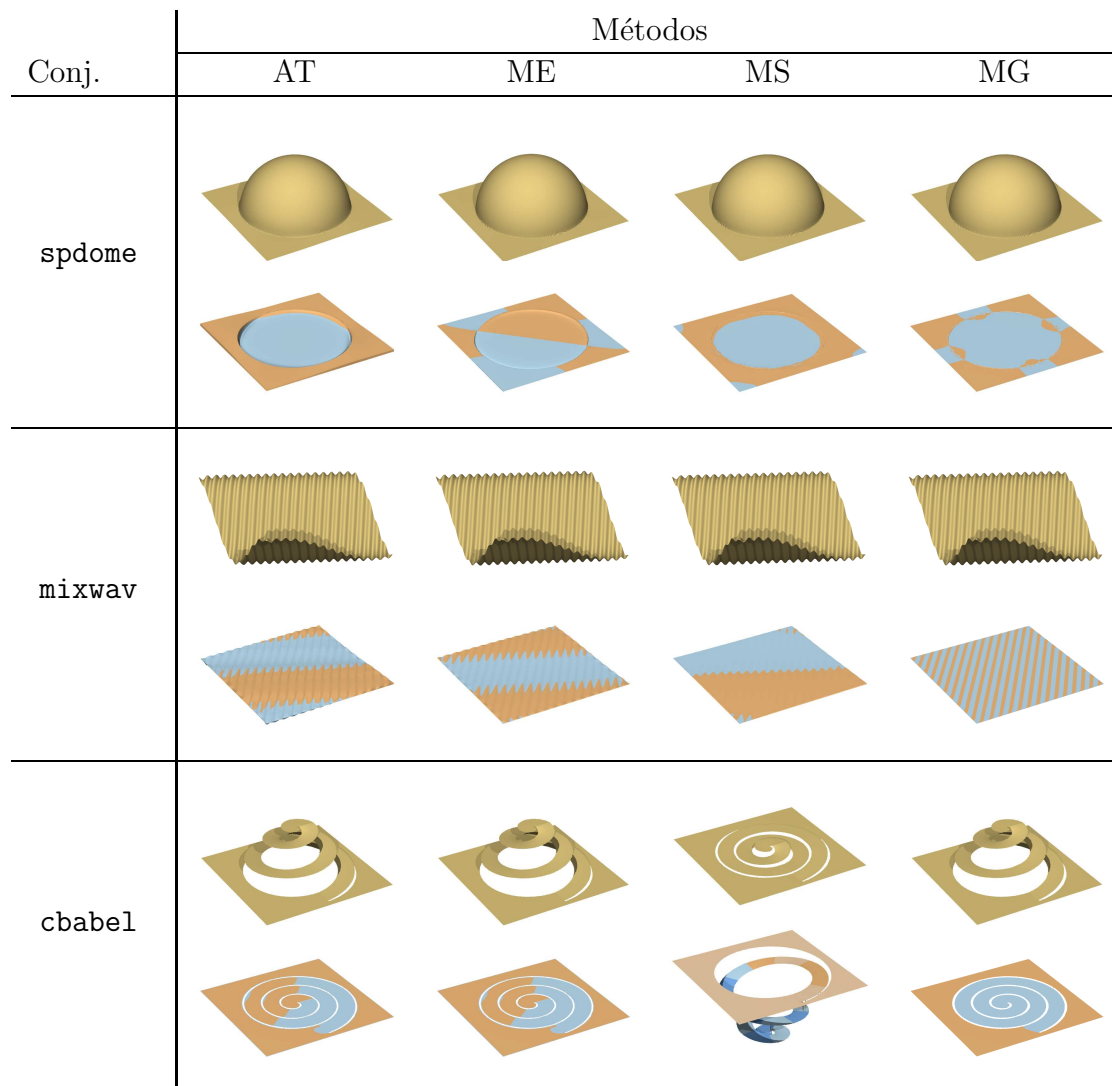


Figura 8.5: Mapas de altura calculados pelos quatro métodos e mapas de erro para os conjuntos de teste `spdome`, `mixwav` e `cbabel` sem adição de ruído. Tons de azul e laranja nos mapas de erro indicam que a altura calculada está respectivamente abaixo ou acima da altura correta.

Tabela 8.3: Erros relativos de RMS de cada método em relação ao mapa de referência, para os testes `spdome`, `mixwav` e `cbabel` sem adição de ruído.

	spdome		mixwav		cbabel	
Método	e	e/R	e	e/R	e	e/R
AT	1.82	5.2%	0.89	2.3%	0.02	0.1%
ME	0.67	1.9%	0.46	1.2%	0.02	0.1%
MS	0.19	0.5%	0.36	0.9%	25.31	134.8%
MG	0.04	0.1%	0.02	0.0%	0.03	0.1%

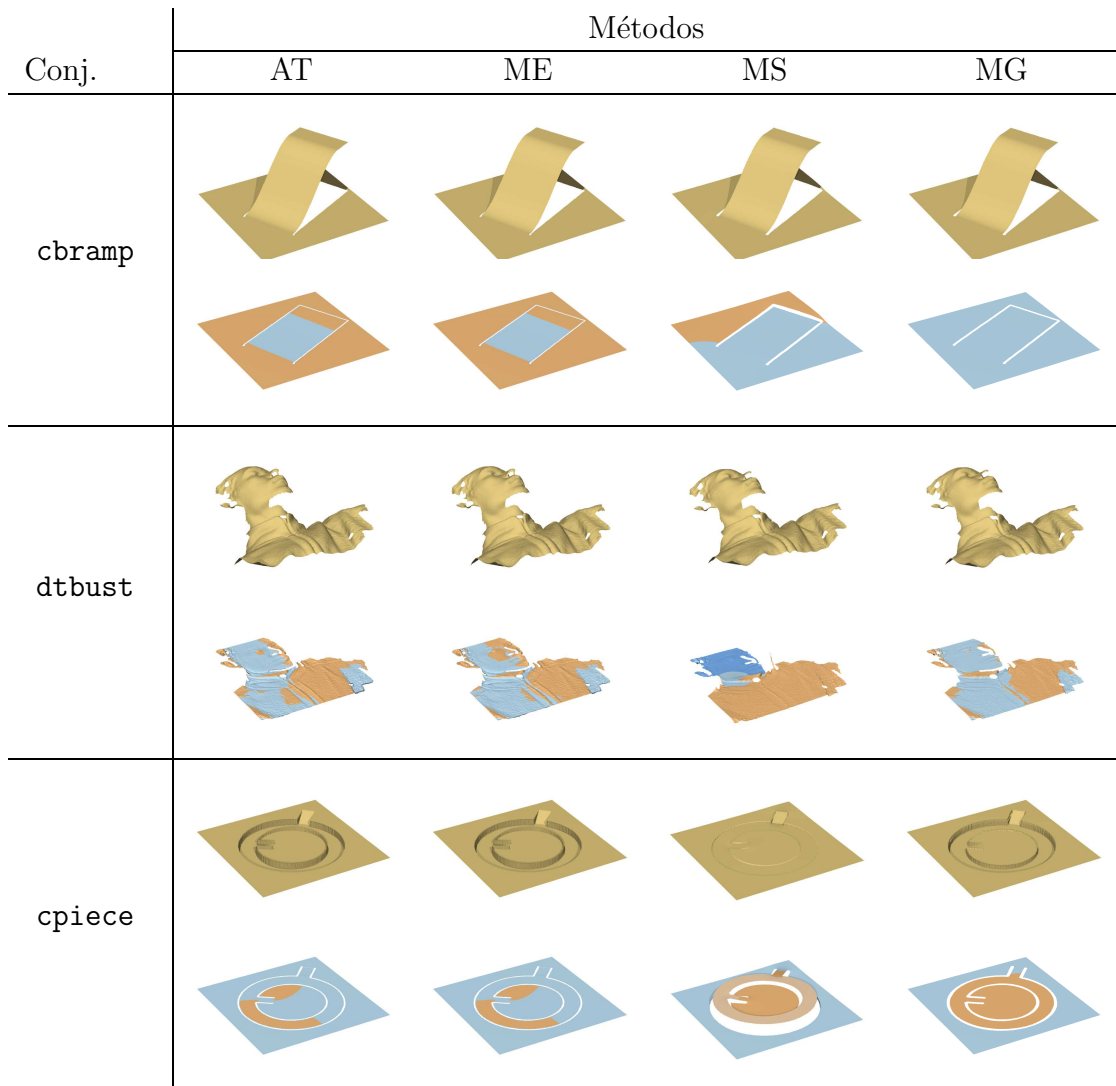


Figura 8.6: Mapas de altura calculados pelos quatro métodos e mapas de erro para os conjuntos de teste **cbramp**, **dtbust** e **cpiece** sem adição de ruído.

Tabela 8.4: Erros relativos de RMS de cada método em relação ao mapa de referência, para os testes **cbramp**, **dtbust** e **cpiece** sem adição de ruído.

	cbramp		dtbust		cpiece	
Método	e	e/R	e	e/R	e	e/R
AT	0.14	0.4%	0.64	2.5%	0.02	0.3%
ME	0.14	0.4%	0.55	2.2%	0.02	0.3%
MS	0.89	2.6%	2.99	12.4%	4.74	123.0%
MG	0.00	0.0%	0.41	1.6%	0.01	0.3%

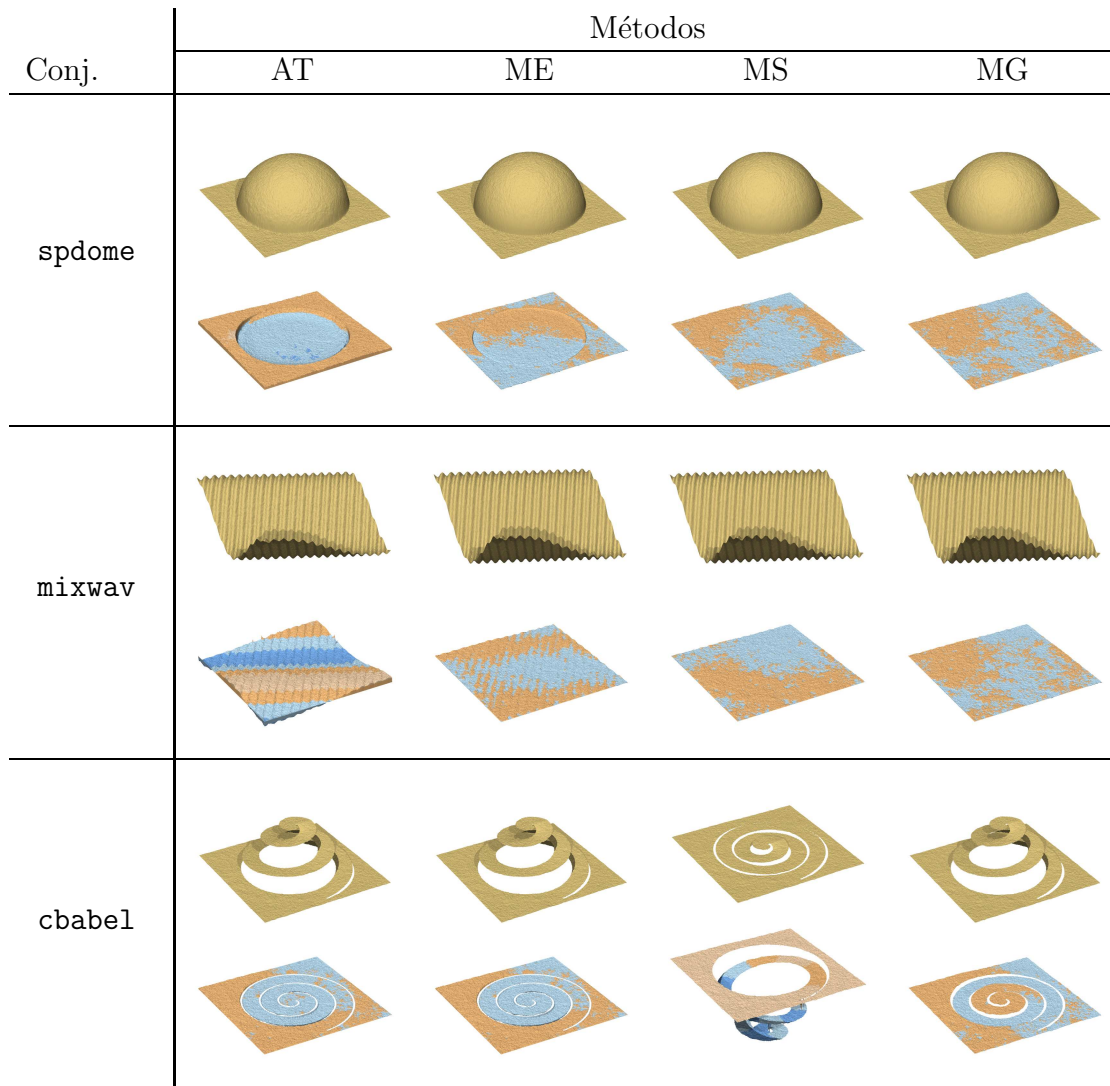


Figura 8.7: Mapas de altura calculados para os quatro métodos e mapas de erro para os conjuntos de teste **spdome**, **mixwav** e **cbabel**, com 30% de ruído gaussiano adicionado.

Tabela 8.5: Erros relativos de RMS de cada método em relação ao mapa de referência, para os conjuntos de teste **spdome**, **mixwav** e **cbabel**, com 30% de ruído gaussiano adicionado.

	spdome		mixwav		cbabel	
Meth.	e	e/R	e	e/R	e	e/R
AT	3.30	9.8%	4.75	13.0%	0.80	3.0%
ME	0.71	2.0%	0.51	1.3%	0.86	3.3%
MS	0.40	1.1%	0.44	1.1%	25.33	134.9%
MG	0.34	1.0%	0.34	0.9%	1.02	3.9%

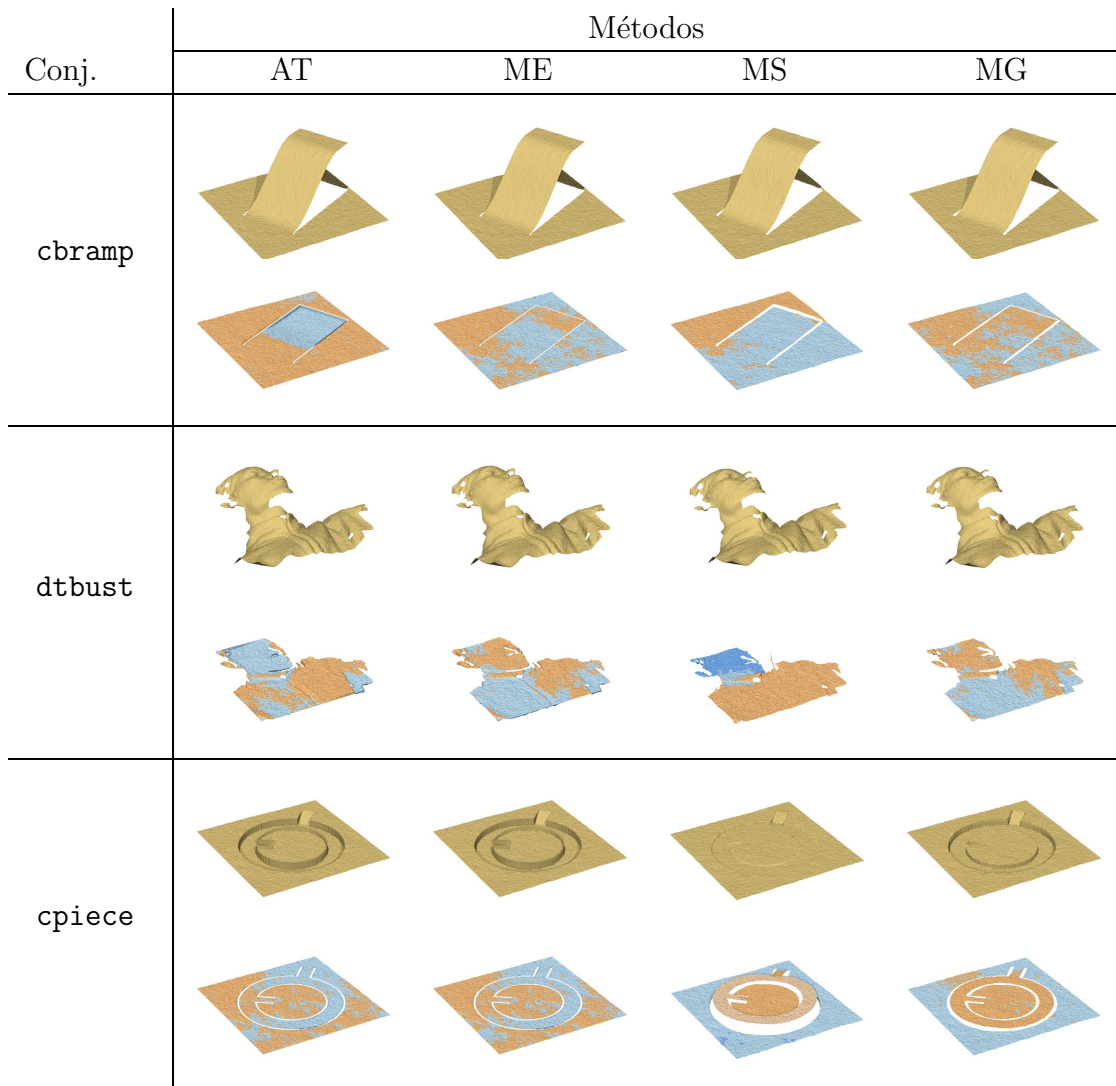


Figura 8.8: Mapas de altura calculados pelos quatro métodos e seus mapas de erro, para os conjuntos de teste **cbramp**, **dtbust** e **cpiece**, com 30% de ruído gaussiano adicionado.

Tabela 8.6: Erros relativos de RMS de cada método em relação ao mapa de referência, para os conjuntos de teste **cbramp**, **dtbust** e **cpiece**, com 30% de ruído gaussiano adicionado.

	cbramp		dtbust		cpiece	
Meth.	e	e/R	e	e/R	e	e/R
AT	1.53	4.5%	1.22	4.9%	0.50	9.2%
ME	0.43	1.2%	0.71	2.8%	0.49	9.1%
MS	0.82	2.4%	2.98	12.4%	4.67	121.3%
MG	0.39	1.1%	0.61	2.4%	0.88	17.6%

As tabelas indicam que a precisão do nosso método MG é equivalente ou melhor ao dos outros três algoritmos. Note que o método MS falha nos conjuntos `cbabel` e `cpiece` devido a perda de conectividade nas escalas menores da pirâmide. O mesmo problema ocorre, em menor grau, no conjunto `dtbust`.

8.5.3 Custo

Para avaliar a eficiência do nosso método, nós medimos o tempo e a quantidade de memória usados na integração de dois conjuntos de teste (`spdome` e `dtface`). As cenas foram amostradas em grades de diversos tamanhos, de 64×64 a 512×512 . Assim como na seção 7.6, para os métodos de Agrawal (AT, ME) consideramos apenas o espaço necessário para armazenar o sistema e o tempo para resolvê-lo; enquanto que para os métodos multi-escala (MS, MG) consideramos o espaço necessário para armazenar todas as malhas $G^{(k)}$ bem como o tempo total para construí-las e para efetuar as iterações de Gauss-Seidel em todos os níveis.

Os custos absolutos de algoritmos diferentes não podem ser comparados diretamente, devido a diferenças substanciais nas implementações e das plataformas de execução. Portanto analisaremos apenas o crescimento dos mesmos em relação ao tamanho do problema. As figuras 8.9, 8.10 e a tabela 8.7 e 8.8 apresentam os resultados dos testes.

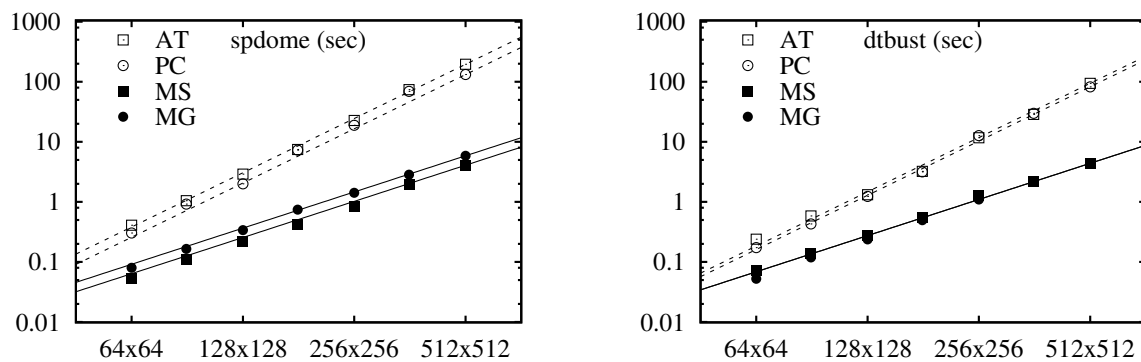


Figura 8.9: Gráficos log-log do tempo de execução de dois métodos de resolução direta (PC,AT) e nossos métodos multi-escala(MS,MG), em segundos, em dois conjuntos representativos de teste e em diversas resoluções. As linhas retas são funções ajustadas $O(N)$ (cheia) e $O(N^{1.5})$ (pontilhada).

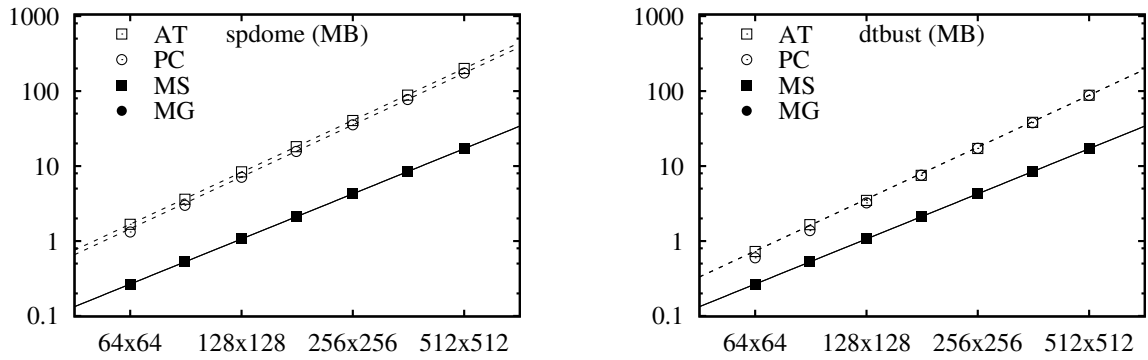


Figura 8.10: Gráficos log-log de uso de memória para a matriz de sistema \mathbf{M} e seu fator triangular \mathbf{U} (se aplicável), em megabytes. As linhas retas são as funções ajustadas $O(N)$ (cheia) e $O(N^{1.15})$ (pontilhada).

Tabela 8.7: Custos de tempo e memória para os métodos PC, AT, MS e MG para o conjunto de teste `spdome`.

spdome								
	tempo				memória			
N	AT	PC	MS	MG	AT	PC	MS	MG
$64 \times 64 = 4096$	0.41	0.30	0.05	0.08	1.7	1.3	0.2	0.8
$90 \times 90 = 8100$	1.05	0.92	0.11	0.17	3.6	3.0	0.5	1.7
$128 \times 128 = 16384$	2.89	1.99	0.22	0.34	8.4	7.1	1.0	3.3
$180 \times 180 = 32400$	7.46	7.27	0.43	0.74	18.1	15.6	1.9	6.6
$256 \times 256 = 65536$	22.67	18.82	0.82	1.42	40.5	35.5	3.9	13.4
$360 \times 360 = 129600$	74.03	69.21	1.94	2.84	88.1	77.2	7.8	26.4
$512 \times 512 = 262144$	195.44	131.84	4.08	5.85	200.2	174.1	15.7	53.5

Tabela 8.8: Custos de tempo e memória para os métodos PC, AT, MS e MG para o conjunto de teste `dtbust`.

dtbust								
	tempo				memória			
N	AT	PC	MS	MG	AT	PC	MS	MG
$64 \times 64 = 4096$	0.24	0.17	0.07	0.05	1.7	1.3	0.2	0.8
$90 \times 90 = 8100$	0.58	0.43	0.14	0.12	3.6	3.0	0.5	1.7
$128 \times 128 = 16384$	1.32	1.23	0.28	0.24	8.4	7.1	1.0	3.3
$180 \times 180 = 32400$	3.22	3.16	0.55	0.50	18.1	15.6	1.9	6.6
$256 \times 256 = 65536$	11.71	12.66	1.27	1.10	40.5	35.5	3.9	13.4
$360 \times 360 = 129600$	28.60	29.81	2.17	2.17	88.1	77.2	7.8	26.4
$512 \times 512 = 262144$	94.23	82.30	4.38	4.40	200.2	174.1	15.7	53.5

Estes resultados mostram que o tempo e de execução e a quantidade de memória usada crescem como $\Theta(N)$ para os métodos multi-escala (MS, MG), mas como $\Theta(N^{1.15})$ e $\Theta(N^{1.5})$ para os métodos de resolução direta (AT, PC).

Nosso método MG requer mais memória e tempo que o método multi-escala simples devido ao armazenamento da malha e sua dizimação. Portanto, podemos afirmar que para mapas suficientemente grandes, o melhor método (mais preciso e mais eficaz) é MS, se o mapa de gradiente for conexo em todas as escalas, e MG caso o contrário.

Parte III

Aplicação

Capítulo 9

Captura da face humana

A captura da geometria tridimensional da face humana (*captura facial*) tem diversas aplicações tais como identificação biométrica, indexação automática de imagens e vídeos, cirurgia plástica, detecção de expressões e modelagem para computação gráfica. Discutiremos neste capítulo a aplicação dos nossos métodos de estéreo fotométrico para captura facial em um dispositivo desenvolvido com esta finalidade, do projeto *PhotoFace* da University of West England. Nossos algoritmos melhoraram muito a qualidade dos resultados em comparação aos métodos usados originalmente.

9.1 Trabalhos anteriores

O método mais comum e direto para captura facial é usando um *scanner* tridimensional. Os *scanners* mais precisos determinam a profundidade de cada ponto de uma cena medindo o tempo de percurso de pulsos *laser* [93]. Estes dispositivos são bastante caros e exigem que a face permaneça estática por um intervalo de tempo considerável. Outros métodos comumente usados para este fim, mas com resolução e aplicabilidade mais limitada, incluem luz estruturada [89], análise de silhuetas [15] e métodos híbridos [2]. Todos estes métodos exigem hardware especializado e volumoso. A técnica de estéreo geométrico tradicional (por disparidade) é pouco usada isoladamente para captura de faces, possivelmente pela dificuldade de encontrar pontos salientes nas partes lisas do rosto.

Yuille *et al.* [113] aplicaram seu método (estéreo fotométrico por DVS com 7 termos) para captura facial. Seus resultados demonstraram a viabilidade de EFIV para esta aplicação.

Schindler [87] implementou o método de Yuille (DVS com 3 termos) para captura facial, usando um monitor LCD, exibindo padrões variados, como fonte de luz variável. Embora seu método seja rápido o suficiente para ser utilizado em tempo real, a precisão

é bastante limitada, o suficiente apenas para video-chats e aplicações similares.

Georgiades *et al.* [40] procuraram obter resultados mais precisos usando um conjunto de 64 fontes de luz pontuais controladas por computador, afixadas a uma armação esférica envolvendo a pessoa. Nessa implementação, a captura das 64 imagens levava em torno de 2 segundos. Para o cálculo das normais, eles utilizaram o método da DVS com 3 termos, excluindo pixels sombreados ou saturados da matriz P por limiares, usando 7 ou mais dessas imagens na reconstrução. Os resultados obtidos foram muito melhores do que os de Yuille *et al.*

Em 2010, Gary Atkinson e Mark Hansen desenvolveram o dispositivo *Photoface* para captura facial de pessoas em movimento por meio de estéreo fotométrico para fins de identificação biométrica [48]. Este trabalho será descrito em detalhes na seção 9.2. O *Photoface* utiliza um número pequeno (quatro) de luzes (unidades de *flash*) essencialmente pontuais com posição conhecida, mas com intensidades pouco uniformes e parcialmente desconhecidas; por essa razão as normais obtidas tinham erros sistemáticos significativos.

Outro método para captura facial em tempo real de pessoas em movimento foi desenvolvido por Vogiatzis *et al.* [102] Eles utilizam estéreo fotométrico multi-espectral como descrito na seção 2.10.5 com fontes pontuais de intensidades conhecidas. Eles calculam as direções das fontes por meio de uma estimativa inicial da geometria da cena, obtida através de técnicas de estéreo geométrico [34]. Eles então determinam a normal de cada pixel supondo acabamento lambertiano pelo método algébrico da seção 2.8.1. Nos pixels onde esta normal é incompatível com a forma e o movimento estimados da face, eles recalculam a normal por otimização não-linear supondo acabamento parcialmente lustroso (modelo de Phong) e parcialmente lambertiano.

9.1.1 Considerações gerais

O acabamento da pele humana não é exatamente lambertiano; ela apresenta reflexos lustrosos de intensidade e grau de polimento bastante variáveis, de ponto a ponto, que às vezes afetam uma fração significativa dos pixels da imagem. Além disso, os tecidos da face são parcialmente translúcidos de modo que parte da luz é espalhada por alguns milímetros abaixo da superfície. O estéreo fotométrico não é aplicável à pupila dos olhos, que é lustrosa e tem albedo efetivo quase nulo. Variações de tom de pele (lábios, manchas) podem ser tratadas como variações de albedo assim como barba curta. Por outro lado, as normais obtidas nas partes cobertas por cabelo e barba mais longos não são confiáveis.

A face humana, vista de posição frontal, pode ser considerada uma superfície suave, sem descontinuidades de profundidade exceto nas narinas e nas orelhas. Estas descontinuidades precisam ser marcadas no mapa de pesos do integrador, caso contrário o mapa de alturas ficará extremamente distorcido.

Apesar dessas complicações, verificamos que o estéreo fotométrico dá resultados bastante satisfatórios, mesmo com gabaritos lambertianos, desde que o número de imagens seja suficientemente grande. Para confirmar esta tese, aplicamos nosso método a um conjunto de 20 imagens do banco de faces B produzido por Georghiades *et al.* [40]. As imagens deste banco, com 640×480 pixels, foram obtidas por uma câmera com resposta linear e estão bem alinhadas. Veja a figura 9.1.

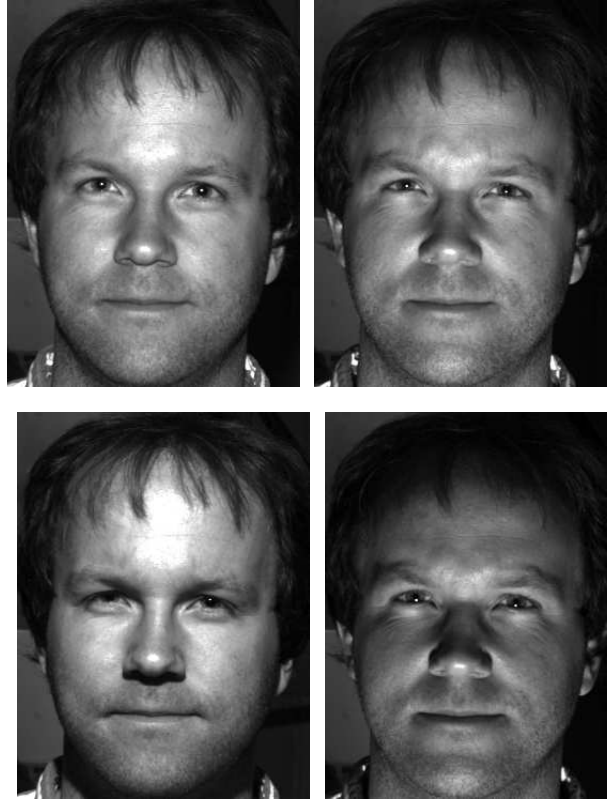


Figura 9.1: Quatro das 64 imagens de um mesmo indivíduo sob iluminações distintas, extraídas da base de dados de Georghiades *et al.* [40].

A figura 9.2(a) mostra as normais calculadas pelo método BGRID do capítulo 5 utilizando $m = 20$ imagens do indivíduo da figura 9.1. A tabela de assinaturas foi gerada amostrando a função vetor de tonalização \mathbf{L} de uma superfície de acabamento lambertiano com m iluminações unidirecionais (cujas direções \vec{u}_i estão especificadas na base de dados). A figura 9.2(c) mostra o mapa de alturas calculado pelo integrador multi-escala MSMAPINTEGRATE do capítulo 7 a partir dessas normais. Infelizmente, o banco de Georghiades não possui os mapas de altura corretos, de modo que não é possível avaliar a precisão do resultado.

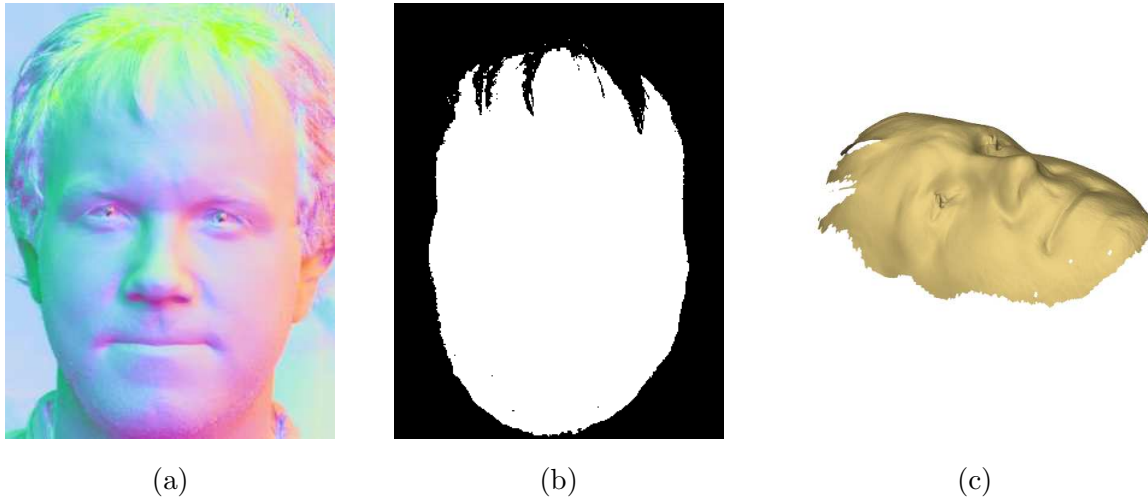


Figura 9.2: Mapa de normais (a) calculado por BGRID a partir de $m = 20$ imagens do indivíduo da figura 9.1, mapa de pesos (b) construído por um editor de imagens, mapa de alturas (c) obtido pelo integrador multi-escala MSMAPITEGRATE com o mapa de peso (b).

9.2 O projeto *Photoface*

O hardware do projeto *Photoface* foi desenvolvido visando a capturar em tempo real, das normais da face de pessoas em movimento, com iluminação ambiente não controlada. Ele consiste em uma armação de alumínio na qual estão montadas uma câmera e quatro lâmpadas *flash*, que são acionadas automaticamente quando uma pessoa chega a uma distância fixa da câmera. Veja a figura 9.3.



Figura 9.3: Dispositivo de captura de normais *Photoface* da UWE.

9.2.1 Captura das imagens

Para garantir o alinhamento das imagens, elas são adquiridas em um intervalo de tempo extremamente curto, com fontes de luz *flash* controladas por computador e uma câmera de alta velocidade. Verificou-se que a taxa de captura mínima necessária para obter quatro imagens adequadas de uma pessoa em movimento era de aproximadamente 150 quadros por segundo. Mais ainda, dada a suposição de intensidade e direção conhecida, é necessário garantir a sincronia precisa do acendimento das luzes e captura das imagens.

As quatro lâmpadas flash (Jessop M100) estão dispostas em retângulo, duas acima e duas abaixo da altura média da face de uma pessoa em pé, a aproximadamente 1 metro na frente da câmera, e a 1 metro da distância da face.

O dispositivo é ativado por um sensor de proximidade por ultrassom, posicionado a um pouco mais de dois metros da câmera, na posição desejada da pessoa. O processamento do sinal do sensor e o acionamento dos flashes e da câmera são feitas por uma placa controladora NI PCI-7811 DIO.

A câmera é uma Basler 504Kc com lentes de 55mm f.56 configurada para produzir imagens de 1280×1024 pixels a 200 quadros por segundo, no formato RGB de 8 bits. As imagens são transmitidas por cabo CameraLink à uma placa de captura PCIe-1429.

A placa de captura e a placa sincronizadora dos *flashes* são conectadas por uma conexão RTSI, e ambas estão conectadas a um computador. Este atua apenas como um “ouvinte,” iniciando o processamento das imagens capturadas ao receber uma notificação da placa controladora. O processo de captura leva em torno de aproximadamente 50 milissegundos; para um observador humano, os quatro disparos parecem simultâneos.

A figura 9.4 mostra uma seqüência de 4 fotos capturadas pelo *Photoface* de um indivíduo caminhando em direção ao dispositivo.

9.2.2 Calibração do dispositivo

Originalmente, o processamento das imagens do *Photoface* começava com uma etapa de eliminação de luz ambiente. Para este fim o dispositivo capturava uma quinta imagem do indivíduo, sem acionar os *flashes*, e esta imagem era então subtraída das outras quatro. Em geral, este termo ambiente era pequeno em comparação às intensidades obtidas com os *flashes*.

A direção \vec{u}_i de cada fonte de luz era determinada previamente por meio de um gabarito lustroso de raio conhecido, colocado na posição aproximada da face. A direção \vec{u}_i era calculada supondo-se que o ponto de intensidade máxima na imagem fosse o ponto onde a normal coincidia com a bissetriz da direção da luz \vec{u}_i e da direção de vista \vec{v} .



Figura 9.4: Quatro imagens de uma pessoa em movimento capturadas pelo *Photoface*

Devido a variações na altura das pessoas e em sua posição lateral no momento de captura, a posição da face nas imagens brutas do *Photoface* é bastante variável. O algoritmo de Lienhart e Maydt [68] é usado para determinar a posição e extensão da face nas imagens. O cálculo das normais era aplicado apenas a essa região relevante (um retângulo de $\approx 400 \times \approx 500$ pixels). Veja a figura 9.5.

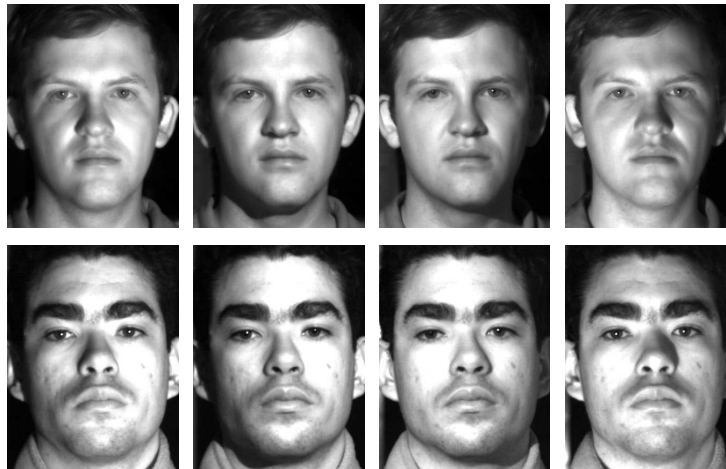


Figura 9.5: Conjuntos de quatro imagens de duas pessoas, capturados pelo dispositivo *Photoface* e recortadas (para fins ilustrativos) de 1280×1024 para 320×420 pixels.

9.2.3 Cálculo das normais

Pela construção do dispositivo, pode-se supor que as fontes são pontuais e têm direção e intensidade conhecida. O algoritmo usado na implementação original escolhia para cada pixel p , as três imagens com maiores intensidades $S_i[p]$. A normal era calculada a partir destes três valores, supondo acabamento lambertiano e luz unidirecional, pelo método da seção 2.8.1. Se o ângulo entre esta normal e a direção da luz \vec{u}_k da imagem S_k de menor intensidade fosse maior que 90° , então o algoritmo considerava que $S_k[p]$ estava afetado por sombras e ignorava esse dado. Caso contrário, o algoritmo recalculava a normal a partir dos quatro valores resolvendo o sistema (2.25) pelo método dos quadrados mínimos (vide seção 2.8.1). Veja a figura 9.6.

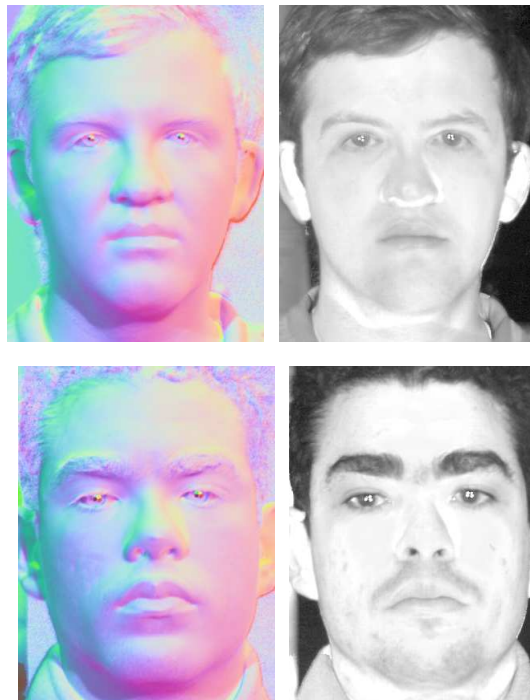


Figura 9.6: Mapas de normais e albedo calculados pelo algoritmo do Photoface para as imagens da figura 9.5.

Na versão original do *Photoface*, o resultado oficial era apenas o mapa de normais. Os autores tentaram obter o mapa de alturas com o integrador de Frankott-Chellapa [37], mas os resultados foram extremamente insatisfatórios, pelas razões descritas no capítulo 7. Veja a figura 9.6 e 9.7.

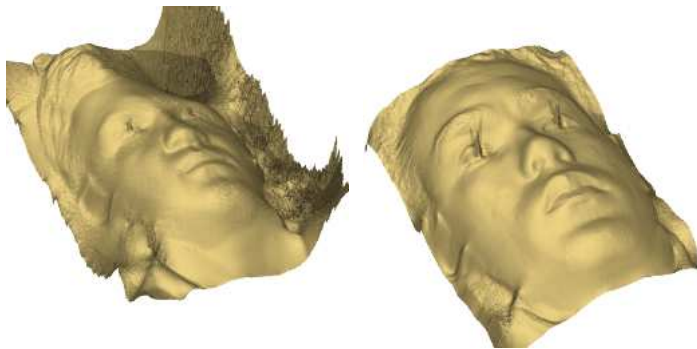


Figura 9.7: Mapas de alturas obtidas integrando os mapas de normais da figura 9.6 pelo método de Frankott-Chellapa [37].

9.3 Melhorando o Photoface

Durante nossa colaboração com a University of West England, aprimoramos os algoritmos do Photoface junto com seus criadores, visando melhorar a qualidade dos resultados.

9.3.1 Calibração da fonte de luz

O algoritmo original do *Photoface* supunha que cada imagem era obtida sob iluminação uniforme por iluminação unidirecional. Entretanto, verificamos que o campo luminoso produzido por cada *flash* tinha variação espacial significativa. Veja a figura 9.8.

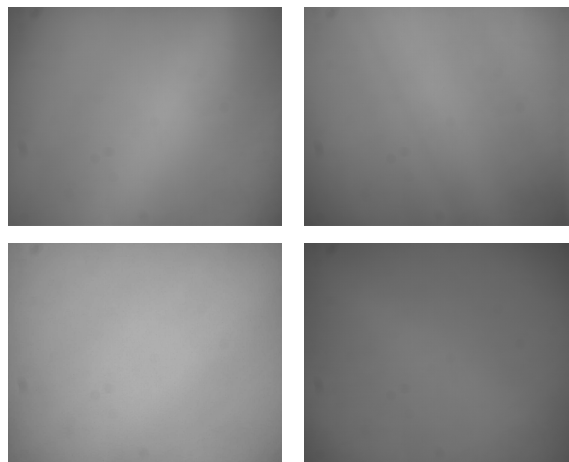


Figura 9.8: Fotos de uma placa plana pintada com tinta *spray* branca fosca, posicionada verticalmente na posição de ativação do sensor, sob as quatro fontes de iluminação do *Photoface*.

Especificamente, a iluminação da área da face era mais intensa na região central da imagem e diminuía na direção das bordas. Este efeito era devido a quatro fatores: *vignetting*

da lente, emissão anisotrópica de luz pela fonte, atenuação da luz inversamente proporcional ao quadrado da distância e variação espacial da direção da luz \vec{u}_i ; sendo que estes dois últimos fatores são devidos à distância relativamente pequena entre a fonte de luz e a cena (cerca de 1m).

Para contornar este problema, construímos uma ferramenta de calibração constituindo de um conjunto de 12 gabaritos instalados sobre uma armação rígida, com protetores individuais contra luz lateral e de fundo. Os gabaritos foram posicionados de maneira a cobrir o campo de visão da câmera. Veja a figura 9.9.

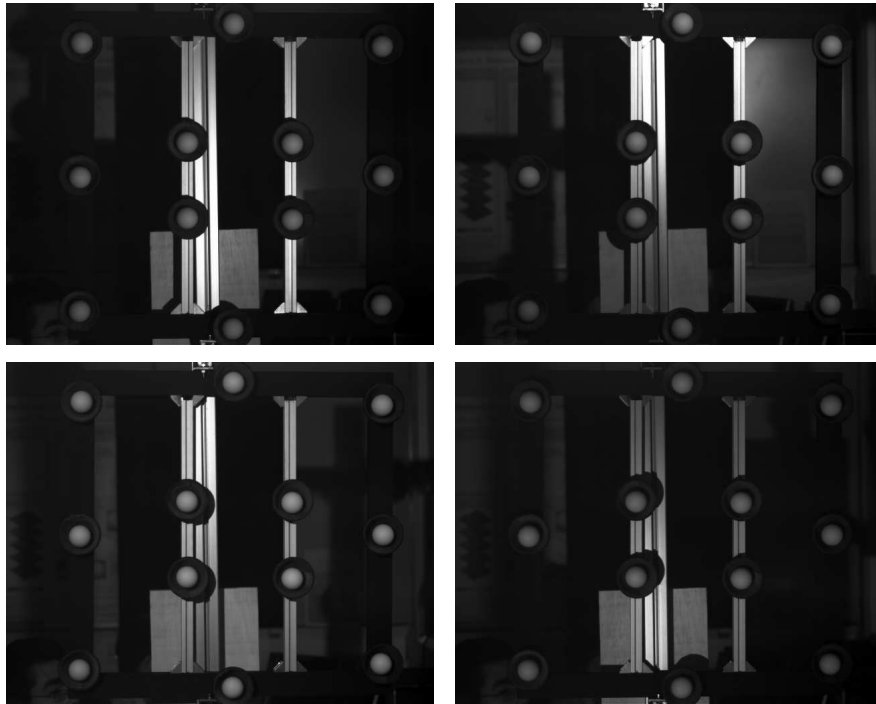


Figura 9.9: Grade de gabaritos sob as 4 fontes de iluminação do Photoface.

Para cada gabarito, ajustamos o modelo **COMPACT** sem o componente de Phong, como descrito no capítulo 4, e criamos uma tabela com 11172 amostras do gabarito virtual.

No decorrer destes ajustes, observamos que a intensidade de cada fonte luminosa apresentava uma variação significativa entre disparos consecutivos. Esta variação introduz uma distorção sistemática das normais. Este problema poderia ser resolvido com a instalação de um gabarito de iluminação permanente, de forma que o mesmo seja fotografado junto com a face à mesma distância. Entretanto, esta solução seria impraticável no contexto da aplicação considerada. Este problema será eventualmente resolvido com o uso de *flashes* mais consistentes.

9.3.2 Cálculo das normais

Para o cálculo das normais, em vez do algoritmo algébrico para superfícies lambertianas, nós usamos o procedimento BGRID do capítulo 5, escolhendo para cada caso a tabela correspondente ao gabarito mais próximo ao centro da face nas imagens. (Dada a quantidade limitada de fontes de luz, não foi possível aplicar a técnica descrita no capítulo 6).

As normais calculadas por este procedimento são significativamente mais corretas do que as obtidas com o algoritmo original, especialmente quando a face está posicionada longe do eixo ótico. Os erros mais significativos ocorrem nas regiões de sombra projetada — ao lado do nariz, abaixo do nariz e no queixo — onde a hipótese dos dados suficientes não é válida; e nos olhos, onde há reflexos especulares. Veja a figura 9.10

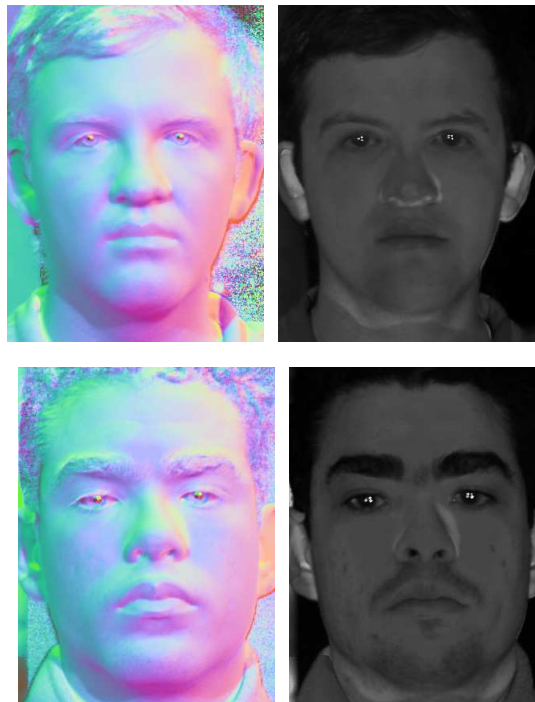


Figura 9.10: Mapas de normais e albedo calculados pelo método BGRID a partir das imagens da figura 9.5, com gabarito virtual COMPACT ajustado a um dos gabaritos da figura 9.9.

9.3.3 Integração do mapa de normais

Outra contribuição nossa para o *Photoface* foi o uso do integrador MSMAPIINTEGRATE do capítulo 7 para obter o mapa de altura no lugar do integrador de Frankott-Chellapa [37]. Veja a figura 9.11.

Observe que os mapas de altura recuperados sem pesos possuem distorções na região do queixo, nariz e testa, normalmente "achatando" a face. Para corrigir este problema,

geramos mapas de pesos, excluindo as regiões de normais espúrias. Infelizmente não desenvolvemos métodos adequados para excluir estas regiões automaticamente, tendo que gerar os mapas de peso manualmente.

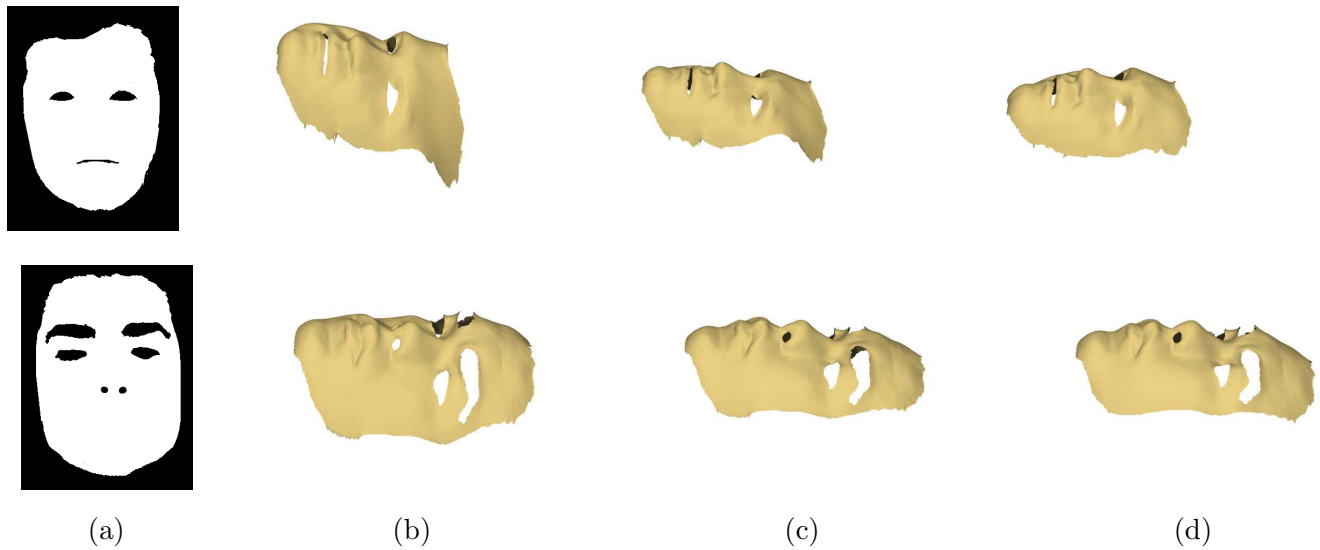


Figura 9.11: Máscaras de peso (a) criadas manualmente e mapas de altura (b–d) calculados a partir dos mapas de normal da figura 9.10: pelo método de Frankott-Chellapa(b), pelo sistema de Poisson sem pesos [6](c) e o método descrito no capítulo 7 (d). Observe que regiões fora do mapa de peso foram excluídas da renderização.

Capítulo 10

Conclusões

10.1 Contribuições

Os métodos que desenvolvemos contribuíram consideravelmente ao estado da arte em estéreo fotométrico, em vários aspectos.

10.1.1 Interpolação e suavização de gabaritos

A qualidade das informações extraídas da imagem do gabarito de iluminações é vital para o sucesso de um método EFBG. Os nossos modelos propostos de harmônicos esféricos (**HARMONIC**), bases radiais (**RADIAL**) e luz compacta única (**COMPACT**) do capítulo 4 permitem evitar defeitos provocados por gabaritos que apresentem irregularidades em suas superfícies, garantindo maior qualidade na recuperação do mapa de normais.

10.1.2 Aceleração de estéreo fotométrico

A técnica de aceleração por grade de baldes bidimensional (**BGRID**) do capítulo 5 proporciona uma aceleração considerável no cálculo de normais, em comparação aos métodos propostos na literatura para este fim; especificamente, *approximate nearest neighbours* (**ANN**), árvore k -dimensional (**KDTREE**) e hashing sensível a localidade (**LSH**). Para tabelas de 10000 entradas podem ser processados conjuntos de 50 ou mais imagens de 2 megapixels em poucos segundos. Além disso, o tempo de execução cresce sub-linearmente com o número de imagens usadas.

10.1.3 Estéreo fotométrico robusto

O método baseado inferência bayesiana que descrevemos no capítulo 6 permite efetuar o cálculo do mapa de normais de uma superfície mesmo na presença de sombras e reflexos.

Embora seja consideravelmente mais lento que o algoritmo de grade de baldes bidimensional, este método permite a utilização do estéreo fotométrico em cenas onde não é possível evitar a presença destes efeitos, ampliando a quantidade de aplicações práticas possíveis.

10.1.4 Integração eficiente e robusta

Os dois integradores multi-escala que desenvolvemos no decorrer desta tese constituem soluções originais, muito mais eficientes que os métodos da literatura e igualmente precisos. O integrador multi-escala MSMAPIINTEGRATE para dados em forma de imagem (capítulo 7) é mais rápido, mas falha com cenas com baixa continuidade, enquanto o algoritmo MSMESHIINTEGRATE para malhas irregulares (capítulo 8), um pouco mais lento, é capaz de lidar com cenas com presença de corredores estreitos e outras regiões fracamente conexas sem degradar a qualidade do resultado.

10.2 Aplicações

Nossa colaboração com o projeto *Photoface* de captura de geometria facial demonstrou as virtudes práticas dos nossos algoritmos nessa aplicação de estéreo fotométrico.

10.3 Publicações

O trabalho desenvolvido nesta tese de doutorado foi publicado em diversas conferências e periódicos, cujos artigos estão enumerados a seguir:

- **Grade de baldes bidimensional**

Saracchini, Stolfi, Leitão – *A Uniform Grid Structure to Speed Up Example-Based Photometric Stereo* – IEEE Transactions of Image Processing, Abril 2011

Saracchini, Stolfi, Leitão - *A Bucket Grid Structure to Speed Up Table Lookup in Gauge-Based Photometric Stereo* - In Proceedings of XX Brazilian Symposium on Computer Graphics and Image Processing, Brasil, 2007

- **Sombras e reflexos lustrosos**

Saracchini, Stolfi, Leitão - *Matching Photometric Stereo Observation Vectors with Shadows and Variable Albedo* - In Proceedings of XXI Brazilian Symposium on Computer Graphics and Image Processing, Brasil, 2008

- **Integração de mapas de gradiente**

Saracchini, Stolfi, Leitão, Atkinson, Smith - *A Robust Multi-Scale Integration Method to Obtain the Depth From Gradient Maps* - Elsevier Computer Vision and Image Understanding, Março 2012

Saracchini, Stolfi, Leitão, Atkinson, Smith - *Multi-Scale Integration of Slope Data on an Irregular Mesh* - In Proceedings of the Fifth Pacific-Rim Symposium on Image and Video Technology (PSIVT2011), Lecture Notes of Computer Science , Vol 7087, Coréia do Sul, Novembro 2011

Saracchini, Stolfi, Leitão, Atkinson, Smith - *Depth from Slope by Weighted Multi-Scale Integration* - XXI British Machine Vision Conference, Reino Unido, Setembro 2010

- **Aplicação: Reconstrução de artefatos arqueológicos**

Saracchini, Stolfi, Leitão - *Shape Reconstruction using a gauge-based photometric stereo method* - In Proceedings of XXXV Latin American Informatics Conference (CLEI), Brasil, Outubro 2009

10.4 Trabalhos Futuros

10.4.1 Inversão analítica

Uma ideia promissora que tivemos, mas ainda não conseguimos finalizar é realizar a inversão da função de assinatura de tonalização $\mathbf{l}(\vec{n})$, extraída do gabarito de iluminação, ajustando uma função aproximadora $F(\mathbf{s}) : \mathbb{S}^{m-1} \rightarrow \mathbb{S}^3$ à função inversa \mathbf{l}^{-1} .

A aproximação direta de uma função de \mathbb{S}^{m-1} para \mathbb{R}^3 por polinômios de grau máximo M em cada coordenada de \mathbf{s} exigiria da ordem de $(M+1)^m$ coeficientes, o que tornaria essa abordagem inviável para mais do que 3-4 imagens. Entretanto, podemos explorar a observação chave do capítulo 5: ou seja, o fato de que o conjunto $\mathbf{l}(\mathbb{H}^2)$ de assinaturas do gabarito é uma variedade bidimensional relativamente plana contida em \mathbb{S}^{m-1} . Podemos então implementar a inversão projetando a assinatura \mathbf{s} no plano P definido no capítulo 5, e escolhendo uma aproximação polinomial nas duas coordenadas $(\mathbf{s} \cdot \mathbf{u} - \mathbf{b}, \mathbf{s} \cdot \mathbf{v} - \mathbf{b})$ da projeção. Nesta abordagem o número de coeficientes fica reduzido a aproximadamente $3(M+1)^2$.

10.4.2 Aceleração de estéreo fotométrico robusta

O método BGRID considera o pixel analisado livre de efeitos indesejados na iluminação. Estamos investigando a utilização de métodos de otimização não-linear [77] para poder

utilizar a técnica de análise bayesiana do capítulo 6 sem a necessidade de verificar todas as entradas da tabela.

10.4.3 Variação espacial de iluminação

Como verificamos no exemplo do Photoface, na prática é muito difícil evitar variações espaciais do fluxo Φ_i . Seria muito útil estender as técnicas de modelagem de gabaritos e os algoritmos de cálculo de normais para campos de iluminação não uniformes, como por exemplo, os fornecidos por um monitor LCD.

10.4.4 Associação com outras abordagens

O EFIV é particularmente útil em regiões sem textura ou na presença de realces lustrosos. Sua associação com métodos de estéreo geométrico ou *Shape from defocusing* permitiria recuperar ou complementar informação de objetos que não podem ser convenientemente abordados por nenhum dos dois métodos isoladamente.

Referências Bibliográficas

- [1] 3dmdface system. Company page at <http://www.3dmd.com/3dmdface.html>, accessed on 2010-04-22, 2010.
- [2] 3dMD. 3dMDFace system. Electronic document at <http://www.3dmd.com/3dmdface.html>, edit date 2010/10/26, last accessed on 2010/10/26, 2010.
- [3] Jens Ackermann, Martin Ritz, André Stork, and Michael Goesele. Removing the example from photometric stereo by example. In *Proceedings of the ECCV 2010 Workshop on Reconstruction and Modeling of Large-Scale 3D Virtual Environments*, 2010.
- [4] A. Agrawal. Matlab/Octave code for robust surface reconstruction from 2d gradient fields. Available from <http://www.umiacs.umd.edu/~aagrawal/software.html>. Accessed on 2010-05-01, 2006. See [6].
- [5] A. Agrawal, R. Chellappa, and R. Raskar. An algebraic approach to surface reconstruction from gradient fields. In *Proc. 2005 Intl. Conf. on Computer Vision (ICCV)*, pages 174–181, 2005.
- [6] A. Agrawal, R. Raskar, and R. Chellappa. What is the range of surface reconstructions from a gradient field? In *Proc. 9th European Conf. on Computer Vision (ECCV)*, volume 3951, pages 578–591, 2006.
- [7] V. Akman, W. R. Franklin, M. Kankanhalli, and C. Narayanaswmi. Geometric computing and uniform grid technique. *Comput. Aided Des.*, 21:410–420, August 1989.
- [8] N. Alldrin, T. Zickler, and D. Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, june 2008.

- [9] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for near neighbor problem in high dimensions. In *Proc. 47th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 459–468, 2006.
- [10] V. Ansan, N. Mangold, P. Masson, E. Gailhardis, and G. Neukum. Topography of valley networks on Mars from Mars Express High Resolution Stereo Camera digital elevation models. *Journal of Geophysical Research (Planets)*, 113:7006, July 2008.
- [11] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the Association of Computing Machinery*, 45(6):891–923, 1998.
- [12] G. A. Atkinson, A. R. Farooq, M. L. Smith, and L. N. Smith. Facial reconstruction and alignment using photometric stereo and surface fitting. In *Proc. 2009 Iberian Conf. on Pattern Recognition and Image Analysis*, 2009.
- [13] G. A. Atkinson and E. R. Hancock. Surface reconstruction using polarization and photometric stereo. In *Proc. 12th Intl. Conf. on Computer Analysis of Images and Patterns (CAIP)*, volume 4673, pages 466–473, 2007.
- [14] G. A. Atkinson and M. L. Smith. Face recognition using photometric stereo. Project page at <http://www.uwe.ac.uk/cems/research/groups/mvl/projects/photoface.shtml>, accessed on 2010-04-22, 2009.
- [15] Hanspeter Pfister Baback Moghaddam, Jinho Lee and Raghu Machiraju. Model-based 3d face capture with shape-from-silhouettes. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, page 20, 2003.
- [16] Svetlana Barsky and Maria Petrou. The 4-source photometric stereo technique for three-dimensional surfaces in presence of highlights and shadows. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1239–1252, 2003.
- [17] R. Basri and D.W. Jacobs. Lambertian reflectance and linear subspaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2):218 – 233, feb 2003.
- [18] Ronen Basri, David Jacobs, and Ira Kemelmacher. Photometric stereo with general unknown lighting. *Intl. Journal of Computer Vision*, 72(3):239–257, 2007.
- [19] J. L. Bentley, J. H. Friedman, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, 1977.

- [20] Fausto Bernardini, Holly Rushmeier, Ioana M Martin, Joshua Mittleman, and Gabriel Taubin. Building a digital model of michelangelo's florentine pieta. *IEEE Computer Graphics and Applications*, 22(1):59–67, 2002.
- [21] R. B. Blackman and John Tukey. *The Measurement of Power Spectra, From the Point of View of Communications Engineering*, volume I. Dover Publications, 1959.
- [22] Ronald N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, 2nd edition, 1986.
- [23] Martin D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press.
- [24] Chris Cason and POV-Team. Persistence of Vision Raytracer. Electronic document at <http://www.povray.org/>, edit date 2008/07/30, last accessed on 2008/10/25, 2008.
- [25] M. Chandraker, S. Agarwal, and D. Kriegman. ShadowCuts: Photometric stereo with shadows. In *CVPR07*, pages 1–8, 2007.
- [26] Chu-Song Chen, Yi-Ping Hung, Chiann-Chu Chiang, and Ja-Ling Wu. Range data acquisition using color structured lighting and stereo vision. *Image and Vision Computing*, 15(6):445 – 456, 1997.
- [27] Tongbo Chen, M. Goesele, and H.-P. Seidel. Mesostructure from specularly. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1825 – 1832, 2006.
- [28] Y. Chen, H. Wang, and Y. Wang. Pyramid surface reconstruction from normal. In *Proc. 3rd IEEE Intl. Conf. on Image and Graphics (ICIG)*, pages 464–467, 2004.
- [29] James J. Clark. Photometric stereo using lcd displays. *Image and Vision Computing (Article in Press)*, November 2008.
- [30] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. McGraw-Hill, 1990.
- [31] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume I. Interscience, New York, 1953.
- [32] John Cristy. ImageMagick: Convert, edit and compose images. Electronic document at <http://www.imagemagick.org>, edit date 2008/10/20, last accessed on 2008/10/25, 2008.

- [33] Jean-Denis Durou, Jean-François Aujol, and Frédéric Courteille. Integrating the normal field of a surface in the presence of discontinuities. In *EMMCVPR '09: Proceedings of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 261–273, Berlin, Heidelberg, 2009. Springer-Verlag.
- [34] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367 – 392, 2004.
- [35] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, June 1981.
- [36] R. Fraile and E. R. Hancock. Combinatorial surface integration. In *Proc. 18th Intl. Conf. on Pattern Recognition (ICPR'06) Volume 1*, pages 59–62, 2006.
- [37] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(4):439–451, 1988.
- [38] Nathan Funk and Yeeh-Hong Yang. Using a raster display for photometric stereo. Technical Report TR06-13, Computing Science, University of Alberta, June 2006.
- [39] A. S. Georgiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:643–660, 2001.
- [40] A.S. Georgiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.
- [41] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S.M. Seitz. Multi-view stereo for community photo collections. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1 –8, oct. 2007.
- [42] D.B. Goldman, B. Curless, A. Hertzmann, and S.M. Seitz. Shape and spatially-varying brdfs from photometric stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):1060 –1071, june 2010.
- [43] Anamaria Gomide and Jorge Stolfi. Bases for non-homogeneous polynomial C_k splines on the sphere. In *Proceedings of LATIN'98 - Latin American Theoretical*

- Informatics Conference*, volume 1380 of *Lecture Notes in Computer Science*, pages 133–140. Springer, 1998.
- [44] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, 51(2):pp. 271–279, 1989.
- [45] Matthias Greuter, Michael Rosenfelder, Michael Blach, and Oliver Bittel. Obstacle and game element detection with the 3d-sensor kinect. In David Obdržálek and Achim Gottscheber, editors, *Research and Education in Robotics - EUROBOT 2011*, volume 161 of *Communications in Computer and Information Science*, pages 130–143. Springer Berlin Heidelberg, 2011. 10.1007/978-3-642-21975-7_13.
- [46] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau. Demosaicking: color filter array interpolation. *IEEE Signal Processing Magazine*, 22:44–54, January 2005.
- [47] Tom S. F. Haines and Richard C. Wilson. Integrating stereo with shape-from-shading derived orientation information. In *British Machine Vision Conf. 2007*, 2007.
- [48] M. F. Hansen, G. A. Atkinson, L. N. Smith, and M. L. Smith. 3d face reconstructions from photometric stereo using near infrared and visible light. *Computer Vision and Image Understanding*, In Press, 2010.
- [49] Hideki Hayakawa. Photometric stereo under a light source with arbitrary motion. *Journal of the Optical Society of America, Series A (JOSAA)*, 11(11):3079–3089, 1994.
- [50] Bryan Henderson. Netpbm home page. Electronic document at <http://netpbm.sourceforge.net/>, edit date 2008/08/01, last accessed on 2008/10/22, 2008.
- [51] C. Hernandez, G. Vogiatzis, G.J. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, oct. 2007.
- [52] Aaron Hertzmann and Steven M. Seitz. Shape and materials by example: A photometric stereo approach. In *Proceedings IEEE CVPR 2003*, volume 1, pages 533–540, 2003.
- [53] Aaron Hertzmann and Steven M. Seitz. Example-based photometric stereo: Shape reconstruction with general, varying BRDFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1254–1264, 2005.

- [54] T. Higo, Y. Matsushita, N. Joshi, and K. Ikeuchi. A hand-held photometric stereo camera for 3-d modeling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1234–1241, 29 2009-oct. 2 2009.
- [55] Tomoaki Higo, Yasuyuki Matsushita, and Katsushi Ikeuchi. Consensus photometric stereo. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 1157–1164, 2010.
- [56] B. K. P. Horn. *Robot Vision*. The MIT Press and McGraw-Hill, 1986.
- [57] B. K. P. Horn. Height and gradient from shading. *Intl. Journal of Computer Vision*, 5(1):37–75, 1990.
- [58] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbour: Towards removing the curse of dimensionality. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing (STOC'98)*, pages 604–613, 1998.
- [59] E. North Coleman Jr. and Ramesh Jain. Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. pages 180–199, 1992.
- [60] M. Kampel and R. Sablatnig. 3D puzzling of archeological fragments. In Danijel Skocaj, editor, *Proc. of 9th Computer Vision Winter Workshop*, pages 31–40, 2004.
- [61] G. Kay and T. Caelli. Estimating the parameters of an illumination model using photometric stereo. *Graphical Models and Image Processing*, 56(5):365–388, 1995.
- [62] D. G. Kirkpatrick. Optimal search in planar subdivisions. *SIAM J. on Computing*, 12:28–35, 1983.
- [63] Soren Konig and Stefan Gumhold. Image-based motion compensation for structured light scanning of dynamic surfaces. *Int. J. Intell. Syst. Technol. Appl.*, 5:434–441, November 2008.
- [64] P. D. Kovesi. `frankotchellappa.m`: The Frankot-Chellappa gradient integrator in MATLAB/Octave. Univ. of Western Australia, <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/Shapelet/>, 2000.
- [65] H. C. G. Leitão, R. F. V. Saracchini, and J. Stolfi. Matching photometric observation vectors with shadows and variable albedo. In *Proc. 21th Brazilian Symp. on Computer Graphics and Image Processing (SIBGRAPI)*, pages 179–186, 2008.
- [66] Helena Cristina G. Leitão and Jorge Stolfi. Digital reconstruction of fragmented artifacts. In Xavier Perrot, editor, *Proc. 8th International Cultural Heritage Informatics Meeting (ICHIM 2004)*, number 2345, pages 1–16, 2004.

- [67] W.-H. Liao, S.J. Aggarwal, and J.K. Aggarwal. The reconstruction of dynamic 3d structure of biological objects using stereo microscope images. *Machine Vision and Applications*, 9:166–178, 1997.
- [68] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I-900 – I-903 vol.1, 2002.
- [69] S. Magda, D.J. Kriegman, T. Zickler, and P.N. Belhumeur. Beyond lambert: reconstructing surfaces with arbitrary brdfs. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 391 –398 vol.2, 2001.
- [70] Stefano Mattoccia, Federico Tombari, and Luigi Di Stefano. Stereo vision enabling precise border localization within a scanline optimization framework. In Yasushi Yagi, Sing Kang, In Kweon, and Hongbin Zha, editors, *Computer Vision – ACCV 2007*, volume 4844 of *Lecture Notes in Computer Science*, pages 517–527. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-76390-1_51.
- [71] G. McGunnigle. Photometric stereo with gradated extended sources for recovery of specular surfaces. *J. Opt. Soc. Am. A*, 27(5):1127–1136, May 2010.
- [72] Daisuke Miyazaki, Kenji Hara, and Katsushi Ikeuchi. Median photometric stereo as applied to the segonko tumulus and museum objects. *International Journal of Computer Vision*, 86:229–242, 2010.
- [73] C. Müller. *Spherical Harmonics*, volume 17. Springer, 1966.
- [74] S. Nayar, K. Ikeuchi, and T. Kanade. Determining shape and reflectance of a hybrid surface by photometric sampling. *IEEE Transactions on Robotics and Automation*, 6(4):418–431, 1990.
- [75] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
- [76] Heung-Sun Ng, Tai-Pang Wu, and Chi-Keung Tang. Surface-from-gradients without discrete integrability enforcement: A Gaussian kernel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):2085–2099, November 2010.
- [77] Danilo Roberto Pereira, Jorge Stolfi, and Rafael Felipe Veiga Saracchini. A new method for nonlinear optimization. Technical Report IC-11-19, Institute of Computing, University of Campinas, November 2011.

- [78] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18:311–317, June 1975.
- [79] B. K. P.Horn. Height and gradient from shading. Technical Report AI Memo 1105, Massachusetts Institute of Technology, 1989.
- [80] R. Pintus, S. Podda, and M. Vanzi. An automatic alignment procedure for a 4-source photometric stereo technique applied to scanning electron microscopy. In *Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE*, pages 1278 –1283, april 2006.
- [81] W. Press, W. Teukolsky, S. Vetterling, and B. Flannery. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 1997.
- [82] D. Reddy, A. Agrawal, and R. Chellappa. Enforcing integrability by error correction using ℓ_1 -minimization. In *Proc. 2009 IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2350–2357, 2009.
- [83] Antonio Roble, Robert Fraile, and Edwin R. Hancock. Surface Integration in Matlab. BitBucket.org Free source code hosting, <http://bitbucket.org/jaxze/surface-integration-in-matlab/overview>, 2010.
- [84] A. Robles-Kelly and E. R. Hancock. Surface height recovery from surface normals using manifold embedding. In *Proc. Intl. Conf. on Image Processing (ICIP)*, 2004.
- [85] Greg Roelofs. Portable Network Graphics: An open, extensible image format with lossless compression. Electronic document at <http://libpng.org/pub/png/>, edit date 2008/09/19, last accessed on 2008/10/22, 2008.
- [86] R. F. V. Saracchini, J. Stolfi, and H. C. G. Leitão. Shape reconstruction using a gauge-based photometric stereo method. In *Proceedings of CLEI*, 2009.
- [87] Grant Schindler. Photometric stereo via computer screen lighting. In *Proc. 4th International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.
- [88] Li Shen, Takashi Machida, and Haruo Takemura. Efficient photometric stereo for three-dimensional surfaces with unknown BRDF. In *Proceedings of the 5th Intl. Conf. on 3-D Digital Imaging and Modeling (3DIM'05)*, pages 326–333, 2005.
- [89] Terence Sim and Takeo Kanade. Combining models and exemplars for face recognition. In *Proceedings of CVPR Workshop on Models versus Exemplars in Computer Vision*, Lecture Notes in Computer Science, 2001.

- [90] G. D. J. Smith and A. G. Bors. Height estimation from vector fields of surface normals. In *Proc. IEEE Intl. Conf. on Digital Signal Processing (DSP)*, pages 1031–1034, 2002.
- [91] M. L. Smith. *Surface Inspection Techniques - Using the Integration of Innovative Machine Vision and Graphical Modelling Techniques*. John Wiley & Sons, 2006.
- [92] Jorge Stolfi. Libpst. Software Library available at <http://www.ic.unicamp.br/~stolfi/links/stolfi/programs/c/JSLIBS/libpst/>, edit date 2011/07/16, last accessed on 2011/11/04, 2011.
- [93] Yasuhito Suenaga and Yasuhiko Watanabe. A method for the synchronized acquisition of cylindrical range and color data. In *MVA*, pages 137–142, 1990.
- [94] J. Sun, M. L. Smith, A. R. Farooq, and L. N. Smith. Concealed object perception and recognition using a photometric stereo strategy. In *Proc. 11th Intl. Conf. on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, volume 5807, pages 445–455, 2009.
- [95] J. Sun, M. L. Smith, L. N. Smith, L. Coutts, R. Dabis, C. Harland, and J. Bamber. Reflectance of human skin using colour photometric stereo - with particular application to pigmented lesion analysis. *Skin Research and Technology*, 14:173–179, 2008.
- [96] H. D. Tagare and R. J. P. de Figueiredo. A theory of photometric stereo for a class of diffuse non-lambertian surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):133–152, 1991.
- [97] D. Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):417–438, 1988.
- [98] Demetri Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2):129–139, March 1986.
- [99] Michael J. Thali, Marcel Braun, and Richard Dirnhofer. Optical 3d surface digitizing in forensic medicine: 3d documentation of skin and bone injuries. *Forensic Science International*, 137(2-3):203 – 208, 2003.
- [100] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *J. Opt. Soc. Am.*, 57(9):1105–1112, Sep 1967.

- [101] T. S. Trowbridge and K. P. Reitz. Average irregularity representation of a rough surface for ray reflection. *J. Opt. Soc. Am.*, 65(5):531–536, May 1975.
- [102] George Vogiatzis and Carlos Hernández. Self-calibrated, multi-spectral photometric stereo for 3d face capture. *International Journal of Computer Vision*, pages 1–13, 2011.
- [103] Lei Wang, J. Shu, T. Emura, and M. Kumagai. A 3d scanning laser rangefinder and its application to an autonomous guided vehicle. In *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, volume 1, pages 331–335 vol.1, 2000.
- [104] Gregory J. Ward. Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph.*, 26:265–272, July 1992.
- [105] T. Wei and R. Klette. Height from gradient using surface curvature and area constraints. In *Proc. 3rd Indian Conf. on Computer Vision, Graphics and Image Processing*, 2002.
- [106] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, 1980.
- [107] R. J. Woodham. Determining surface curvature with photometric stereo. In *Proceedings of the 1989 IEEE Intl. Conf. on Robotics and Automation*, volume 1, pages 36–42, 1989.
- [108] R. J. Woodham. Gradient and curvature from the photometric stereo method, including local confidence estimation. *Journal of the Optical Society of America, Series A*, 11(11):3050–3068, 1994.
- [109] Z. Wu and L. Li. A line-integration based method for depth recovery from surface normals. *Computer Vision, Graphics and Image Processing*, 43(1):53–66, 1988.
- [110] Oliver Wulf and Bernardo Wagner. Fast 3d scanning methods for laser measurement systems. *Scanning*, 1(section 2):3–8, 2003.
- [111] C. Young, D. K. Buck, and A. C. Collins. POV-Ray - Persistence of Vision Raytracer, version 3.6.1. Available at <http://www.povray.org>, last accessed on 2007/12/08.
- [112] Chanki Yu, Yongduek Seo, and Sang Lee. Photometric stereo from maximum feasible lambertian reflections. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios,

- editors, *Computer Vision – ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 115–126. Springer Berlin / Heidelberg, 2010.
- [113] A.L. Yuille, D. Snow, R. Epstein, and P.N. Belhumeur. Determining generative models of objects under varying illumination: Shape and albedo from multiple images using svd and integrability. *International Journal of Computer Vision*, 35:203–222, 1999. 10.1023/A:1008180726317.
- [114] Alan Yuille and Daniel Snow. Shape and albedo from multiple images using integrability. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 97)*, pages 158–164, 1999.
- [115] Carlos Elias Arminio Zampieri and Jorge Stolfi. Image retrieval by multi-scale interval distance estimation. Technical Report IC-09-39, Institute of Computing, University of Campinas, October 2009.
- [116] Lin Zhong and James J. Little. Photometric stereo via locality sensitive high-dimension hashing. In *Proceedings of the Second Canadian Conf. on Computer and Robot Vision (CRV’05)*, pages 104–111, 2005.