

**Anselmo Antunes
Montenegro**

**Reconstrução de cenas a
partir de imagens através de
escultura do espaço por
refinamento adaptativo**

TESE DE DOUTORADO

**DEPARTAMENTO DE INFORMÁTICA
Programa de Pós-graduação em
Informática**

Rio de Janeiro
Setembro de 2003



Anselmo Antunes Montenegro

**Reconstrução de cenas a partir de imagens
através de escultura do espaço por
refinamento adaptativo**

Tese de Doutorado

Tese apresentada ao Programa de Pós-graduação em In-
formática do Departamento de Informática da PUC como
parte dos requisitos parciais para obtenção do título de
Doutor em Informática

Orientador: Prof. Marcelo Gattass
Co-Orientador: Prof. Paulo Cezar Pinto Carvalho

Rio de Janeiro
Setembro de 2003



Anselmo Antunes Montenegro

**Reconstrução de cenas a partir de imagens
através de escultura do espaço por
refinamento adaptativo**

Tese apresentada ao Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC como parte dos requisitos parciais para obtenção do título de Doutor em Informática aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marcelo Gattass

Orientador

Departamento de Informática — PUC

Prof. Paulo Cezar Pinto Carvalho

Co-Orientador

Departamento de Informática — PUC

Prof. Manuel Menezes de Oliveira Neto

Instituto de Informática - UFRGS

Prof. Luis Carlos Pacheco R. Velho

Instituto de Matemática Pura e Aplicada - IMPA

Prof. Waldemar Celes Filho

Departamento de Informática - Puc-Rio

Prof. Marcelo de Andrade Dreux

Departamento de Engenharia Mecânica - Puc-Rio

Prof. Ney Augusto Dumont

Coordenador Setorial do Centro Técnico Científico —

PUC

Rio de Janeiro, 12 de Setembro de 2003

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Anselmo Antunes Montenegro

Graduou-se em Informática pela Universidade Federal Fluminense – UFF. Obteve o título de Mestre em Informática pela Pontifícia Universidade Católica do Rio de Janeiro – Puc-Rio na área de Computação Gráfica.

Ficha Catalográfica

Montenegro, Anselmo Antunes

Reconstrução de cenas a partir de imagens através de escultura do espaço por refinamento adaptativo/ Anselmo Antunes Montenegro; orientador: Marcelo Gattass; co-orientador: Paulo Cezar Pinto Carvalho. — Rio de Janeiro : PUC, Departamento de Informática, 2003.

v., 196 f. il. ; 30 cm

1. Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Teses. 2. Escultura do espaço 3. Coloração de voxels 4. Reconstrução volumétrica 5. Modelagem baseada em imagens I. Gattass, Marcelo. II. Carvalho, Paulo Cezar Pinto. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

Aos meus pais, Henrique e Selma.

Agradecimentos

Em primeiro lugar, a Deus, por ter me dado vida e por permitir que eu possa aprender um pouco mais a cada instante.

Aos meus pais, pelo amor e carinho que me deram e por me ajudarem em todos os momentos da minha vida.

A toda a minha família, por todos os momentos que passamos juntos.

Ao meu padrinho e avô Cecílio (in memoria), que me despertou o interesse pelo estudo e pela leitura.

À Geiza por ter estado sempre a meu lado e me incentivado a progredir cada vez mais.

Aos meus orientadores Marcelo Gattass e Paulo Cezar Pinto Carvalho, sem os quais este trabalho jamais teria sido concluído. Agradeço-lhes por todos os ensinamentos que me passaram nestes últimos anos, e por terem me ajudado a amadurecer um pouco mais como pesquisador e ser humano.

Ao Prof. Luiz Velho pela chance de poder desenvolver um trabalho que culminou nesta tese e pelas valiosas sugestões e incentivos.

Ao Prof. Manuel Menezes de Oliveira Neto por todas as sugestões e críticas que foram essenciais para o aprimoramento deste trabalho, assim como a toda banca examinadora.

Ao Prof. Luiz Satoru Ochi da UFF por ter me iniciado como pesquisador, meu eterno agradecimento.

À CAPES que me propiciou a bolsa, fundamental para a realização desse trabalho.

Ao Tecgraf pela ajuda técnica e financeira, além de me permitir crescer profissionalmente.

Ao Visgraf e ao IMPA que forneceram todo o ambiente necessário à conclusão deste trabalho.

Aos meus amigos Flávio Szenberg (TeCGraf) e Fábio Franco (Visgraf) por terem colaborado significativamente para o desenvolvimento deste trabalho.

Aos amigos do Visgraf e TeCGraf que colaboraram direta e indiretamente para o desenvolvimento deste trabalho e pelos vários momentos de descontração que juntos compartilhamos.

E finalmente, a todas as outras pessoas que aqui não foram mencionadas mas que foram importantes em algum momento da minha vida.

Resumo

Montenegro, Anselmo Antunes; Gattass, Marcelo; Carvalho, Paulo Cezar Pinto. **Reconstrução de cenas a partir de imagens através de escultura do espaço por refinamento adaptativo.** Rio de Janeiro, 2003. 196p. Tese de Doutorado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A reconstrução de cenas a partir de imagens tem recebido, recentemente, grande interesse por parte dos pesquisadores das áreas de visão computacional, computação gráfica e modelagem geométrica. Várias são as suas aplicações como, por exemplo, modelagem de objetos a partir de imagens, construção de ambientes virtuais e telepresença.

Dentre os métodos que têm produzido bons resultados na reconstrução de cenas a partir de imagens, podemos destacar aqueles que se baseiam em algoritmos de *Escultura do Espaço*. Tais técnicas procuram determinar quais são os elementos, em uma representação volumétrica do espaço da cena, que satisfazem um conjunto de restrições fotométricas impostas por um conjunto de imagens. Uma vez determinados, tais elementos volumétricos são coloridos de modo que reproduzam as informações fotométricas nas imagens de entrada, com uma certa margem de tolerância especificada com base em critérios estatísticos.

Neste trabalho, investigamos o emprego de técnicas utilizadas em visualização no desenvolvimento de métodos de escultura do espaço. Como resultado, propomos um método por refinamento adaptativo que trabalha sobre espaços de reconstrução representados através de subdivisões espaciais. Tal método é capaz de realizar o processo de reconstrução de modo mais eficiente, empregando esforços proporcionais às características locais da cena, que são descobertas à medida em que a reconstrução é realizada.

Finalmente, avaliamos a qualidade e a eficiência do método proposto, com base em um conjunto de resultados obtidos através de um sistema de reconstrução de objetos que utiliza imagens capturadas por *webcams*.

Palavras-chave

Escultura do Espaço; coloração de voxels; reconstrução volumétrica; modelagem baseada em imagens.

Abstract

Montenegro, Anselmo Antunes; Gattass, Marcelo; Carvalho, Paulo Cezar Pinto. . Rio de Janeiro, 2003. 196p. PhD. Thesis — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The reconstruction of scenes from images has received special attention from researchers of the areas of *computer vision*, *computer graphics* and *geometric modeling*. As examples of application we can mention image-based scene reconstruction, modeling of complex "as-built" objects, construction of virtual environments and telepresence.

Among the most successful methods used for the reconstruction of scenes from images are those based on Space Carving algorithms. These techniques reconstruct the shape of the objects of interest in a scene by determining, in a volumetric representation of the scene space, those elements that satisfy a set of photometric constraints imposed by the input images. Once determined, each photo-consistent element is colorized according to the photometric information in the input images, in such a way that they reproduce the photometric information in the input images, within some pre-specified error tolerance.

In this work, we investigate the use of rendering techniques in space carving methods. As a result, we propose a method based on an adaptive refinement process which works on reconstruction spaces represented by spatial subdivisions. We claim that such method can reconstruct the objects of interest in a more efficient way, using resources proportional to the local characteristics of the scene, which are discovered as the reconstruction takes place.

Finally, we evaluate the quality and the efficiency of the method based on the results obtained from a reconstruction device that works with images captured from webcams.

Keywords

Space Carving; voxel coloring; volumetric reconstruction; scene reconstruction; image based modeling.

Conteúdo

1	Introdução	12
2	Uma revisão em reconstrução volumétrica de cenas	17
2.1	Introdução	17
2.2	Reconstrução baseada em silhuetas	17
2.3	Reconstrução através de foto-consistência	22
2.4	Coloração de voxels	23
2.5	Escultura do espaço (<i>Space Carving</i>)	31
2.6	Variações	41
3	Técnicas de visualização aplicadas à Escultura do Espaço	53
3.1	Introdução	53
3.2	Alguns aspectos pouco explorados	53
3.3	Processamento no espaço das imagens	55
3.4	Processamento no espaço da cena	56
4	Escultura do espaço no espaço da cena	81
4.1	Introdução	81
4.2	Registro	82
4.3	Uso de silhuetas na reconstrução	87
4.4	Tratamento da visibilidade	90
4.5	Determinação dos voxels foto-consistentes	93
4.6	Atribuição de cor aos voxels	102
4.7	Algoritmo	103
4.8	Análise	104
5	Escultura do espaço através de refinamento adaptativo	105
5.1	Motivação	106
5.2	Alguns trabalhos anteriores	108
5.3	Representação adaptativa do espaço de reconstrução	110
5.4	Escultura de cenas em espaços representados através de octrees	113
6	Implementação do sistema	129
6.1	Calibração das câmeras	130
6.2	Câmera Tsai x Câmera OpenGL	133
6.3	Calibração do fundo da cena	135
6.4	Segmentação com base em mapas de intervalos de confiança	137
6.5	Reconstrução	138
7	Resultados	139
7.1	Introdução	139
7.2	Análise qualitativa	148
7.3	Análise em função da resolução	162
7.4	Análise do algoritmo EERA nos diferentes níveis de refinamento	165

7.5	Análise do teste estatístico baseado em limiares locais	171
7.6	Análise de desempenho	175
8	Conclusões e trabalhos futuros	180
8.1	Conclusões e comentários finais	180
8.2	Trabalhos futuros	184

Lista de Figuras

1.1	Etapas de um processo de reconstrução	14
2.1	Envoltória visual inferida	18
2.2	Coloração de voxels	25
2.3	Determinação da visibilidade	30
2.4	Exemplo de aplicação do método de coloração de voxels	31
2.5	Teorema do subconjunto	35
2.6	Câmeras consideradas em uma varredura para um determinado voxel.	39
2.7	GVC-IB e GVC-LDI	45
2.8	Projeção das texturas no espaço da cena	46
2.9	Refinamento	47
2.10	Exemplos de shuffle transforms	50
3.1	Superfície no interior de um voxel	56
3.2	Planos de referência	58
3.3	Mapeamento de textura sobre uma superfície	59
3.4	Mapeamento projetivo de textura	59
3.5	Amostragem acima do limite de Nyqvist	69
3.6	Amostragem abaixo do limite de Nyqvist	70
3.7	Pré-filtragem	72
3.8	Mapeamentos não-afins	73
3.9	Filtros gaussianos elípticos	77
3.10	Mipmap	78
4.1	Projeção de uma imagem em um plano de referência	82
4.2	Imagens tomadas de um modelo sintético	86
4.3	Imagens projetadas sobre um plano de referência no espaço da cena	86
4.4	Segmentação do objeto nas imagens	89
4.5	Complementariedade entre a envoltória fotométrica (photo hull) e envoltória visual (visual hull).	89
4.6	Mapas de visibilidade projetados sobre um plano de referência no espaço da cena	93
5.1	Espaço de reconstrução representado por voxels.	105
5.2	Coloração através de refinamento. Figura obtida de [52]. (a) - reconstrução com perdas; (b) - reconstrução correta; (c) - diferença	110
5.3	Representação através de octrees	113
5.4	Coloração em uma octree	114
5.5	Varredura de planos em uma octree	118
5.6	Número de planos em cada nível de resolução	119
5.7	Células interceptadas por um plano de varredura. As células em tom escuro são as células ativas, as em tom mais claro são células já avaliadas.	120

5.8	Imagens projetadas em diferentes níveis de resolução	121
6.1	Imagem do sistema	129
6.2	Processo de reconstrução	130
6.3	Padrão de calibração	130
6.4	Resultado da calibração das câmeras do sistema	132
6.5	Sistemas de coordenadas utilizados na OpenGL e no método de Tsai	134
7.1	Dado I - AI - Imagens de entrada	142
7.2	Dado I - AI - Imagens segmentadas	143
7.3	Dado II - Dolphins - Imagens de entrada	144
7.4	Dado II - Dolphins - Imagens de segmentação	145
7.5	Dado III - Tigre - Imagens do objeto e da segmentação	146
7.6	Dado IV - Dino - Imagens do objeto e da segmentação	147
7.7	Reconstrução do dado I através do algoritmo EEEI	149
7.8	Reconstrução do dado I através do algoritmo EEEC	150
7.9	Reconstrução do dado I através do algoritmo EERA	151
7.10	Reconstrução do dado II através do algoritmo EEEI	152
7.11	Reconstrução do dado II através do algoritmo EEEC	153
7.12	Reconstrução do dado II através do algoritmo EERA	154
7.13	Reconstrução do dado III através do algoritmo EEEI	155
7.14	Reconstrução do dado III através do algoritmo EEEC	156
7.15	Reconstrução do dado III através do algoritmo EERA	157
7.16	Reconstrução do dado IV através do algoritmo EEEI	158
7.17	Reconstrução do dado IV através do algoritmo EEEC	159
7.18	Reconstrução do dado IV através do algoritmo EERA	160
7.19	Voxels classificados como não visíveis	161
7.20	Comparação entre os diversos métodos no níveis de resolução 6x64 e 128x128.	163
7.21	Comparação entre os diversos métodos no níveis de resolução 6x64 e 128x128.	164
7.22	Estágios de refinamento de 4 a 8 da reconstrução do dado I através do algoritmo EERA.	167
7.23	Estágios de refinamento de 5 a 8 da reconstrução do dado II através do algoritmo EERA.	168
7.24	Estágios de refinamento de 5 a 8 da reconstrução do dado III através do algoritmo EERA.	169
7.25	Estágios de refinamento de 5 a 8 da reconstrução do dado IV através do algoritmo EERA.	170
7.26	Resultado da aplicação dos algoritmos para diferentes níveis de significância do teste estatístico proposto no capítulo 5	173
7.27	Resultado da aplicação dos algoritmos para diferentes níveis de significância do teste estatístico proposto no capítulo 5	174

Lista de Tabelas

2.1	Esquema de atualização dos rótulos da octree	21
2.2	Coloração de Voxels em Multiresolução x Coloração de Voxels Original. Os experimentos de Prock foram realizados em uma SGI O2 R5000 de 200 Mhz.	48
2.3	Classificação dos algoritmos	52
7.1	Resultado da aplicação do algoritmo EEEl na reconstrução do dado I	176
7.2	Resultado da aplicação do algoritmo EEEc na reconstrução do dado I	176
7.3	Resultado da aplicação do algoritmo EEEA na reconstrução do dado I	176
7.4	Resultado da aplicação do algoritmo EEEl na reconstrução do dado II	177
7.5	Resultado da aplicação do algoritmo EEEc na reconstrução do dado II	177
7.6	Resultado da aplicação do algoritmo EEEA na reconstrução do dado II	177
7.7	Resultado da aplicação do algoritmo EEEl na reconstrução do dado III	178
7.8	Resultado da aplicação do algoritmo EEEc na reconstrução do dado III	178
7.9	Resultado da aplicação do algoritmo EEEA na reconstrução do dado III	178
7.10	Resultado da aplicação do algoritmo EEEl na reconstrução do dado IV	179
7.11	Resultado da aplicação do algoritmo EEEc na reconstrução do dado IV	179
7.12	Resultado da aplicação do algoritmo EEEA na reconstrução do dado IV	179

Ó Shariputra, forma nada mais é do que vazio, vazio nada mais é do que forma; Assim, forma é vazio, vazio é forma.

**Diálogo entre o Bodhisattva Avalokiteshvara e o Arhat
Shariputra, *Sutra do Coração*.**

1

Introdução

A reconstrução de cenas a partir de imagens é um dos problemas mais importantes em visão computacional e tem recebido atualmente um grande interesse por parte dos pesquisadores das áreas de computação gráfica e modelagem geométrica.

A origem da pesquisa em reconstrução de cenas pode ser relacionada, de certa maneira, aos primeiros estudos sobre o processo de percepção de forma nos sistemas visuais de animais superiores e de seres humanos. Nesta fase inicial, pesquisadores das áreas de inteligência artificial, psicologia e neurologia buscavam encontrar modelos que pudessem explicar como o cérebro humano é capaz de reconstituir e interpretar as informações sobre a forma de uma cena codificadas através de um par de imagens [6, 8]. A maioria dos primeiros modelos propostos procurava descrever uma abstração da arquitetura existente nos sistemas visuais biológicos através de um conjunto de interações entre sensores e um sistema de processamento visual primitivo. Nesta fase, era bastante comum a modelagem de tais sistemas através de redes neuronais ou através de sistemas cooperativos, os quais podem ser compreendidos como casos particulares de redes de autômatos celulares [7].

À medida que conhecimentos mais profundos sobre tais processos foram sendo adquiridos, houve uma mudança gradual do enfoque para a investigação de técnicas que pudessem ser empregadas no auxílio à execução de tarefas que requerem algum tipo de conhecimento sobre as formas existentes no espaço de trabalho.

Os primeiros métodos para determinação da forma de uma cena a partir de imagens foram naturalmente influenciados pelos primeiros trabalhos sobre os sistemas visuais biológicos, sendo a maior parte deles baseados na visão estéreo [9, 10, 14, 15, 17]. Esta abordagem foi e ainda é, em certo nível, adotada principalmente por pesquisadores das áreas de visão computacional e robótica.

O processo de reconstrução de forma através de visão estereoscópica

requer a solução de um problema de correspondências entre feições existentes em duas ou mais imagens obtidas de uma cena a partir de pontos de vista distintos. Uma vez determinadas estas correspondências, e conhecidos os parâmetros intrínsecos e extrínsecos das câmeras que forneceram as imagens, é possível calcular, através de um processo de triangulação, as coordenadas tridimensionais dos pontos da cena associados às feições consideradas. Em uma etapa posterior, efetua-se a interpolação dos pontos 3D obtidos com base em alguma restrição sobre a forma da superfície (normalmente uma restrição de suavidade), de maneira que a geometria e a topologia que descrevem a cena possam ser reconstituídas. Finalmente, com o objetivo de tornar as reconstruções mais realistas, é efetuado um processo de colorização através da aplicação de uma combinação das texturas provenientes das diferentes imagens. Para maiores detalhes ver [35, 36, 62].

Dentre as principais limitações associadas aos métodos baseados em visão estéreo, podemos citar: reconstrução restrita a superfícies $2, 5D$, aplicabilidade a cenas cujos efeitos de oclusão e paralaxe são limitados e a necessidade de se introduzir na formulação do problema conhecimentos sobre a forma da cena através da adição de restrições. Algumas destas restrições são inerentes à geometria do problema como, por exemplo, a restrição epipolar, enquanto que outras são adicionadas com o objetivo de garantir a unicidade da solução ou possibilitar o desenvolvimento de métodos simples e eficientes.

Paralelamente, pesquisadores das áreas de computação gráfica e modelagem geométrica começaram a se interessar por técnicas de modelagem a partir de imagens. Inicialmente, as técnicas principais envolviam a determinação do espaço de ocupação da cena através da interseção de volumes cônicos, os quais eram determinados por raios passando pelos conjuntos de pixels delimitados pelas silhuetas dos objetos nas imagens e seus respectivos centros de projeção.

Podemos citar, como exemplos clássicos, os métodos propostos por Martin e Aggarwal [11], Chien [12, 18] e Potmesil [23]. Estes primeiros trabalhos limitavam bastante as vistas de entrada, já que requeriam procedimentos de interseção relativamente complexos.

Aos poucos, os novos métodos propostos começaram a aproveitar todo o potencial fornecido pelo desenvolvimento tecnológico relacionado à aquisição de imagens, se libertando definitivamente do paradigma fornecido pela visão estéreo binocular. Surgem então diversos métodos baseados em visão estéreo capazes de determinar mapas de profundidade a partir de múltiplos pontos de vista [47, 48, 49, 76].

Um outro grande avanço surgiu com o aprimoramento da tecnologia de aquisição de dados 3D, seja através de técnicas de estéreo ativo [33], seja através do surgimento de sistemas de varredura 3D de alta precisão [19, 24, 37, 39, 40, 41, 42, 43]. A partir deste momento, os métodos de reconstrução baseados em imagens de profundidade (*range images*) se tornaram bastante utilizados, principalmente quando são necessárias reconstruções detalhadas e com grande precisão.

Com o surgimento de câmeras digitais de baixo custo e de boa qualidade, renasce o interesse pela reconstrução de cenas a partir de fotos provenientes de câmeras calibradas. Devido às limitações existentes nas técnicas baseadas em estéreo, foram propostas novas alternativas que trabalham no espaço da cena e não requerem a solução do problema de correspondências como um pré-requisito.

Seitz e Dyer [44], provavelmente se inspirando nas técnicas de reconstrução volumétrica iniciais, propuseram uma nova abordagem baseada em uma representação do espaço da cena através de voxels na qual não é necessário o conhecimento explícito das silhuetas dos objetos, pois são utilizados apenas critérios de foto-consistência para determinar a pertinência de um voxel à cena. Como os demais métodos, a técnica proposta por Seitz restringe-se apenas à reconstrução de superfícies opacas. Além disso, introduz uma nova restrição: a de que as cenas a serem reconstruídas possam ser modeladas por superfícies lambertianas. A grande inovação introduzida por estes métodos é sua capacidade de integrar, em um único arcabouço, as diversas etapas necessárias à reconstrução que normalmente são efetuadas separadamente nos demais métodos de reconstrução. Tanto a reconstrução da geometria e da topologia de uma cena, quanto a sua colorização, são efetuadas simultaneamente no decorrer do processo (Figura 1.1).



Figura 1.1: Etapas de um processo de reconstrução

A primeira implementação do método de coloração de voxels se restringia a configurações de câmera que se posicionavam fora do fecho convexo da cena. Esta restrição foi introduzida com o intuito de simplificar o problema de determinação da visibilidade, um aspecto fundamental para a realização da técnica. Através desta restrição é possível determinar uma

ordenação global dos voxels em relação ao conjunto de câmeras permitindo desta forma que o método possa ser realizado em um único passo. Em seguida, Kutulakos e Seitz [63] desenvolveram uma teoria sobre reconstrução de formas com base na satisfação de restrições de foto-consistência e generalizaram o método proposto para configurações de câmeras arbitrárias. Esta generalização passou a ser conhecida como Escultura do Espaço (*Space Carving*).

Ultimamente, algumas variações desta técnica têm sido propostas, dentre as quais podemos destacar as seguintes: a *Coloração de Voxels Generalizada (Generalized Voxel Coloring)* de Slabaugh [64], que propõe novas estruturas de dados para o tratamento da visibilidade dos voxels; a *Coloração de Voxels por Hipóteses Múltiplas* de Eisert [55], que propõe um novo algoritmo em dois passos, um envolvendo uma construção de uma lista de hipóteses e um outro no qual é efetuada uma remoção de hipóteses não consistentes com os dados de entrada; a *Visão Estereoscópica Aproximada através de Múltiplos Pontos de Vista (Approximated N-view Stereo)* de Kutulakos [65] que procura resolver o problema de como determinar uma família de formas que aproximam incrementalmente a forma de uma cena 3D arbitrária através de uma nova definição de foto-consistência baseada em uma classe de transformações, as *shuffle transforms*. Outras variantes importantes serão vistas na revisão bibliográfica apresentada no restante do documento.

As versões mais recentes foram propostas por Broadhurst e Cipolla [66, 71, 72] e paralelamente por Kutulakos e Bhotika [73], baseando-se em uma formalização estatística e probabilística para a modelagem do problema de reconstrução de cenas através de representações volumétricas.

Neste trabalho, apresentamos um novo método de escultura do espaço baseado no uso combinado de técnicas de mapeamento projetivo de textura e subdivisão adaptativa do espaço. Tal método é capaz de lidar apropriadamente com os problemas de amostragem existentes no processo de reconstrução, além de empregar esforços proporcionais às características locais da cena, que são descobertas à medida em que a reconstrução é realizada. O principal argumento desta tese é a de que métodos de escultura do espaço baseados no uso combinado de informações fotométricas e de segmentação, juntamente com estratégias de refinamento adaptativo, são capazes de produzir eficientemente reconstruções de boa qualidade, mesmo quando são utilizadas poucas imagens e sistema de aquisição mais modestos. A validade desta afirmação foi comprovada através os resultados obtidos através de sistema de reconstrução 3D baseado em *webcams*.

Apresentamos também as seguintes contribuições secundárias:

- Um estudo sistemático da aplicação de técnicas de visualização no desenvolvimento de métodos de escultura do espaço que trabalham diretamente no espaço da cena. Em particular, investigamos em detalhes o uso da técnica de mapeamento projetivo como mecanismo de registro de diversas informações no espaço da cena, como a visibilidade dos elementos do espaço, a segmentação dos objetos de interesse e a quantidade de ruído introduzido pelos sensores das câmeras usadas na aquisição das imagens.
- Um estudo sobre os testes estatísticos adequados para a determinação da foto-consistência dos elementos do espaço na presença de diferentes taxas de ruído em cada um dos sensores das câmeras.
- O desenvolvimento de um sistema de reconstrução 3D com base em *webcams* e diferentes estratégias para lidar com suas limitações.

O restante do trabalho está organizado da seguinte forma: no capítulo 2 apresentamos os principais conceitos e técnicas sobre reconstrução volumétrica de cenas, além de apresentar, de modo resumido, alguns dos trabalhos mais importantes no assunto; no capítulo 3, descrevemos como a técnica de mapeamento projetivo pode ser utilizada para registrar as informações necessárias à realização dos métodos de escultura do espaço no espaço em que se encontram os objetos de interesse, e discutimos suas diversas implicações; no capítulo 4, descrevemos o funcionamento dos métodos de reconstrução que trabalham diretamente no espaço 3D com base nas técnicas descritas no capítulo 3; no capítulo 5, apresentamos o método de reconstrução proposto que se baseia em uma estratégia de escultura do espaço por refinamento adaptativo; no capítulo 6, descrevemos o sistema de reconstrução 3D baseado em *webcams* que utiliza as técnicas de reconstrução baseadas em escultura do espaço, além de apresentarmos os principais problemas encontrados e as respectivas técnicas utilizadas para resolvê-los; no capítulo 7, descrevemos os resultados obtidos através de reconstruções a partir de imagens obtidas de cenas sintéticas e reais e finalmente, no capítulo 8, apresentamos as conclusões finais sobre o trabalho e alguns dos possíveis tópicos de pesquisa que podem ser investigados em trabalhos futuros.

2

Uma revisão em reconstrução volumétrica de cenas

2.1

Introdução

Os métodos que serão apresentados neste capítulo se caracterizam basicamente por descrever a cena a ser reconstruída através de uma representação volumétrica. Normalmente, em Computação Gráfica, associamos o termo volumétrico a algum tipo de amostragem. Nesta exposição, adotaremos a interpretação de Slabaugh et al. [74], o qual considera como sendo volumétrica uma representação da ocupação espacial da cena sem considerar a existência de uma amostragem subjacente. Por este motivo podemos incluir nesta categoria os primeiros métodos de reconstrução de forma que descreviam a cena obtida através da interseção de volumes geométricos.

Agruparemos os métodos de reconstrução volumétrica de cenas em duas grandes categorias que se impuseram naturalmente à medida em que as técnicas foram sendo desenvolvidas:

- métodos baseados em silhuetas.
- métodos baseados em critérios de foto-consistência.

Uma excelente compilação das técnicas de reconstrução volumétrica pode ser encontrado no trabalho de Slabaugh, Culbertson e Malzbender [74].

2.2

Reconstrução baseada em silhuetas

Os primeiros métodos de reconstrução volumétrica buscavam aproximar uma estrutura denominada Envoltória Visual (*Visual Hull*), a qual é definida como sendo a forma maximal que reproduz as silhuetas do objeto original quando renderizada a partir de todos os pontos de vista posicionados fora do seu fecho convexo. O conceito de Envoltória Visual foi estabele-

cido por Laurentini e é a base para a maioria das técnicas de reconstrução de cenas baseadas em silhuetas [31].

Na prática não é possível utilizar um conjunto contendo um número infinito de imagens, o que nos obriga a buscar uma aproximação da *Envoltória Visual* através da determinação do volume de ocupação da cena com base em um conjunto de restrições estabelecidas pela segmentação do objeto de estudo nas imagens. Em geral, isto é feito através de um processo de interseção de volumes capaz de determinar o que chamamos de *Envoltória Visual Inferida* (Figura 2.1). As imagens são segmentadas binariamente em regiões de fundo e regiões do objeto. As regiões contendo as projeções do objeto, juntamente com os centros de projeção associados, determinam volumes cônicos cuja interseção no espaço 3D determina uma aproximação para a envoltória visual.

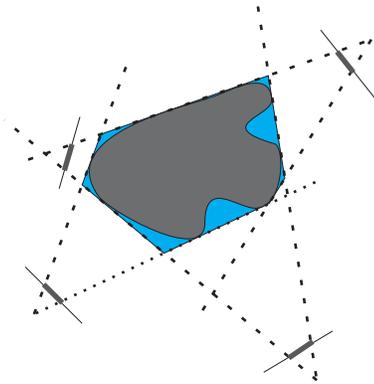


Figura 2.1: Envoltória visual inferida

As principais propriedades que relacionam o Fecho Visual Aproximado à forma original do objeto de estudo são as seguintes:

- é uma aproximação que engloba a forma real do objeto.
- o seu tamanho decresce à medida que mais imagens são consideradas.
- fornece uma aproximação para o fecho convexo da cena.

Os métodos apresentados a seguir se baseiam no processo de interseção volumétrica, variando-se apenas o tipo da estrutura de dados utilizada na representação dos volumes cônicos e os métodos utilizados no cálculo de interseções.

2.2.1

Representação por segmentos de volume

Martin e Aggarwal [11] foram os primeiros a propor um método para reconstrução volumétrica de cenas a partir de silhuetas. Inicialmente, obtém-se as segmentações dos objetos de interesse nas imagens, sobre as quais é realizada uma análise de componentes conexas de forma a determinar as silhuetas.

Após a determinação das silhuetas constrói-se uma *estrutura de paralelogramos* no espaço 3D com base na combinação das projeções ortográficas das silhuetas de duas imagens arbitrariamente escolhidas.

Em um passo seguinte, constrói-se uma representação por *volumes segmentados* o qual é composto por um conjunto de segmentos de reta paralelos a um dos eixos de um sistema de coordenadas local escolhido adequadamente. A estrutura de dados utilizada para representar os volumes segmentados é organizada de forma hierárquica em relação a cada uma das componentes do sistema de coordenadas local adotado, de forma a facilitar o processo de interseção de segmentos.

Após esta etapa, a representação por volumes segmentados é refinada através de silhuetas provenientes de outros pontos de vista, permitindo assim que uma representação poliédrica adequada da superfície possa ser obtida.

O método de Martin e Aggarwal é essencialmente um método geométrico, cuja descrição se torna um pouco complexa sem o auxílio de figuras. Para maiores detalhes, encorajamos o leitor a verificar o trabalho original.

2.2.2

Representação por octrees

Chien [12, 18] foi o primeiro a sugerir a utilização de octrees para a representação do volume de ocupação de cenas reconstruídas por métodos baseados em silhuetas. O método por ele proposto consiste na utilização de três imagens binárias obtidas a partir de três câmeras paralelas ortogonais, as quais são convertidas em estruturas quadtree, e posteriormente recombinadas de forma a obter uma representação da envoltória visual através de uma octree. As principais limitações do método proposto por Chien são a restrição do número de imagens utilizadas, a restrição de ortogonalidade dos eixos óticos e a utilização exclusiva de projeção paralela.

Apesar de conter severas restrições, o método proposto por Chien foi de suma importância para o aprimoramento das técnicas de reconstrução

volumétrica ao introduzir a utilização de estruturas espaciais adaptativas capazes de permitir que o processo de reconstrução fosse realizado com maior eficiência e simplicidade. Um aprimoramento das técnicas propostas por Chien pode ser encontrado no trabalho de Veenstra [20] que é capaz de gerar uma reconstrução do fecho visual de uma cena a partir de 13 pontos de vista ortográficos pré-definidos.

Potmesil [23] introduziu uma nova técnica capaz de reconstruir cenas através de imagens geradas por projeções perspectivas a partir de pontos de vista posicionados arbitrariamente no espaço. Em um primeiro momento, calcula-se uma representação através de uma estrutura *octree* de cada volume cônico determinado pelas silhuetas e seus respectivos centros de projeção. Em uma segunda etapa, calcula-se a interseção entre todos os cones definidos pelas *octrees*, gerando assim um modelo global sobre o qual é efetuado uma análise de componentes conexas 3D com o objetivo de rotular objetos individuais. Uma vez obtido o volume de ocupação da cena, associa-se um conjunto de vetores normais ao objeto, calculados com base na topologia da *octree*. Finalmente, efetua-se um processo de texturização, no qual um mecanismo de combinação é utilizado nas regiões da superfície do objeto que sofrem influência de mais de uma textura. Srivastava [27] propõe um trabalho similar no qual os contornos são aproximados por polígonos que, por sua vez, são decompostos em componentes convexas de forma a facilitar os testes de interseção entre as projeções das células da *octree* e as regiões delimitadas pelas silhuetas.

Um grande passo no desenvolvimento das técnicas de reconstrução a partir de silhuetas foi dado por Richard Szeliski [29]. Szeliski aproveitou de maneira eficaz as propriedades das estruturas *octree* de forma a explorar intensamente a coerência existente entre formas obtidas em diversos níveis de resolução. Outras contribuições importantes deste trabalho foram a proposta de um método automático para segmentação do objeto através de subtração de imagens, além de um mecanismo bastante prático para determinação da orientação do sistema de captura, o qual era composto por um prato rotativo e uma câmera fixa.

Quando propôs este novo método, Szeliski tinha dois objetivos principais a serem atingidos: processar cada imagem a partir do momento em que estivesse disponível e produzir rapidamente um modelo em baixa resolução, o qual poderia ser refinado incrementalmente à medida em que novas imagens fossem capturadas pelo sistema. Um outro objetivo de Szeliski era investigar métodos altamente paralelizáveis.

Basicamente, o algoritmo constrói uma representação do volume de ocupação do objeto de interesse através de um procedimento de refinamento efetuado sobre uma *octree*, cujas células são rotuladas da seguinte forma:

- negra - célula que está totalmente contida no objeto
- branca - célula que está completamente fora do objeto
- cinza - célula ambígua.

Primeiramente, assume-se que a cena seja representada por um volume inicial o qual é definido por uma célula negra ou por um pequeno conjunto de células negras, determinando, por exemplo, um volume de $8 \times 8 \times 8$ elementos. Para cada nova imagem adquirida, projeta-se todos os cubos associados às células negras(ativas) da *octree* na nova imagem e verifica-se se eles recaem completamente dentro ou completamente fora da silhueta. O novo rótulo da célula é determinado de acordo com a tabela 2.1.

cor antiga \Rightarrow resultado \Downarrow	negra	cinza	branca
dentro	negra	cinza	branca
ambíguo	cinza	cinza	branca
fora	branca	branca	branca

Tabela 2.1: Esquema de atualização dos rótulos da octree

Após um determinado nível de resolução ter sido processado, o que corresponde a uma revolução completa do objeto em relação à câmera fixa, refina-se as células cinzas subdividindo-as em oito novas células as quais passam a ser rotuladas como negras. As novas células negras assim obtidas são processadas exatamente como as do nível anterior.

2.2.3

Representação por voxels

Uma alternativa à representação do espaço da cena através de estruturas espaciais pode ser obtida através de representações volumétricas regulares.

Massone [16] foi um dos primeiros a propor um método que determinava a envoltória visual a partir de um volume descrito por voxels. Uma das características mais notáveis deste método era sua capacidade de reconstruir cenas reais(não sintéticas) cujas imagens eram obtidas através de

câmeras *Vidcon*. Este foi, na realidade, o primeiro método a lidar tanto com projeções paralelas quanto com projeções perspectivas.

Um pouco mais tarde, Fromherz [32] retoma a idéia de Massone e descreve um método, restrito a projeções ortográficas, que trabalha sobre um volume de voxels cujo tamanho das projeções são da escala dos pixels das imagens. Cada uma das silhuetas é obtida através de um processo de segmentação automática, assim como no método de Szeliski. Posteriormente tal método foi aprimorado através da introdução de um passo de refinamento do modelo com base em informações de luminância [34]. Em cada iteração, os voxels pertencentes à superfície do modelo inicial obtido são projetados em pares de imagens seqüenciais em uma seqüência de rotação. Se as luminâncias dos pixels em cada uma das regiões de projeção de um voxel diferem por uma quantidade maior que um determinado valor constante, então o voxel é removido da cena. Pela primeira vez é proposto um mecanismo para descarte de voxels com base na verificação de algum tipo de consistência fotométrica.

A reconstrução volumétrica de cenas a partir de seqüências de vídeo foi introduzida por Moezzi [38], que desenvolveu um sistema de 17 câmeras centralizadas em uma cena dinâmica contida em uma região volumétrica de $1m \times 1m \times 2m$. Em uma etapa de pré-processamento, cada quadro é segmentado em objeto e fundo gerando-se, desta forma, uma seqüência de imagens binárias. Em seguida é efetuada a etapa de interseção dos volumes com o objetivo de reconstruir o envoltória visual e por último, aplica-se um passo adicional para extração da superfície poligonal do volume obtido, sobre a qual são aplicadas texturas provenientes das imagens de entrada. Em [45] foi proposta uma melhoria no procedimento de colorização.

2.3

Reconstrução através de foto-consistência

O surgimento dos métodos de reconstrução volumétrica baseados em foto-consistência foi uma conseqüência natural do desenvolvimento das técnicas de reconstrução baseadas em silhuetas. Podemos observar claramente que os trabalhos de Szeliski e principalmente Fromherz, com sua função de consistência baseada em luminância, tiveram importância fundamental no surgimento dos métodos baseados em escultura do espaço que hoje conhecemos.

O grande diferencial entre os métodos baseados em silhuetas e os métodos baseados em foto-consistência é a capacidade destes últimos em li-

dar verdadeiramente com o problema de estereoscopia. Os métodos baseados em silhuetas, na verdade, só são capazes de reconstruir objetos da cena cuja segmentação fundo/objeto for explícita, permitindo assim a determinação do espaço de ocupação da cena através da interseção de volumes. É óbvio que estes métodos não são capazes de reconstruir corretamente objetos descritos por superfícies com concavidades, já que estas informações não podem ser capturadas por silhuetas.

Os métodos de reconstrução volumétrica baseados em foto-consistência não só são capazes de reconstruir cenas com geometrias arbitrárias como também são capazes de gerar um modelo completamente colorido sem necessidade de uma etapa de texturização adicional. Todo o processo é realizado através de consultas que verificam se um determinado elemento é capaz de explicar as cores nas regiões do conjunto de imagens em que se encontra visível.

Por outro lado, o problema de estereoscopia é um problema muito mais difícil, sendo bastante sensível a erros sistemáticos como erros de calibração, ruído, aliasing e etc. Contudo, a vantagem em relação aos métodos baseados somente em silhuetas é bastante grande, e o que é mais animador, nada impede que as duas técnicas sejam combinadas de forma a auxiliar o processo de reconstrução.

Na seção seguinte iremos descrever os métodos de *Coloração de Voxels* e *Escultura do espaço*, assim como suas diversas variações encontradas na literatura afim. Os conceitos de reconstrução com base em foto-consistência serão introduzidos de forma intuitiva através do método de coloração de voxels, enquanto que uma abordagem mais formal da teoria será descrita na seção que trata do método de Escultura do Espaço.

2.4

Coloração de voxels

O método proposto por S. Seitz e C. Dyer [44, 56] trata o problema de reconstrução de uma cena a partir de fotos como um problema de coloração de voxels. Primeiramente, descreveremos o problema, introduzindo as notações necessárias. Em seguida avaliaremos cada um dos aspectos importantes associados ao problema e finalmente especificaremos o algoritmo capaz de solucioná-lo.

2.4.1

Definição do problema

Primeiramente, considera-se que a cena tridimensional S , a ser reconstruída, esteja contida em um subconjunto fechado $U \subset \mathbb{R}^3$, o qual é representado através de um conjunto de voxels \mathcal{V} , onde cada voxel $v \in \mathcal{V}$ ocupa um volume homogêneo do espaço e possui uma única cor. Identificamos S com sua representação discreta induzida por \mathcal{V} , com a exigência de que todo voxel $v \in S$ seja completamente opaco. Além disso, utilizamos a notação $cor(v, S)$ para representar a cor de um voxel v em uma cena S .

O conjunto de imagens é representado por $I = \{I_1, \dots, I_n\}$. A cor de um pixel p em uma imagem I_i é representada pela notação $cor(p, I_i)$. De maneira análoga, $C = \{C_1, \dots, C_n\}$ denota o conjunto de todas as câmeras a partir das quais o conjunto de imagens I foi obtido. Dado um pixel p de uma imagem I_i qualquer e uma cena S , chamamos de $S(p)$ um voxel de S que se projeta em p .

Definição 2.4.1 (Cena completa) *Uma cena S é completa em relação a um conjunto de imagens I quando, para toda imagem I_i e para todo pixel $p \in I_i$, existe um voxel v tal que $v = S(p)$ [44].*

Definição 2.4.2 (Cena consistente) *Uma cena completa S é considerada consistente em relação a um conjunto de imagens I quando, para toda imagem I_i e para todo pixel $p \in I_i$, $cor(p, I_i) = cor(S(p), S)$ [44].*

Definição 2.4.3 (Problema) *Seja \mathcal{V} o conjunto de voxels que corresponde à discretização do espaço $U \subset \mathbb{R}^3$ no qual S está contida. Além disso, seja dada uma coleção de imagens I obtidas a partir de um conjunto de câmeras calibradas C cujos centros de projeção se encontram em pontos cp_i tais que $cp_i \in \mathbb{R}^3 - U$. O problema de coloração do conjunto de voxels \mathcal{V} consiste em obter uma atribuição de cores a cada um dos elementos $v \in \mathcal{V}$ de tal forma que o conjunto \mathcal{V} , quando renderizado a partir de cada uma das câmeras C_i , reproduza as imagens I_i .*

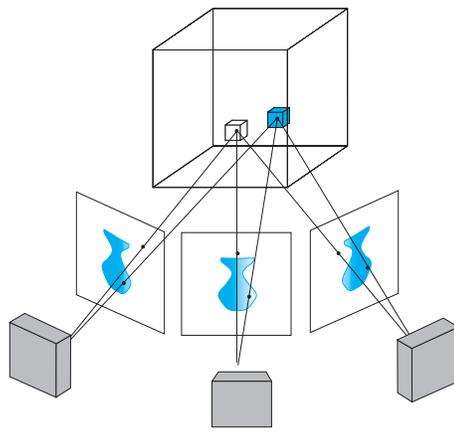


Figura 2.2: Coloração de voxels

Uma vez que o problema de coloração de voxels tenha sido especificado, devemos considerar as seguintes questões:

- Existe uma solução?
- A solução obtida é única?
- Como encontrá-la?

2.4.2 Restrições

Iremos primeiramente analisar em que condições o problema definido acima admite uma solução. O fato de considerarmos que cada voxel possui uma única cor associada nos leva a restrições importantes sobre as propriedades físicas da cena. De fato, só conseguiremos apresentar soluções para cenas cujas projeções de seus pontos possuam uma mesma cor em todas as imagens em que forem visíveis. Esta igualdade é definida a menos de erros introduzidos pelo sistema de aquisição como, por exemplo, ruído, *aliasing* e erros de calibração, os quais não levaremos em consideração neste momento. Logo, a reconstrução fica limitada a superfícies denominadas *lambertianas*, nas quais cada ponto emite a mesma radiação luminosa em todas as direções.

Em segundo lugar, para que a forma obtida possa reproduzir as imagens de entrada quando visualizada, é necessário que a discretização do volume seja feita com uma resolução compatível com as mesmas, isto é, as regiões associadas à projeção de um voxel devem ser aproximadamente iguais a um pixel. Isto pode ser resolvido, de forma aparentemente simples, através de um aumento arbitrário da resolução (mais tarde veremos que

o problema não é tão simples assim). Mais uma vez, com o intuito de simplificar a análise, iremos considerar, assim como no trabalho clássico de Seitz[44], que a discretização é realizada com uma resolução grande o suficiente para que um voxel possa ser aproximado satisfatoriamente através de seu centróide.

Supondo-se que a superfície da cena se comporte aproximadamente como uma superfície lambertiana, podemos perceber, de forma intuitiva, que sempre é possível apresentar uma solução, bastando para isso indicar o subconjunto de voxels de \mathcal{V} cujas projeções, nas imagens em que são visíveis tenham uma mesma cor.

2.4.3 Determinação da visibilidade

Uma questão importantíssima para a análise do método de coloração de voxels é a de como determinar a visibilidade dos voxels durante o processo de reconstrução. No caso geral, a visibilidade de um voxel v em relação a um ponto de vista associado a uma câmera específica C_i depende diretamente da ausência de voxels opacos entre o segmento de reta que liga V ao centro ótico de C_i . Logo, a determinação da visibilidade envolve, em um caso mais genérico, algum procedimento capaz de determinar a interseção entre o raio de projeção e os elementos que compõem a cena parcial em um dado instante do processo de coloração.

Uma forma de se evitar o tratamento da determinação da visibilidade por meios mais sofisticados é através do conhecimento de uma relação de ordem existente entre os voxels e o conjunto de câmeras consideradas.

Através de uma ordenação dos elementos de \mathcal{V} em relação ao sistema de câmeras, podemos percorrer os voxels no sentido dos mais próximos às câmeras para os mais distantes. Assim, a visibilidade dos voxels mais próximos as câmeras sempre fica determinada antes da visibilidade dos voxels mais distantes. Com base neste artifício não corremos o risco de que a visibilidade de um voxel específico venha a ser modificada posteriormente por uma operação de remoção de voxels, o que invalidaria as decisões já tomadas.

Com o objetivo de facilitar o processo de determinação da visibilidade dos voxels e para permitir que o algoritmo possa ser efetuado em um único passo, Seitz [44] propõe que o processo de reconstrução de cenas através de coloração de voxels seja restrito a configurações de câmera que satisfaçam uma restrição de ordenação da visibilidade.

Definição 2.4.4 (Restrição de ordenação da visibilidade) *Existe uma norma $\|\bullet\|$ tal que, para todos os pares de pontos Q e Q' pertencentes à cena e para todas as imagens de entrada I , Q oculta Q' , em relação ao conjunto I , somente se $\|Q\| \leq \|Q'\|$ [44].*

Infelizmente, esta norma nem sempre existe, o que significa que uma ordenação global nem sempre pode ser determinada. No entanto, existem casos práticos em que a restrição de ordenação de visibilidade pode ser satisfeita. Seitz mostra em seu artigo que sempre é possível determinar uma ordenação global dos voxels em relação ao sistema de câmeras se o volume ocupado pela cena se encontra totalmente fora do fecho convexo determinado pelos centros de projeção das câmeras.

2.4.4 Invariantes à coloração e unicidade

Considerando a questão da unicidade, podemos verificar que, infelizmente, existem várias soluções possíveis para um determinado conjunto de entrada para o problema. Podemos construir exemplos em que vários conjuntos diferentes de voxels, contidos no espaço que envolve a cena, podem ser capazes de reproduzir as imagens de entrada através de uma atribuição adequada de cores. Logo, precisamos adicionar alguma restrição ao problema de forma que a solução a ser obtida seja única.

Uma forma de garantirmos a unicidade da solução do problema é impor a condição de que esta venha a ser composta por elementos que satisfaçam alguma propriedade que seja *invariante* em todas as cenas consistentes.

Um exemplo de invariante é o chamado *invariante forte* o qual é formado por elementos que estejam presentes em todas as possíveis cenas consistentes. A grande dificuldade associada aos invariantes fortes é que estes são raros e nem sempre existem em número suficiente para obtermos uma reconstrução completa com relação ao conjunto de imagens de entrada.

Seitz propõe que, ao invés de trabalharmos com invariantes rígidos, os quais não nos garantem a existência de uma solução viável, devemos procurar *invariantes à coloração* os quais são definidos da seguinte forma:

Definição 2.4.5 (Invariante à coloração) *Um voxel v é invariante à coloração em relação ao conjunto de imagens de entrada I , se para todo par de cenas consistentes S e S' , se $v \in S \cap S'$ então $cor(v, S) = cor(v, S')$ [44].*

Em outras palavras, um voxel invariante à coloração não precisa estar presente em todas as cenas capazes de reproduzir os dados de entrada, contudo, sua cor deve ser sempre a mesma em cada uma delas em que estiver presente.

Identificaremos agora que elementos de \mathcal{V} satisfazem as condições necessárias para serem considerados invariantes à coloração.

Seja p um pixel pertencente a uma imagem I_i . Definimos o voxel v_p como o voxel mais próximo ao conjunto de câmeras dentre os voxels pertencentes ao conjunto $\{S(p) \mid S \text{ é uma cena consistente}\}$. Afirmamos que v_p é um invariante a coloração.

Prova 2.4.6 *Suponha que $v_p \in S$, onde S é alguma cena consistente; então $v_p = S(p)$. Isto é verdade já que se $v_p \neq S(p)$ então $S(p)$ seria mais próximo ao sistema de câmeras que v_p , o que é impossível pela definição de v_p . Logo, v_p tem sempre a mesma cor em todas as cenas foto-consistentes S em que estiver presente, pois $\text{cor}(v_p, S) = \text{cor}(S(p), S) = \text{cor}(p, I_i)$.*

Logo, uma coloração de voxels a partir de um conjunto de imagens I_1, \dots, I_m de uma cena S é definida como:

$$\bar{S} = \{v_p \mid p \in I_i, 1 \leq i \leq m\} \quad (2.4.1)$$

Em seu artigo[44], Seitz prova que uma coloração de voxels definida acima satisfaz as seguintes propriedades:

- \bar{S} é uma cena consistente.
- Todo $v \in \bar{S}$ é invariante à coloração.
- \bar{S} pode ser obtido a partir de qualquer conjunto C de câmeras que satisfaça a restrição de ordenação da visibilidade.

As provas de cada uma destas afirmações podem ser encontradas no trabalho original.

2.4.5

Cálculo da coloração de voxels

Quando a configuração das câmeras satisfaz a restrição de ordenação da visibilidade, podemos afirmar que a norma que mede a distância dos pontos no espaço às câmeras determina uma partição do conjunto de voxels \mathcal{V} em um conjunto de camadas determinada pelas seguintes expressões:

$$\mathcal{V}_C^d = \{v \mid \|v\|_C = d\} \quad (2.4.2)$$

$$\mathcal{V} = \bigcup_{i=1}^r \mathcal{V}_C^{d_i} \quad (2.4.3)$$

onde d_1, \dots, d_r é uma seqüência crescente de números reais.

O algoritmo consiste, então, em percorrer os voxels em cada uma das camadas, em uma ordem crescente de distância, projetando-os nas imagens em que estão visíveis e avaliando-os segundo sua consistência. Se um voxel projetado é considerado consistente então ele é colorido com base em uma função das cores obtidas das imagens em que se encontra visível. Caso contrário, atribui-se uma cor transparente, o que corresponde a removê-lo do modelo.

Na prática, a visibilidade pode ser calculada através da associação de um mapa de visibilidade Mv_i para cada imagem I_i (Figura 2.3). No início do algoritmo, atribui-se o valor 0 para cada uma das posições dos mapas de visibilidade. Quando um voxel v é avaliado, determina-se para cada I_i a projeção $proj_{I_i}(v)$ do centróide de v em I_i . Se o valor na posição dada por $proj_{I_i}(v)$ em Mv_i for igual a zero então o voxel é visível, caso contrário, ele está ocluso. Os mapas de visibilidade são atualizados cada vez que um voxel é considerado consistente, bastando para isso atribuir o valor 1 à posição associada à projeção do centróide de v em cada uma das imagens em que estiver visível.

Devido ao fato de que as imagens da cena não refletem uma cena completamente lambertiana, além da presença de artifícios causados por ruído e quantização, é necessário introduzir medidas estatísticas que meçam a verossimilhança da consistência de um voxel. Seitz propõe que a consistência do voxel seja decidida pelo teste da razão da verossimilhança baseada na estatística:

$$\lambda_k(V) = \frac{(n-1)s^2}{\sigma_0^2} \quad (2.4.4)$$

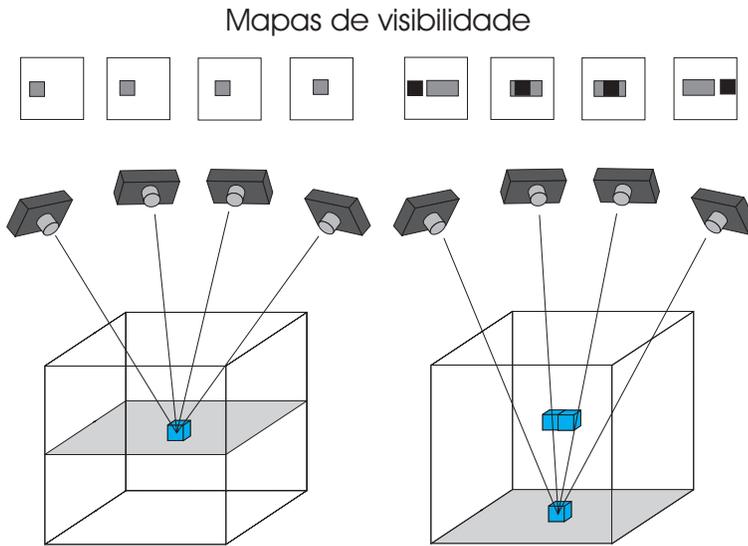


Figura 2.3: Determinação da visibilidade

onde k é o número de imagens em que o voxel está visível, s o desvio padrão calculado sobre o conjunto de cores em $proj_{I_i}(v)$, n a cardinalidade deste conjunto de cores e σ_0 o desvio padrão de uma distribuição normal representando os erros introduzidos pelo sensor. Sob hipótese de consistência, $\lambda_k(V)$ tem distribuição χ^2_{n-1} . Se $\lambda_k \geq T$, onde T é um limite global calculado na distribuição χ^2 para um nível de significância escolhido, então o voxel é considerado consistente, caso contrário é inconsistente e deve ser removido do modelo. Abaixo descrevemos o algoritmo de coloração de voxels (Algoritmo 1).

Algoritmo 1 Coloração de Voxels

```

 $S \leftarrow \emptyset$ 
para  $i = 1, \dots, r$  faça
  para cada  $V \in \mathcal{V}_C^{d_i}$  faça
    projete  $V$  em  $I_1, \dots, I_n$  e calcule  $\lambda_k(V)$ 
    se  $\lambda_k(V) \leq T$  então
       $S \leftarrow S \cup \{V\}$ 
    fim se
  fim para
fim para

```

Antes de encerrarmos a descrição do método proposto por Seitz, seria importante lembrar que uma outra hipótese foi assumida para que pudéssemos aplicar as idéias aqui descritas. Esta hipótese é a de que as colorações dos diferentes voxels sejam totalmente independentes, hipótese esta que permite que o problema possa ser resolvido através de simples consultas locais sobre a foto-consistência de cada voxel sem nos obrigar

a utilizar métodos mais sofisticados para determinar uma coloração que levasse em consideração a coerência espacial existente entre elementos vizinhos. Iremos em seguida rever a maior parte destas idéias dentro de um esquema formal, o qual define uma teoria sobre forma com base em foto-consistência.

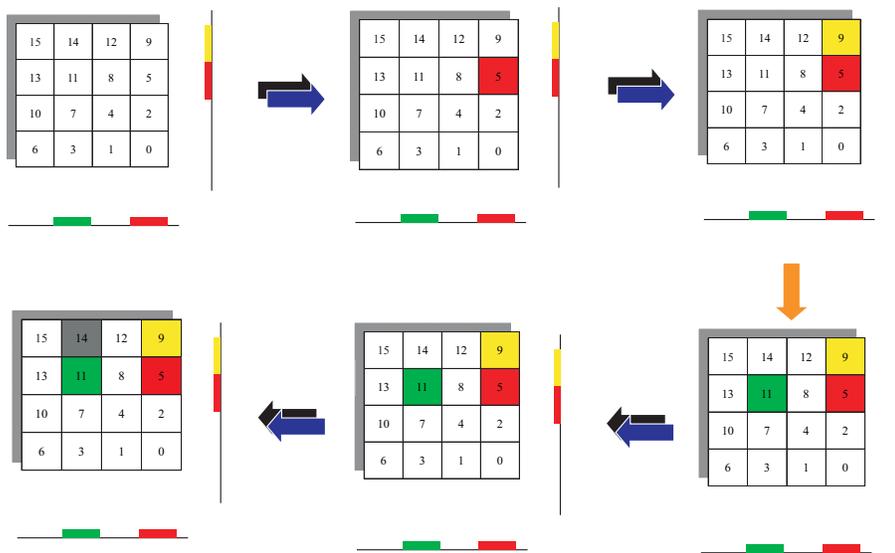


Figura 2.4: Exemplo de aplicação do método de coloração de voxels

2.5 Escultura do espaço (Space Carving)

A teoria de reconstrução de forma através de escultura do espaço, estabelecida por Seitz e Kutulatos [63], define precisamente o que podemos inferir sobre a forma de uma cena a partir de uma coleção de imagens obtidas em condições gerais e não controladas.

Apesar de boa parte das idéias apresentadas nesse trabalho já estarem presentes no trabalho sobre coloração de voxels de Seitz e Dyer, podemos destacar algumas diferenças fundamentais, como a apresentação de uma análise sobre reconstrução de cenas totalmente independente de algoritmos e a proposta de um método baseado na teoria subjacente, capaz de reconstruir cenas a partir de configurações de câmeras posicionadas arbitrariamente no espaço que contém a cena. Este novo algoritmo, denominado *Escultura do Espaço*, pode ser visto como uma generalização do algoritmo de Coloração de Voxels para configurações de câmeras arbitrárias.

Uma outra característica importante é a ênfase na reconstrução de cenas com base em um mínimo possível de restrições ou de conhecimentos

sobre a cena. Algumas restrições, utilizadas normalmente em métodos convencionais de reconstrução, não são levadas em consideração como, por exemplo, restrições sobre a geometria e topologia da cena, restrições sobre o posicionamento das câmeras, existência de feições específicas nas imagens de entrada (como pontos, arestas, cantos, linhas, contornos, textura) e conhecimento à priori de correspondências.

Nos trabalhos anteriores, jamais se cogitou a possibilidade de se resolver o problema de reconstrução de cenas sem supor ao menos uma das condições descritas acima. Porém, Kutulakos e Seitz demonstraram que isto pode ser feito nos casos em que a radiância dos pontos da cena a ser reconstruída pertence à classe de funções de radiância *localmente computáveis*, que definiremos nesta seção.

Toda a teoria de Kutulakos e Seitz é baseada na formulação do problema de reconstrução de formas como um problema de satisfação de um conjunto de restrições. É mostrado que um conjunto de fotos de uma cena rígida determina um conjunto de restrições que devem ser satisfeitas por qualquer cena que se projete nestas fotos. Kutulakos e Seitz, ao investigar a classe de equivalência das formas que reproduzem as imagens de entrada, provaram a existência de um membro especial desta classe, chamado *Photo Hull*, que pode ser calculado através do algoritmo denominado Escultura do Espaço e que engloba todas as possíveis formas foto-consistentes. A introdução do conceito de *Photo Hull* tem importância similar à introdução do conceito de *Fecho Visual* por Laurentini [31].

Veremos como as restrições impostas pelas imagens obtidas de uma cena determinam a classe de equivalência das formas que as reproduzem e como a relação existente entre elementos desta classe, e suas propriedades, nos permite especificar um algoritmo capaz de determinar o *Photo Hull*.

2.5.1

Restrições fotométricas

Seguindo a notação utilizada em [63], uma forma \mathcal{V} é definida como um conjunto fechado de pontos do \mathbb{R}^3 visto por um conjunto de n câmeras C_1, \dots, C_n em $\mathbb{R}^3 - \mathcal{V}$. Definimos a *radiância de um ponto* p na superfície $Surf(\mathcal{V})$ da forma com sendo a função $rad_p(\xi)$ que associa a cada raio orientado ξ a cor da luz refletida por p ao longo de ξ . Uma *descrição forma-radiância* nada mais é que uma forma \mathcal{V} a qual associamos uma função de radiância para cada ponto p em $Surf(\mathcal{V})$. Esta descrição é suficiente para reproduzir uma foto da cena a partir de qualquer ponto de vista.

Kutulakos afirma que uma imagem de uma cena 3D particiona o espaço de todas as possíveis descrições *forma-radiância* em duas famílias: aquelas que reproduzem a imagem e aquelas que não a reproduzem. Esta restrição é caracterizada pelos conceitos de foto-consistência hierarquicamente definidos abaixo.

Definição 2.5.1 (Foto-consistência de um ponto) *Seja U um subconjunto arbitrário do \mathbb{R}^3 . Um ponto $p \in U$ visível a partir de C_i é foto-consistente com a foto em C_i quando (i) p não se projeta em um pixel de fundo, e (ii) a cor na projeção de p é igual a $rad_p(p\vec{c})$. Se p não é visível a partir de C_i , então ele é trivialmente foto-consistente em relação à foto C_i . [63]*

Definição 2.5.2 (Foto-consistência de uma descrição forma-radiância) *Uma descrição forma-radiância de uma cena é foto-consistente com uma foto em C_i , quando todos os pontos visíveis a partir de C_i são foto-consistentes e todo pixel que não pertence ao fundo é a projeção de um ponto de \mathcal{V} . [63]*

Definição 2.5.3 (Foto-consistência de uma forma) *Uma forma \mathcal{V} é foto-consistente com um conjunto de fotos quando existe uma atribuição de funções de radiância aos pontos visíveis de \mathcal{V} que torna a descrição forma-radiância resultante consistente com todas as fotos. [63]*

Basicamente existem dois tipos de restrições determinadas pelas imagens de entrada sobre a forma de uma cena foto-consistente:

- restrições impostas pela segmentação fundo/objeto das imagens da cena.
- restrições impostas pela coerência entre as cores dos pontos da cena vistos pelas imagens.

A primeira restrição impede que um ponto de uma cena \mathcal{V} se projete em um pixel pertencente ao fundo de uma imagem. Logo, uma imagem I_i tomada de uma câmera com centro de projeção C_i restringe o espaço que contém a cena foto-consistente a um cone determinado pelos raios que passam pelos pontos na região delimitada pela silhueta do objeto na imagem. A interseção dos cones determinados por cada uma das imagens segmentadas define a envoltória visual, como já vimos anteriormente.

Infelizmente, a envoltória visual se degenera no \mathbb{R}^3 na ausência de informações provenientes de uma segmentação fundo/objeto, o que nos obriga a considerar o emprego de um outro tipo de restrição.

A restrição imposta pelas cores da cena vista pelas imagens só é válida quando a cena em questão emite cor de forma coerente. Isto é o caso quando a função de radiância definida sobre a superfície da cena pertence à classe das funções de radiância *localmente computáveis*.

Definição 2.5.4 (Função de radiância localmente computável)

Uma função de radiância é denominada localmente computável se o seu valor em um ponto específico independe dos valores nos demais pontos na superfície.

Isto significa que efeitos de iluminação global como inter-reflexões, sombras e transparência devem poder ser considerados irrelevantes.

Para cenas cuja função de radiância é localmente computável podemos supor a existência de um critério simples capaz de medir a de foto-consistência de um ponto da cena com base nas cores emitidas em cada imagem e suas respectivas direções.

Definição 2.5.5 *Um algoritmo $consiste_K()$ toma como entrada um conjunto de pelo menos $K \leq N$ cores, col_1, \dots, col_k , K vetores ξ_1, \dots, ξ_k e as posições das fontes de luz (caso não lambertiano), e decide se é possível para um único ponto da superfície emitir luz na cor col_i na direção ξ_i simultaneamente para todos os $i = 1 \dots K$. Assume-se que $consiste_K$ seja monotônico, o que significa que $consiste_K(col_1, col_2, \dots, col_j, \xi_1, \xi_2, \dots, \xi_j)$ implica em $consiste_K(col_1, col_2, \dots, col_{j-1}, \xi_1, \xi_2, \dots, \xi_{j-1})$ para toda permutação de $1, \dots, j$. [63]*

Através do critério de foto-consistência estabelecido anteriormente, podemos determinar quando uma cena descrita através de uma representação forma-radiância é consistente com o conjunto de imagens de entrada. Precisamos agora especificar um método capaz de construir uma possível forma foto-consistente em relação a uma coleção de imagens de uma cena. Isto será feito com base em uma análise de propriedades que relacionam uma forma com subconjuntos de si mesma e com as imagens de entrada.

Lema 2.5.6 (Lema da visibilidade) *Seja p um ponto sobre a superfície de \mathcal{V} , e seja $Vis_{\mathcal{V}}(p)$ a coleção de todas as fotos de entrada nas quais p não se encontra oculto devido à \mathcal{V} . Se $\mathcal{V}' \subset \mathcal{V}$ é uma forma que também contém p em sua superfície então $Vis_{\mathcal{V}}(p) \subseteq Vis_{\mathcal{V}'}(p)$. [63]*

Lema 2.5.7 (Lema da não-foto-consistência) *Se $p \in Surf(\mathcal{V})$ não é foto-consistente com um subconjunto de $Vis_{\mathcal{V}}(p)$, então ele não é foto-consistente com $Vis_{\mathcal{V}}(p)$. [63]*

Em poucas palavras, o *lema da visibilidade* e o *lema da não-foto-consistência* demonstram que a visibilidade e a não foto-consistência de uma cena expressam uma certa forma de monotonicidade. O lema da visibilidade indica que a visibilidade de um ponto na superfície de uma descrição de forma \mathcal{V} sempre aumenta à medida em que \mathcal{V} se torna menor. Analogamente, o lema da não foto-consistência afirma que uma parte da cena que não é foto-consistente com um subconjunto das imagens de entrada não pode ser foto-consistente com o conjunto completo. Estes dois lemas nos levam ao teorema do subconjunto o qual nos fornece a chave para um algoritmo incremental capaz de produzir uma forma foto-consistente.

Teorema 2.5.8 (Teorema do subconjunto) *Se $p \in Surf(\mathcal{V})$ não é foto-consistente, então nenhum subconjunto foto-consistente de \mathcal{V} pode conter p . [63]*

Prova 2.5.9 *Seja $\mathcal{V}' \subset \mathcal{V}$ uma forma que contém um ponto p . Se p está na superfície de \mathcal{V} então p tem que estar na superfície de \mathcal{V}' . Pelo lema da visibilidade segue-se que $Vis_{\mathcal{V}}(p) \subset Vis_{\mathcal{V}'}(p)$. Além disso, pelo lema da não-foto-consistência e pelo fato de que a radiância em p independe da radiância nos demais pontos, se p não é foto-consistente com em relação a $Vis_{\mathcal{V}}(p)$ então ele não pode ser foto-consistente com $Vis_{\mathcal{V}'}(p)$. Desta forma nenhum subconjunto de \mathcal{V} foto-consistente pode conter p .*

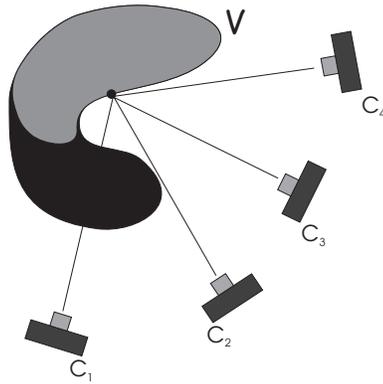


Figura 2.5: Teorema do subconjunto

Em resumo, o teorema do subconjunto indica que a não foto-consistência em um ponto de uma forma determina a não foto-consistência de uma família completa de formas.

2.5.2 O Photo Hull

A partir de uma estimativa inicial \mathcal{V} que contenha a cena a ser reconstruída, é possível produzir uma forma foto-consistente através da aplicação sucessiva de operações de remoção de voxels não foto-consistentes na superfície de \mathcal{V} . Iremos agora mostrar que uma forma foto-consistente assim obtida possui propriedades especiais. Na verdade, esta forma converge para o que chamamos de *Photo Hull*.

Teorema 2.5.10 (Teorema do Photo Hull) *Seja \mathcal{V} um subconjunto arbitrário do \mathbb{R}^3 . Se \mathcal{V}^* é a união de todas as formas foto-consistentes em \mathcal{V} então todo ponto na superfície de \mathcal{V}^* é foto-consistente. A forma \mathcal{V}^* é denominada Photo Hull.[63]*

Prova 2.5.11 (Por contradição) *Suponha que p é um ponto sobre a superfície de \mathcal{V}^* que não é foto-consistente. Se $p \in \mathcal{V}^*$ então existe uma forma foto-consistente, $\mathcal{V}' \subset \mathcal{V}^*$, que também contem p em sua superfície. Pelo teorema do subconjunto segue-se que \mathcal{V}' não é foto-consistente.[63]*

O teorema acima descrito estabelece a relação existente entre o *Photo Hull* e a classe de formas foto-consistentes com um conjunto de imagens de uma cena. Podemos afirmar que o *Photo Hull* determina um limite superior justo para todas as soluções foto-consistentes com um conjunto de imagens e pode ser vista como a maior forma foto-consistente que pode ser obtida.

Podemos observar agora que os invariantes à coloração utilizados no algoritmo de Coloração de Voxels são um caso particular do *Photo-Hull* para o problema em que as câmeras satisfazem a restrição da ordenação de visibilidade.

Apesar de ter sido provado aqui que o Photo Hull satisfaz os critérios de foto-consistência, não foi mostrado que o mesmo define um conjunto fechado satisfazendo assim a definição de forma especificada no início da seção. A prova de que o Photo Hull consiste em um conjunto fechado é apresentada no trabalho original. Para questões práticas basta sabermos que, no caso de representações discretas, o Photo Hull satisfaz as condições necessárias para ser considerado uma forma.

2.5.3

Algoritmo

O algoritmo de escultura do espaço é baseado no *Teorema do Subconjunto* o qual nos indica que, dado um volume inicial que contenha a cena, é possível determinar uma forma foto-consistente através da remoção incremental de elementos não foto-consistentes em sua superfície. Sabemos pelo *Teorema do Photo Hull* que este processo converge para a forma maximal desejada.

O algoritmo para a determinação do *Photo Hull*, descrito abaixo (Algoritmo 2), é similar ao algoritmo de coloração de voxels, excetuando-se o fato de que a questão da visibilidade não é mencionada explicitamente.

Algoritmo 2 Escultura do Espaço

Estabeleça um volume inicial \mathcal{V} que contenha a cena.

$fotoconsistência \leftarrow 1$

repita

$L \leftarrow Surf(\mathcal{V})$

repita

Escolha um voxel $v \in L$ ainda não visitado

Projete v em todas as imagens pertencentes a $Vis_L(v)$.

Determine a foto-consistência de v usando $consiste_K(col_1, col_2, \dots, col_j, \xi_1, \xi_2, \dots, \xi_j)$

se v não for foto-consistente **então**

$\mathcal{V} \leftarrow \mathcal{V} - \{v\}$

$fotoconsistência \leftarrow 0$

senão

$fotoconsistência \leftarrow 1$

fim se

até que $fotoconsistência = 0 \wedge L = \emptyset$

até que $fotoconsistência = 1$

$\mathcal{V}^* \leftarrow \mathcal{V}$

retorne \mathcal{V}^*

Implementação

O tratamento da visibilidade é uma questão importante no algoritmo de escultura do espaço, já que não admitimos restrição alguma sobre a configuração das câmeras.

Kutulakos propõe uma solução elegante para a determinação da visibilidade no caso de configurações genéricas a qual se baseia em várias iterações de um algoritmo denominado Varredura por planos, similar ao algoritmo de Coloração de Voxels. Tal algoritmo é aplicado a cada uma das seis direções

principais até que nenhuma remoção tenha sido efetuada, garantindo dessa forma a convergência do processo.

Primeiramente determina-se um volume inicial \mathcal{V} que envolve a cena a ser reconstruída, juntamente com um plano de varredura Π posicionado totalmente à frente de \mathcal{V} . Move-se Π em direção a \mathcal{V} até que Π corte um subconjunto de voxels de \mathcal{V} .

Em seguida, determina-se o conjunto de voxels na interseção de Π e \mathcal{V} . Para cada um dos voxels na interseção, calcula-se sua foto-consistência em relação ao conjunto de imagens em que se encontra visível.

O mesmo esquema de mapas de visibilidade adotado no algoritmo de coloração de voxels pode ser utilizado para manter a informação sobre a visibilidade dos voxels.

Os voxels que não satisfazem os critérios de foto-consistência são removidos; caso contrário são coloridos de acordo com as imagens em que estão visíveis. Após este passo, os mapas de visibilidade são atualizados convenientemente de forma que representem coerentemente a nova superfície visível.

Finalmente, efetua-se uma movimentação do plano em direção ao volume por uma distância de comprimento igual a uma das dimensões de um voxel, iniciando-se, logo em seguida, uma nova iteração. O processo termina quando o plano Π está posicionado atrás do volume inicial. Os passos que descrevemos encontram-se codificados em pseudo-linguagem no Algoritmo 3.

Através do algoritmo *Varredura por Planos* podemos calcular uma reconstrução parcial em relação a um subconjunto de câmeras que se posiciona atrás de um plano perpendicular a uma das direções principais. Contudo, isto não é suficiente para termos uma reconstrução global correta da forma da cena desejada.

Kutulakos propõe então que efetuemos uma seqüência de aplicações do algoritmo *varredura por planos* em cada uma das seis direções principais utilizando-se um conjunto de câmeras distinto, em cada aplicação, de forma a evitar que uma mesma câmera seja considerada repetidamente em varreduras realizadas em diferentes direções (Figura 2.6).

Ao final das varreduras em cada uma das seis direções principais determina-se o conjunto de voxels que foram avaliados em mais de uma varredura e efetua-se uma checagem de consistência considerando-se todo o conjunto de câmeras em que estão visíveis. O processo termina quando nenhuma remoção de voxels é efetuada em uma determinada iteração (ver Algoritmo 4).

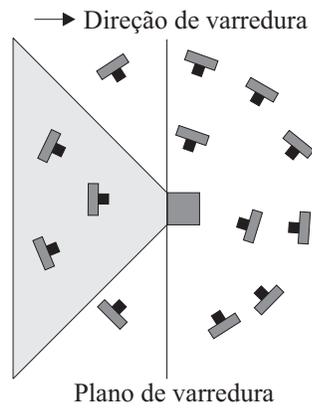


Figura 2.6: Câmeras consideradas em uma varredura para um determinado voxel.

A implementação da checagem de consistência pode ser efetuada de forma similar ao algoritmo de Coloração de Voxels. Uma análise interessante sobre estatísticas utilizadas para a checagem de consistência em algoritmo de escultura do espaço pode ser encontrada na tese de Broadhurst [72].

Algoritmo 3 Varredura por planos

Estabeleça um volume inicial \mathcal{V} que contenha a cena.

Posicione um plano de varredura Π à frente do volume inicial \mathcal{V} .

Mova Π em direção a \mathcal{V} até que eles se interceptem

repita

Calcule $\Pi \cap \mathcal{V}$

para cada voxel $v \in \Pi \cap \mathcal{V}$ **faça**

Sejam c_i, \dots, c_j as câmeras à frente de Π para as quais v se projeta em pixels não marcados.

Determine a foto-consistência de v usando $consiste_K(col_1, col_2, \dots, col_j, \xi_1, \xi_2, \dots, \xi_j)$

se v for inconsistente **então**

$\mathcal{V} \leftarrow \mathcal{V} - \{v\}$

senão

marque os pixels em que v se projeta.

fim se

fim para

Mova o plano um voxel em direção ao \mathcal{V}

até que \mathcal{V} esteja a frente de Π

Algoritmo 4 Escultura do Espaço por Múltiplas Varreduras

Determine um volume inicial \mathcal{V} que contenha a cena.

repita

Aplique o algoritmo *Varredura por Planos* em cada uma das seis direções principais e atualize \mathcal{V} adequadamente.

para cada voxel em (\mathcal{V}) cuja consistência foi avaliada em mais de uma varredura por planos **faça**

Sejam c_i, \dots, c_j as câmeras que foram utilizadas na checagem de consistência de v em alguma varredura de planos.

Determine a foto-consistência de v usando $consiste_K(col_1, col_2, \dots, col_j, \xi_1, \xi_2, \dots, \xi_j)$

fim para

até que nenhum voxel tenha sido removido em nenhuma das etapas

$\mathcal{V}^* \leftarrow \mathcal{V}$

2.6 Variações

Diversas variações sobre os métodos de Coloração de Voxels e Escultura do Espaço podem ser encontradas na literatura. Normalmente estas variações consistem em modificações e especializações com base em algum dos seguintes aspectos:

- tipo de representação utilizada (octree, voxels).
- tipo de função de consistência.
- tratamento da visibilidade.
- abordagem determinística ou probabilística.
- utilização de métodos de refinamento através de otimização.
- uso de espaços volumétricos alternativos.

Iremos apresentar aqui algumas das principais variações encontradas na literatura sobre Escultura do Espaço e Coloração de Voxels. Com exceção de Broadhurst e Cipolla [71], os quais introduziram mudanças significativas na modelagem do problema ao proporem uma abordagem probabilística, a maioria dos métodos mantém a estrutura básica proposta por Seitz, Kutulakos e Dyer.

2.6.1 Coloração de Voxels baseada em múltiplas hipóteses

O método de Coloração de Voxels baseada em múltiplas hipóteses foi proposto por Eisert et al. [55]. Uma hipótese nada mais é que uma possível atribuição de cor a um voxel específico. Diferentemente do método de coloração de voxels, esta técnica é capaz de lidar com quaisquer configurações de câmeras possibilitando reconstruções de cenas arbitrárias. Uma outra característica importante a ser mencionada é a ênfase de Eisert na combinação das técnicas baseadas em critérios de foto-consistência e na segmentação dos objetos de interesse nas imagens de entrada para obtenção de reconstruções de boa qualidade.

O método se subdivide em três etapas principais:

- Determinação do volume inicial que contém a cena.
- Atribuição de hipóteses a cada voxel do volume inicial.
- Verificação da consistência das hipóteses em relação à cada ponto de vista seguido de sua eventual remoção ou validação.

O primeiro passo consiste em determinar um volume que contenha a cena a ser reconstruída. Como no método de Seitz, o volume é determinado por uma caixa envolvente discretizada, gerando um conjunto \mathcal{V} de voxels onde cada um deles é referenciado através da notação v_{lmn} , onde l , m e n são índices em uma matriz tridimensional. Ao contrário da coloração de voxels, atribui-se inicialmente a cada voxel uma coloração transparente.

Na etapa seguinte, atribui-se a cada voxel um conjunto de hipóteses onde a k -ésima hipótese atribuída ao voxel V_{lmn} é dada por

$$H_{lmn}^k = (R(X_i, Y_i), G(X_i, Y_i), B(X_i, Y_i)) \quad (2.6.1)$$

onde (X_i, Y_i) são as coordenadas da projeção perspectiva do centro do voxel na i -ésima imagem e R, G e B especificam as componentes de cor associadas à hipótese. Uma hipótese H_{lmn}^k é atribuída a um voxel v_{lmn} , se a sua cor associada for consistente em relação a pelo menos duas imagens I_i e I_j . Nesta etapa não é levada em consideração a visibilidade dos voxels, já que a geometria ainda não é conhecida.

Finalmente, efetua-se um passo de remoção das hipóteses, levando-se em consideração a visibilidade de cada um dos voxels na cena em construção. Para cada ponto de vista, associamos um mapa de visibilidade que guarda o índice do voxel visível para cada pixel.

Os voxels não transparentes (aos quais foram atribuídas hipóteses) na superfície do volume envolvente são tomados como uma estimativa inicial da geometria da cena. Então, para cada vista separadamente, é feita uma checagem de consistência das hipóteses de cada voxel na superfície visível, isto é, a cor de cada hipótese é comparada com a cor da projeção do voxel visível pela câmera associada ao ponto de vista corrente. Se a diferença entre a cor associada à uma hipótese e a cor do pixel correspondente à projeção do voxel for maior que um certo valor arbitrário, então a hipótese é descartada. Se não restar hipótese alguma para um determinado voxel então associamos a ele uma cor transparente e atualizamos a superfície visível em relação àquela vista. Efetua-se o mesmo processo várias vezes para cada vista até que não haja nenhuma remoção de hipóteses. O conjunto de voxels opacos (não transparentes) resultante determina a coloração de voxels desejada.

Uma diferença fundamental entre o método de Seitz e o método de Eisert é que no método de Seitz a checagem de consistência é feita considerando-se todas as imagens simultaneamente, enquanto que no método de múltiplas hipóteses checa-se a consistência em relação a uma única imagem por vez. A estratégia do método de múltiplas hipóteses simplifica a determinação

das superfícies de visibilidade já que elas são calculadas em relação a um único ponto de vista através de uma simples varredura no sentido da menor distância para a maior distância em relação às câmeras. Por outro lado é necessário um processamento adicional já que todos os voxels devem ser testados, inclusive aqueles pertencentes ao interior do volume.

2.6.2 GVC(Generalized Voxel Coloring)

O método GVC [57] consiste em uma implementação alternativa para o algoritmo de escultura do espaço no qual a visibilidade é calculada de forma exata, isto é considerando-se todos os pontos de vista em que um voxel esteja visível durante a verificação de sua foto-consistência. É necessário esclarecer que a primeira versão do algoritmo de escultura do espaço [50], contemporânea ao surgimento do método GVC, não era capaz de lidar com a visibilidade de forma exata, utilizando somente um subconjunto das imagens em que um voxel está visível durante a checagem da foto-consistência. Somente quando Kutulakos e Seitz publicaram o trabalho intitulado *A Theory of Shape by Space Carving* [63] é que a questão do tratamento exato da visibilidade foi definitivamente incorporada ao algoritmo de Escultura do Espaço.

Existem duas variantes do algoritmo GVC, cada qual utilizando uma diferente estrutura de dados para lidar com o problema de determinação da visibilidade (Figura 2.7). A primeira delas, chamada GVC-IB, utiliza uma estrutura chamada buffer de itens (*item buffers*), a qual armazena, para cada pixel p de cada imagem $I_i \in I$, o voxel que está visível a partir de p . A segunda versão, denominada GVC-LDI, é mais sofisticada e armazena para cada pixel p de cada imagem $I_i \in I$ uma lista ordenada por profundidade de todos os voxels que se projetam em p . Esta estrutura na verdade é conhecida em computação gráfica por *Layered Depth Images*[51].

GVC-IB

Primeiramente, associa-se um identificador único a cada voxel. Em seguida, um buffer de itens é calculado para cada imagem através da rasterização de cada voxel por meio de um algoritmo similar ao *z-buffer*. Cada elemento do buffer de itens armazena um identificador e um valor de distância em relação à câmera associada. No momento em que um voxel é

rasterizado é possível determinar quais são os pixels que estão contidos em sua projeção.

A determinação do conjunto de pixels $Vis(v)$ para os quais um voxel v está visível é simples. Basta projetar v na imagem e comparar o identificador no buffer de itens associado a cada pixel na projeção de v com o identificador de v . Os pixels para os quais for verificada a igualdade entre os identificadores são inseridos no conjunto $Vis(v)$ sobre o qual é calculado a foto-consistência.

A operação de remoção de voxels torna os buffers de itens desatualizados, o que leva a necessidade de recalculá-los freqüentemente. Como isso é um procedimento custoso, deixamos os buffers de itens desatualizados na maior parte do tempo, o que não causa dano algum ao processo, já que a informação armazenada sobre a visibilidade indicará que alguns voxels estarão visíveis em apenas um subconjunto do conjunto total de câmeras onde a visibilidade é verificada. Como o algoritmo de coloração de voxels é conservativo, jamais irá remover voxels que façam parte da cena. No final do processo, nenhuma remoção ocorrerá e a consistência será calculada usando as informações atualizadas e exatas sobre a visibilidade dos voxels.

GVC-LDI

O GVC-LDI funciona de maneira similar ao anterior porém, como armazenamos uma lista completa de todos os voxels que se projetam em um determinado pixel de uma imagem, então a atualização dos voxels visíveis, após operações de remoção, é imediata, bastando para isso retirar o elemento da cabeça da lista associada aos pixels afetados.

A vantagem do GVC-LDI é a de que as informações sobre a visibilidade dos voxels está sempre atualizada, o que nos leva a remover voxels mais rapidamente que o GVC-IB. Por outro lado, a estrutura de dados utilizada no GVC-LDI é extremamente grande e requer um espaço de armazenamento muito maior que a estrutura utilizada pela versão GVC-IB.

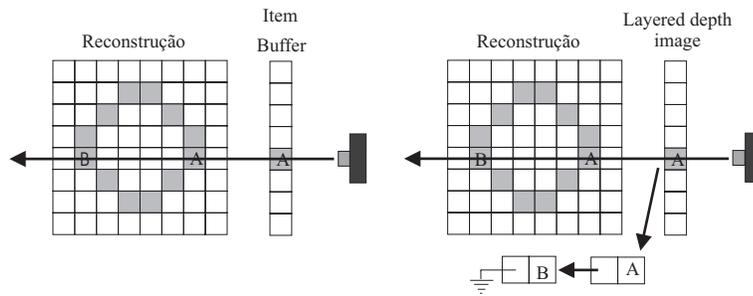


Figura 2.7: GVC-IB e GVC-LDI

2.6.3 Multiresolução de Prock

A tarefa de reconstruir cenas através de imagens requer um grande esforço computacional mesmo quando realizada através de técnicas simples e diretas como a coloração de voxels. Apesar de possuírem complexidade $O(nm)$, onde n é o número de voxels e m o número de imagens, os algoritmos de coloração de voxels envolvem um processamento árduo devido ao fato de trabalharem sobre dados volumétricos, os quais são sempre bastante grandes.

A. Prock e C. Dyer foram os primeiros a enfrentar o desafio de reduzir o tempo de processamento envolvido nos métodos de coloração de voxels, tendo como objetivo aproximar o tempo total de processamento ao necessário para reconstruções em tempo real [52].

Apesar de não alcançarem totalmente seu objetivo, já que o método proposto leva em torno de 0.5s a 1s para efetuar uma reconstrução, podemos afirmar sem dúvida que foi um grande avanço em relação ao algoritmo clássico proposto por Seitz em direção a reconstruções em tempo real.

A idéia de Prock para reduzir o tempo total de processamento é baseada em três idéias chaves:

- uso de texturização em hardware.
- aproveitamento da coerência espacial.
- aproveitamento da coerência temporal.

Texturização em hardware

A cada iteração do algoritmo de coloração de voxels necessitamos projetar centenas de milhares ou às vezes milhões de voxels pertencentes a uma camada do volume. Uma abordagem mais eficiente consiste em aproximar

uma camada de voxels por um plano, o qual é projetado como um todo sobre as imagens de entrada. Isto pode ser feito de forma mais eficiente via texturização, através da qual projetamos as imagens de entrada sobre o plano correspondente a uma camada de voxels como se fossem verdadeiros *slides*. O procedimento de texturização não introduz nenhum custo adicional, já que pode ser efetuado eficientemente através das facilidades existentes nas placas gráficas aceleradas por hardware.

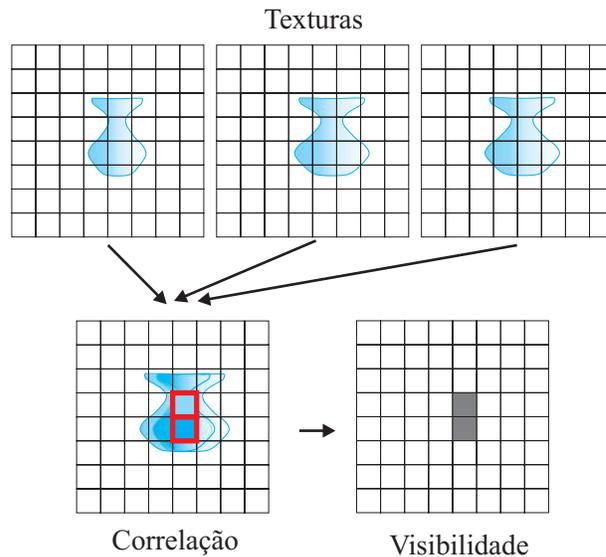


Figura 2.8: Projeção das texturas no espaço da cena

Refinamento adaptativo

A principal motivação para a utilização de reconstrução por refinamento adaptativo se baseia no fato de que, apesar de representarmos o espaço que contém a cena através de uma representação volumétrica enumerativa, estamos, na realidade, interessados em determinar a superfície que envolve o volume de ocupação da cena. A superfície que determina a forma da cena, mesmo que representada através de voxels, contém um número de elementos pequeno quando comparado ao número de elementos presente no espaço de busca inicial.

Por este motivo, uma maior eficiência pode ser obtida se pudermos concentrar esforços especificamente nas regiões do espaço em que a superfície da cena está contida, ao mesmo tempo em que minimizamos o número de operações nas regiões consideradas vazias.

Obviamente, por não conhecermos a cena, não podemos determinar, logo de início, quais são estes espaços vazios, de tal forma que possamos efetuar de imediato uma subdivisão espacial adaptativa.

Prock propõe uma estratégia em que o algoritmo de coloração é aplicado sobre um conjunto inicial de voxels em baixa resolução sendo o resultado assim obtido fornecido como entrada para um processo de refinamento. O processo de refinamento consiste em subdividir cada voxel consistente em oito novos voxels, os quais são armazenados em uma lista. Percorre-se cada elemento da lista, na ordem determinada pela subdivisão em camadas do volume em função de sua distância ao conjunto de câmeras, verificando-se se cada um deles deve ser removido ou colorido. Em seguida, aplica-se o algoritmo de refinamento recursivamente sobre os elementos da lista que foram considerados consistentes.

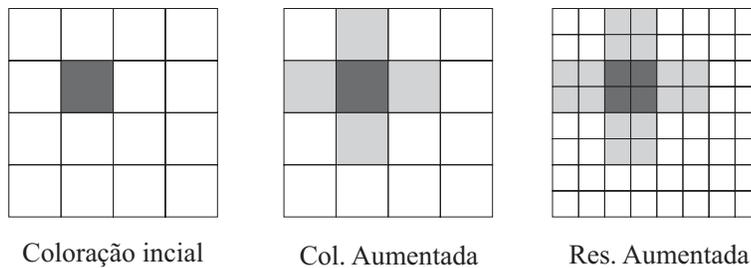


Figura 2.9: Refinamento

Através desta representação em diversos níveis de resolução, podemos tornar o algoritmo muito mais eficiente, ainda que preservando todas as facilidades de manipulação fornecidas por uma representação volumétrica enumerativa.

Uma questão importante a se considerar é de que ao diminuirmos a resolução do espaço de representação corremos risco de perder partes da cena. Isto ocorre já que, quando diminuirmos a resolução, estamos aumentando a região delimitada por um voxel. Isto aumenta a probabilidade de que as projeções de cada um destes elementos contêm mais pixels não consistentes, o que aumenta a possibilidade de que um voxel venha a ser descartado, apesar de conter fragmentos da superfície da cena em seu interior. No trabalho original podemos encontrar uma estimativa do quanto da cena é perdido ao se reduzir a resolução do espaço que contém a cena.

Prock, baseando-se na existência de uma coerência espacial entre partes vizinhas da cena, propõe uma heurística para evitar este problema. A idéia consiste em subdividir não só cada voxel v considerado consistente mas também todos aqueles que estiverem em uma vizinhança de v . De forma

a não introduzir, no processo, um custo adicional muito grande, utiliza-se uma vizinhança de norma unitária.

Ao adotar uma estratégia de refinamento em que os voxels podem ter dimensões arbitrárias, torna-se claro que a aproximação de um voxel por seu centróide não mais é adequada. Prock adota uma aproximação determinada por uma região planar paralela à direção de varredura passando pelo centro do voxel. Apesar de ser uma aproximação melhor que a anterior, já que leva em consideração que a projeção de um voxel nas imagens é dada por uma região poligonal e não por um ponto, ainda temos problema quando o voxel tem todas as três dimensões não desprezíveis.

Resolução	Original	Multiresolução	<i>Speedup</i>
32	0.9291s	0.7509s	1.23
64	5.777s	1.512s	3.82
128	48.33s	4.093s	11.8
256	335.8s	15.70s	21.4
512	2671s	64.98s	41.1

Tabela 2.2: Coloração de Voxels em Multiresolução x Coloração de Voxels Original. Os experimentos de Prock foram realizados em uma SGI O2 R5000 de 200 Mhz.

Uma limitação importante do método de Prock é a necessidade de se iniciar o algoritmo com uma boa estimativa da cena. De certa forma, a estimativa inicial deve conter uma boa "semente", capaz de gerar a cena completa com todos os detalhes significativos à medida em que é refinada nos estágios posteriores. Caso alguma parte da cena não esteja presente na estimativa inicial, então este erro será propagado para os demais estágios de refinamento, causando um comprometimento do restante do processo.

Tratamento de cenas dinâmicas

Prock foi além da reconstrução de cenas estáticas e ousou propor uma abordagem para a reconstrução de cenas dinâmicas. Utilizando as técnicas descritas acima, Prock explora a existência de coerência temporal entre quadros consecutivos para tornar ainda mais eficiente o processo de reconstrução.

Sua estratégia consiste em utilizar uma reconstrução em baixa resolução obtida a partir de um conjunto de quadros em um instante de tempo t_k como ponto de partida para reconstrução da cena em um instante t_{k+1} .

Assim como há problemas de perda de elementos em cenas estáticas devido à redução da resolução, também há problemas semelhantes e de certa maneira ainda mais graves quando se trata de cenas dinâmicas. Infelizmente, se os objetos se moverem muito rapidamente através da cena, então a reconstrução em baixa resolução fornecida como volume inicial para a reconstrução em um instante posterior pode não conter partes importantes. Isto pode ser resolvido, com um pouco mais de dificuldade através de um processo de acompanhamento (*tracking*) do movimento dos objetos de interesse de forma a adicionar regiões ao volume inicial para que este leve em consideração o movimento dos objetos.

Ao invés de utilizar um procedimento de *tracking* sofisticado, Prock sugere a utilização da mesma heurística de busca local em uma vizinhança adotada para o caso de cenas estáticas. No entanto, a utilização desta estratégia fica limitada a cenas cujos objetos não tenham velocidade superior a um certo limiar. Alguns resultados da aplicação desta técnica sobre cenas dinâmicas são apresentados no trabalho original.

2.6.4

Visão estéreo aproximada

Neste trabalho, Kutulakos [65] propõe uma solução para o problema de visão estéreo aproximada através de vários pontos de vista (*approximated n-view stereo*) o qual consiste em recuperar uma seqüência de reconstruções volumétricas que aproximam incrementalmente uma cena 3D com forma arbitrária.

A solução apresentada para este problema é baseada em uma nova função de consistência baseada em uma transformação denominada *shuffle transform*.

Definição 2.6.1 (Shuffle Transform) *Uma transformação 2D $T : I_1 \rightarrow I_2$ que leva os pixels de uma imagem I_1 nos pixels de uma imagem I_2 é uma r -Shuffle transform, se e somente se, podemos encontrar para cada pixel $j \in I_2$ um pixel $i \in I_1$ com a mesma cor que j em um raio de vizinhança r em torno das coordenadas de j . A constante r é denominada raio de dispersão da transformação T .*

Em outras palavras uma *Shuffle Transform* nada mais é que uma reorganização dos pixels de uma imagem dentro de um raio de vizinhança limitado. Além de representar transformações paramétricas, uma *shuffle transform* também é capaz de representar transformações envolvendo descarte e

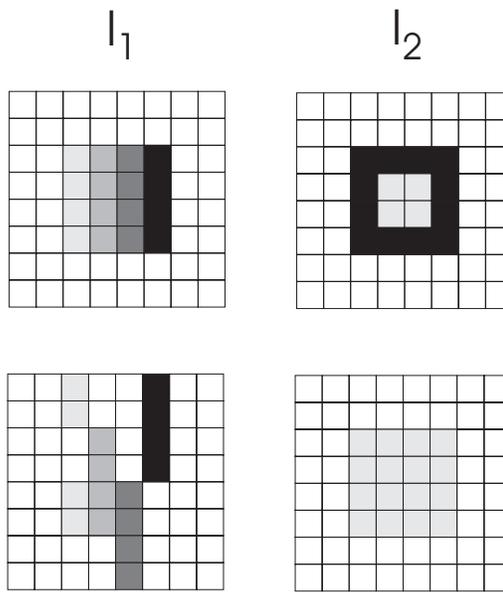


Figura 2.10: Exemplos de shuffle transforms

substituição de elementos, como podemos ver na coluna à direita na figura 2.10.

Através de *shuffle transforms* podemos definir um novo tipo de consistência que nos permite caracterizar uma família de formas consistentes com as imagens, a menos de uma transformação.

Definição 2.6.2 (r-Consistência) *Um volume \mathcal{V} é r-consistente se para toda imagem de entrada I_i existe uma r-shuffle $T : I_i \rightarrow I'_i$ que torna \mathcal{V} consistente com as fotos I'_i, \dots, I'_m .*

O restante do método é similar ao método de escultura do espaço. A única diferença está na utilização da função de consistência baseada em *shuffle transforms*. Na prática, a checagem de consistência é implementada através de um algoritmo capaz de determinar a existência de pelo menos um pixel com a mesma cor em cada um dos discos de raio r em torno das projeções de um voxel específico v . Quando esta existência é confirmada então dizemos que as cores associadas à projeção de um voxel v são consistentes a menos de uma *r-shuffle*.

Kutulakos afirma que seu método é aplicável à reconstrução na presença de erros de calibração, trata adequadamente a questão da utilização de imagens discretas e permite reconstruções incrementais, chegando mesmo a definir um *Espaço de Escala Foto Consistente*.

Apesar da enorme versatilidade do método proposto algumas questões importantes foram deixadas em aberto:

1. como determinar o raio de dispersão mínimo que leva a uma reconstrução válida.
2. como aplicar as técnicas ao problema de reconstrução/autocalibração a partir de n imagens.
3. como desenvolver um algoritmo adaptativo em multiresolução através do qual diferentes partes de uma mesma cena possam ser aproximadas por elementos de diferentes escalas.

2.6.5

Outras Variantes

Uma variação bastante importante dos métodos de escultura do espaço/coloração de voxels se baseia em procedimentos de otimização para o refinamento de uma solução inicial através da minimização de *erros de reprojeção*. O erro de reprojeção é compreendido como o erro determinado pelas diferenças entre as imagens de entrada e as imagens geradas pela renderização da reconstrução obtida a partir dos mesmos pontos de vistas originais.

O objetivo destes métodos é lidar com o problema de que, em alguns casos, um único limiar (*threshold*) não é suficiente para se obter boas reconstruções, por exemplo, quando a cena não se comporta perfeitamente como uma superfície lambertiana. O que ocorre é que com o aumento do limiar pode-se obter soluções de baixa qualidade, por outro lado, com a redução deste mesmo valor, partes importantes da cena acabam sendo removidas.

O processo de refinamento por otimização utiliza *thresholds* especialmente adequados para cada uma das regiões da cena, gerando assim modelos que se ajustam melhor à forma desejada.

Diferentemente do que ocorre nos métodos mais convencionais, o processo de otimização pode tanto remover quanto adicionar voxels com o objetivo de minimizar o erro de projeção. Um exemplo deste método é o método de Slabaugh [67]. Um outro trabalho dentro desta classe de métodos é o trabalho de De Bonet e Paul Viola que propuseram um método chamado *Roxels Algorithm*, o qual tenta determinar reconstruções de cenas com valores de opacidade arbitrários [58].

Outras variações importantes encontradas na literatura são as que utilizam espaços de voxels alternativos, como os trabalhos de Saito [59], Kimura [60] e Slabaugh [64]. Vale a pena mencionar também o trabalho

de Vedula [68] para cenas dinâmicas que utiliza um espaço de voxels 6D conectando dois espaços de voxels 3D consecutivos no tempo. Um pouco mais distantes em filosofia, mas também relacionados, são os trabalhos baseados em interfaces de propagação e Level Sets como [25, 53, 61].

Uma nova variação muito importante é a classe dos métodos probabilísticos de escultura do espaço [73, 66, 71, 72], os quais não serão descritos neste trabalho. Abaixo apresentamos uma pequena tabela que compara alguns dos métodos apresentados.

	câmeras em posições arbitrárias	multires.	cenas dinâmicas	prob.	visibil. exata
Coloração de Voxels	não	não	não	não	ok
Escultura do Espaço	ok	não	não	não	ok
GVC-IB	ok	não	não	não	não
GVC-LDI	ok	não	não	não	ok
Múlt. hipóteses	ok	não	não	não	ok
Coloração de Voxels em Multiresolução	não	ok	ok	não	ok
Shuffle Transforms	ok	possível	não	não	ok
Escultura do espaço Probabilística	ok	não	não	ok	ok

Tabela 2.3: Classificação dos algoritmos

3

Técnicas de visualização aplicadas à Escultura do Espaço

3.1

Introdução

No capítulo anterior apresentamos alguns dos principais conceitos sobre reconstrução volumétrica de cenas a partir de imagens, além de expor alguns dos principais métodos de coloração de voxels e escultura do espaço encontrados na literatura. Neste capítulo iremos descrever a aplicação de técnicas tradicionalmente utilizadas em síntese de imagens para o tratamento de algumas questões relevantes aos métodos de escultura do espaço. Especificamente, discutiremos a realização do processamento diretamente no espaço da cena, ao contrário da forma convencional, que trabalha no espaço das imagens.

Uma das vantagens em se utilizar técnicas de visualização para o tratamento de alguns dos subproblemas encontrados nos métodos de escultura do espaço é a de que estas técnicas, na maioria das vezes, se encontram implementadas em *hardware* altamente otimizado, permitido assim a construção de algoritmos concisos e eficientes.

Alguns dos trabalhos que também seguiram esta linha de abordagem foram o trabalho de Prock [52], o primeiro a sugerir a utilização de técnicas de síntese de imagens para o desenvolvimento de algoritmos eficientes para o problema de coloração de voxels e o trabalho de Sainz et al. [77] que, apesar de ter sido desenvolvido de forma completamente independente de nosso trabalho, possui com este vários pontos em comum, ainda que diferenças importantes possam ser encontradas, como veremos no próximo capítulo.

3.2

Alguns aspectos pouco explorados

Os primeiros trabalhos sobre a reconstrução de cenas baseados em escultura do espaço deram grande ênfase aos aspectos conceituais do proble-

ma, o que é natural em contribuições seminais [44, 50, 56]. O objetivo principal destes trabalhos era introduzir na literatura uma nova técnica, de fácil compreensão e de simples implementação, que pudesse se apresentar como uma nova alternativa ao conjunto de técnicas já existentes [37, 39, 40, 41, 42, 43] para a reconstrução de cenas a partir de imagens.

Em um segundo momento, enquanto alguns trabalhos continuaram a busca por um modelo teórico definitivo para o problema [55, 63], outros começaram a investigar questões um pouco mais específicas como, por exemplo, o uso de diferentes estatísticas na determinação da foto-consistência [71], o estudo de espaços volumétricos alternativos [58, 69, 70, 68] e a introdução de modelos probabilísticos [71, 72, 73]. Além disso, à medida em que a teoria sobre reconstrução a partir de imagens se tornou mais sólida, alguns problemas importantes do ponto de vista mais prático também começaram a ser focalizados, como o aumento de eficiência dos algoritmos [52], o uso de estruturas de dados específicas para manutenção das informações de visibilidade [57], a busca por robustez em face a erros de calibração [65] e o estudo de técnicas para o refinamento da reconstrução [67].

Entretanto, apesar de todo este esforço, alguns aspectos relevantes, até mesmo do ponto de vista conceitual, ainda permanecem pouco explorados. Como exemplos, podemos citar a relação existente entre o cálculo da foto-consistência e os problemas clássicos de reconstrução e amostragem de imagens, cuja compreensão é fundamental para que o resultado final do processo não seja afetado por erros provenientes de *aliasing*, o que muitas das vezes pode ser crítico quando somado aos erros inerentes à calibração e a ruído introduzido nas imagens pelos sistemas de aquisição.

Um outro aspecto não muito explorado é o de como integrar as informações sobre a segmentação do objeto nas imagens de entrada de modo coerente com as técnicas de síntese de imagens que aqui descreveremos. Mostraremos, no capítulo seguinte, que a integração destas informações pode ser extremamente útil ao processo de reconstrução, e, em alguns casos, fundamental, como no desenvolvimento de algoritmos adaptativos, o qual será abordado no capítulo 5.

Nas seções seguintes, iremos descrever alguns dos problemas encontrados nas implementações mais ingênuas do algoritmo de escultura do espaço, que se caracterizam basicamente por trabalhar no espaço das imagens, e discutir como estes problemas podem ser solucionados ao se trabalhar no espaço da cena, através da utilização de técnicas empregadas tradicionalmente em síntese de imagens.

3.3

Processamento no espaço das imagens

Como vimos no capítulo anterior, os algoritmos de escultura do espaço procuram determinar a forma de uma cena através da avaliação da foto-consistência de cada um dos voxels que compõem o volume de reconstrução inicial, o que é feito por meio do cálculo de uma medida estatística entre os pixels pertencentes às projeções destes mesmos voxels em cada uma das imagens. Logo, a projeção dos voxels, o registro e a rasterização de seus elementos constituem passos fundamentais para a determinação de quais são os elementos que fazem parte da cena a ser reconstruída.

Devido às transformações de câmera e de perspectiva, os voxels que compõem o volume de reconstrução inicial se projetam em forma de quadriláteros arbitrários, tornando não trivial tanto o registro quanto a rasterização dos pixels correspondentes.

Podemos contornar este problema ao assumir que os voxels são pequenos o suficiente para que possam ser aproximados por seus centróides; como resultado, a projeção de um voxel em uma imagem se reduz a um pixel, o que torna o registro praticamente imediato. Apesar de bastante simples, esta solução introduz problemas graves de amostragem, principalmente quando as imagens são geradas através de projeções perspectivas. Além disso, neste caso, o registro das informações de visibilidade com base em centróides não reflete corretamente a visibilidade dos voxels fazendo com que muitos dos que deveriam ser considerados não visíveis acabam sendo erroneamente tratados como visíveis.

O posicionamento das câmeras a distâncias arbitrárias também pode acrescentar uma nova variável ao problema, já que se torna difícil, neste caso, especificar uma resolução única para a discretização do espaço 3D de forma que a aproximação pontual dos voxels seja adequada para cada uma das câmeras utilizadas.

Como podemos perceber, o processamento no espaço das imagens desencadeia uma série de problemas que podem comprometer seriamente a qualidade da reconstrução obtida, e até mesmo sua correção, se cuidados necessários não forem tomados e se certas condições não forem consideradas.

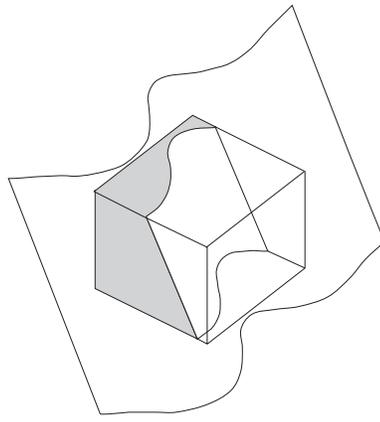


Figura 3.1: Superfície no interior de um voxel

3.4

Processamento no espaço da cena

Com o objetivo de evitar os problemas causados pelo processamento no espaço da imagem, Prock propôs que a escultura do espaço fosse realizada diretamente no espaço da cena.

Prock percebeu que da mesma forma como podemos trabalhar no espaço das imagens reprojando voxels nas imagens de entrada, podemos trabalhar no espaço da cena através da projeção das imagens de entrada sobre superfícies passando pelo interior dos voxels (Figura 3.1). Entretanto, para que possamos adotar esta estratégia, precisamos primeiramente especificar quais são as superfícies sobre as quais as imagens de entrada devem ser projetadas.

Infelizmente não temos a menor idéia sobre quais são estas superfícies, o que nos obriga a verificar a foto-consistência em todas as possíveis superfícies que passam por cada um dos voxels do volume de reconstrução inicial, o que é obviamente impossível.

Na verdade, poderíamos estimar as superfícies do objeto passando pelo interior dos voxels com base na triangulação dual da envoltória visual [78], porém, por enquanto, não iremos explorar esta possibilidade, a qual levaria a uma nova classe de algoritmos de reconstrução. Na seção sobre trabalhos futuros discutiremos mais sobre esta possibilidade. Um solução mais simples pode ser obtida através da aproximação da geometria de um fragmento de superfície, potencialmente foto-consistente, por um plano cujo posicionamento e orientação devem ser especificados convenientemente. Obviamente, temos total liberdade para a escolha destes parâmetros em cada voxel. No entanto, o processamento pode tornar-se extremamente caro

se for necessário projetar as imagens de entrada em um plano distinto para cada voxel que vier a ser avaliado. Logo, lucraremos muito mais se pudermos avaliar a foto-consistência de um conjunto de voxels com base na projeção do conjunto de imagens em uma única superfície.

Podemos alcançar este objetivo tomando a direção de varredura como a orientação dos planos que irão compor o conjunto de superfícies candidatas. Desta forma, um único plano determina todos os fragmentos de superfícies candidatas para um conjunto de voxels que se encontram a uma distância fixa das câmeras de entrada. Desta forma, um plano com mesma orientação da direção de varredura, devidamente transladado, aproxima uma camada de voxels induzida pela partição do espaço e pela função de distância em relação ao conjunto de câmeras.

Várias escolhas podem ser feitas com relação ao posicionamento, sendo que nenhuma delas é significativamente mais vantajosa que as demais. Na maioria das vezes opta-se pela posição central (passando pelo centro do voxel), já que é a posição mais neutra, ou a posição frontal (a qual corresponde à face frontal do voxel), que por sua vez, é a superfície mais próxima ao conjunto de câmeras consideradas em uma varredura, no caso do algoritmo de escultura do espaço, ou do conjunto completo de câmeras, no caso do algoritmo de coloração de voxels (Figura 3.2).

Desta forma, definimos o conjunto de superfícies de projeção como um conjunto de planos π_k , os quais denominamos *planos de referência*, com mesma orientação da direção de varredura e com distância $dist(k)$ crescente em relação ao conjunto de câmeras consideradas.

A interseção de cada π_k com o conjunto de voxels V determina, no espaço da cena, um conjunto de regiões planares $RP = \{rp_{ijk} | rp_{ijk} \in \pi_k \cap V\}$ sobre as quais a etapa de determinação da foto-consistência deve ser efetuada. Por enquanto, não nos preocuparemos em como a amostragem deve ser feita sobre as imagens projetadas em cada uma destas regiões pois estamos interessados apenas, no momento, em como as imagens de entrada podem ser registradas no espaço da cena.

Ainda que não seja exata, esta solução é bastante superior à aproximação pontual descrita anteriormente. Uma de suas vantagens é a de que, quando as imagens são projetadas em um plano de referência π_k , o registro entre elas é automaticamente determinado em cada uma das regiões planares associadas, já que cada uma destas regiões recebe a contribuição fotométrica correspondente às projeções dos respectivos voxels no espaço da imagem. Uma outra vantagem é que o registro da visibilidade também pode ser feito de forma correta, ao contrário do que ocorre com o processamento

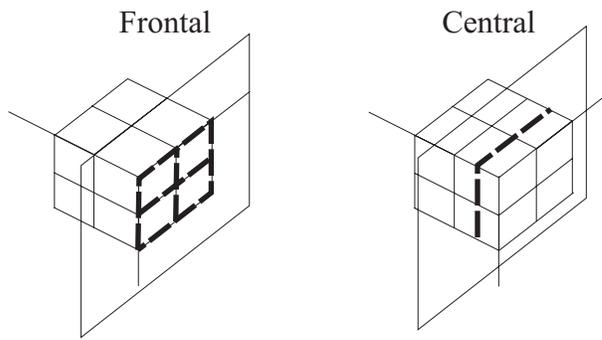


Figura 3.2: Planos de referência

no espaço das imagens.

As vantagens associadas à projeção das imagens no espaço da cena não se restringem somente à solução do problema de registro; ela também permite que um número grande de voxels possa ser processado de forma mais eficiente. Através desta estratégia podemos substituir a projeção e rasterização de milhões de voxels no espaço das imagens pela projeção e rasterização de apenas algumas centenas de imagens no espaço da cena, o que corresponde a um ganho considerável em eficiência.

Matematicamente, o mapeamento das imagens no espaço da cena corresponde a uma operação de *warping projetivo*, cuja complexidade pode ser proibitiva se considerarmos a utilização de técnicas de filtragem e anti-aliasing, mesmo quando implementada através de algoritmos que trabalham separadamente em cada uma das orientações naturais das imagens. Uma alternativa mais promissora consiste em efetuar o *warping projetivo* das imagens em hardware gráfico, através da projeção das mesmas como mapas de textura sobre a superfície determinada pelo plano de referência.

Esta técnica, conhecida como mapeamento projetivo de texturas, será utilizada neste trabalho para registrar, no espaço da cena, a maior parte das informações necessárias à reconstrução por escultura do espaço. Por este motivo, iremos dedicar algumas seções na descrição detalhada desta técnica, de seus principais problemas e de suas implicações diretas ou indiretas sobre os métodos de reconstrução que aqui serão apresentados.

3.4.1

Mapeamento projetivo de texturas

O *mapeamento projetivo de texturas*, introduzido por Segal et al. [28], é uma generalização do mapeamento de texturas clássico utilizado desde os primórdios da computação gráfica. Basicamente, sua idéia consiste em

projetar uma textura sobre uma superfície da cena, de forma análoga a uma projeção de slides (Figura 3.3). Várias são as suas aplicações, dentre as quais podemos destacar a produção de efeitos especiais de iluminação e a geração de sombras (*shadow maps*), os quais incrementam consideravelmente o realismo de cenas 3D produzidas sinteticamente (Figura 3.4).

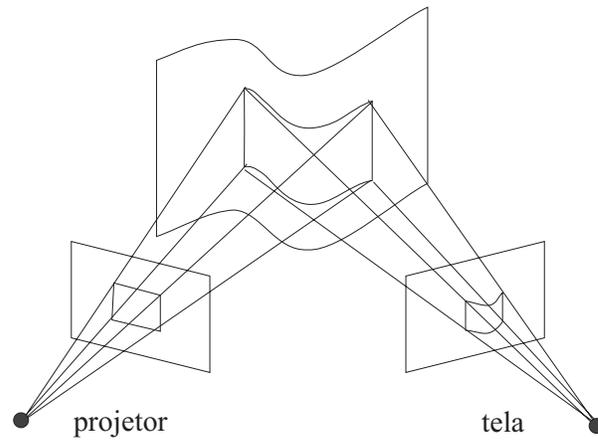


Figura 3.3: Mapeamento de textura sobre uma superfície

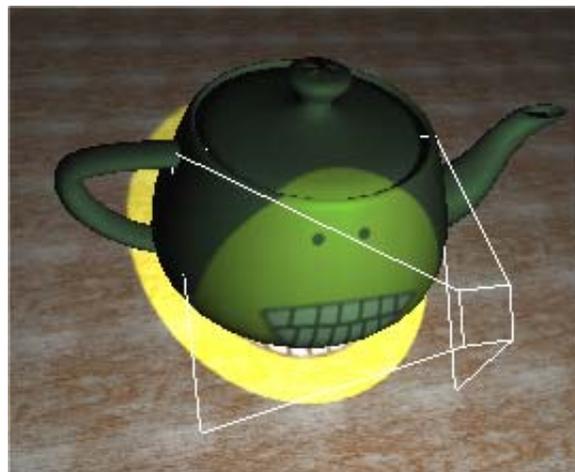
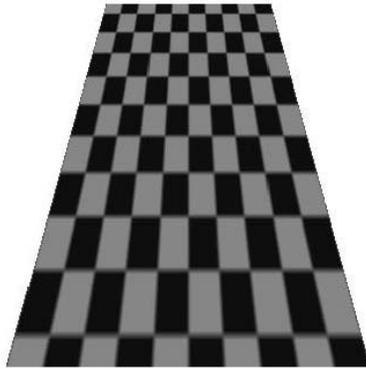


Figura 3.4: Mapeamento projetivo de textura

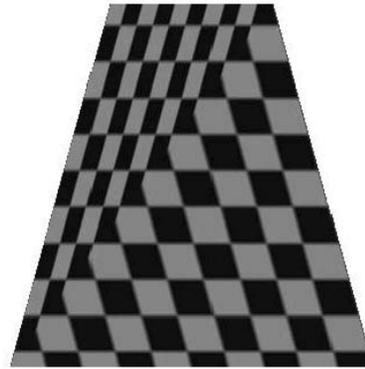
No mapeamento projetivo de textura, a relação entre as coordenadas no espaço de textura e as coordenadas no espaço do objeto são dadas por uma transformação projetiva, enquanto que no mapeamento de textura convencional esta relação é dada por uma transformação afim.

Consideremos, por exemplo, um mapeamento de textura projetivo bidimensional. Sejam (s, t, q) as coordenadas de textura homogêneas de um ponto p pertencente a uma região poligonal P , as quais são determinadas a partir da interpolação das coordenadas de textura homogêneas nos vértices de P . As coordenadas de textura reais correspondentes ao ponto p são obtidas então através da projeção das coordenadas de textura projetadas $(s/q, t/q, 1)$ no plano de suporte da textura. A coordenada q neste caso nada mais é que a distância do ponto em relação ao plano de suporte da textura.

As imagens abaixo ilustram as diferenças entre realizar a interpolação de coordenadas no espaço projetivo e no espaço real.



3.5(a): Mapeamento projetivo correto



3.5(b): Mapeamento projetivo incorreto

Para que o mapeamento projetivo possa ser realizado é necessário então atribuir coordenadas homogêneas de textura aos vértices que definem a primitiva a ser texturizada. A biblioteca OpenGL [1], por exemplo, adota o paradigma de projetor de slides sintético para representar o processo de mapeamento projetivo de textura. Isto significa que podemos definir uma matriz *modelview* e uma matriz de projeção a um projetor de slides que modela o efeito de um mapeamento projetivo de texturas, assim como fazemos na definição de uma câmera sintética. Desta forma, a relação entre o sistema de coordenadas do projetor e da câmera ficam determinadas de forma bastante consistente com a arquitetura da biblioteca.

Para que as coordenadas homogêneas fiquem completamente especificadas, a biblioteca OpenGL fornece um mecanismo para atribuir automaticamente coordenadas de textura aos vértices de uma determinada primitiva

com base em outros atributos de tais vértices. No caso de mapeamento projetivo de textura os dois modos que podem ser utilizados são o *eye linear texture generation* e o *object linear texture generation*, os quais atribuem coordenadas de textura aos vértices de um primitiva a partir de suas coordenadas no sistema de coordenadas da câmera e do objeto respectivamente. Em ambos os casos, cada uma das componentes das coordenadas de textura de um dado vértice são avaliadas através do valor fornecido por uma equação do plano no vértice em questão.

A atribuição automática de coordenadas de textura gera coordenadas de textura mesmo para pontos fora da região de projeção. Neste caso, como as coordenadas calculadas estão fora do domínio no qual a textura está definida, o processo de mapeamento de textura funcionará por replicação, o que significa que ele atribuirá as cores do elemento de textura mais próximo das coordenadas calculadas originalmente. Isto significa que a textura será estendida por toda a primitiva, o que não é adequado para os nossos propósitos. Para evitar este efeito é preciso criar uma borda transparente de modo que a textura projetada fique limitada a área de projeção do projeto sintético.

Este foi o principal motivo que nos levou a abandonar o procedimento de atribuição automática de coordenadas de textura da biblioteca OpenGL, já que seria necessário adicionar bordas transparentes às imagens originais, durante a criação da textura, para que o mapeamento desejado pudesse ser obtido. Optamos então por utilizar o próprio mapeamento de textura convencional do OpenGL o que, no entanto, fez com que fosse necessário determinar explicitamente as coordenadas dos vértices que definem a área projetada, no sistema de coordenadas do mundo, juntamente com as suas coordenadas de textura. As coordenadas de textura de x_t e y_t de cada vértice são determinadas de modo idêntico ao mapeamento de textura convencional, enquanto que a coordenada z_t deve ser igual a zero, uma vez que trabalhamos com texturas bidimensionais. A coordenada w_t por sua vez é determinada calculando-se a distância do vértice em relação ao plano de suporte da textura, o que equivale a calcular uma equação do plano no vértice em questão, exatamente como é feito nos modos *eye linear* e *object linear* do processo de atribuição automática de coordenadas de textura da biblioteca OpenGL. Para maiores detalhes ver [28].

3.4.2

Reamostragem em mapeamentos de textura

A reamostragem de uma mapa de textura é necessária porque, em geral, não há uma associação direta entre um elemento de textura (*texel*) e as coordenadas de textura associadas a um pixel da tela, uma vez que os reticulados definidos no espaço da tela e no espaço de textura podem diferir tanto em resolução quanto em geometria.

A forma mais ingênua de se efetuar este procedimento consiste em determinar, para um determinado pixel da tela com coordenadas p_s , o elemento do mapa de textura cujas coordenadas se encontram mais próximas das coordenadas de textura p_t e em seguida atribuir os valores das suas componentes de cor e opacidade às componentes do pixel correspondente.

Esta solução, apesar de simples, pode produzir efeitos indesejáveis, principalmente quando o mapa de textura apresenta regiões com frequências muito elevadas quando comparadas à taxa de amostragem. Este fenômeno, conhecido como *aliasing*, está intrinsecamente associado ao fato de precisarmos representar sinais contínuos de forma discreta em sistemas digitais.

Os aspectos relacionados à reconstrução e amostragem de imagens normalmente são estudadas dentro do contexto de *processamento de sinais*. De fato, o conceito de sinal é bastante adequado para o tratamento de imagens, já que estas podem ser vistas como grandezas fotométricas que variam em um subconjunto do espaço.

Por este motivo, iremos rever alguns dos principais conceitos e ferramentas utilizadas em processamento de sinais para que possamos efetuar uma análise sobre o processo de reamostragem que deve ser realizado durante o mapeamento projetivo de texturas.

Processamento de imagens como sinais

A grande maioria dos sinais provenientes do universo físico são de natureza contínua, como por exemplo, as imagens capturadas por uma câmera, o que nos obriga a encontrar métodos capazes de representá-los de forma adequada para que possamos processá-los em sistemas digitais, os quais, por definição, trabalham em domínios discretos.

Por esta razão, é fundamental, em processamento de sinais digitais, a análise dos procedimentos para obtenção de representações discretas de sinais contínuos, assim como dos procedimentos que buscam reconstituir um sinal contínuo a partir de representações discretas. Os primeiros procedimentos são conhecidos como *amostragem* e os últimos como *reconstrução*.

Dizemos que uma reconstrução é exata quando ela é capaz de reconstituir um sinal original, a partir de sua representação discreta, sem perdas de informação. Caso contrário dizemos que ela é uma reconstrução apenas aproximada.

A análise das relações entre a representação e a reconstrução de um sinal contínuo, e em que condições estes processos podem ser efetuados sem perda significativa de informação, é de suma importância para a área de processamento de sinais, e normalmente requer um tratamento mais rigoroso através de modelos e ferramentas matemáticas.

Modelos matemáticos para sinais

Um sinal pode ser modelado matematicamente através de uma função da forma $f : R^n \rightarrow V$, isto é, através de uma função que varia em um espaço n -dimensional tomando valores em um espaço vetorial.

Definição 3.4.1 Um espaço de sinais S é um subespaço vetorial do espaço de funções $f : U \in R^n \rightarrow V^m$, no qual U, V e m são fixados.

Os *modelos funcionais* são os modelos matemáticos mais adequados ao estudo de sinais, já que são capazes de capturar, com bastante naturalidade e precisão, o conceito de grandezas que variam ao longo de um determinado domínio. É importante ressaltar que isto é válido apenas nos casos em que os sinais correspondem a processos determinísticos; quando a variação da grandeza associada ao sinal ocorre de maneira não determinística, então a modelagem através de *processos estocásticos* é a mais adequada.

O modelo matemático de espaços de sinais é capaz de representar a maior parte dos sinais encontrados no universo físico. Entretanto, alguns sinais importantes não podem ser modelados através de funções, como é o caso dos *impulsos*, os quais precisam ser modelados através de formalismos mais sofisticados, como o das *distribuições*, que por sua vez, são generalizações matemáticas do conceito de função. Apesar desta dificuldade é possível, sem um envolvimento técnico muito rigoroso, aproximar um impulso através do limite de uma seqüência de sinais pulso [46].

Definição 3.4.2 Um sinal pulso pode ser modelado como uma função

$$p_a(t) = \begin{cases} 0 & t > |a|, \\ 1 & t \leq |a| \end{cases} \quad (3.4.1)$$

Assim, podemos definir um sinal impulso, também chamado δ de *Dirac* através do seguinte limite:

$$\delta(t) = \lim_{n \rightarrow \infty} \frac{1}{n} p_{1/n}(t) \quad (3.4.2)$$

O sinal impulso tem um papel fundamental na análise e descrição de sinais, pois podemos caracterizar qualquer função $f(t)$ através de uma seqüência de impulsos devidamente modulados e transladados.

$$f(x) = \int_{-\infty}^{+\infty} f(t) \delta(x - t) dt \quad (3.4.3)$$

Os modelos funcionais diferem entre si de acordo com a interpretação associada ao domínio e ao contradomínio das funções utilizadas na modelagem. Os modelos funcionais mais comumente utilizados em processamento de sinais são os modelos *espaciais* e modelos *espectrais* (ou de freqüências).

Nos *modelos funcionais espaciais*, como o próprio nome indica, o domínio representa a região do espaço ou intervalo de tempo em que a grandeza varia, enquanto que o contradomínio representa o conjunto de valores que ela pode assumir. Por exemplo, um modelo funcional para uma imagem na tela de um computador pode ser descrito através de uma função $f : U \subset R^2 \rightarrow R^3$, onde o domínio é um subconjunto do plano e o contradomínio é formado por um espaço vetorial tridimensional que caracteriza um espaço de cor formado pelas componentes vermelha, verde e azul.

Os *modelos funcionais espectrais*, apesar de serem um pouco menos intuitivos, estão diretamente relacionados à forma como percebemos a grande maioria dos sinais provenientes do universo físico, já que o sistema sensorial humano percebe sinais, como por exemplo, sons e cores, através de diferentes freqüências. Existem vários modelos espectrais dependendo dos operadores utilizados para detectar e analisar as freqüências dos sinais de interesse. O modelo espectral mais comum é o estabelecido com base na chamada *transformada de Fourier*.

Transformada de Fourier

Definição 3.4.3 *A Transformada de Fourier consiste em um operador unário, $F : S \rightarrow S'$, entre dois espaços de sinais, caracterizado por um núcleo periódico $e^{-2\pi i w x}$ com freqüência w , capaz de detectar as freqüências de um sinal não necessariamente periódico.*

$$\hat{f}(w) = (Ff)(x) = \int_{+\infty}^{-\infty} f(x)e^{-2\pi iwx} dx \quad (3.4.4)$$

Podemos entender a transformada de Fourier como um operador que mede a densidade de uma determinada frequência w em um sinal $f(x)$, visto que a integral do produto $f(x)e^{-2\pi iwx}$ retorna uma medida da ressonância entre as componentes exponenciais de $f(x)$, com frequência w , e o núcleo exponencial da transformada.

A transformada de Fourier é inversível, sendo sua inversa dada por

$$f(x) = (F^{-1}f)(w) = \int_{+\infty}^{-\infty} \hat{f}(w)e^{2\pi iwx} dw, \quad (3.4.5)$$

o que significa que uma sinal pode ser reconstruído através de uma soma infinita de sinais exponenciais periódicos com frequência w e amplitude $\hat{f}(w)$.

Uma das desvantagens da transformada de Fourier é sua incapacidade de indicar a localização espacial de uma determinada frequência w existente em um sinal s . Quando isto é necessário precisamos adotar outros operadores como por exemplo, aqueles associados às *transformadas de Fourier com janelas* ou às *transformadas de wavelets* [54].

A Transformada de Fourier aqui apresentada é definida no domínio contínuo, porém sabemos que, na prática, os sinais devem ser representados discretamente para que possam ser processados em sistemas digitais, o que implica que, para analisar sinais desta natureza, precisamos trabalhar com versões discretas desta mesma transformada. Para a exposição que faremos a seguir, a transformada de Fourier contínua é suficiente, bastando para isso, termos em mente que estamos analisando sinais originalmente contínuos. Para maiores detalhes sobre transformadas de Fourier discretas ver [46].

Operações com sinais

Ao trabalharmos com espaços de sinais, obviamente estamos interessados em caracterizar as operações que sobre eles podem ser efetuadas.

Definição 3.4.4 *Uma operação em um espaço de sinais é um mapeamento $T : R^m \times S^n \rightarrow S'$ em um espaço de sinais S' , possivelmente distinto.*

Por definição, um espaço de sinais herda as operações fundamentais de um espaço vetorial: a operação de soma $S \times S \rightarrow S$, dada por $(f + g)(t) = f(t) + g(t)$ e a operação de multiplicação por escalar $R \times S \rightarrow S$, definida por $(\lambda f)(t) = \lambda f(t)$. Quando os sinais possuem estrutura de multiplicação,

como por exemplo, sinais que assumem valores em R^n ou C^n , podemos definir facilmente a operação de multiplicação de sinais $S \times S \rightarrow S$ como $(fg)(x) = f(x)g(x)$.

Uma operação importantíssima no estudo de sinais é a operação de *filtragem*.

Definição 3.4.5 *Um filtro é uma operação unária $L : S \rightarrow S$ que mapeia um sinal $s_0(x)$ em um sinal $s_1(x)$.*

Uma outra interpretação pode ser feita através do conceito de um sistema que recebe como entrada um sinal s_0 e retorna como saída um sinal s_1 . Dentre os diversos tipos de filtros podemos destacar dois casos bastante importantes: os *filtros espacialmente invariantes* e os *filtros lineares*.

Definição 3.4.6 *Um filtro L é espacialmente invariante se, para todo sinal $s(x)$, $(Ls)(x - \alpha) = L(s(x - \alpha))$, o que significa que filtrar um sinal e então transladá-lo é equivalente a transladar um sinal e então filtrá-lo.*

Definição 3.4.7 *Um filtro L é linear se, para todo sinal $s(x)$, $L(\lambda s(x)) = \lambda L(s(x))$ e se para quaisquer sinais $s_0(x)$ e $s_1(x)$, $L(s_0 + s_1)(x) = L(s_0(x)) + L(s_1(x))$.*

Os *filtros lineares espacialmente invariantes* são extremamente importantes pois podem ser caracterizados através de sua *resposta de impulso* ou *função de espalhamento* $h(t) = L(\delta)$, isto é, através da resposta produzida pelo filtro ao ser aplicado a um sinal impulso. A transformada de Fourier $\hat{h}(t)$ da resposta de impulso de um sinal é conhecida como *função de transferência*.

Como conseqüência, as operações de filtragem de um sinal envolvendo filtros lineares espacialmente invariantes podem ser representadas através da seguinte operação

$$L(f(x)) = \int_{-\infty}^{+\infty} f(t)L(\delta(x - t))dt = \int_{-\infty}^{+\infty} f(t)h(x - t)dt \quad (3.4.6)$$

A segunda integral define uma operação $S \times S \rightarrow S$ denominada *produto de convolução* ou simplesmente *convolução*.

$$f * g = \int_{-\infty}^{+\infty} f(t)g(x - t)dt \quad (3.4.7)$$

No caso dos filtros lineares espacialmente invariantes, a relação entre a operação de multiplicação no domínio espacial e a convolução no domínio da

freqüência é estabelecida pela transformada de Fourier. Esta característica é extremamente útil para que possamos relacionar o comportamento de certas operações nos domínios espacial e espectral.

$$f(x)g(x) \longleftrightarrow \hat{f}(w) * \hat{g}(w) \quad (3.4.8)$$

$$f(x) * g(x) \longleftrightarrow \hat{f}(w)\hat{g}(w) \quad (3.4.9)$$

Uma outra classe de filtros que devemos mencionar é a classe dos *filtros adaptativos*, que desempenham um papel bastante significativo no processo de reamostragem envolvendo mapeamentos genéricos, como por exemplo, os mapeamentos projetivos.

Finalmente, devemos lembrar que os filtros também podem ser classificados segundo a natureza do suporte de sua resposta de impulso. Neste caso, eles podem ser classificados em filtros com *resposta de impulso finita* ou com *resposta de impulso infinita*. Estes últimos precisam ser truncados para que possam ser utilizados em sistemas digitais, o que gera um problema importante, já que os filtros ideais de reconstrução, como veremos a seguir, não possuem resposta de impulso com suporte limitado.

Reconstrução e amostragem de sinais

A maneira mais comum de se efetuar a amostragem de um sinal s é através de uma *amostragem pontual uniforme*. Matematicamente, isto pode ser feito multiplicando-se s por um trem de impulsos

$$i_{\Delta t}(x) = \sum_{k \in \mathbb{Z}} \delta(x - k\Delta t), \quad (3.4.10)$$

onde $\delta(t)$ são impulsos unitários, e Δt é o *intervalo de amostragem*. O trem de impulsos normalmente é conhecido como *função pente* ou *senal de amostragem*. A transformada de Fourier de uma função pente é uma outra função pente

$$i_{1/\Delta t}(x) = \frac{1}{\Delta t} \sum_{k \in \mathbb{Z}} \delta(x - \frac{k}{\Delta t}), \quad (3.4.11)$$

o que significa que o intervalo de amostragem de uma função pente no domínio do espectro é igual ao inverso do seu intervalo de amostragem no domínio do espaço (Figura 3.5(b)).

No domínio espectral a amostragem pontual de um sinal s corresponde ao produto de convolução da resposta espectral do sinal \hat{s} com a função de transferência da função pente $\hat{i}(x)$. A convolução tem como efeito a replicação do espectro de freqüências do sinal original, ao longo do domínio do espectro, nas posições correspondentes a cada um dos impulsos que formam a função pente (Figura 3.5(c)).

Conseqüentemente, para reconstruirmos um sinal, precisamos remover as altas freqüências introduzidas pelo processo de amostragem que não correspondem às freqüências originais do sinal, o que pode ser feito através de um processo de filtragem.

A *filtragem ideal* é feita através da multiplicação do espectro de freqüências do sinal amostrado por um *filtro box* devidamente posicionado e modulado, o que no domínio espectral corresponde a um produto de convolução do sinal original com uma função $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$, que por sua vez, corresponde à transformada de Fourier inversa do filtro box (Figura 3.5(d)).

Uma vez removidas as freqüências espúrias, introduzidas pelo processo de amostragem, podemos reconstruir o sinal original através da transformada de Fourier inversa do sinal filtrado.

Quando a amostragem causa uma mescla das informações de alta freqüência com as informações de baixa freqüência no espectro do sinal amostrado, então não é possível reconstruir um sinal, mesmo através da utilização de um filtro de reconstrução ideal; este fenômeno é o que conhecemos por *aliasing* (Figura 3.6(c)).

De fato, segundo o *teorema de amostragem*, só podemos reconstruir um sinal com exatidão, quando

$$\Delta t \geq \frac{1}{2\omega}, \quad (3.4.12)$$

ou de forma equivalente,

$$\frac{1}{\Delta t} \leq 2\omega, \quad (3.4.13)$$

isto é, quando a freqüência de amostragem é duas vezes maior que a freqüência máxima presente no sinal. Infelizmente, os sinais podem ter freqüências arbitrariamente altas, o que torna inviável e em alguns casos impossível, a amostragem acima do limite de Nyquist. Nestes casos, o máximo que podemos fazer é tentar reduzir ao máximo os efeitos provocados por aliasing através de técnicas baseadas em filtragem conhecidas como técnicas *anti-aliasing*.

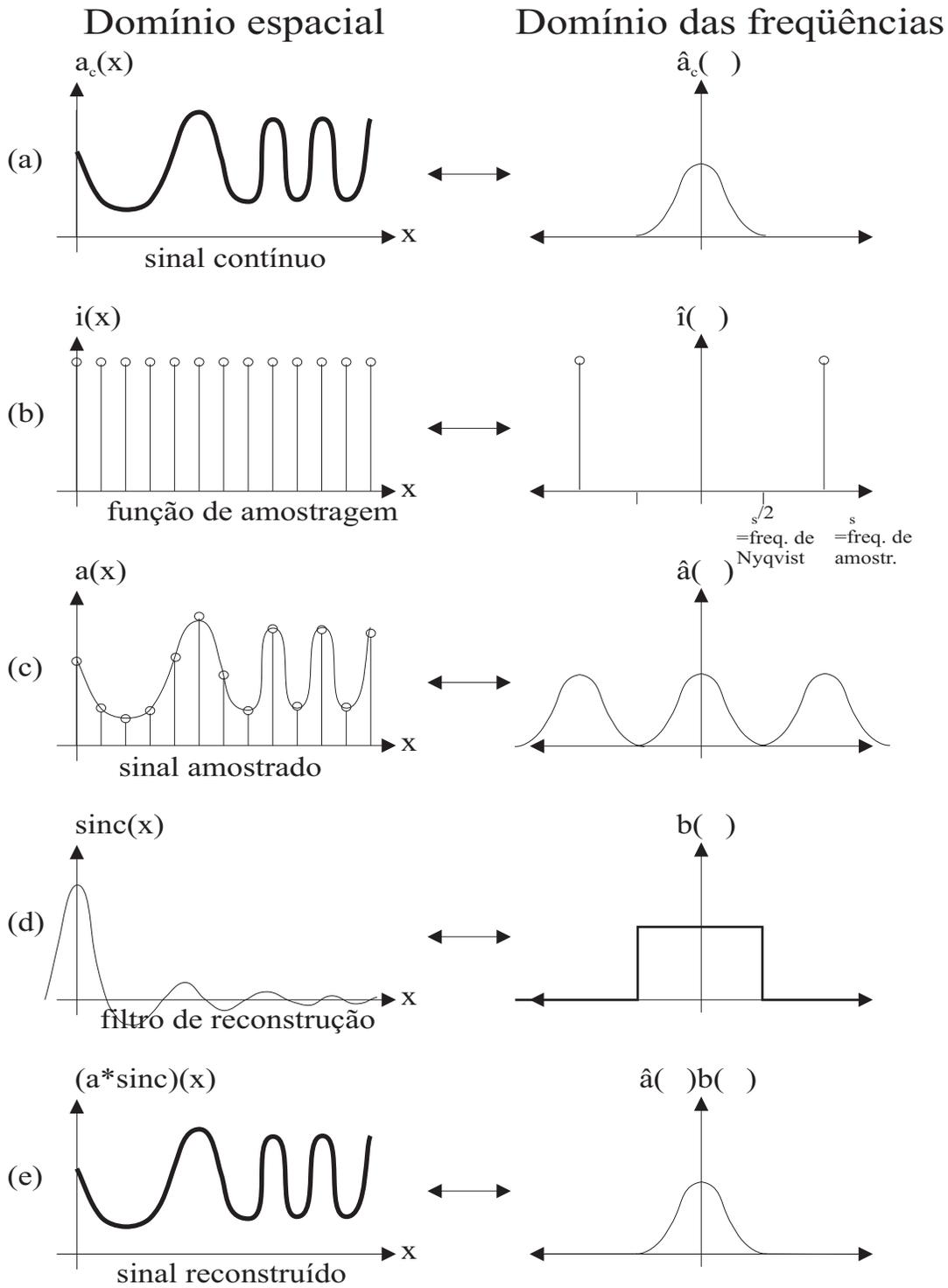


Figura 3.5: Amostragem acima do limite de Nyquist

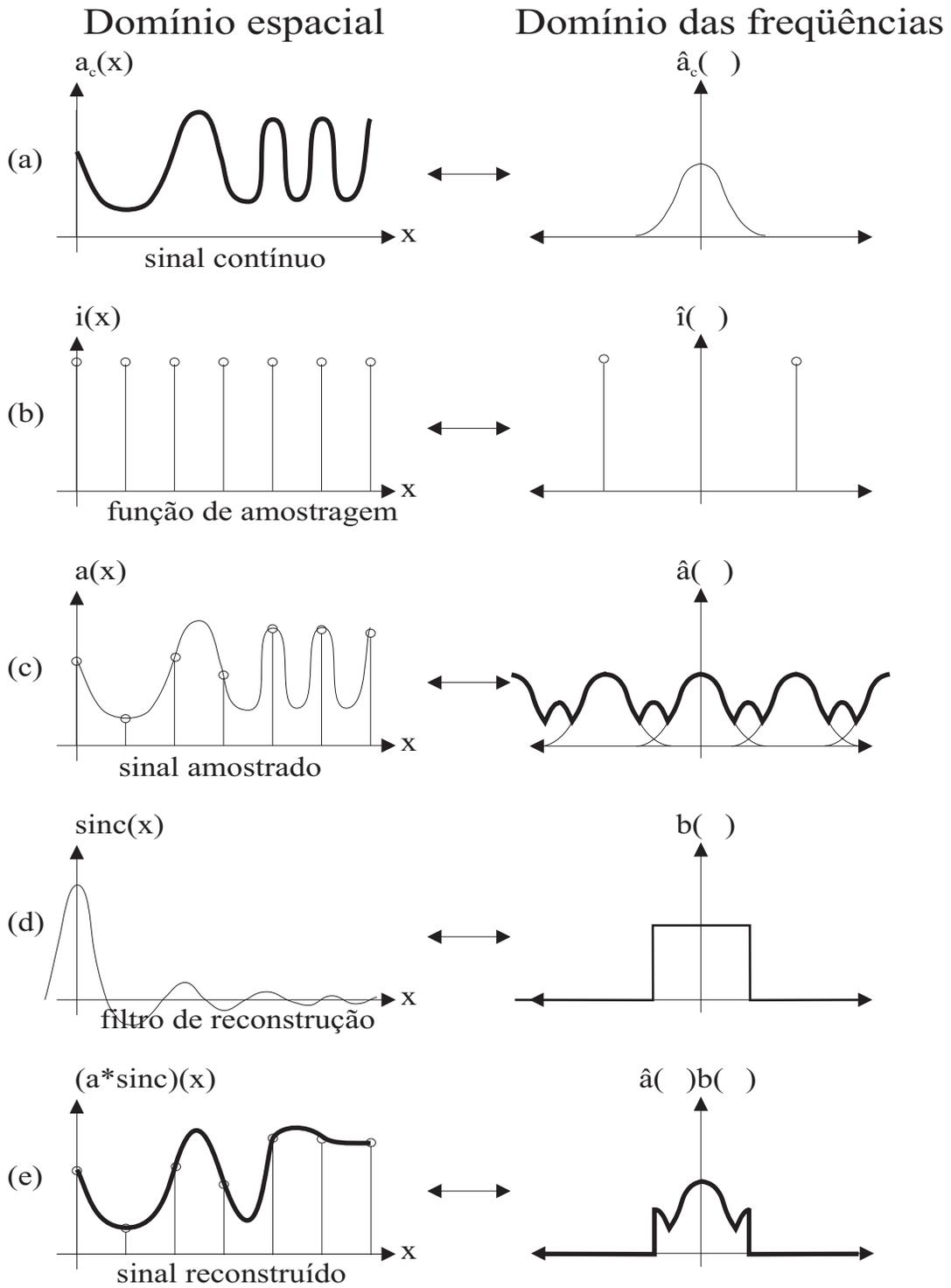


Figura 3.6: Amostragem abaixo do limite de Nyquist

Técnicas anti-aliasing

Basicamente existem duas formas de se erradicar ou pelo menos reduzir o problema de aliasing: através do *aumento da frequência de amostragem* ou através de *pré-filtragem* do sinal.

A primeira técnica consiste em aumentar a frequência de amostragem de tal forma que ela se aproxime do limite de Nyquist, permitindo assim que o sinal possa ser reconstruído, com o mínimo de aliasing possível, através de interpolação com filtros de reconstrução apropriados (no caso ideal isto significa efetuar uma convolução do sinal com uma função $\text{sinc}(x)$).

Ainda que seja capaz de reduzir a presença de aliasing na reconstrução de um sinal, o aumento da frequência de amostragem por si próprio não é capaz de evitá-lo por completo, já que um sinal pode ter frequências arbitrariamente altas. Além disso, o aumento excessivo da frequência de amostragem pode ter conseqüências drásticas tanto em relação ao aumento do espaço em memória para o armazenamento da representação do sinal, quanto ao custo computacional para processá-lo.

A única técnica capaz de evitar aliasing por completo, ao menos em teoria, é a pré-filtragem, a qual consiste em aplicar um *filtro passa baixa* no sinal original com o objetivo de remover as altas frequências antes de amostrá-lo (Figura 3.7).

Infelizmente, filtrar o sinal no domínio contínuo envolve a solução de integrais que podem ser arbitrariamente complexas dependendo do tipo de função associada, além de que, em alguns casos, o filtro utilizado pode não ser espacialmente invariante.

Por estes motivos, muitas das vezes, na prática, opta-se por resolver o problema de aliasing, através de *pós-filtragem*, na qual a filtragem ocorre após uma superamostragem do sinal.

Muitas das vezes, o sinal com o qual trabalhamos já é um sinal amostrado que, por algum motivo, precisa sofrer uma nova operação de amostragem. Este tipo de procedimento é muito comum em processamento digitais de sinais, estando diretamente relacionado aos procedimentos de mapeamento de texturas. Por esta razão, iremos investigá-lo em maiores detalhes na próxima seção.

Reamostragem de sinais

A *reamostragem* é o processo pelo qual um sinal contínuo é reconstruído a partir de um sinal discreto, podendo eventualmente sofrer uma transformação de um espaço de origem para um espaço destino, antes

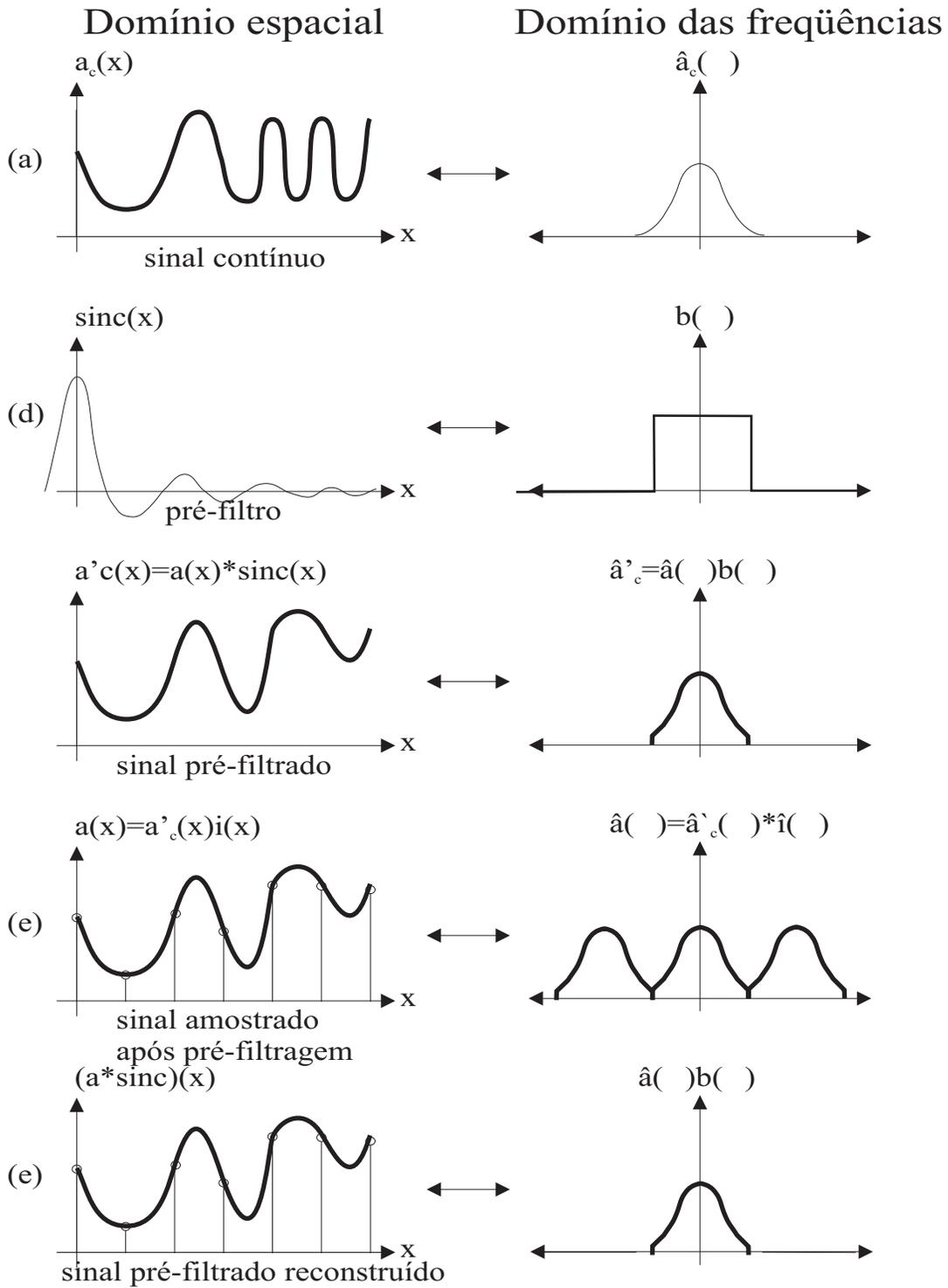


Figura 3.7: Pré-filtragem

de ser novamente amostrado, na maioria das vezes, com uma taxa de amostragem diferente da taxa de amostragem original. Um exemplo típico de reamostragem ocorre quando mapeamos uma textura em um polígono definido no espaço da tela.

O procedimento de reamostragem normalmente segue o seguinte esquema:

1. Reconstrução do sinal contínuo a partir do sinal discreto original.
2. Mapeamento do sinal reconstruído para o espaço de destino.
3. Pré-filtragem.
4. Amostragem.

Quando o mapeamento do espaço de origem para o espaço de destino é de natureza afim, então a teoria sobre *amostragem pontual uniforme*, apresentada anteriormente pode ser utilizada sem maiores dificuldades, já que nestes casos, a filtragem pode ser caracterizada através de *filtros lineares espacialmente invariantes*. Por outro lado, quando trabalhamos com mapeamentos mais genéricos, como por exemplo, mapeamentos projetivos, então os filtros em geral não são espacialmente invariantes, o que torna o processo um pouco mais sofisticado (Figura 3.8).

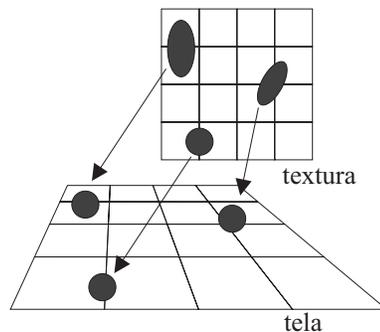


Figura 3.8: Mapeamentos não-afins

Seja $f(\mathbf{u})$ um sinal discreto definido em espaço de origem com coordenadas $\mathbf{u} = (u, v)$, o qual deve sofrer um processo de reamostragem para que possa ser mapeado em um sinal $g(\mathbf{x})$, também discreto, definido em um espaço de destino com coordenadas $\mathbf{x} = (x, y)$. O espaço de origem e o espaço de destino estão relacionados entre si através de um mapeamento direto $\mathbf{x} = m(\mathbf{u})$ e de um mapeamento inverso $\mathbf{u} = m^{-1}(\mathbf{x})$. Seja ainda $r(\mathbf{u})$ um filtro de reconstrução e $h(\mathbf{x})$ um pré-filtro a ser utilizado no processo

sinal discreto original	$f(\mathbf{u})$
reconstrução	$f_c(\mathbf{u}) = f(\mathbf{u}) * r(\mathbf{u}) = \sum_{\mathbf{k} \in Z^n} f(\mathbf{k})r(\mathbf{u} - \mathbf{k})$
mapeamento	$g_c(\mathbf{x}) = f_c(m^{-1}(\mathbf{x}))$
pré-filtragem	$g'_c = g_c(\mathbf{x}) * h(\mathbf{x}) = \int_{R^n} g_c(\mathbf{t})h(\mathbf{x} - \mathbf{t})d\mathbf{t}$
amostragem final	$g(\mathbf{x}) = g'_c(\mathbf{x})i(\mathbf{x})$

de reamostragem. Então, $g(\mathbf{x})$ pode ser obtido a partir de $f(u)$ através dos seguintes passos:

Expandindo a equação final $g(x)$ e reorganizando as operações de forma conveniente, podemos reduzir o conjunto de passos anterior a uma operação envolvendo o sinal original discreto $f(\mathbf{u})$ e um único filtro $\rho(\mathbf{x}, \mathbf{k})$, denominado *filtro de reamostragem*.

$$\begin{aligned}
g(\mathbf{x}) &= g'_c(\mathbf{x}) \\
&= \int_{R^n} f_c(m^{-1}(\mathbf{t}))h(\mathbf{x} - \mathbf{t})d\mathbf{t} \\
&= \int_{R^n} h(\mathbf{x} - \mathbf{t}) \sum_{\mathbf{k} \in Z^n} f(\mathbf{k})r(m^{-1}(\mathbf{t}) - \mathbf{k})d\mathbf{t} \\
&= \sum_{\mathbf{k} \in Z^n} f(\mathbf{k})\rho(\mathbf{x}, \mathbf{k})
\end{aligned}$$

$$\text{onde : } \rho(\mathbf{x}, \mathbf{k}) = \int_{R^n} h(\mathbf{x} - \mathbf{t})r(m^{-1}(\mathbf{t}) - \mathbf{k})d\mathbf{t} \quad (3.4.14)$$

O filtro de reamostragem $\rho(\mathbf{x}, \mathbf{k})$ varia espacialmente, determinando o peso de uma amostra na posição \mathbf{k} , no espaço de origem, correspondente a uma posição \mathbf{x} , no espaço de destino. Na equação anterior, $\rho(\mathbf{x}, \mathbf{k})$ é definido no espaço de destino, como uma integral do produto do pré-filtro com o filtro de reconstrução transformado para o espaço de destino. Porém, através de uma troca de variáveis, fazendo $\mathbf{t} = m(\mathbf{u})$, podemos defini-lo no espaço de origem como uma integral do produto do pré-filtro transformado para o espaço de origem com o filtro de reconstrução.

$$\rho(\mathbf{x}, \mathbf{k}) = \int_{R^n} h(\mathbf{x} - m(\mathbf{u}))r(\mathbf{u} - \mathbf{k}) \left| \frac{\partial m}{\partial \mathbf{u}} \right| d\mathbf{u} \quad (3.4.15)$$

Como podemos ver através da equação 3.4.15, a reamostragem de sinais envolvendo mapeamentos não-afins pode ser relativamente complexa, o que significa que na prática precisamos efetuar algumas simplificações tanto no mapeamento do espaço de origem para o espaço de destino, quanto nos filtros de reconstrução.

O mapeamento projetivo $m(\mathbf{u})$ pode ser aproximado em uma vizinhança local \mathbf{u}_0 através de um mapeamento afim

$$m_{\mathbf{u}_0}(\mathbf{u}) = \mathbf{x}_0 + (\mathbf{u} - \mathbf{u}_0)J_{\mathbf{u}_0}, \quad (3.4.16)$$

onde

$$J_{\mathbf{u}_0} = \frac{\partial m}{\partial \mathbf{u}}(\mathbf{u}_0) \quad (3.4.17)$$

é a matriz jacobiana em \mathbf{u}_0 .

Como o pré-filtro $h(\mathbf{x} - m^{-1}(\mathbf{x}))$ atribui um peso maior às regiões mais próximas a \mathbf{u}_0 , onde a aproximação é mais precisa, e um peso menor às regiões mais distantes, onde a aproximação não é tão precisa, podemos utilizar a aproximação linear para o mapeamento sem causar maiores danos ao filtro de reamostragem.

Com efeito, o filtro de reamostragem pode ser expresso como

$$\rho_{\mathbf{u}_0}(\mathbf{x}_0, \mathbf{k}) = \int h(\mathbf{x} - m_{\mathbf{u}_0}(\mathbf{u}))r(\mathbf{u} - \mathbf{k})|J_{\mathbf{u}_0}|d\mathbf{u}. \quad (3.4.18)$$

Podemos também aproximar ou os filtros de reconstrução, ou os pré-filtros, através de impulsos, conforme o efeito causado pelo mapeamento envolvido no processo de reamostragem.

Quando o mapeamento causa uma ampliação da imagem no espaço de destino, então a forma e tamanho do filtro de reamostragem são dominados pelo filtro de reconstrução. Neste caso, podemos aproximar o pré-filtro através de um impulso, isto é, fazendo $h = \delta$. Desta forma, o filtro de reamostragem se reduz a

$$\rho(\mathbf{x}, \mathbf{k}) \approx \left| \frac{\partial m}{\partial \mathbf{u}} \right| r(m^{-1}(\mathbf{x}) - \mathbf{k}) \quad (3.4.19)$$

o qual é um filtro de reconstrução não transformado, centrado em $m^{-1}(\mathbf{x})$, com um suporte independente do mapeamento e de tamanho em torno de 1 a 4 pixels. Estes filtros são conhecidos em processamento de imagens como *filtros de magnificação* ou *filtros de interpolação*, sendo exemplos clássicos os filtros box, os filtros bilineares e os filtros definidos por b-splines cúbicas.

Por outro lado, quando o mapeamento causa uma redução da imagem no espaço de destino, então a forma e tamanho do filtro de reamostragem são dominados pelo pré-filtro, o que nos permite aproximar o filtro de reconstrução por um impulso, isto é, fazendo $r = \delta$. Neste caso, o filtro

de reamostragem pode ser expresso da seguinte forma:

$$\rho(\mathbf{x}, \mathbf{k}) \approx \left| \frac{\partial m}{\partial \mathbf{u}} \right| h(\mathbf{x} - m(\mathbf{k})) \quad (3.4.20)$$

o qual é um pré-filtro centrado em $\mathbf{m}^{-1}(\mathbf{x})$, devidamente transformado segundo o mapeamento m . Estes filtros são conhecidos como *filtros de redução* ou *filtros de decimação*. O suporte deste tipo de filtro tem área inversamente proporcional ao fator de escala do mapeamento; como os fatores de escala podem assumir valores arbitrários, então o número de pixels no suporte pode variar de um a até alguns milhões.

As implementações de mapeamentos de textura normalmente determinam que sejam especificados filtros, possivelmente distintos, para os casos de magnificação e de redução, os quais são selecionados durante o processo de reamostragem de acordo com o fator de escala determinado pelo mapeamento.

Este tipo de procedimento pode causar alguns problemas em duas dimensões já que neste caso o fator de escala não é dado por um único escalar e sim por uma matriz 2×2 , a matriz jacobiana da transformação. Por conseguinte, o mapeamento pode causar a ampliação em uma direção e uma redução em outra, o que em geral é resolvido, de maneira *ad hoc*, através da seguinte estratégia: utiliza-se o filtro de magnificação quando ambas as direções são ampliadas, caso alguma das direções seja reduzida, utiliza-se um filtro de redução. Quando uma das direções sofre uma redução e a outra sofre uma magnificação, a forma do filtro de reamostragem resultante pode ficar tão estreita que alguns pixels podem ser perdidos. Neste caso, costuma-se obrigar o filtro a amostrar ao menos um elemento.

Heckbert propôs a solução deste problema através da utilização de *filtros gaussianos elípticos* (Figura 3.9), os quais são capazes de fornecer um tratamento unificado para a operação de reamostragem pois conseguem transitar suavemente entre os casos de magnificação e redução [26]. Infelizmente, a maioria das implementações em placas gráficas não adota filtros tão sofisticados, pois a utilização destes, além de dificultar o projeto do hardware, implicaria em um custo computacional bastante alto, inviável para processamento em tempo real. A maioria dos métodos de filtragem disponíveis em placas gráficas ainda é baseada em filtros mais simples, como os filtros box e os filtros bilineares, juntamente com o uso de heurísticas para a escolha do tipo de filtragem utilizada, isto é, de redução ou de ampliação.

Em nossa implementação utilizamos os mecanismos de filtragem disponíveis nos procedimentos de mapeamento de textura fornecidos pela

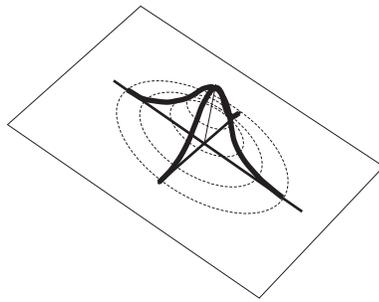


Figura 3.9: Filtros gaussianos elípticos

OpenGL, a qual determina que sejam especificados filtros para os casos de filtragem de magnificação e de redução. Para cada um destes casos é possível efetuar uma escolha entre dois tipos básicos de filtragem: um filtro baseado em vizinhança mais próxima e um filtro linear.

O filtro baseado em vizinhança mais próxima, na verdade, um filtro box, retorna o elemento de textura cujas coordenadas se encontram mais próximas das coordenadas de textura associadas a um determinado elemento da tela. Os resultados produzidos pela aplicação deste tipo de filtro são extremamente suscetíveis a aliasing, o que nos impede de utilizá-lo nos procedimentos de mapeamento projetivo de texturas necessários ao registro das imagens de entrada no espaço da cena.

O filtro linear, definido por uma máscara 2×2 , efetua uma interpolação bilinear entre os elementos de textura mais próximos das coordenadas de textura correspondentes ao pixel desejado. Este filtro pode ser utilizado sem grandes problemas quando a imagem associada à textura é ampliada, porém, quando a imagem é reduzida, então os resultados provenientes de sua aplicação não são muito melhores que os obtidos através da aplicação de filtros box, sendo também bastante suscetíveis aos efeitos causados por aliasing. Isto ocorre porque os suportes dos filtros de redução devem variar de acordo com o fator de escala, não podendo ser especificados através de suportes de tamanho fixo, com por exemplo um suporte definido por uma máscara 2×2 .

Um possível melhoramento pode ser obtido através do uso de *mipmapping*, o qual é um método de anti-aliasing que possibilita a aproximação da operação de filtragem, sobre um conjunto de texels que são mapeados em um determinado pixel, através de uma única operação de amostragem efetuada sobre uma pirâmide de mapas de textura pré-filtrados.

A estrutura piramidal utilizada pelo processo de mipmapping é formada por uma seqüência de n imagens em resolução decrescente, cada uma



Figura 3.10: Mipmap

delas determinando um nível de detalhe para o mapa de textura original (Figura 3.10). Cada um dos n níveis de detalhe são especificados da seguinte forma: o nível 0, que corresponde à base da pirâmide, é determinado pelo mapa de textura original; um nível i , tal que $0 < i < n$ é obtido através de um subamostragem do nível $i - 1$, onde uma combinação de texels vizinhos do nível $i - 1$ contribui para a geração de um único texel no nível i (normalmente esta combinação é dada por uma média de quatro texels vizinhos); o último nível, isto é, o nível n , corresponde ao mapa de textura no qual uma de suas dimensões possui apenas um elemento, obtido através da subamostragem do mapa de textura no nível $n - 1$.

O mecanismo subjacente ao processo de mipmapping procura determinar o nível de detalhe λ mais adequado para uma determinada operação de reamostragem, através de uma análise da região do mapa de textura que contribui para um determinado pixel no espaço da imagem. Um maneira simples de aproximar tal contribuição pode ser feita através da medida da maior aresta pertencente ao quadrilátero correspondente à projeção de um pixel no mapa de textura em questão. Uma outra estratégia se baseia no valor absoluto máximo entre as diferenciais $\partial u/\partial x$, $\partial v/\partial x$, $\partial w/\partial x$, $\partial u/\partial y$, $\partial v/\partial y$, $\partial w/\partial y$, que refletem as taxas de variação instantâneas das coordenadas de textura em função das variações instantâneas das coordenadas x e y no espaço da tela.

O valor de λ é utilizado para determinar o nível no qual deve ser realizada a amostragem na estrutura piramidal. O nível de detalhe mais adequado é aquele que possibilita uma taxa de amostragem mais próxima de um pixel para cada amostra, ou preferencialmente, de dois pixels para uma amostra, satisfazendo assim o limite de Nyquist. Quando isto ocorre, a reamostragem é realizada de forma bastante eficiente ao mesmo tempo em

que os efeitos de aliasing são adequadamente atenuados.

A amostra desejada é especificada por uma tripla (u, v, λ) . No entanto, como λ é um valor fracionário, precisamos efetuar alguns cálculos para determinar o nível apropriado na estrutura piramidal. A solução adotada consiste em amostrar os texels nos níveis de detalhe, acima e abaixo de λ , determinados por interpolações bilineares em torno das posições determinadas respectivamente por $(u, v, \lfloor \lambda \rfloor)$ e $(u, v, \lceil \lambda \rceil)$. A amostra resultante é então calculada através de uma interpolação linear entre as duas amostras obtidas, com base na distância de cada nível de textura a λ .

O processo de mipmapping tem a grande vantagem de realizar a amostragem em tempo constante, independente do grau de redução causado pelo mapeamento. No entanto, também apresenta alguns problemas, em especial, o excesso de suavização que pode ocorrer em determinadas regiões.

A causa deste problema está no fato de que, no mecanismo de mipmapping, somos obrigados a aproximar a área relativa à contribuição do mapa de textura para um determinado pixel através de um região quadrada. Esta aproximação normalmente não é adequada quando a área de contribuição real possui mais elementos de textura em uma direção do que na outra, o que é bastante comum quando o polígono texturizado é quase ortogonal ao plano de projeção, causando um borramento excessivo de certas características presentes no mapa de textura.

Existem alguns métodos que procuram evitar o fenômeno de suavização excessiva como por exemplo o *Ripmapping* [21, 30] e o método das *Tabelas de Áreas Somadas (Summed-area Tables)* [13]. Estes métodos são classificados como métodos de filtragem anisotrópicos, sendo capazes de recuperar valores de textura definidos em áreas não quadradas, o que os permite trabalhar com taxas de amostragem diferentes em direções distintas. A grande desvantagem associada a estes métodos é que eles só conseguem efetuar filtrações anisotrópicas, de forma eficiente, em direções preferencialmente horizontais e verticais. Uma outra característica inconveniente é a de que estes métodos requerem um espaço de memória bem maior que o necessário para o armazenamento da estrutura piramidal utilizada em mipmapping.

Um método mais robusto e, na verdade, mais eficiente também, é o *método de filtragem anisotrópica irrestrita* que trabalha sobre a própria estrutura utilizada em *mipmapping*. A diferença chave é a de que utilizamos várias amostras da estrutura do mipmap, ao invés de somente uma, para aproximar a área de contribuição do mapa de textura para um dado pixel, a qual corresponde à cobertura de tal pixel no mapa de textura. O valor

de λ , neste tipo de filtragem é dado pelo menor lado do quadrilátero associado à área de cobertura de um pixel no mapa de textura, o que faz com a região filtrada seja menor, causando, conseqüentemente, um menor borramento. O maior lado do quadrilátero determina o que chamamos *linha de anisotropia*, a qual passa pelo interior da região a ele associada, podendo assumir qualquer direção, sendo este o motivo pelo qual esse esquema de filtragem é denominado irrestrito. Um fator de anisotropia 2:1 significa que duas amostras são tomadas ao longo da direção por ela determinada. Já se encontram disponíveis filtragens com fator de anisotropia 8:1, como por exemplo, a que podemos encontrar nas GeForce 3 [2]. Em nosso trabalho não investigamos a utilização de filtragem anisotrópica.

Devido às limitações associadas à utilização de filtros de reamostragem bastante aquém dos filtros de reamostragem ideais, adicionamos um procedimento de *superamostragem* às técnicas de mipmapping e filtragem linear, com o objetivo de amenizar um pouco mais os efeitos causados por aliasing.

No próximo capítulo iremos aplicar as técnicas aqui descritas na construção de um algoritmo de escultura do espaço que trabalha diretamente no espaço da cena, em contraposição aos algoritmos de escultura do espaço tradicionais, que trabalham no espaço das imagens.

4

Escultura do espaço no espaço da cena

4.1

Introdução

Com base nas técnicas apresentadas no capítulo anterior, iremos descrever, neste capítulo, um método de escultura do espaço que trabalha diretamente no espaço da cena. Este método apresenta características similares ao esquema de varredura de planos de Kutulakos e Seitz, porém possui como diferença fundamental a existência de uma *etapa de registro*, através da qual todas as informações necessárias ao processo de escultura do espaço são registradas sobre cada um dos planos de referência no espaço da cena. Implementamos esta etapa através do mapeamento de textura básico fornecido pela biblioteca gráfica OpenGL, sem utilizar os procedimentos para geração automática de coordenadas de textura, pois optamos por calculá-las manualmente.

O *tratamento da visibilidade* e a *determinação da foto-consistência*, possuem diferenças importantes em relação às respectivas etapas existentes no algoritmo anterior. Estas diferenças se devem, em grande parte, à necessidade de manipulação das informações de forma distinta e também ao fato de que o processamento no espaço da cena nos permite explorar novas funcionalidades e potenciais inexistentes no método anterior.

Iremos apresentar, em detalhe, nas próximas seções, cada uma das cinco etapas fundamentais que compõem o método de escultura do espaço que trabalha no espaço da cena: o registro das imagens no espaço da cena, a incorporação das informações de segmentação do objeto nas imagens de entrada, o tratamento da visibilidade, a determinação da foto-consistência dos voxels, e a atribuição de suas informações fotométricas.

Por último, ao final do capítulo, apresentaremos uma síntese do método e faremos uma análise sucinta sobre suas características.

4.2 Registro

Após descrever, no capítulo anterior, as ferramentas e conceitos associados à técnica de mapeamento projetivo de texturas, podemos finalmente apresentar uma das etapas mais importantes do método de escultura do espaço no espaço da cena: a etapa de registro das informações.

O registro das imagens de entrada, sobre um plano de referência π_k , no espaço de cena, é dado por um conjunto de imagens projetadas I_p onde cada imagem projetada I_{p_i} corresponde ao mapeamento projetivo de uma imagem I_i sobre o plano π_k .

Primeiramente, identificamos cada imagem $I_i \in I_p$, e sua respectiva câmera C_i , com um *projektor de textura* P_i . Cada uma das imagens I_i é representada por um mapa de textura RGBA, onde os canais r , g e b são utilizados para armazenar as componentes de cor e o canal a , isto é, o canal de opacidade, é utilizado, como veremos em detalhe mais tarde, para armazenar as segmentações dos objetos de interesse. Em seguida, para cada projetor P_i , determinamos sua área de projeção definida por um polígono cujas coordenadas de seus vértices são dadas pela interseção entre o frustum associado a P_i e o plano de referência considerado. Uma vez determinados cada um dos polígonos P_{o_i} , calculamos as coordenadas homogêneas de textura de cada um de seus vértices $v_{i0}..v_{i3}$, através do cálculo das suas distâncias ao plano do projetor. Finalmente, geramos uma imagem projetada I_{p_i} , para cada imagem de entrada I_i , desenhando os polígonos P_{o_i} apropriadamente texturizados, sobre o plano de referência π_k (Figura 4.1).

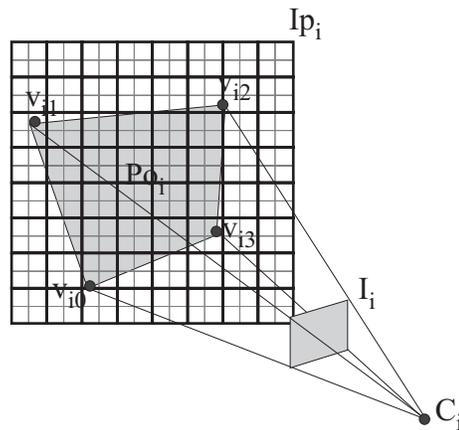


Figura 4.1: Projeção de uma imagem em um plano de referência

O desenho de P_{o_i} sobre o plano π_k , segundo o mapeamento proje-

tivo correspondente, pode ser obtido através da passagem para o *pipeline* de visualização, das coordenadas de cada um de seus vértices v_{ij} em coordenadas do mundo (globais), juntamente com suas coordenadas de textura (x_t, y_t, z_t, w_t) .

As coordenadas de textura x_t e y_t , de cada vértice, são dadas pelas coordenadas dos cantos das imagens normalizadas para o intervalo $[0, 1]$. Como a OpenGL requer a especificação de uma coordenada z_t (utilizada no caso de mapeamentos de textura 3D), precisamos lhe atribuir o valor 0, visto que estamos trabalhando apenas com texturas bidimensionais. As coordenadas homogêneas w_t são determinadas pela distância entre o vértice e o plano de suporte da textura, que por sua vez é o próprio plano de projeção da câmera que obteve a imagem a ser registrada. Ilustramos abaixo um trecho de código que desenha um polígono com uma textura mapeada projetivamente.

```
void DrawTexturizedPolygon(tpTexture * texture, float coord[4][4])
{
    /* a coordenada de textura homogenea esta em coord[...] [3] */

    float wCorr = texture->wCorr,
        hCorr = texture->hCorr;

    glEnable(GL_TEXTURE_2D);
    glBegin(GL_QUADS);
        glNormal3f(0.0,0.0,1.0);
        glTexCoord4d(0.0,0.0,0.0,coord[0][3]);
        glVertex4f(coord[0][0],coord[0][1],coord[0][2],1);
        glTexCoord4d(wCorr*coord[1][3],0.0,0.0,coord[1][3]);
        glVertex4f(coord[1][0],coord[1][1],coord[1][2],1);
        glTexCoord4d(wCorr*coord[2][3],hCorr*coord[2][3],0.0,coord[2][3]);
        glVertex4f(coord[2][0],coord[2][1],coord[2][2],1);
        glTexCoord4d(0.0,hCorr*coord[3][3],0.0,coord[3][3]);
        glVertex4f(coord[3][0],coord[3][1],coord[3][2],1);
    glEnd();
    glDisable(GL_TEXTURE_2D);
}
```

Na figura 4.3 ilustramos o registro de um conjunto de imagens obtidas de uma cena sintética (Figura 4.2) sobre um plano de referência do espaço de reconstrução. Observe que as imagens na figura são distorcidas corretamente de acordo com o mapeamento projetivo.

As imagens projetadas I_{p_i} são produzidas com um fator de superamostragem n . Em outras palavras, são geradas com uma resolução

n vezes maior que a resolução da fatia de voxels correspondente no espaço volumétrico discretizado. Como mencionamos anteriormente, a superamostragem é realizada pra compensar a utilização de filtros não adequados, além de permitir uma maior robustez em face dos erros numéricos presentes nos cálculos das coordenadas das amostras obtidas das imagens projetadas Ip_i , assim como no cálculo das áreas de projeção associadas aos polígonos Po_i .

Um fator de superamostragem igual a 2 é bastante razoável. Por exemplo, se o volume de reconstrução inicial possui 128^3 voxels, então geramos as imagens com o dobro da resolução, isto é, 256×256 . Desta forma, associamos quatro amostras a cada região planar, as quais serão utilizadas durante o processo de avaliação da foto-consistência. Naturalmente, podemos utilizar um fator de superamostragem maior, porém, devemos lembrar que há um compromisso entre a resolução das imagens e o tempo de processamento necessário à execução da etapa de determinação da foto-consistência, pois esta depende do número de amostras em cada região planar associada a um determinado voxel. De fato, a resolução máxima do volume de reconstrução inicial é limitada pela resolução máxima com a qual as imagens podem ser capturadas, pois a utilização de uma resolução maior do que a das imagens de entrada não acrescenta nenhuma informação adicional, sendo completamente inócua.

Ao final da geração de uma imagem projetada Ip_i , precisamos copiá-la do framebuffer para uma região em memória convencional de forma que possamos liberar este último para a geração de uma nova imagem Ip_j . Devemos nos lembrar, entretanto, que a operação de cópia do *framebuffer* para regiões em memória convencional ainda é bastante lenta nas arquiteturas atuais, o que pode aumentar significativamente o tempo total de processamento.

Uma possível solução seria a utilização de um framebuffer específico para a geração de cada uma das imagens projetadas, o que pode ser feito, sem maiores dificuldades através de *pbuffers(pixel buffers)*, os quais são *framebuffers* que podem ser criados em memória de vídeo conforme a necessidade. Embora bastante interessante, esta estratégia introduz alguns inconvenientes: primeiro, a criação de um framebuffer distinto para cada imagem projetada a ser produzida gera um gasto adicional de memória de vídeo, o que não é desejável, já que gastamos boa parte desta memória na criação dos mapas de textura; segundo, o acesso a elementos individuais em memória de vídeo é extremamente lento, sendo completamente inadequado a nossos propósitos.

Uma alternativa muito superior, porém não adotada neste trabalho,

consiste em efetuar todo o cálculo da estatística, utilizada na determinação da foto-consistência, diretamente na placa de vídeo, à medida em que as informações necessárias são registradas através de sua projeção sobre o plano de referência. Isto pode ser feito através de programação sobre fragmentos (*pixel shading*, *pixel programming* ou *fragment shading*) [79], a qual nos permite efetuar processamentos diretamente sobre os fragmentos durante a etapa de rasterização. Obviamente, para que esta estratégia possa funcionar, é necessário que todas as informações necessárias ao cálculo da estatística utilizada na determinação da foto-consistência, como as informações fotométricas, de visibilidade e eventualmente de segmentação, sejam passadas para o *pipeline* de renderização como mapas de textura que devem ser projetados no plano de referência.

As linguagens de programação disponíveis para a programação sobre fragmentos ainda não se encontram definitivamente padronizadas, estando bastante vinculadas aos detalhes de cada arquitetura existente. Esta diversidade torna bastante difícil a implementação de forma independente de hardware, ainda que alguns esforços tenham sido realizados com o objetivo de criar camadas de abstração, como por exemplo, a linguagem Cg [3] e o trabalho de Tuler e Celes, com base na linguagem Lua [4], o qual, no entanto está voltado por enquanto, somente à programação sobre vértices. Este foi um dos motivos pelo qual preferimos não adotar neste momento, a estratégia de programação em placa, ainda que um ganho considerável em eficiência poderia ter sido alcançado através de sua utilização. Apesar disso, prevemos a investigação deste aspecto em trabalhos futuros, haja visto a velocidade com a qual avança o desenvolvimento de processadores gráficos.

Em seguida, iremos descrever como o tratamento das demais informações deve ser realizado, para que finalmente possamos efetuar a determinação da foto-consistência dos voxels através de um teste de hipóteses estatístico.



Figura 4.2: Imagens tomadas de um modelo sintético

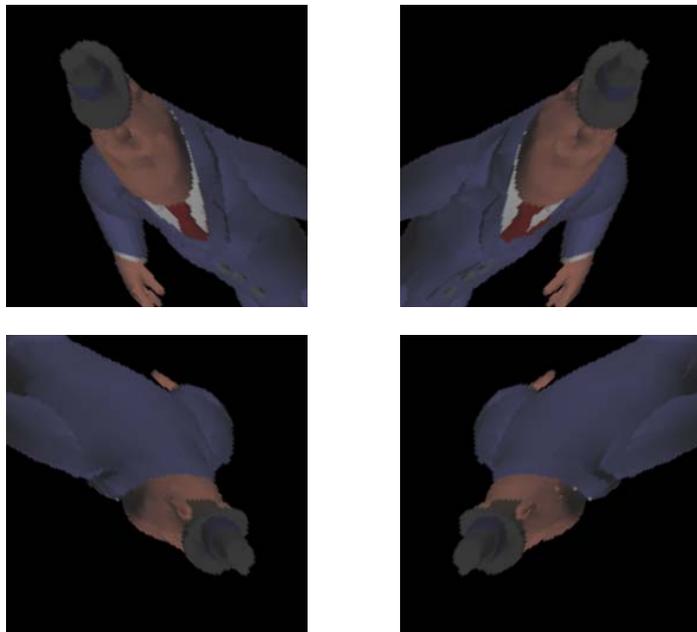


Figura 4.3: Imagens projetadas sobre um plano de referência no espaço da cena

4.3

Uso de silhuetas na reconstrução

Uma das vantagens normalmente associadas aos métodos de escultura do espaço é sua capacidade de reconstruir uma cena com base apenas em informações fotométricas, sem a necessidade de conhecimentos sobre a segmentação dos objetos de interesse nas imagens de entrada.

Neste trabalho, argumentamos que o conhecimento sobre a segmentação do objeto nas imagens, apesar de não ser obrigatoriamente necessário, é extremamente útil para o processo de reconstrução, chegando mesmo a ser fundamental nos casos em que se deseja obter soluções de alta qualidade.

Nos métodos de escultura do espaço não temos como garantir a precisão geométrica da solução obtida em relação ao modelo original, pois tais métodos se baseiam apenas em critérios fotométricos, o que significa que as soluções obtidas apenas reproduzem as imagens de entrada a menos de uma tolerância pré-estabelecida. De fato, as reconstruções produzidas pelos métodos de escultura do espaço são reconstruções maximais, cujas geometrias podem ser bastante diferentes da geometria da cena original.

Diversos fatores podem determinar o quão diferente uma reconstrução obtida através de escultura do espaço pode ser da cena original. Um dos principais é a escolha do limiar utilizado na determinação da foto-consistência. Quanto mais tolerante for o limiar escolhido, maior se tornará a forma da cena reconstruída em comparação com a forma da cena original; por outro lado, quanto mais rigoroso for o limiar, mais próxima da forma original será a forma obtida através do processo de reconstrução.

Com base neste comportamento, podemos imaginar que basta escolhermos um limiar pequeno o suficiente para conseguirmos uma reconstrução com geometria próxima o suficiente da geometria da cena original. No entanto, na prática, isto não é tão simples assim, pois a escolha de um limiar muito pequeno pode causar uma perda parcial ou até mesmo total dos elementos que compõem a cena. Ainda mais surpreendente é o fato de que, mesmo quando escolhemos um limiar igual a zero, não há garantia de que as partes que não se perderam coincidam com a superfície da cena original.

A explicação para este fenômeno está na ambigüidade intrínseca ao problema de determinação de forma a partir de imagens nos métodos de reconstrução passivos. Não há como garantir a posição exata de um determinado fragmento da superfície de uma cena lambertiana, na presença de ambigüidades geradas por regiões homogêneas correspondentes nas ima-

gens. Na verdade, esta afirmação, é um enunciado informal de um teorema que foi demonstrado por Kanade, Baker, e Sim em um trabalho recente [80].

Logo, a escolha inadequada do limiar pode fazer com que a reconstrução se torne extremamente tolerante a desvios, implicando, desta forma, na geração de soluções de baixa qualidade, tanto em termos fotométricos quanto geométricos; não obstante, um limiar extremamente rigoroso pode causar perdas na reconstrução final, as quais são inaceitáveis na maioria das vezes. Este problema tem sido objeto de estudo de alguns trabalhos, como o de Broadhurst [72], que propôs um método extremamente interessante, baseado na utilização de múltiplos limiares durante o processo de reconstrução, o qual por outro lado, requer métodos mais sofisticados e custosos para visualização do modelo obtido.

Um outro fator importante é o número de imagens utilizadas no processo de reconstrução. Devemos lembrar que, de acordo com o lema da *não-foto-consistência*, quanto maior o número de imagens, menor é a chance de que um elemento não pertencente a cena real venha a se tornar foto-consistente. Logo, quanto mais imagens pudermos utilizar no processo de reconstrução maiores são as chances de que a geometria da forma obtida se aproxime da geometria da cena real.

Apesar de teoricamente parecer uma boa estratégia, o uso de muitas imagens nem sempre é possível, principalmente quando o sistema de aquisição não dispõe de muitos recursos. Nestes casos, podemos compensar a falta de informações fotométricas através do uso de informações sobre a segmentação dos objetos de interesse nas imagens, desde que estas sejam obtidas com um certo grau de precisão (Figura 4.4).

De fato, as informações provenientes da segmentação dos objetos nas imagens são complementares às informações fotométricas. As informações provenientes da segmentação determinam de forma bastante precisa a geometria da cena nas regiões do espaço que se projetam em alguma das silhuetas dos objetos de interesse nas imagens. No entanto, não fornecem nenhuma pista sobre a geometria das regiões que se projetam no interior dos objetos. Na verdade, tais informações determinam somente a envoltória visual da cena, que por sua vez não captura formas com curvatura negativa. Já as informações fotométricas são capazes de restringir, de modo bastante razoável, a forma das regiões da cena que se projetam no interior das silhuetas dos objetos. Por outro lado, não delimitam com muita precisão as superfícies de separação entre os objetos de interesse e o fundo da cena. Este tipo de comportamento pode ser observado facilmente através das imagens na figura 4.5. Pelos motivos acima apresentados, propomos, neste trabalho,



Figura 4.4: Segmentação do objeto nas imagens

a combinação de ambas as informações com o objetivo de alcançarmos melhores resultados na reconstrução.

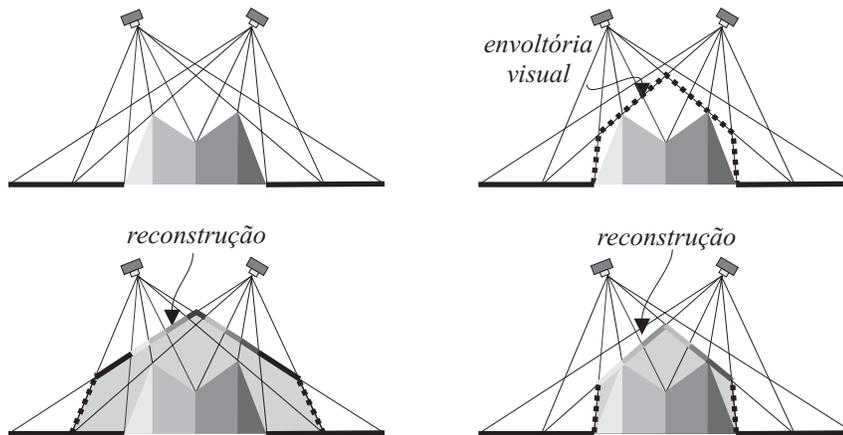


Figura 4.5: Complementariedade entre a envoltória fotométrica (photo hull) e envoltória visual (visual hull).

Não é muito difícil incorporar o conhecimento sobre a segmentação do objeto nas imagens de entrada nos mecanismos de checagem de consistência existentes nos métodos de escultura do espaço. Porém, precisamos especificar uma forma através da qual estas informações possam ser introduzidas, de modo natural, no esquema de processamento no espaço da cena.

A solução aqui apresentada é bastante simples e consiste em inserir o conhecimento das silhuetas do objeto nos canais de opacidade de cada uma das imagens de entrada, os quais por sua vez, são registrados em mapas de

textura, em conjunto com as informações fotométricas. Logo, a segmentação é mapeada no espaço da cena juntamente com as informações fotométricas através do procedimento de mapeamento projetivo de texturas.

Esta solução é semelhante à adotada no trabalho de Sainz et al. [44], mas de certa forma diferente, pois este último emprega técnicas de decalque (*stenciling*) em combinação com testes sobre o canal de opacidade (*alpha tests*) para inibir o desenho dos fragmentos que não correspondem aos objetos de interesse nas imagens projetadas. Em nossa implementação achamos este processo desnecessário, já que as informações de segmentação são transportadas juntamente com as informações fotométricas para a memória convencional, bastando um simples teste booleano para que possamos selecionar somente as regiões relevantes, isto é, as que não correspondem ao fundo da cena.

Pelos motivos apresentados acima, argumentamos que o uso de informações de segmentação no auxílio ao processo de escultura do espaço ainda tem importância considerável, principalmente quando poucas imagens são utilizadas. Além disso, quando buscamos efetuar a aquisição de um objeto específico da cena então esta abordagem é a mais adequada. No entanto, o mesmo não pode ser dito quando a reconstrução deve capturar toda a cena, de forma indistinta, isto é, tanto o fundo quanto os objetos à sua frente.

4.4

Tratamento da visibilidade

Como vimos anteriormente, é possível armazenar e manipular as informações sobre a visibilidade dos voxels, de maneira simples e eficiente, através da associação de um mapa de visibilidade a cada uma das câmeras utilizadas no processo de reconstrução. Apesar de simples e eficaz, esta estratégia precisa ser ligeiramente modificada para que possa ser utilizada quando o processamento é realizado no espaço da cena. Mais especificamente, precisamos transportar as informações de visibilidade para os planos de referência, o que também será feito através da técnica de mapeamento projetivo de texturas (Figura 4.6).

Propomos que os mapas de visibilidade Mv_i sejam representados através de mapas de textura, os quais devem ser projetados sobre o plano de referência corrente para que possamos registrar as informações de visibilidade às informações fotométricas. As imagens geradas sobre o plano de referência, por sua vez, são armazenadas em memória convencional, nos

mapas de visibilidade projetados Mvp_i , para que posteriormente possam ser utilizadas na etapa de determinação da foto-consistência.

Primeiramente, precisamos especificar como as informações nos mapas de visibilidade serão codificadas nos mapas de textura. Propomos a utilização de um canal de cor e um canal de opacidade em um mapa de textura RGBA, para armazenar a informação de visibilidade. Podemos utilizar a cor vermelha opaca, isto é $(r=255, g=0, b=0, a=1)$, para representar regiões ocultas e a cor negra transparente, isto é, $(r=0, g=0, b=0, a=0)$, para representar as regiões visíveis. A diferença nos canais de opacidade é estabelecida para que possamos, futuramente, ter controle sobre o efeito da operação de filtragem nos elementos próximos à fronteira entre as regiões visíveis e não visíveis em cada um dos mapas. Isto será importante para que possamos detectar aonde perdemos precisão na codificação da informação sobre visibilidade devido ao processo de reamostragem.

Durante o início do processo de escultura do espaço, todos os mapas de visibilidade são representados por mapas de textura completamente negros, o que significa que todos os voxels estão visíveis. À medida em que os voxels consistentes são detectados, tais mapas precisam ser atualizados para que possam refletir a nova configuração. Este procedimento deve ser realizado ao final da etapa de avaliação da foto-consistência dos voxels associados a um dado plano de varredura.

O processo de atualização consiste, basicamente, na alteração dos valores no canal vermelho e no canal de opacidade dos texels pertencentes às regiões correspondentes à projeção dos novos voxels foto-consistentes em cada um dos mapas de textura nos quais se encontram visíveis. Devemos estar atentos para o fato de que esta modificação deve ser feita no espaço das imagens e não no espaço da cena, pois é nele que os mapas de textura estão definidos. Isto pode ser feito através do desenho dos voxels detectados, em cor vermelha opaca, sobre cada um dos mapas de textura visualizados de acordo com a orientação e com a transformação projetiva da câmera correspondente.

Infelizmente, os mecanismos disponíveis para o desenho em mapas de textura não são nada simples. Em nosso caso, para atualizarmos um mapa de textura, temos que mapear projetivamente a textura atual sobre o plano de referência corrente, desenhar os novos voxels foto-consistentes com a cor apropriada, isto é, vermelha totalmente opaca, e em seguida copiar o conteúdo do framebuffer de volta para a região na memória de vídeo alocada para a textura.

Uma outra opção consiste em utilizar os novos mecanismos de desenho

direto em texturas (*rendering to texture*), que estão disponíveis nas novas implementações da OpenGL. Infelizmente, é consenso atual de que os mecanismos de desenho direto em texturas, pelo menos nas placas Nvidia, ainda são mais lentos do que simples cópias realizadas entre buffers e o mapa de textura, quer sejam *framebuffers* ou *pbuffers*.

Por outro lado, a solução baseada na cópia do conteúdo do *framebuffer* atualizado para a região do mapa de textura não produz soluções corretas para o nosso problema, pois cada vez que projetamos uma textura, para que possamos atualizá-la, causamos alterações em seus valores devido à operação de filtragem. A abordagem correta envolve um procedimento de correção dos valores, antes da cópia para o mapa de textura, de modo que os mapas sempre contenham valores constantes iguais a 0 ou 255. Experimentamos um procedimento de correção através de operação de mapeamento de cores que, no entanto, demonstrou ser excessivamente lento.

Por causa de todos estes problemas, adotamos uma estratégia bem simples, que apesar de não ser tão rápida, pelo menos nos fornece um tratamento correto para o problema de atualização dos mapas de visibilidade. Em primeiro lugar, associamos a cada mapa de visibilidade Mv_i uma imagem Imv_i que armazena o seu conteúdo em memória convencional. Então, para que possamos atualizar um determinado mapa de visibilidade Mv_i , precisamos apenas desenhar os novos voxels foto-consistentes detectados, visualizados segundo a câmera C_i correspondente, e copiar o conteúdo do framebuffer para uma imagem temporária em memória convencional. Em seguida, efetuamos uma operação de combinação entre a imagem temporária e a imagem Imv_i que representa o mapa de visibilidade, e finalmente, copiamos o conteúdo desta imagem para o mapa de textura em questão. Desta forma, eliminamos os problemas causados pela operação de filtragem que ocorriam na abordagem anterior.

Outras abordagens

Sainz et al. [77] propuseram uma solução diferente baseada em mapas de sombra (*shadow maps*), para determinar as regiões não-visíveis em relação a cada uma das câmeras durante o processo de avaliação da foto-consistência dos voxels interceptados por um determinado plano de referência.

Esta solução, apesar de bastante elegante, requer que todo o conjunto de voxels identificados como consistentes, em um determinado estágio do processo, sejam renderizados de forma que sua sombra possa ser determi-

nada sobre o plano de referência corrente, o que, dependendo da resolução do espaço de reconstrução, pode ser mais lento do que a atualização de um mapa de textura, estratégia esta que foi adotada em nosso trabalho.

Devemos nos lembrar que, nestes casos, não é possível a criação de listas de polígonos (*display lists*), o que tornaria o processo bastante eficiente, uma vez que o conjunto de voxels, e conseqüentemente o conjunto de polígonos, muda dinamicamente a cada nova interação. Um outro problema é que esta estratégia, de certa maneira, cria uma não uniformidade nos custos associados ao processamento de cada um dos planos de referência, já que os que se encontram próximos a cena serão processados de forma mais rápida que os planos mais distantes, quando boa parte do volume de reconstrução já tiver sido detectada.

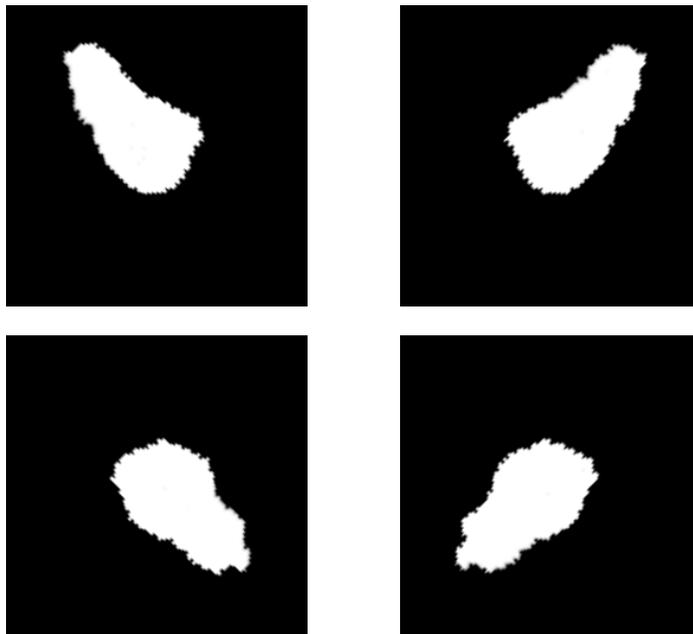


Figura 4.6: Mapas de visibilidade projetados sobre um plano de referência no espaço da cena

4.5

Determinação dos voxels foto-consistentes

A determinação dos voxels foto-consistentes é a etapa mais importante nos métodos de escultura do espaço, pois é através dela que podemos determinar quais são os elementos do espaço de reconstrução inicial que compõem a cena que desejamos reconstruir. Basicamente, tal processo procura verificar se as informações fotométricas provenientes das imagens, associadas a um determinado voxel, são coerentes segundo algum critério

pré-estabelecido. Nas subseções seguintes discutiremos em detalhes como este procedimento pode ser realizado.

4.5.1

Informações utilizadas no processo de avaliação da foto-consistência

Em nosso método de escultura do espaço utilizamos três tipos de informação para determinar a foto-consistência de um voxel: as informações fotométricas e as informações de segmentação do objeto nas imagens, armazenadas nas imagens projetadas I_{p_i} , e as informações sobre a visibilidade de cada elemento do espaço de reconstrução, armazenadas nos mapas de visibilidade projetados M_{vp_i} . Cada uma destas informações são reamostradas e registradas de modo apropriado durante o processo de mapeamento projetivo sobre os planos de referência, entretanto devemos nos lembrar que as mesmas são alteradas por causa das operações de filtragem, o que requer um cuidado especial quando tais informações forem utilizadas.

Primeiramente, a operação de filtragem altera sensivelmente os valores nos mapas de visibilidade e de opacidade, nos elementos próximos às transições entre regiões de classificação. Nos mapas de visibilidade, os elementos próximos às transições recebem uma mistura de valores correspondentes a visibilidade e não visibilidade, enquanto que nos mapas de segmentação, os elementos próximos às arestas recebem uma mescla de valores que correspondem a pertencer ao interior do objeto e a pertencer ao fundo da cena. Logo, é necessária a aplicação de uma função de reclassificação que consiga determinar, com um certo grau de precisão, quais eram as regiões originais antes do processo de filtragem.

Alguns artefatos e efeitos extremamente indesejáveis podem ser introduzidos na cena reconstruída caso esta reclassificação não seja realizada de forma adequada. Por exemplo, a reclassificação errônea dos mapas de segmentação pode fazer com que partes da região de fundo apareçam na reconstrução como objeto, caso não seja utilizado um limiar muito rígido para o estabelecimento do que consideramos como foto-consistente. Da mesma forma, estruturas muito sutis podem não aparecer no final da reconstrução, caso elementos pertencentes aos objetos de interesse sejam reclassificados como elementos do fundo da cena. Já a classificação inadequada dos mapas de visibilidade pode fazer com que voxels ocultos sejam considerado visíveis, também levando a classificações de foto-consistência incorretas. Além disso, voxels visíveis também podem ser considerados não visíveis fazendo com que partes da superfície da cena sejam classificadas como regiões não visíveis,

não recebendo assim a atribuição de cores adequada. Através de procedimentos empíricos, descobrimos que uma função de separação com um limiar igual a 0.9 do valor máximo igual a 255, é capaz segmentar as regiões de modo apropriado.

Um outro problema é que a filtragem também afeta os valores fotométricos nas imagens projetadas através de convoluções. Enquanto que no interior dos objetos de interesse isto é desejável, podemos ter problema nas bordas, já que isto acaba levando cores de fundo para o interior do objeto. Infelizmente este é um problema difícil de resolver, o qual envolveria processos de desconvolução nas regiões próximas às bordas, ou então a utilização de filtros seletivos, dos quais não temos conhecimento de sua existência em hardware gráfico. Uma possível solução, por outro lado, consistiria em pré-segmentar as imagens de entrada antes da definição dos mapas de textura correspondentes. Desta forma as regiões de fundo seriam representadas por uma cor negra completamente transparente ($r = 0, g = 0, b = 0, a = 0$), o que nos permitiria utilizar um processo de correção, ainda que aproximado.

Como a modificação nas informações fotométricas está diretamente relacionada à modificação no canal de opacidade, o qual armazena a segmentação do objeto de interesse nas imagens, podemos efetuar uma operação de correção com o objetivo de atenuar o efeito indesejável causado pelo processo de filtragem. Uma vez conhecidas as regiões correspondentes aos objetos de interesse, através dos mapas de segmentação reclassificados, podemos realizar tal correção através da divisão dos valores nos canais r , g e b pelo valor do canal de opacidade normalizado $a/255$. Desta forma os novos valores das componentes de cor de cada uma das amostras são dadas por:

$$r' = \frac{r}{a/255}, g' = \frac{g}{a/255}, b' = \frac{b}{a/255} \quad (4.5.1)$$

Podemos perceber que, no interior dos objetos de interesse, o canal de opacidade não sofre nenhuma modificação, isto é seu valor é sempre igual a 255 ($a/255 = 1$), o que significa que nenhuma correção é realizada. Quando ($0 < a/255 < 1$), os valores fotométricos são compensados para atenuar as modificações causadas pela média das cores originais com a cor negra totalmente transparente.

4.5.2

Determinação da foto-consistência em uma região planar

A avaliação da foto-consistência de um voxel v é realizada através da análise de um conjunto de elementos e_{st} determinados pela discretização da região planar rp associada a v . O número de elementos na região planar é definido a partir do fator de superamostragem com o qual as imagens são geradas. Em cada um destes elementos individuais, a análise é realizada através de um conjunto de amostras am_{st}^n tomadas das diferentes informações registradas sobre rp , as quais são provenientes de cada uma das câmeras. As coordenadas s e t de cada uma das amostras são definidas no sistema de coordenadas das imagens projetadas no plano de referência, enquanto que o índice n indica que a amostra é obtida das informações correspondentes à câmera C_n . Cada uma das amostras am_{st}^n carrega em si três tipos de informação: a visibilidade $am_{st}^n.vis$ de um elemento específico e_{st} , em relação à câmera C_n , proveniente do mapa projetado Mvp_n ; as intensidades nos canais r, g, b da imagem projetada Ip_n , nas coordenadas (s, t) , representadas respectivamente por $am_{st}^n.r$, $am_{st}^n.g$ e $am_{st}^n.b$; e a informação $am_{st}^n.a$, que caracteriza o elemento segundo sua pertinência a uma região correspondente aos objetos de interesse ou à região de fundo na imagem projetada Ip_n .

4.5.3

Classificação de um voxel segundo sua visibilidade

Primeiramente, determinamos quais são os elementos ev^{st} , associados a um determinado voxel v , que se encontram visíveis em pelo menos uma das imagens. Isto pode ser feito através da observação do valor do campo de visibilidade $am_{st}^n.vis$ das amostras correspondentes a cada um deles. Caso não haja nenhum elemento visível, em pelo menos uma das imagens de entrada, determinamos o voxel correspondente como trivialmente foto-consistente.

4.5.4

Classificação de um voxel com base nas informações de segmentação

Em seguida, verificamos se cada um dos elementos visíveis da região planar corresponde a uma região associada aos objetos de interesse em todas as imagens projetadas, isto é, se $am_{st}^i.a = 1$, para $i = 0, \dots, n$, em cada elemento visível ev^{st} . Caso esta condição não seja satisfeita, consideramos o

voxel como não-foto-consistente. Podemos ser menos rígidos, determinando como não-foto-consistente somente os voxels que possuam um número de elementos pertencentes a regiões de fundo maior que um determinado percentual do total de elementos na superamostragem.

4.5.5

Classificação de um voxel com base em critérios fotométricos

Caso não seja possível descartar a foto-consistência de um voxel com base nas informações de segmentação, precisamos determinar sua foto-consistência com base na coerência entre as informações fotométricas das amostras associadas a cada um dos elementos pertencentes a sua região planar.

Relembramos que um voxel visível foto-consistente é um voxel que, na ausência de ruído e de efeitos de quantização, se projeta em um conjunto de pixels de mesma cor. Considerando o processamento no espaço da cena, nas mesmas condições ideais, podemos afirmar, de modo equivalente, que um voxel é foto-consistente se existe um fragmento de superfície em seu interior, no qual as imagens de entrada se projetam gerando imagens idênticas.

Nos casos ideais, a avaliação da foto-consistência de um voxel visível se reduz a um simples teste de comparação entre as subimagens geradas pela reprojeção das imagens de entrada sobre a região planar associada. Caso as subimagens sejam iguais, classificamos o voxel como foto-consistente, caso contrário, o consideramos como não-foto-consistente. Na prática não podemos nos basear em um teste tão simples assim, pois a probabilidade de que as imagens geradas pela reprojeção das imagens de entrada sobre uma superfície da cena sejam iguais é muito pequena na presença de ruído e outros artefatos introduzidos pelos sistemas de aquisição. Por este motivo, precisamos efetuar testes estatísticos, os quais são capazes de levar em consideração a existência de variações introduzidas pelo sistema de aquisição em imagens, que em teoria, provêm de uma mesma origem. Para isto, no entanto, é necessário que compreendamos o processo de formação das imagens de entrada em termos probabilísticos e não determinísticos.

Dado um conjunto de imagens obtidas pela reprojeção das imagens de entrada sobre uma superfície da cena, devemos interpretar as sutis diferenças entre os valores de intensidade das componentes de cor de elementos correspondentes, não como um conjunto de valores distintos sem nenhuma relação entre si, e sim, como diferentes realizações de um conjunto de variáveis aleatórias caracterizadas por uma certa distribuição.

Desta forma, as diferenças detectadas entre as imagens reprojeta- das, sobre uma superfície da cena, que no caso ideal deveriam ser idênticas, po- dem ser explicadas pelo desvio natural existente na distribuição de probabi- lidade que caracteriza as variáveis aleatórias que modelam o comportamento em cada sensor das câmeras. Obviamente, se estes valores são muito diferen- tes então podemos suspeitar de uma certa inconsistência, o que significa que provavelmente tal superfície não pertence aos objetos que geraram as ima- gens de entrada. Vejamos agora, como a determinação da foto-consistência de um voxel pode ser realizada através de um teste de hipótese estatístico.

Teste de hipótese para determinação da foto-consistência

Para facilitar as notações iremos efetuar as análises supondo que as informações fotométricas são descritas por um única componente de cor, já que a extensão para três ou mais componentes pode ser feita de forma trivial, se considerarmos que elas são independentes, hipótese que assumiremos aqui.

Seja e_{st} o elemento na posição (s, t) em uma região planar rp associada a um voxel v . Modelamos a informação fotométrica em cada imagem Ip_i , na posição correspondente a e_{st} como uma realização de uma variável aleatória X_i , com distribuição normal, de média μ igual à cor do objeto e variância σ_i^2 , correspondente ao ruído introduzido pelo respectivo sensor. Consideramos, assim, que as variâncias são possivelmente diferentes para cada imagem, devido à diferença dos sensores. Neste modelo, o valor de μ é considerado desconhecido, mas os valores de σ_i^2 , para cada i , são conhecidos (estimados, por exemplo, a partir de um conjunto de observações de uma mesma cena, tomadas a partir de cada câmera).

Para verificar se as informações fornecidas pelas diversas imagens Ip_i na posição (s, t) são consistentes, devemos testar a hipótese de que, de fato, as médias correspondentes às diversas observações sejam iguais. Assim, nosso problema é o de, dadas observações X_1, X_2, \dots, X_n , independentes, com distribuição $N(\mu_i, \sigma_i^2)$, testar a hipótese $H_0 : \mu_1 = \mu_2 = \dots = \mu_n$.

Inicialmente, devemos estimar, sob a hipótese H_0 , a média comum a todas as distribuições. Intuitivamente, tomar simplesmente a média aritmética de X_1, X_2, \dots, X_n não é adequado devido à diferença entre os erros nas distribuições.

De fato, uma melhor estimativa para a média μ pode ser obtida através do *estimador de máxima verossimilhança*, o qual retorna um valor que maximiza a probabilidade de obtermos as amostras observadas, na ordem

particular em que os elementos da mesma aparecem. Em nosso caso, a função de verossimilhança é dada por

$$f(x, \mu) = \frac{1}{\sqrt{2\pi}\sigma} \int -e^{\sum (X_i - \mu)^2 / (\sigma_i^2)} dx_i \quad (4.5.2)$$

a qual tem valor máximo quando μ satisfaz

$$\frac{\partial \sum \left(\frac{X_i - \mu}{\sigma_i} \right)^2}{\partial \mu} = 0, \quad (4.5.3)$$

Logo, o estimador de máxima verossimilhança é dado por

$$X' = \frac{\sum_{i=1}^n X_i / \sigma_i^2}{\sum_{i=1}^n 1 / \sigma_i^2} \quad (4.5.4)$$

A estatística utilizada no teste para validação da hipótese de igualdade entre as médias das variáveis aleatórias associadas a cada amostra é dada por $\sum_{i=1}^n \left(\frac{X_i - X'}{\sigma_i} \right)^2$. A distribuição desta estatística é dada pelo teorema abaixo:

Teorema 4.5.1 *Sejam X_1, X_2, \dots, X_n independentes, com $X_i \sim N(\mu, \sigma_i^2)$. Então $\sum_{i=1}^n \left(\frac{X_i - X'}{\sigma_i} \right)^2 \sim \chi_{n-1}^2$, onde $X' = \frac{\sum_{i=1}^n X_i / \sigma_i^2}{\sum_{i=1}^n 1 / \sigma_i^2}$.*

Prova 4.5.2 *Defina $Y_i = \frac{X_i - \mu}{\sigma_i}$, $i = 1, \dots, n$ (evidentemente, Y_1, \dots, Y_n são independentes com distribuição $N(0, 1)$). Então*

$$\begin{aligned} X' &= \frac{\sum_{i=1}^n (\mu + \sigma_i Y_i) / \sigma_i^2}{\sum_{i=1}^n 1 / \sigma_i^2} \\ &= \mu + \underbrace{\frac{\sum Y_i / \sigma_i}{\sum 1 / \sigma_i^2}}_{Y'} \end{aligned}$$

Logo,

$$\begin{aligned} \sum_{i=1}^n \left(\frac{X_i - X'}{\sigma_i} \right)^2 &= \\ \sum_{i=1}^n \left(\frac{\sigma_i Y_i + \mu - \mu - Y'}{\sigma_i} \right)^2 &= \sum_{i=1}^n (Y_i - Y' / \sigma_i)^2. \end{aligned}$$

Portanto, temos que provar que Y_1, \dots, Y_n iid $N(0, 1)$ implica em

$$\sum_{i=1}^n \left(Y_i - \frac{Y'}{\sigma_i} \right)^2 \sim \chi_{n-1}^2.$$

Seja A uma matriz ortogonal em que a primeira linha é $\left(\frac{1/\sigma_1}{\sqrt{\sum 1/\sigma_i^2}}, \frac{1/\sigma_2}{\sqrt{\sum 1/\sigma_i^2}}, \dots, \frac{1/\sigma_n}{\sqrt{\sum 1/\sigma_i^2}} \right)$ e seja

$$\begin{pmatrix} Z_1 \\ \vdots \\ Z_n \end{pmatrix} = A \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

Como A é ortogonal, $\sum_{i=1}^n Z_i^2 = \sum_{i=1}^n Y_i^2$. Além disso, Z_1, \dots, Z_n são *iid* $N(0, 1)$. Observamos, ainda, que

$$\begin{aligned} Z_1 &= \frac{\sum_{i=1}^n (1/\sigma_i) Y_i}{\sqrt{\sum 1/\sigma_i^2}} \\ &= \sqrt{\sum 1/\sigma_i^2} \cdot Y' \end{aligned}$$

Agora:

$$\begin{aligned} \sum_{i=1}^n (Y_i - Y'/\sigma_i)^2 &= \sum Y_i^2 - 2Y' \underbrace{\sum Y_i/\sigma_i}_{Y' \sum 1/\sigma_i^2} + Y'^2 \sum 1/\sigma_i^2 \\ &= \sum Y_i^2 - 2Y'Y' \sum 1/\sigma_i^2 + Y'^2 \sum 1/\sigma_i^2 \\ &= \sum Y_i^2 - Y'^2 \sum 1/\sigma_i^2 \\ &= \sum_{i=1}^n Z_i^2 - \underbrace{(Y' \sqrt{\sum 1/\sigma_i^2})^2}_{Z_1^2} \\ &= \sum_{i=2}^n Z_i^2 \sim \chi_{n-1}^2 \end{aligned}$$

O teste para verificar a validade da hipótese $H_0 : \mu_0 = \mu_1 = \dots = \mu_n$ consiste em determinar se o valor da estatística calculada $Z = \sum \left(\frac{X_i - X'}{\sigma_i} \right)^2$ está contido em uma região crítica RC dada por $[0, \chi_0^2]$, tal que $P(Z \in RC) = P(Z < \chi_0^2) = \alpha$, onde α é um valor de significância. Se $\chi_0^2 \in RC$, aceitamos H_0 , caso contrário, a rejeitamos.

Para verificarmos a foto-consistência de um voxel, levando em consideração todos os elementos e_{uv} , podemos efetuar um teste de hipótese com base em uma estatística dada por

$$\mathbf{Z} = \frac{1}{c \cdot d} \sum_{s=0}^c \sum_{t=0}^d Z_{st}, \text{ onde} \quad (4.5.5)$$

$$\mathbf{Z}_{st} = \sum \left(\frac{X_{ist} - X'_{st}}{\sigma_{ist}^2} \right)^2 \quad (4.5.6)$$

Como a soma de distribuições χ^2 é uma outra soma de distribuições χ^2 , então,

$$\mathbf{Z} \sim \chi_{cd(n-1)}^2 \quad (4.5.7)$$

Finalmente, o teste sobre a validade da hipótese $H_0 : \mu_{0st} = \mu_{1st} = \dots = \mu_{nst}$, para cada elemento e_{st} , na região planar associada ao voxel v , pode ser feito, da mesma forma, através de um teste de hipóteses baseado na estatística $\chi_{cd(n-1)}^2$ para um determinado nível de significância.

Implementação dos testes estatísticos

Para que possamos implementar os testes apresentados neste capítulo, precisamos primeiramente estimar o ruído introduzido pelos sensores das câmeras durante o processo de aquisição das imagens. Isto pode ser feito, de modo simples, tomando um conjunto relativamente grande de imagens através de cada uma das câmeras e estimando a variância a partir dos valores observados em cada um dos pixels. Desta forma, para cada câmera construímos um mapa de variâncias para cada pixel que por sua vez está associado a cada sensor.

O segundo passo, este um pouco mais complexo, consiste em codificar as informações sobre a variância em mapas de textura para que possamos registrar tais informações com as demais informações nos planos de referência no espaço da cena. Podemos por exemplo, codificar as variâncias estimadas nos dois canais não utilizados nos mapas de textura que representam os mapas de visibilidade. Obviamente, este processo requer uma operação de quantização, e por sua vez, de interpolação, no momento em que as informações forem resgatadas para utilização nos testes que avaliam a foto-consistência de um voxel através de elementos de uma região planar.

Um dos problemas a serem investigados em trabalhos futuros é o efeito das operações de filtragem sobre as variâncias codificadas nos mapas de textura. Isto pode ter um impacto significativo quando as imagens são mapeadas nos planos de referência com resoluções muito diferentes da original, o que certamente ocorrerá se adotarmos estratégias por refinamento adaptativo que utilizam representações das imagens de entrada em múltiplas resoluções.

Como veremos no capítulo 7, na seção 7.5, os resultados produzidos pelo método estatístico aqui descrito não foram totalmente satisfatórios. Isto se deve ao fato de que há outras fontes de erro, além do ruído da câmera. Por este motivo, utilizamos como alternativa, um método empírico, mais simples, que proporcionou resultados razoáveis.

O teste utilizado baseia-se apenas no desvio detectado entre as amostras visíveis. Para cada elemento visível ev^{st} na região planar que não se projeta em uma região de fundo, colhemos um conjunto de amostras visíveis, relativas a cada uma das imagens projetadas. Em seguida, calculamos o desvio padrão dp_{st} para cada uma das componentes de cor do conjunto de amostras de cada um dos elementos. Finalmente, calculamos a média dpm dos desvios padrão de cada conjunto de amostras am_{st}^n e comparamos com um limiar escolhido previamente. Caso o valor dado pela média dos desvios padrão for menor que o limiar escolhido, aceitamos o voxel como foto-consistente, caso contrário o rejeitamos. Este teste está descrito em pseudo-linguagem no algoritmo 5.

Algoritmo 5 Avalia foto-consistência(rp, Ip, Mvp)

Determine o conjunto de elementos visíveis $Ev = \{ev_{st} \in rp \mid \exists n, Mvp_{nst} = 1\}$
se $Ev = \emptyset$ **então**
 retorne CONSISTENTE
fim se
para cada $ev_{st} \in E_v$ **faça**
 se $\exists am_{st}^n$ de ev_{st} tal que $Ip_{nst.a} \leq tolfund$ **então**
 retorne INCONSISTENTE
 fim se
 $dp_{st} = \frac{1}{n-1} \sum_{l=1}^n (am_{st}^l - \overline{am}_{st})^2$
 fim para
 $dpm = \frac{1}{|Ev|} \sum_{s,t} (dp_{st})$
 se $dpm < limiar$ **então**
 retorne CONSISTENTE
 senão
 retorne INCONSISTENTE
fim se

4.6

Atribuição de cor aos voxels

Após avaliar a foto-consistência de um voxel, precisamos lhe atribuir as informações fotométricas, de acordo com a sua classificação. De fato, a atribuição das informações fotométricas é também parte importante

do processo de reconstrução, já que as cenas ou objetos que desejamos reconstruir não são definidos somente por sua geometria, mas também por suas cores.

A atribuição das informações fotométricas de um voxel pode ser feita de forma bastante simples, com base em um pequeno conjunto de regras. Quando um determinado voxel é considerado não-foto-consistente, então simplesmente atribuímos o valor zero a seu valor de opacidade, o que significa que ele se tornará invisível na cena. Por sua vez, quando um voxel é considerado foto-consistente, mas não está visível em nenhuma das câmeras, então, podemos atribuir qualquer conjunto de cores a cada um dos canais que definem seus atributos de cor. Por exemplo, podemos atribuir o valor 128 a cada um dos canais de cor e o valor 1 ao canal de opacidade. Finalmente, quando o voxel é foto-consistente e visível em alguma das câmeras, atribuímos a cada uma das suas componentes de cor uma média dos valores nas componentes de cor de cada uma das amostras tomadas da região planar utilizada na determinação da foto-consistência. Obviamente, neste caso, atribuímos um valor igual a 1 à componente de opacidade do voxel em questão.

4.7

Algoritmo

Após descrever e analisar cada uma das etapas isoladamente, podemos integrá-las na forma de um algoritmo. Abaixo especificamos o algoritmo de coloração de voxels no espaço da cena (Algoritmo 6).

4.7.1

Complexidade

Não é difícil verificar que a complexidade do algoritmo é idêntica à do algoritmo original no espaço da imagens [44]. Não podemos, porém, deixar de observar que o algoritmo no espaço da cena conforme foi implementado é mais lento que o algoritmo original justamente pelo fato de precisarmos realizar cópias do *framebuffer* para a memória convencional.

Algoritmo 6 Varredura de Planos no Espaço da Cena(n, I, C, limiar)

Determine um conjunto de voxels V que contenha a cena.

Limpe os mapas de visibilidade Mv_i .

para cada plano de referência π^k , que corta uma camada de voxels V_k a uma distância $dist(k)$ do conjunto de câmeras C **faça**

Determine o conjunto de imagens projetadas Ip .

Determine o conjunto de mapas de visibilidade projetados Mvp .

para cada voxel $v_{ijk} \in V^k$ **faça**

Determine a região planar $rp_{ijk} = v_{ijk} \cap \pi_k$.

$v.fconsist = \text{AvaliaFotoConsistência}(rp_{ijk}, Ip, Mvp, \text{limiar})$

se $v.fconsist = \text{CONSISTENTE}$ **então**

Atribua a média das componentes de cor das amostras de rp_{ijk} às respectivas componentes de cor de v_{ijk} .

senão

Atribua o valor zero à opacidade de v_{ijk} .

fim se

fim para

Atualize cada um dos mapas de visibilidade Mv_i .

fim para

4.8

Análise

O processamento do espaço da cena, fundamentado na técnica de mapeamento projetivo de textura, permite um tratamento unificado e eficiente das operações de registro, reamostragem e manipulação das informações necessárias à avaliação da foto-consistência dos elementos que compõem o volume de reconstrução inicial. Apesar das vantagens introduzidas por esta técnica e de sua relativa eficiência, podemos perceber, que o mecanismo básico ainda é o mesmo do método original, isto é, está fundamentado em uma seqüência de consultas efetuadas de forma exaustiva sobre um espaço uniformemente particionado.

O ponto chave, nesta afirmação, é o de que os métodos de escultura do espaço que conhecemos, à exceção do método de Prock, não são capazes de aproveitar, em sua totalidade, as informações sobre a organização espacial da cena que podem ser inferidas a partir das imagens de entrada, as quais podem tornar o processo bem mais eficiente.

No capítulo seguinte, iremos propor um método de escultura do espaço por refinamento adaptativo com base em uma subdivisão hierárquica do espaço através de uma octree e multiresolução das imagens de entrada.

5

Escultura do espaço através de refinamento adaptativo

Os métodos de escultura do espaço apresentados no segundo capítulo trabalham sobre uma representação do espaço de reconstrução determinada por uma *subdivisão espacial regular*. Embora seja simples, e permita a utilização de uma grande variedade de algoritmos, esta forma de estruturação do espaço, normalmente representada por uma matriz de voxels, possui algumas desvantagens que em certos casos podem tornar bastante difícil a aplicação do método.

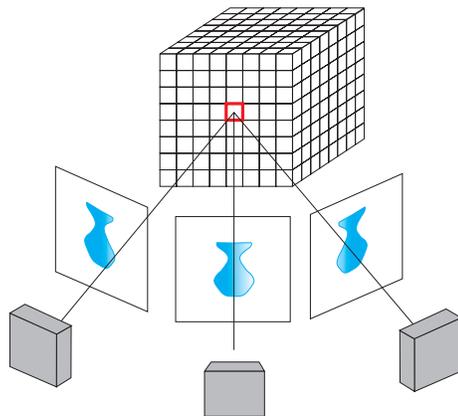


Figura 5.1: Espaço de reconstrução representado por voxels.

Em primeiro lugar, a representação do espaço de reconstrução através de um conjunto de voxels requer a utilização de um espaço de armazenamento cujo tamanho é proporcional à resolução com a qual se deseja reconstruir a cena, o que pode inviabilizar a utilização de métodos de escultura do espaço quando são necessárias reconstruções de alta qualidade. Além disso, nestes casos, é preciso avaliar a foto-consistência de um número gigantesco de elementos, tornando o processamento extremamente caro e conseqüentemente, lento.

Muitas das vezes, podemos classificar regiões extensas no espaço da cena de uma só vez, bastando para isso, tomar decisões com base nas propriedades fotométricas locais de tais regiões, as quais são determinadas

pelas câmeras e pelas imagens de entrada. Por exemplo, em muitos casos, grandes regiões vazias no espaço da cena podem ser facilmente detectadas verificando-se se a sua projeção nas imagens corresponde a um conjunto de pixels de fundo, não sendo necessária a avaliação da foto-consistência individual de um conjunto enorme de pequenos elementos estruturais em seu interior.

Obviamente, para que possamos efetuar processamentos sobre regiões extensas, como mencionamos acima, é necessário saber muito bem quais são estas regiões e como elas estão organizadas, o que infelizmente só é possível quando já tivermos conhecimento sobre como os elementos da cena se distribuem no espaço. Por outro lado, podemos descobrir as propriedades do espaço da cena de forma progressiva, desde que estructuremos o espaço de reconstrução convenientemente e adotemos mecanismos capazes de aproveitar tal estrutura.

Com base nesta idéia, iremos propor neste capítulo um método de escultura do espaço por refinamento adaptativo no qual o espaço de reconstrução é representado através de uma subdivisão espacial adaptativa. Mostraremos que tal método é capaz de realizar o processo de reconstrução de maneira mais eficiente, empregando esforços proporcionais às propriedades locais da cena, as quais são descobertas à medida em que a reconstrução é realizada.

5.1

Motivação

Sabemos que, quanto mais conhecimento tivermos sobre a estrutura do espaço no qual um determinado problema está definido, mais facilmente poderemos traçar estratégias capazes de tomar decisões que levem a soluções eficientes. Um exemplo típico são os problemas de busca em um conjunto de dados espaciais. Quanto melhor estruturados estiverem estes dados, levando em consideração as propriedades do espaço a que pertencem e as próprias relações de vizinhança existentes entre si, mais eficientes serão as operações capazes de identificar elementos com as características desejadas. Além disso, mais facilmente poderemos efetuar modificações locais sobre ela como, por exemplo, a remoção e a inserção de dados.

Os métodos de escultura do espaço estão interessados essencialmente na determinação dos elementos foto-consistentes na superfície dos objetos de interesse no espaço da cena. Conseqüentemente, podemos de certo modo, compreendê-los como métodos que procuram encontrar, em um determinado

espaço de busca, um conjunto de elementos que satisfazem determinadas propriedades. Conseqüentemente, quanto mais informações tivermos sobre esse espaço de busca, isto é, sobre o espaço de reconstrução, mais facilmente poderemos encontrar os elementos relevantes e conseqüentemente efetuar a reconstrução da cena.

É bastante evidente que a estruturação obtida através de uma matriz de voxels apenas estabelece uma organização conveniente do espaço, não incorporando em si mesma nenhuma informação sobre as propriedades locais associadas às regiões da cena. O máximo que os métodos de escultura do espaço conseguem extrair de tais estruturas, com o objetivo de auxiliar o processo de reconstrução, são as relações de ordem, e em alguns casos, de vizinhança, herdadas diretamente das propriedades geométricas e topológicas intrínsecas a uma partição uniforme de um subconjunto do espaço 3D euclidiano.

Se conseguíssemos estruturar o espaço de reconstrução de maneira que pudéssemos empregar esforços computacionais proporcionais as características locais das regiões do espaço em questão, inferidas a partir das informações codificadas nas imagens de entrada, então poderíamos atingir um grau de eficiência maior que o alcançado através de um mecanismo de busca exaustiva em uma partição uniforme. Na verdade, isto pode ser feito através do que chamamos de *método adaptativo*.

Um *método adaptativo* é um método capaz de se ajustar de acordo com as propriedades locais do espaço associado à definição do problema, quer seja através de mudanças de estratégia, quer seja através do emprego de esforços e recursos proporcionais às características locais das regiões pertencentes a tal espaço.

A primeira impressão que temos é a de que é fundamental o conhecimento prévio da organização e das características locais do espaço de definição do problema para que possamos utilizar métodos adaptativos.

Com relação ao problema de reconstrução de cenas através de escultura do espaço, podemos argumentar que, a princípio, não temos nenhum conhecimento sobre as propriedades do espaço de reconstrução, o que nos impede de estruturá-lo de modo mais inteligente, tornando impossível, desta forma, a realização de um processo de busca mais eficiente. Segundo este raciocínio, estamos limitados à estruturação do espaço de busca através de uma partição uniforme e a realização de um processo de busca exaustiva.

Entretanto, podemos observar que a estrutura da cena pode ser determinada progressivamente através de um processo de *refinamento* no qual a reconstrução é realizada inicialmente sobre uma representação bastante

grosseira do espaço de reconstrução inicial, que por sua vez é refinado em certas regiões à medida em que as propriedades do espaço que contém a cena são determinadas com base nos critérios de foto-consistência.

Desta forma, a estrutura do espaço de busca, isto é, do espaço formado pelos elementos que devem ser classificados, é determinada concomitantemente com a própria reconstrução da cena, tornando o processo muito mais ágil, já que deste modo podemos empregar esforços proporcionais às características locais da cena em questão.

Antes de analisarmos os requisitos necessários ao desenvolvimento de um método de escultura do espaço por refinamento adaptativo, iremos descrever, de forma bem sucinta, alguns dos trabalhos que abordaram previamente este aspecto do problema e quais as dificuldades por eles encontradas.

5.2

Alguns trabalhos anteriores

Até onde temos conhecimento, apenas três trabalhos abordaram, em diferentes níveis de profundidade, o desenvolvimento de métodos para reconstrução de cenas a partir de imagens por refinamento adaptativo: o trabalho de Szeliski [29], o trabalho de Prock [52] e o trabalho de Kutulakos [65].

Na verdade, apenas Prock [52] realmente propôs um método de escultura do espaço por refinamento adaptativo. O método de reconstrução de Szeliski[29], apesar de propor um esquema de refinamento adaptativo, não se baseia em critérios de foto-consistência, utilizando apenas informações provenientes de silhuetas, o que não o caracteriza, desta forma, como um verdadeiro método de escultura do espaço. Kutulakos[65], por sua vez, não chegou definitivamente a desenvolver um método de refinamento adaptativo, apesar de estabelecer as bases para isso, ao propor a avaliação da foto-consistência de elementos não pontuais através de *shuffle transforms*. De fato, em seu trabalho ele afirma que a aplicação de *shuffle transforms* no desenvolvimento de tais métodos ainda é um tópico em aberto, o qual necessita ser investigado com maior profundidade.

A utilização de estratégias de refinamento adaptativo na reconstrução de cenas parece ter surgido juntamente com o trabalho de Szeliski, o qual, na época, apresentou resultados bastante surpreendentes. No entanto, o que parecia ser uma idéia bastante promissora, acabou sendo praticamente

esquecida quando os métodos baseados em foto-consistência começaram a ser adotados.

Uma possível explicação para a não utilização desta técnica está no fato de que a teoria sobre forma baseada em foto-consistência está fundamentada sobre o conceito matemático de ponto, enquanto que os métodos de escultura do espaço por refinamento adaptativo envolvem a avaliação de foto-consistência sobre elementos que dificilmente podem ser aproximados através de elementos pontuais. De fato, a avaliação da foto-consistência de elementos de natureza não pontuais é um problema bastante difícil, cuja compreensão é fundamental para que possamos chegar a uma boa modelagem para o problema de reconstrução volumétrica de cenas através de refinamento adaptativo.

Apesar das dificuldades descritas acima, Prock propôs um método de coloração de voxels por refinamento adaptativo que utiliza uma coloração obtida em baixa resolução como ponto de partida para a obtenção de colorações em resoluções mais altas. Sem entrar em muitos detalhes, o método funciona da seguinte forma: efetua-se uma coloração de voxels sobre um espaço de reconstrução em baixa resolução e em seguida, seleciona-se os voxels que foram considerados foto-consistentes. Estes voxels são subdivididos em oito novos voxels sobre os quais é efetuado um novo passo de coloração de voxels. O processo é aplicado recursivamente até que tenha-se alcançado a resolução desejada.

Prock imediatamente percebeu que esta estratégia não levava a resultados corretos, já que permitia que regiões contendo fragmentos de superfície foto-consistente pudessem ser eliminadas durante o processamento nos níveis de resolução mais grosseiros. Para contornar este problema, que está diretamente relacionado à dificuldade de se definir foto-consistência em regiões extensas do espaço da cena, Prock utilizou uma estratégia de busca local para evitar a perda de voxels consistentes durante as etapas realizadas em baixa resolução. Sua idéia consiste em subdividir e processar, em uma resolução mais alta, não somente os voxels considerados consistentes em um determinado nível de resolução, mas também todos os voxels que estiverem em uma vizinhança unitária dos voxels escolhidos (Figura 5.2).

Aparentemente, para simplificar este processo de busca, Prock não utilizou uma estruturação adaptativa do espaço da cena. Simplesmente, a cada novo nível de refinamento, todo espaço é discretizado uniformemente com uma resolução maior, embora somente os elementos selecionados no nível anterior sejam considerados na nova etapa de coloração de voxels.

Apesar de produzir resultados muito bons, o método proposto por

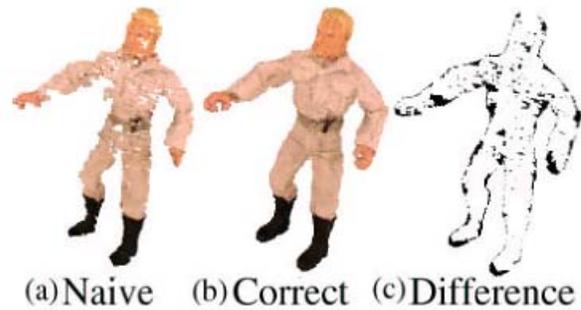


Figura 5.2: Coloração através de refinamento. Figura obtida de [52]. (a) - reconstrução com perdas; (b) - reconstrução correta; (c) - diferença

Prock nos obriga a iniciar o processo de reconstrução com uma discretização do espaço fina o suficiente para que o processo de coloração seja capaz de capturar ao menos um voxel no interior de cada componente conexa correspondente a cada objeto de interesse na cena. Denominamos tais voxels de *sementes* da reconstrução. Caso algumas destas sementes sejam perdidas logo no início do processo, o objeto correspondente não aparecerá no final da reconstrução.

Iremos agora descrever o método de escultura do espaço através de refinamento que propomos neste trabalho, começando pela descrição do esquema utilizado para representar o espaço de reconstrução. De forma surpreendente, a escolha deste esquema nos dará pistas sobre como tratar a questão sobre a avaliação da foto-consistência de elementos não pontuais.

5.3

Representação adaptativa do espaço de reconstrução

Um dos pontos chaves para o desenvolvimento de um método de escultura do espaço por refinamento adaptativo está na escolha do esquema utilizado para representar o espaço de reconstrução. Para que possa permitir que a reconstrução seja realizada de forma incremental e adaptativa, ao mesmo tempo em que preserve as vantagens associadas aos métodos de escultura do espaço, o esquema de representação utilizado deve possuir as seguintes propriedades:

Poder de expressão: o esquema de representação deve ser capaz de representar a cena reconstruída com grau de precisão arbitrário; além disso, deve ser o mais genérico possível, não impondo restrições sobre a topologia

da cena a ser reconstruída.

Concisão: a representação deve ser feita de forma concisa aproveitando a coerência espacial existente entre as regiões do espaço no qual a cena está definida.

Capacidade de representar relações de ordem: esta propriedade é útil para o desenvolvimento de eficientes algoritmos de escultura do espaço. Vimos anteriormente que o processo de escultura do espaço pode ser desempenhado de forma muito mais fácil quando existem relações de ordem bem definidas entre os elementos que compõem o espaço de reconstrução. Logo, é importante que o esquema permita que tais relações sejam extraídas de maneira simples e eficiente, sem um aumento da complexidade dos algoritmos baseados em varreduras por planos.

Dinamismo e adaptabilidade: o esquema de representação deve ser dinâmico, de maneira que possa acomodar novas informações que são inferidas no decorrer do processo de escultura do espaço. Além disso, os elementos estruturais utilizados na representação devem se adaptar, da melhor forma possível, às características do modelo que procuram representar.

Os esquemas de representação do espaço que satisfazem com maior naturalidade as propriedades descritas acima são os esquemas baseados em *subdivisão espacial adaptativa*. Tais esquemas se baseiam na propriedade fundamental de que o número de elementos necessários para representar um sólido, através de uma subdivisão do espaço, é proporcional à sua área e não ao seu volume. Logo, enquanto os esquemas baseados em enumeração exaustiva utilizam um número de elementos proporcional a r^3 , onde r é a resolução utilizada, os esquemas baseados em subdivisão adaptativa utilizam um número de elementos proporcional a r^2 .

Computacionalmente, é comum representar os esquemas de subdivisão espacial adaptativa através de estruturas espaciais hierárquicas como, por exemplo, *octrees*, *bintrees*, *k-d trees*, etc. Essas estruturas são capazes de representar uma região do espaço adaptativamente, com base em um processo de subdivisão seletiva, segundo o qual são refinadas somente as subregiões estritamente necessárias. Além disso, por serem dinâmicas e de fácil atualização, são capazes de capturar a organização do espaço de forma progressiva, tornando possível o processo de reconstrução por refinamento.

Por outro lado, devido ao fato de permitirem configurações mais genéricas, tais estruturas não são tão eficazes quanto as estruturas baseadas em particionamentos regulares, na representação das relações de ordem existentes entre os elementos primitivos utilizados na descrição do espaço.

Nestas últimas, uma simples função, definida sobre os índices associados aos seus elementos primitivos, é capaz de nos informar imediatamente quando um determinado elemento está à frente de um outro, segundo uma certa direção. Da mesma forma, nas estruturas espaciais hierárquicas, as relações de vizinhança entre os elementos não são tão explícitas quanto nas estruturas determinadas por partições uniformes.

Tanto as relações de ordem quanto de vizinhança podem ser extraídas a partir das estruturas hierárquicas através de processos de busca, que apesar de não serem tão eficientes quanto uma simples operação sobre índices, não são extremamente custosas, sendo normalmente descritas através de operações de percorrimento em árvores. Veremos mais adiante que tais processos de busca serão fundamentais para que possamos determinar quais elementos da estrutura devem estar ativos em cada uma das iterações do nosso algoritmo de escultura do espaço por refinamento adaptativo.

Em nosso trabalho, representaremos o espaço de reconstrução através de uma octree, que apesar de ser uma estrutura simples, é poderosa o suficiente para atender os requisitos necessários para o desenvolvimento de um método por refinamento adaptativo.

As *octrees* são estruturas espaciais hierárquicas que se baseiam em uma subdivisão recursiva do espaço de interesse em oito octantes, a qual é representada através de uma árvore 8-ária (Figura 5.3). Quando uma região do espaço é representada através de uma octree, cada um dos oito octantes iniciais pode ser classificado em *cheio*, *vazio* ou *parcialmente cheio*, dependendo da sua interseção com o objeto em questão. Os octantes parcialmente cheios devem ser subdivididos em novos oito octantes, os quais são classificados recursivamente da mesma forma que os 8 primeiros. O processo termina apenas quando todos os octantes são considerados homogêneos ou até que um grau de subdivisão máximo, pré-estabelecido, tenha sido alcançado. Na estrutura resultante, os octantes parcialmente cheios compõem os nós internos da árvore, enquanto que os octantes cheios e vazios determinam as suas folhas.

Através do uso de uma estrutura octree, é possível desenvolver um método de escultura do espaço capaz de concentrar esforços nas regiões próximas à superfície dos objetos de interesse, a qual por sua vez, é o alvo principal dos métodos de escultura do espaço, apesar de toda sua conotação volumétrica.

Desta forma, grandes regiões vazias ou grandes espaços pertencentes ao interior do objeto são determinadas logo nos estágios iniciais do processo de reconstrução, sendo representadas através de células grandes de

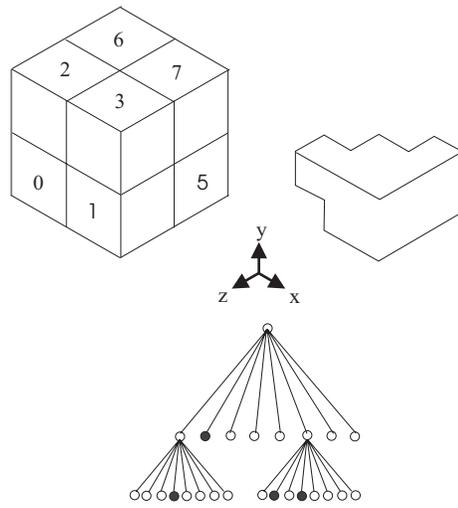


Figura 5.3: Representação através de octrees

baixa resolução. Por sua vez, as regiões correspondentes a fragmentos de superfícies foto-consistentes tendem a ser identificadas em estágios posteriores e representadas através de células de maior resolução.

É possível que outras estruturas espaciais sejam capazes de permitir um maior grau de adaptabilidade e concisão, como por exemplo as árvores baseadas em subdivisão binária (*bintrees*) e *árvores k-d*. Todavia, deixaremos a investigação da representação do espaço de reconstrução através de tais estruturas para trabalhos futuros.

Veremos agora como estas estruturas se encaixam no esquema de refinamento adaptativo proposto e quais as suas implicações sobre o método.

5.4

Escultura de cenas em espaços representados através de octrees

Os primeiros métodos de reconstrução volumétrica a utilizar octrees para representar os espaços de reconstrução foram os métodos de reconstrução a partir de silhuetas. Citamos como exemplos os métodos propostos nos trabalhos de Chien[18], Potmesil [23], Veenstra [20], Srivastava[27], e principalmente Szeliski [29].

Neste trabalho, de forma similar aos trabalhos mencionados acima, apresentamos um método de reconstrução a partir de imagens que trabalha sobre espaços volumétricos representados através de octrees, o qual, no entanto, utiliza para reconstruir a cena de interesse, tanto informações

fotométricas, quanto informações provenientes da segmentação do objeto de interesse nas imagens (Figura 5.4).

É bastante comum que os algoritmos desenvolvidos para solucionar um determinado problema sejam influenciados pela escolha da estrutura de dados utilizada para representar o espaço no qual tal problema é definido. Por esta razão, o método que propomos possui muitos pontos em comum com os esquemas utilizados para construção de representações através de octrees.

A idéia fundamental do método consiste em descrever inicialmente o espaço de reconstrução através de uma célula, correspondente à raiz de uma octree, cujo volume é especificado de maneira que englobe a cena que desejamos reconstruir. Em seguida, procuramos classificá-la segundo a sua foto-consistência em relação às imagens nas quais se encontra visível, o que nem sempre é possível, por motivos que descrevemos posteriormente. Se conseguirmos classificar a célula inicial, quer seja como foto-consistente, quer seja como não-foto-consistente, então podemos encerrar o processo de reconstrução, bastando apenas atribuir as informações fotométricas apropriadas à célula em questão. Por outro lado, quando não for possível classificá-la com precisão, o que ocorre com grande probabilidade no início do processo, precisamos subdividi-la uniformemente em oito novas células as quais devem sofrer um novo processo de classificação. Este procedimento é então repetido sucessivamente até que toda a região do espaço seja classificada adequadamente, conforme sua foto-consistência em relação às imagens de entradas, ou até que o nível máximo de subdivisão da árvore tenha sido atingido.

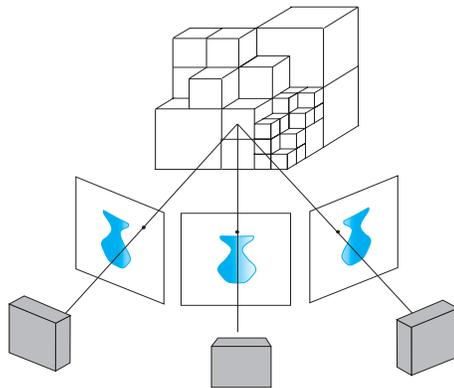


Figura 5.4: Coloração em uma octree

A aplicação deste método de reconstrução requer o tratamento das seguintes questões que serão abordadas nas seções seguintes.

- Como classificar uma célula com base em sua foto-consistência.
- Como efetuar a classificação do espaço de reconstrução sistematicamente.
- Como atribuir as informações fotométricas às células classificadas.

5.4.1

Como classificar uma célula da octree

Como vimos no capítulo 2, os métodos de escultura do espaço tradicionais procuram classificar o conjunto de voxels representando uma sub-região de interesse do espaço 3D em duas categorias: *voxels foto-consistentes* e *voxels não-foto-consistentes*. Analogamente, o método de escultura do espaço por refinamento adaptativo que propomos deve ser capaz de classificar as células da octree em células foto-consistentes e células não-foto-consistentes.

O grande desafio que devemos enfrentar, ao propor um método de reconstrução que trabalha sobre um espaço representado através de uma octree, é o de como avaliar a foto-consistência nas regiões do espaço correspondentes a cada célula sem precisar avaliar um conjunto muito grande de pontos em seu interior.

Antes de tomar qualquer decisão, precisamos primeiramente analisar se as classes de foto-consistência utilizadas nos métodos tradicionais são capazes de representar todas as circunstâncias que surgem em um processo de escultura do espaço por refinamento.

Devemos nos lembrar que, no caso geral, a detecção de não-foto-consistência em um pequeno conjunto de amostras no interior de uma região extensa não nos permite afirmar com certeza absoluta que toda a região é inconsistente, pois existe a possibilidade de que haja um outro conjunto de amostras capaz de satisfazer as restrições determinadas pelas imagens nas quais elas se encontram visíveis. Além disso, a região a ser avaliada pode conter uma combinação de partes que correspondem a regiões vazias com partes que correspondem a regiões do objeto, tornando ainda mais difícil o processo de classificação.

Ambas as questões não eram tão relevantes quando utilizávamos uma representação do espaço da cena através de voxels de dimensões desprezíveis, pois nestes casos a teoria sobre foto-consistência se comporta de maneira adequada.

Como descrevemos anteriormente, apenas Kutulakos propôs um critério, baseado nas *shuffle transforms*, capaz de classificar regiões extensas

apropriadamente. Por outro lado, o critério por ele utilizado é extremamente conservativo, uma vez que requer que a projeção de uma região do espaço nas imagens de entrada não contenha cores similares em uma vizinhança em torno de seus centros. Além disso, a determinação do tamanho adequado para as regiões de vizinhança, dentro do contexto de métodos por refinamento adaptativo, é um problema que ainda se encontra totalmente em aberto.

Ao invés de buscarmos critérios complexos para determinação da foto-consistência em regiões não pontuais, propomos uma abordagem mais simples, a qual, no entanto, é capaz de manter o aspecto conservativo dos métodos de escultura do espaço. Simplesmente criamos um novo grupo de classificação para as células da octree: o grupo das células *indefinidas*, as quais contemplam aquelas que não podem ser classificadas nem como foto-consistentes nem como não-foto-consistentes.

Conseqüentemente, as células da octree passam a ser classificadas da seguinte forma:

- células foto-consistentes.
- células não-foto-consistentes.
- células indefinidas.

Em nosso esquema de classificação, o conceito de célula foto-consistente é análogo ao conceito de foto-consistência existente na teoria desenvolvida por Seitz e Kutulakos. Por outro lado, restringimos o conceito de célula não-foto-consistente apenas às células de mais alta resolução que falharam no teste de foto-consistência ou então às células, neste caso de qualquer nível, que se projetam completamente em regiões de fundo.

As células indefinidas correspondem às células cuja foto-consistência foi invalidada pela análise, mas que de certa forma, não podemos afirmar com certeza absoluta que não há em seu interior nenhum fragmento de superfície capaz de satisfazer as restrições impostas pelas imagens. Isto pode ocorrer como conseqüência dos seguintes eventos: primeiro, quando a célula contém uma combinação de pontos que pertencem ao espaço vazio com pontos que pertencem ao objeto; segundo, quando o critério de foto-consistência não pode afirmar com precisão se a célula é foto-consistente ou não. O último caso acontece quando os elementos utilizados no procedimento de avaliação da foto-consistência não são suficientemente representativos ou então, quando o valor de foto-consistência obtido é muito próximo ao limiar de separação das classes.

Quando uma célula indefinida é detectada, devemos refiná-la, isto é, subdividi-la em oito novas células de forma que estes novos elementos gerados possam ser processados novamente em uma nova iteração do algoritmo de escultura do espaço em um nível de refinamento mais elevado.

Através desta estratégia não precisamos nos preocupar em utilizar um conjunto de amostras representativas para a célula que desejamos classificar, já que nos casos de não-foto-consistência, com exceção do caso em que a célula corresponde a uma região vazia, sempre efetuamos uma subdivisão da mesma até que a amostragem utilizada seja representativa o suficiente para que possamos efetuar uma classificação fidedigna. Em nosso método, a representatividade só é garantida quando chegamos ao nível de resolução máximo.

Por esta razão, podemos classificar as células da octree da mesma forma que classificamos voxels no método não adaptativo, isto é, através da avaliação da foto-consistência de um conjunto de amostras em uma região planar determinada pela interseção entre as células com um plano passando pelo seus centros e com a mesma orientação da direção de varredura.

5.4.2

Classificação do espaço de reconstrução

Como vimos anteriormente, a estratégia baseada em varredura de planos no espaço da cena é uma das mais eficientes para determinar a foto-consistência em um conjunto de regiões planares com uma mesma orientação. No entanto, para que possamos adotar esta mesma solução, para o caso de regiões planares associadas às células de uma octree, precisamos levar em consideração os seguintes problemas: primeiro, as células são geradas progressivamente, o que nos impede de utilizar um único passo do algoritmo de varredura de planos; segundo, o conceito de camadas, fundamental para o algoritmo de varredura de planos, não é muito bem definido em uma octree, uma vez que as células tem tamanho diferentes e se distribuem pelos diferentes níveis da estrutura (Figura 5.5).

Ambos os problemas podem ser resolvidos de maneira simples através da aplicação de um passo do algoritmo de varredura de planos para cada nível de refinamento. Logo, diferentemente dos métodos tradicionalmente utilizados para obtenção de representações através de octrees, não exploramos a estrutura em profundidade, mas sim em largura, o que significa que todos os nós de um determinado nível são processados antes de processarmos os nós do nível de resolução seguinte.

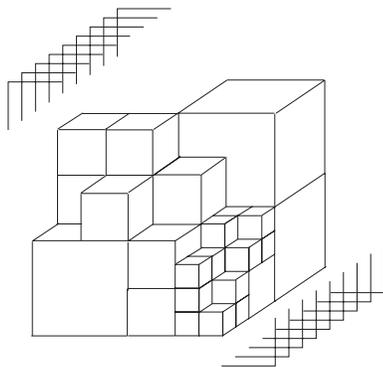


Figura 5.5: Varredura de planos em uma octree

O número de planos varridos em uma determinada aplicação do algoritmo de varredura de planos é igual ao número de camadas definidas na estrutura utilizada para representar o espaço de reconstrução. Como não é possível identificar camadas em uma octree, a não ser em casos muito particulares, determinamos o número de planos varridos através do número de camadas existente em uma partição regular com resolução igual à resolução do nível corrente na octree. Por exemplo, o número de planos no nível zero da octree é igual a um, pois existe uma única camada em uma partição regular equivalente; no nível 1, o número planos é igual a dois, já que existem duas camadas na partição regular equivalente, e assim por diante (Figura 5.6). Desta forma, o número de planos em uma aplicação do algoritmo de varredura de planos depende do nível de resolução corrente do processo, sendo então proporcional a 2^κ , onde κ é o nível da octree.

Como nas octrees não existem estruturas equivalentes às camadas, como existem nas representações baseadas em subdivisão regular, precisamos especificar um mecanismo para determinar quais são as células ativas durante uma iteração correspondente a determinação da foto-consistência em um determinado plano de referência em um certo nível de resolução. A solução encontrada também é bastante simples e consiste em percorrer a árvore em profundidade, determinando quais são as células que são interceptadas pelo plano de referência corrente que ainda não foram avaliadas (obviamente, isto exclui as células que já foram checadas nos níveis anteriores)(Figura 5.7).

Assim como nos métodos de escultura do espaço não adaptativos, a realização do algoritmo de varredura de planos requer a definição das seguintes etapas: registro das informações no espaço da cena, tratamento da visibilidade, determinação da foto-consistência e colorização dos elementos estruturais. A maioria destes procedimentos requer algumas modificações e

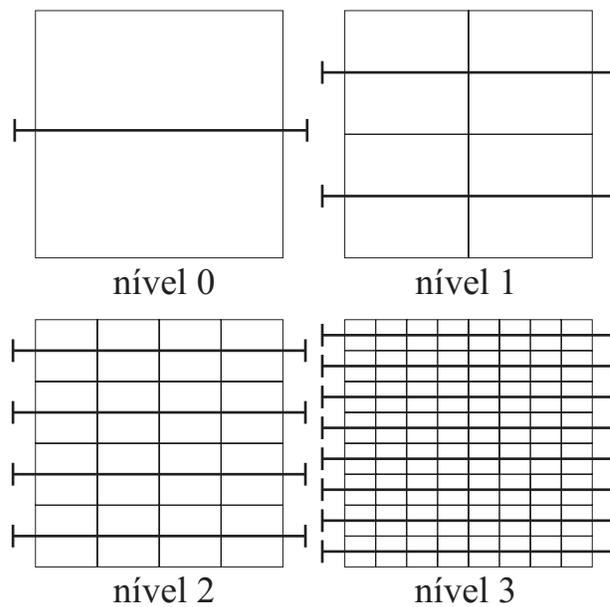


Figura 5.6: Número de planos em cada nível de resolução

adaptações para que possamos adotá-los no método de escultura do espaço adaptativo, as quais serão vistas em seguida.

Registro

O registro das informações no espaço da cena pode ser realizado de modo similar ao que é realizado no método de escultura do espaço não adaptativo, isto é, através do mapeamento projetivo das imagens de entrada sobre os planos de referência utilizados em cada iteração do algoritmo de varredura de planos em cada nível de refinamento. No entanto, uma atenção especial deve ser tomada em relação à resolução com a qual cada uma das imagens projetadas devem ser geradas.

As resoluções das imagens projetadas dependem do número de amostras utilizadas no processo de avaliação de foto-consistência, as quais por sua vez estão diretamente relacionadas ao número de células cortadas pelo plano de referência correspondente. Como o número de células varia de acordo com o nível de resolução com o qual trabalhamos, então, a resolução de uma imagem projetada deve ser proporcional ao nível de refinamento sobre a qual trabalha a etapa de varredura de planos corrente.

Obviamente, podemos gerar todas as imagens projetadas com a resolução máxima. No entanto, isto não parece sensato pois implica na diminuição do número de amostras para cada célula, à medida em que a resolução da representação do espaço aumenta. Uma estratégia mais

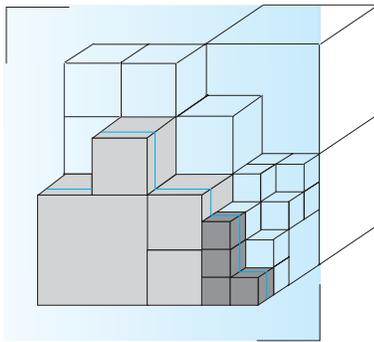


Figura 5.7: Células interceptadas por um plano de varredura. As células em tom escuro são as células ativas, as em tom mais claro são células já avaliadas.

razoável consiste em gerar imagens projetadas com um número de amostras que se mantém constante (ou que aumente), em cada célula, à medida em que aumenta o nível de refinamento da representação do espaço de reconstrução.

Uma justificativa para a não utilização de imagens com máxima resolução, nas etapas iniciais do processo de varredura de planos, é a de que nestes estágios dificilmente haverá correlação das imagens de entrada em uma dada região planar, implicando em um desperdício de esforço computacional causado pela utilização de um número excessivo de amostras. Por este motivo preferimos manter um número de amostras constante para cada célula, em todos os níveis de resolução, o que significa que teremos uma amostragem menos densa no início do processo e mais densa nos estágios de maior refinamento. Um valor razoável, por exemplo, é o de quatro amostras para cada célula (Figura 5.8).

A geração de imagens nas diferentes resoluções pode ser realizada de maneira simples através da variação do tamanho da janela (*viewport*) associada ao canvas (*framebuffer*) no qual as imagens projetadas são produzidas. Como as imagens projetadas são geradas com tamanho inferior ao das imagens originais, com exceção das geradas no último nível de refinamento, é fundamental a utilização do processo de mipmapping durante o mapeamento projetivo, já que tal mapeamento se caracteriza com um mapeamento de redução.

O fato de produzirmos as imagens em resolução proporcional ao nível de refinamento da representação do espaço de reconstrução tem implicações fundamentais sobre a etapa de determinação da foto-consistência das células, principalmente em relação à escolha do limiar de classificação, uma vez que neste caso, as amostras sofrem alterações consideráveis em

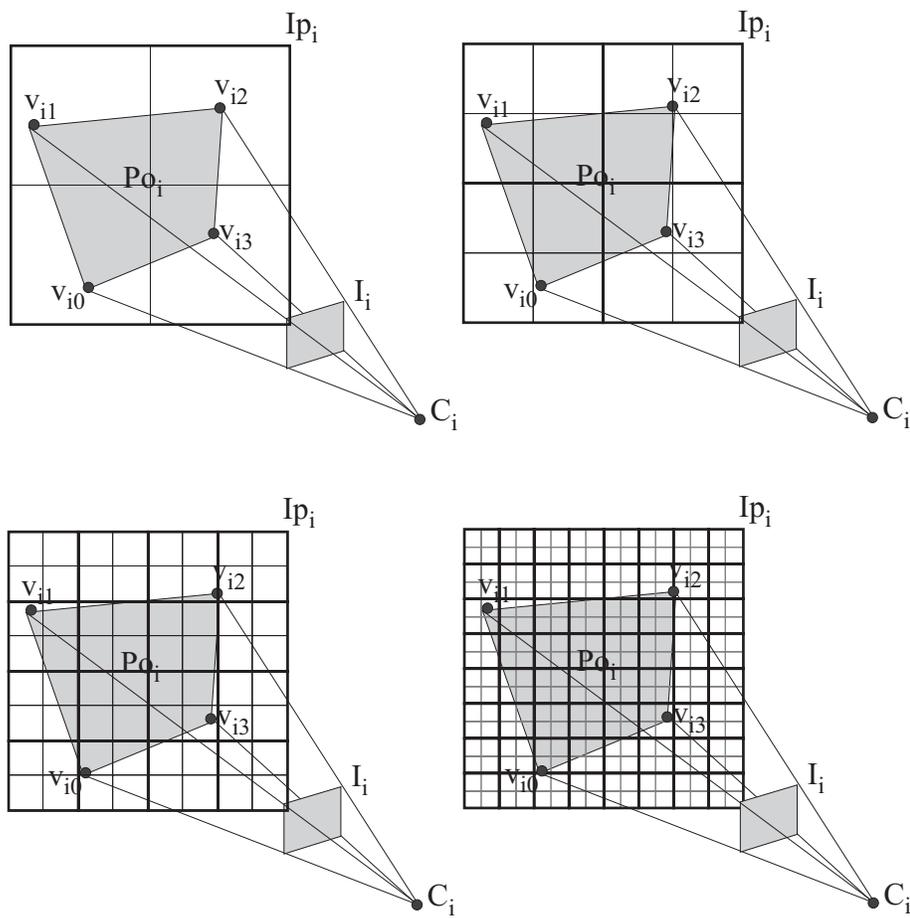


Figura 5.8: Imagens projetadas em diferentes níveis de resolução

função do processo de filtragem. Iremos investigar este aspecto com mais detalhes na seção que aborda a determinação de foto-consistência.

Uso das informações sobre a segmentação dos objetos de interesse nas imagens

A utilização das informações sobre a segmentação dos objetos de interesse nas imagens de entrada é fundamental no método de escultura do espaço adaptativo que propomos. Em nosso método, somente através de uma análise do espaço de reconstrução, com base nas informações sobre segmentação, é que podemos detectar e remover grandes regiões vazias existentes no espaço da cena, visto que a análise de foto-consistência, por si somente, não nos dá garantia sobre a não-foto-consistência de uma determinada região, a menos que estejamos no estágio de refinamento máximo.

Uma outra função importante das informações sobre segmentação é a

de que elas funcionam como um dos mecanismos reguladores do processo de refinamento. Graças à reamostragem das informações de segmentação através de mipmapping, podemos garantir que o espaço é refinado sempre que necessário, independentemente do grau de refinamento inicial do espaço de reconstrução, diferentemente do que ocorre no método de Prock.

Por exemplo, vejamos o que ocorre quando representamos o espaço de reconstrução inicial através de uma única célula englobando toda a cena. Neste caso, os mapas de segmentação serão projetados sobre o plano de referência correspondente após terem sido filtrados e subamostrados. Logo, os valores associados à classificação dos elementos na região planar correspondente, em função dos mapas de segmentação, não serão homogêneos, o que levará a determinação de tal célula como indefinida. Em outras palavras, isto significa que foi detectada, na célula, a presença de elementos do fundo da cena mesclados com elementos pertencentes aos objetos de interesse. Conseqüentemente, em função desta classificação tal célula será obrigatoriamente subdivida, o que fará com que o processo de escultura do espaço tenha prosseguimento.

Tratamento da visibilidade

O esquema adotado para a determinação da visibilidade das células da octree é similar ao esquema adotado no algoritmo de escultura do espaço não adaptativo no espaço da cena. Isto significa que precisamos definir mapas de visibilidade, através de mapas de textura, para cada uma das câmeras, os quais são registrados sobre os planos de referência à medida em que as iterações do algoritmo de varredura do espaço são realizadas.

A única diferença importante, que deve ser ressaltada, é a de que precisamos reconstruir os mapas de visibilidade a cada nova etapa do algoritmo de varredura de planos correspondente a cada novo nível de refinamento. Isto é necessário porque as informações no mapa de visibilidade não são mais válidas quando mudamos de um nível para o outro.

Os mapas de visibilidade, definidos por mapas de bits que são atualizados de forma sincronizada com o processo de varredura, apenas refletem a visibilidade dos voxels no plano de referência corrente. Quando mudamos de um nível para o outro, as informações de visibilidade não mais são válidas pois apenas refletem a visibilidade do último plano de varredura do nível de resolução anterior. Como neste novo nível podemos encontrar novos voxels consistentes que se posicionam a frente de tal plano, isto é, do último plano de referência do nível anterior, então não podemos nos guiar por tais

informações. Portanto, precisamos limpar os mapas de visibilidade antes da execução do algoritmo de varredura de planos em um novo nível de refinamento.

Uma consequência imediata deste fato é a de que os mapas de visibilidade são reconstruídos a cada nova execução do algoritmo de varredura de planos em nível de resolução mais refinado. Por esta razão, na etapa de atualização dos mapas de visibilidade, que ocorre após a determinação da foto-consistência das células ativas interceptadas por um determinado plano de referência, precisamos considerar não somente as novas células foto-consistentes, mas também todas as células foto-consistentes que interceptam o plano de referência corrente que foram determinadas nos níveis de refinamento anteriores.

Determinação da foto-consistência

O procedimento utilizado para determinar a foto-consistência das células de uma octree é similar aos procedimentos vistos nos capítulos 4 e 5, utilizados no método de escultura do espaço não adaptativo para determinar a foto-consistência de um conjunto de voxels.

Entretanto, existe uma diferença fundamental causada pelo fato de que, no método de escultura do espaço por refinamento adaptativo, as imagens projetadas são geradas através de um processo de reamostragem baseado em *mipmapping*, o que significa que cada um de seus elementos é, por si próprio, uma combinação de elementos do nível de resolução mais alto da representação. Mais importante ainda é o fato de que o número de elementos em tal combinação depende diretamente da resolução com a qual as imagens são geradas. Obviamente isto tem um impacto direto sobre a forma como a foto-consistência deve ser calculada, pois cada amostra, tomada para uma determinada célula, neste caso, é uma combinação de elementos.

Para que possamos determinar como a foto-consistência deve ser realizada de maneira que leve em consideração este fato, precisamos primeiramente compreender qual é o efeito da operação de filtragem por *mipmapping* sobre a variância das amostras tomadas para um dado elemento, já que é através dela que estimamos o quão próximo suas médias se encontram, o que por sua vez nos permite afirmar através de um teste de comparação com um limiar pré-estabelecido, se as observações são foto-consistentes ou não. Vimos anteriormente que, a grosso modo, a operação de filtragem efetua uma média sobre um conjunto de elementos de textura em uma vizinhança

determinada de modo aproximado, pela região correspondente ao mapeamento de um pixel no espaço da textura. Intuitivamente, tal operação, causa uma redução no desvio das amostras observadas. Logo, para que possamos compensar esta redução é necessário reduzir o limiar que caracteriza a foto-consistência de maneira proporcional à redução observada.

Para ajustar o limiar nos estágios em que trabalhamos com imagens em resolução mais baixa, usamos o fato de que, se X_1, X_2, \dots, X_n são independentes, com mesma variância σ^2 , sua média \bar{X} tem variância igual a σ^2/n . Em nosso caso, consideramos que cada amostra em um dado nível de resolução seja a média de quatro amostras no nível superior. Isto significa que em um dado nível, cada amostra é uma média de $4^{(nimax-ni)}$ amostras, onde $nimax$ é o nível máximo de resolução e ni é o nível corrente. Portanto, a variância calculada para um conjunto de amostras representadas por X_1, X_2, \dots, X_n , em um dado nível ni de fato deve ser igual a $\sigma^2/4^{(nimax-ni)}$, onde σ é o desvio calculado com base nos valores observados.

Por exemplo, no teste utilizado no capítulo 4, o limiar em um dado nível de refinamento ni deve ser proporcional ao desvio padrão correspondente àquele nível. Logo deve ser igual a $limiar/2^{(nimax-ni)}$, onde $limiar$ é o limiar no nível mais alto de resolução.

Na verdade, em nosso caso, não podemos assumir, a princípio, que os elementos que compõem uma determinada amostra são independentes, e de fato não são, já que a convolução, realizada durante a reamostragem, faz com que cada uma delas receba contribuições de amostras vizinhas. Logo, devemos estar conscientes de que tal ajuste é apenas uma aproximação.

Atribuição de cores às células da octree

Esta é uma das questões mais complexas no método de escultura do espaço por refinamento adaptativo, e que precisa de um estudo mais profundo para ser tratada de forma apropriada. Em nossa implementação simplesmente colorimos as células da mesma forma que colorimos voxels de dimensões pequenas, isto é, através de uma média das cores das amostras tomadas para cada elemento pertencente à região planar associada. Entretanto, sabemos que esta solução pode produzir resultados bastante insatisfatórios quando aplicada a alguns casos particulares, como, por exemplo, quando poucas câmeras enxergam uma região com geometria quase planar que, no entanto, possui informações de cor de grande variabilidade. Nestes casos, há uma probabilidade razoável de que tal região seja determinada como foto-consistente nos níveis de resolução mais baixos o que levaria a

uma atribuição de cores inadequada, já que a média de uma região de grande variabilidade não seria representativa o suficiente. Apesar deste problema, na prática os resultados são bastante satisfatórios, e o que é mais importante: tal comportamento não é simples consequência do acaso e sim da própria estrutura intrínseca do método proposto.

Dois fatores são cruciais para a obtenção de resultados corretos no processo de colorização utilizado: o emprego da estratégia por refinamento adaptativo, combinada com o uso de informações de segmentação em resoluções incrementais, e o uso de limiares que se adequam aos diferentes níveis de refinamento no teste de checagem da foto-consistência dos elementos do espaço de reconstrução.

O primeiro fator automaticamente favorece a subdivisão dos elementos próximos à superfície do objeto durante o processamento nos estágios de refinamento mais grosseiros. Isto se deve ao fato de que, nestes estágios, os valores provenientes dos mapas de segmentação projetados, observados em tais elementos, são determinados por uma combinação (causada pela operação de filtragem) de valores que correspondem ao fundo e aos objetos de interesse. Logo, é esperado que nos estágios mais grosseiros tais regiões sejam consideradas duvidosas, o que fará com que sejam obrigatoriamente subdividas. Este aspecto justifica tanto o uso de informações de segmentação quanto o emprego de representações em multiresolução das informações utilizadas na avaliação de foto-consistência.

O segundo fator dificulta a classificação de células de baixa resolução como foto-consistentes, pois os limiares são bastante baixos no início do processo, compensando desta forma a diminuição dos desvios causadas pela operação de filtragem. Logo, células de baixa resolução somente são classificadas como foto-consistentes se houver uma correlação muito grande entre as informações registradas, o que dificilmente acontecerá a menos que a variabilidade da região seja muito pequena e a superfície da cena coincida com o plano de varredura.

Obviamente, o procedimento que utilizamos pode falhar em alguns casos patológicos. Por este motivo, seria melhor encontrarmos uma solução que cobrisse todos os casos possíveis. Infelizmente, a validação da determinação de uma célula foto-consistente de baixa resolução envolve o conhecimento sobre a variabilidade nas regiões das imagens registradas na resolução original, o que tornaria o processo extremamente caro se fôssemos obrigados a projetar as imagens sempre na resolução máxima. Uma possível estratégia seria calcular as informações de variabilidade no espaço das imagens e armazená-las através de estruturas compactas como quadrees, que pudessem

ser recuperadas eficientemente. O processo resultante seria dado então por uma combinação de processamento no espaço da cena com processamento no espaço da imagem. Isto é um tema que teremos que investigar futuramente. Abaixo descrevemos o algoritmo para avaliação da foto-consistência das células nos diferentes níveis de refinamento de uma octree (Algoritmo 7) e em seguida apresentamos o algoritmo de escultura do espaço por refinamento adaptativo.

Algoritmo 7 AvaliaFotoconsistênciaII(*limiar, nível, rp, Ip, Mvp*)

Determine o conjunto $E_v = \{ev_{st} \in rp \mid \exists n, Mvp_{nst} = 1\}$

se $E_v = \emptyset$ **então**

retorne CONSISTENTE

fim se

para cada $ev_{st} \in E_v$ **faça**

se $\exists am_{st}^n$ de ev_{st} tal que $Ip_{nst.a} \leq tolfund$ e $nível < NÍVELMÁXIMO$

então

retorne INDEFINIDA

senão

retorne INCONSISTENTE

fim se

$$dp_{st} = \frac{1}{n-1} \sum_{l=1}^n (am_{st}^l - \overline{am}_{st})^2$$

fim para

$$dpm = \frac{1}{|E_v|} \sum_{s,t} (dp_{st})$$

se $dpm < limiar$ **então**

retorne CONSISTENTE

senão se $nível < NÍVELMÁXIMO$ **então**

retorne INDEFINIDA

senão

retorne INCONSISTENTE

fim se

5.4.3 Algoritmo

Algoritmo 8 Escultura do espaço por refinamento adaptativo

$nível \leftarrow octree$

Inicie $nó.célula$ com o volume de reconstrução inicial.

repita

$nível \leftarrow 0$

Limpe os mapas de visibilidade.

para cada plano de varredura π_k , $k = 2^{nível}..0$ **faça**

Determine o conjunto de imagens projetadas I_p na resolução apropriada.

Determine o conjunto de mapas de visibilidade projetados Mvp na resolução apropriada.

para cada célula c na octree corrente tal que $(c \cap \pi_k \neq \emptyset)$ e $(c.fconsist = \text{NÃOAVALIADA})$ **faça**

$rp \leftarrow c \cap \pi_k$

$limiar \leftarrow limiarmáximo / 2^{(NÍVELMÁXIMO - nível)}$

$c.fconsist \leftarrow \text{AvaliaFotoconsistênciaII}(limiar, nível, rp, I_p, Mvp)$

se $c.fconsist = \text{CONSISTENTE}$ **então**

Atribua as cores apropriadas à c .

senão se $c.fconsist = \text{INDEFINIDA}$ **então**

Crie oito nós filhos de $nó$.

Subdivide c em 8 novas células.

Rotule cada nova célula como NÃOAVALIADA .

Atribua as novas células aos respectivos filhos de $nó$.

fim se

fim para

Atualize os mapas de visibilidade.

fim para

$nível \leftarrow nível + 1$

até que que nenhuma célula tenha sido subdividida ou $nível < \text{NÍVELMÁXIMO}$

5.4.4 Observação

É possível tornar o algoritmo acima ainda mais eficiente se o aplicarmos sobre uma estimativa do espaço de reconstrução dada pelo fecho visual representado através de uma octree. Como o fecho visual é utilizado apenas como estimativa, não precisamos determiná-lo através de processamento no espaço da cena, bastando calculá-lo diretamente a partir das imagens de entrada. Um algoritmo bastante eficiente que retorna o fecho visual representado através de uma octree é o algoritmo de Szelisk [29]. Na verdade, em

nossa implementação, adotamos esta estratégia, isto é calculamos o fecho visual no espaço das imagens, através de um algoritmo semelhante ao de Szeliski e então, sobre tal estimativa, aplicamos o algoritmo de escultura do espaço por refinamento adaptativo, o que nos retorna o fecho fotométrico ou *Photo Hull*.

6

Implementação do sistema

Como resultado dos estudos deste trabalho, implementamos um protótipo de um sistema de reconstrução 3D com base nos métodos de escultura do espaço. Nosso sistema é bastante simples e de baixo custo, consistindo de um computador Pentium III, munido de uma placa GeForce II e um conjunto de quatro webcams Logitech conectadas por um hub USB, colocadas nos vértices superiores de um cubo de 1x1x1m(Figura 6.1).



Figura 6.1: Imagem do sistema

O processo de reconstrução envolve cinco etapas fundamentais: a calibração das câmeras, que pode ser efetuada uma única vez, desde que a posição e a orientação das mesmas não sejam alteradas; a calibração do fundo que tem como o objetivo permitir a extração das regiões delimitadas pelas silhuetas do objeto; a captura das imagens e a segmentação dos objetos de interesse; e finalmente, o processo de reconstrução propriamente dito, através dos métodos de escultura do espaço aqui apresentados (Figura 6.2).

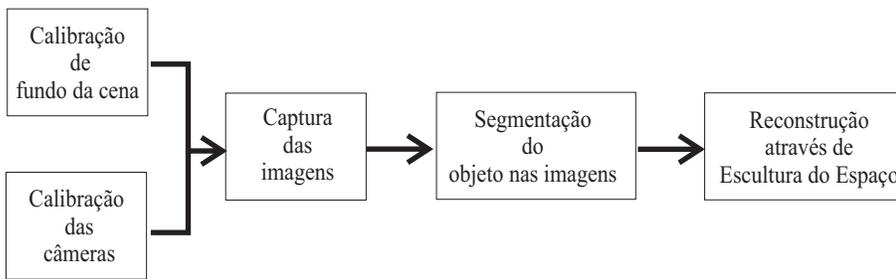


Figura 6.2: Processo de reconstrução

Um dos maiores desafios na construção deste sistema foi lidar com os problemas causados pelo uso de webcams, as quais possuem resolução muito baixa, além de introduzirem muito ruído nas imagens capturadas.

Vejam como foram implementados cada uma das etapas associadas ao processo de reconstrução no sistema construído.

6.1 Calibração das câmeras

O método utilizado para calibrar as câmeras é baseado no processo de calibração automático proposto por Szenberg em sua tese de doutoramento [75]. O método de Szenberg é capaz de calibrar uma câmera automaticamente através do reconhecimento de um modelo descrito por um conjunto de segmentos de reta. Em nosso caso utilizamos o padrão mostrado na figura 6.3.

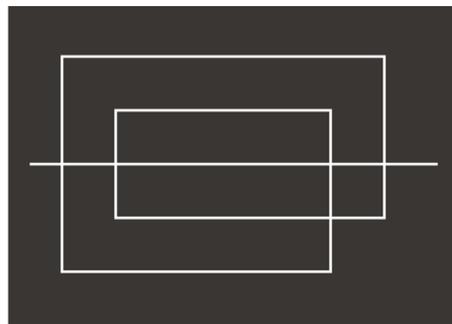


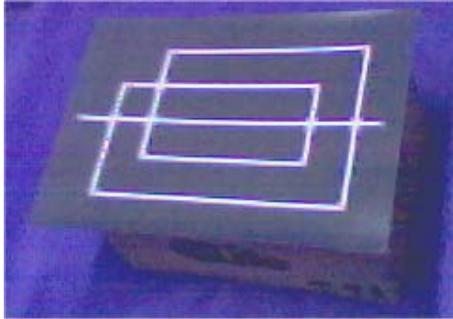
Figura 6.3: Padrão de calibração

Primeiramente, a imagem contendo o modelo é filtrada, através de um filtro *laplaciano da gaussiana*, com o objetivo de realçar os pixels pertencentes aos segmentos de reta de interesse. Em seguida, uma etapa de busca é efetuada com o objetivo de detectar segmentos de retas longos, a partir de pequenos segmentos de reta com mesma orientação, os quais são analisados

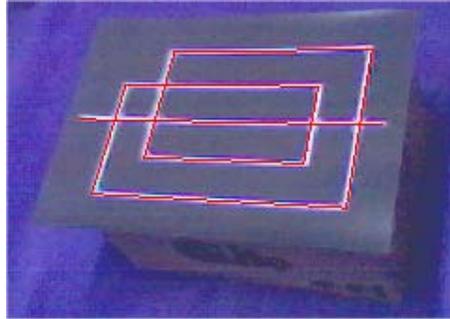
com base em uma árvore de interpretações. O resultado desta análise retorna os segmentos reconhecidos, isto é, aqueles que satisfazem as restrições utilizadas na caracterização do modelo. Posteriormente, é calculada uma transformação projetiva que mapeia os segmentos do modelo nos segmentos reconhecidos, através da qual pode-se detectar novos segmentos ou reajustar os antigos, tanto na própria imagem usada na calibração quanto em uma nova imagem obtida a partir de uma ligeira modificação na câmera. Finalmente, com base em um conjunto de pontos de calibração, obtidos a partir da interseção dos segmentos reconhecidos e ajustados, alimentamos o método clássico de calibração de câmeras de Tsai [22], que nos retorna então os parâmetros extrínsecos e intrínsecos desejados.

Uma das características mais notáveis do método de Szenberg é sua capacidade de ajustar a calibração de uma câmera que teve seus parâmetros ligeiramente alterados como por exemplo em uma seqüência de imagens de vídeo, bastando para isso, efetuar somente um passo de reajuste dos segmentos reconhecidos e uma nova iteração do método de Tsai. Em nosso caso não precisamos desta funcionalidade, porém ela pode ser utilizada com bastante sucesso em um método de reconstrução que envolva reposicionamento de câmeras.

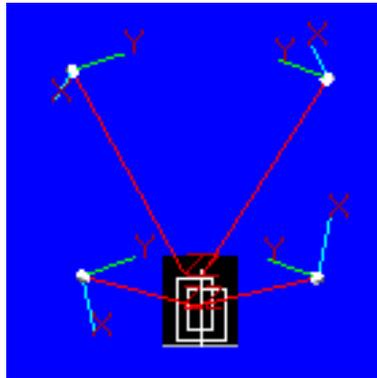
O método proposto por Szenberg também tem como vantagens a sua extrema robustez. Antes de sua utilização, experimentamos o método de calibração da biblioteca OpenCV [5], o qual não foi capaz de nos fornecer resultados satisfatórios. O nível de ruído presente nas imagens tornava impossível a detecção de todos os pontos de calibração necessários, inviabilizando assim o restante do processo. Quando, por outro lado, começamos a adotar o método de Szenberg, não mais precisamos nos preocupar com este problema já que os pontos de calibração são gerados automaticamente a partir do reconhecimento do modelo, o qual pode ser feito mesmo quando apenas um subconjunto dos segmentos que o compõem são detectados.



6.4(a): Um dos padrões de calibração



6.4(b): Padrão calibrado



6.4(c): Imagem tridimensional do posicionamento das câmeras

Figura 6.4: Resultado da calibração das câmeras do sistema

6.2

Câmera Tsai x Câmera OpenGL

Em nosso método, é necessário determinar as matrizes de transformação de câmera e de projeção, usadas pela OpenGL, a partir dos parâmetros intrínsecos e extrínsecos produzidos pelo método de calibração de Tsai. Vejamos como isso pode ser feito.

6.2.1

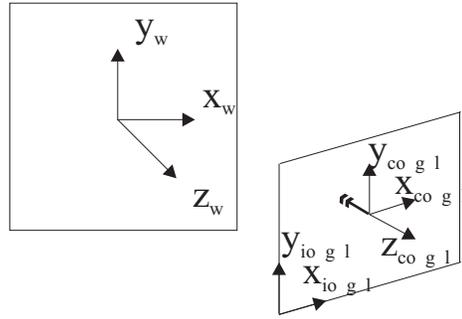
Determinando a matriz de transformação de câmera da OpenGL

Um cuidado especial deve ser tomado no cálculo das matrizes de transformação de câmera da OpenGL a partir dos parâmetros de rotação e translação fornecidos pelo método de Tsai. A relação entre a matriz de transformação de câmera da OpenGL, normalmente conhecida como *modelview matrix*, e os parâmetros extrínsecos determinados pelo método de Tsai, dependem dos sistemas de coordenadas nos quais estão definidas as coordenadas dos pontos de calibração no mundo e nas imagens.

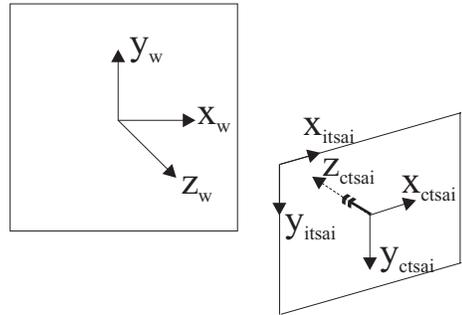
Tanto o sistema de coordenadas adotado pelo método de Tsai, quanto o adotado pela OpenGL, são dextrógiros. Por conseguinte, esperar-se-ia que os parâmetros de rotação e translação, obtidos a partir da calibração determinada pelo método de Tsai, pudessem automaticamente determinar uma *modelview* válida para o sistema de visualização da OpenGL. No entanto, a convenção adotada pela OpenGL é a de que a câmera aponta para o sentido negativo do eixo Z (Figura 6.5(a)), enquanto que, no sistema adotado por Tsai, considera-se que a câmera aponta para o sentido positivo do eixo Z . Além disso, o sistema de coordenadas da imagem, no método de Tsai é diferente do utilizado na OpenGL. Para que possa ser consistente com um sistema dextrógiro, o eixo Y no sistema de coordenadas da imagem adotada pelo método de Tsai aponta para baixo, ao invés de para cima, como na OpenGL (Figura 6.5(b)).

Por causa destas diferenças, os parâmetros de rotação r_1, r_2, \dots, r_9 e de translação t_x, t_y, t_z somente determinam uma matriz compatível com o sistema utilizado pela OpenGL, quando as coordenadas dos pontos de calibração nas imagens, forem obtidas de acordo com o sistema de coordenadas no espaço da imagem adotado pelo método de Tsai. Ainda assim, a matriz produzida a partir dos parâmetros calculados pelo método de Tsai não é exatamente uma matriz de transformação OpenGL, pois os eixos Y e Z encontram-se invertidos. A correção a ser efetuada é bastante

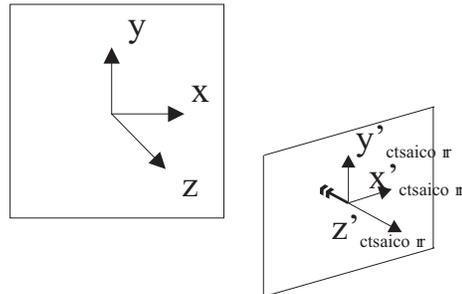
simples: basta inverter os eixos Y e Z (Figura 6.5(c)), operação esta que pode ser representada matricialmente através da expressão abaixo:



6.5(a): Sistema de coordenadas OpenGL



6.5(b): Sistema de coordenadas do método de Tsai



6.5(c): Sistema de coordenadas de Tsai corrigido

Figura 6.5: Sistemas de coordenadas utilizados na OpenGL e no método de Tsai

$$\begin{aligned}
 modelview &= M_{tsai}MC \\
 \begin{pmatrix} r_1 & -r_2 & -r_3 & t_x \\ r_4 & -r_5 & -r_6 & t_y \\ r_7 & -r_8 & -r_9 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} &= \begin{pmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 & \qquad \qquad \qquad (6.2.1)
 \end{aligned}$$

6.2.2

Determinando a matriz de projeção da OpenGL

A matriz de projeção da OpenGL pode ser montada manualmente com base nos parâmetros intrínsecos produzidos pela rotina de calibração. No entanto, como esta matriz é uma matriz normalizada, é bem mais simples utilizar as próprias rotinas da OpenGL dedicadas a sua especificação.

Em nossa implementação especificamos a matriz de projeção de forma direta, através da chamada da função *gluPerspective*, passando como argumentos os valores do *campo de visão*, da *razão de aspecto*, do *plano mais próximo* e do *plano mais distante*. O *campo de visão* e *razão de aspecto* são obtidos a partir das equações abaixo:

$$\begin{aligned} \text{campo de visão} &= \frac{2 * \arctan(\text{altura da imagem} * 1/2) * \pi}{\text{distância focal} * 180} \\ \text{razão de aspecto} &= \frac{\text{largura da imagem}}{\text{altura da imagem}} \end{aligned} \quad (6.2.2)$$

Já o *plano mais próximo* e o *plano mais distante* podem ser especificados arbitrariamente, desde que o frustum determinado contenha a cena desejada.

6.3

Calibração do fundo da cena

A calibração do fundo da cena é uma das etapas mais importantes para a realização do processo de reconstrução no sistema que propomos, pois é através dela que podemos segmentar o objeto de interesse nas imagens de entrada.

Para que possamos segmentar os objetos de interesse nas imagens, é necessário modelar o fundo da cena visto por cada uma das câmeras. Isto pode ser feito, para cada câmera, através da captura de imagens da cena sem os objetos que desejamos reconstruir. Esta abordagem, no entanto, não funciona, pois as intensidades de cada uma das componentes de cor dos pixels de uma determinada imagem são modificadas a cada instante de tempo por ruído aleatório, tornando impossível a utilização de um modelo determinístico. Por esta razão, precisamos modelar as imagens do fundo da cena estatisticamente.

O valor de uma dada componente de cor, digamos ρ em um certo pixel de uma imagem com ruído, pode ser modelado, apropriadamente, através

de uma variável aleatória com uma distribuição aproximada normal com uma média μ e variância σ^2 desconhecidas.

Como não conhecemos os parâmetros da distribuição de tal variável aleatória, precisamos estimá-los de alguma forma através de um conjunto de amostras. Em nosso caso, optamos por modelar cada uma das imagens do fundo da cena, através de matrizes onde cada elemento é um intervalo de confiança *ico* para a média μ em cada pixel. Cada um destes intervalos de confiança é calculado com base no erro entre uma estimativa pontual \bar{X} , dada pela média de um conjunto de n amostras tomadas em diferentes instantes de tempo, e a média populacional μ . Este erro, segundo o *Teorema Central do Limite*, tem sua distribuição aproximada dada por $e = (\bar{X} - \mu) \sim N(0, \sigma_{\bar{x}}^2)$ onde, $\sigma_{\bar{x}}^2 = \sigma^2/n$

Logo, a probabilidade de que tal erro e seja menor que uma determinada magnitude, dada por um múltiplo do desvio padrão amostral, é

$$P(|e| < \delta\sigma_{\bar{x}}) = P(|\bar{X} - \mu| < \delta\sigma_{\bar{x}}) = NC,$$

onde NC é o nível de confiança.

Reescrevendo esta afirmação probabilística como

$$P(\bar{X} - \delta\sigma_{\bar{x}} < \mu < \bar{X} + \delta\sigma_{\bar{x}}) = NC \quad (6.3.1)$$

podemos extrair o intervalo com o nível de confiança NC desejado

$$ico = \bar{X} \pm \delta\sigma_{\bar{x}} \quad (6.3.2)$$

Suponha agora que temos n componentes de cor para cada elemento de uma imagem, e que desejamos encontrar intervalos de confiança para cada componente ρ^k de maneira que a probabilidade de que todos eles contenham as respectivas médias populacionais seja igual a um determinado valor p . Neste caso, considerando que os erros são independentes temos que

$$\begin{aligned}
\prod_{k=1}^n P(|e_{\rho^k}| < \delta\sigma_{x_{\rho^k}}) &= \prod_{k=1}^n P(|\overline{X_{\rho^k}} - \mu_{\rho^k}| < \delta\sigma_{x_{\rho^k}}) & (6.3.3) \\
&= \prod_{k=1}^n P(\overline{X_{\rho^k}} - \delta\sigma_{x_{\rho^k}} < \mu_{\rho^k} < \overline{X_{\rho^k}} + \delta\sigma_{x_{\rho^k}}) \\
&= p
\end{aligned}$$

Se assumirmos que as probabilidades são as mesmas para cada componente de cor, então cada intervalo pode ser obtido de modo análogo com um nível de confiança $NC = 1/\sqrt{p}$.

No caso em que trabalhamos com as três componentes r , g e b , os intervalos de confiança que contêm simultaneamente as médias μ_r , μ_g e μ_b , em um dado pixel, com um nível de confiança de 95%, são dados respectivamente por:

$$\begin{aligned}
icor &= \overline{Xr} \pm 2,39\sigma_{\overline{xr}} \\
icog &= \overline{Xg} \pm 2,39\sigma_{\overline{xg}} \\
icob &= \overline{Xb} \pm 2,39\sigma_{\overline{xb}}
\end{aligned}$$

Devido ao ruído excessivo introduzido nas imagens obtidas através de webcams, é necessário obter um grande número de amostras temporais para que possamos obter estatísticas confiáveis para cada pixel. O ruído é notavelmente percebido em regiões de alta frequência, onde praticamente é impossível gerar informações com variabilidade bem comportada.

6.4

Segmentação com base em mapas de intervalos de confiança

A captura das imagens é realizada através de uma média temporal para que possamos eliminar parte do ruído aleatório introduzido pelo sistema de aquisição. Em nosso caso, utilizamos uma média de quatro quadros para cada imagem capturada.

Após a captura das imagens de entrada, precisamos segmentar os objetos de interesse com base nas matrizes de intervalos de confiança que modelam as imagens de fundo vistas por cada uma das câmeras.

A segmentação é bastante simples, consistindo em teste que verifica se os valores das componentes de cor de cada pixel da imagem capturada estão dentro do intervalo de confiança correspondente. Caso o teste seja bem

sucedido, isto é, se o valor da componente de cor de um pixel estiver dentro do intervalo de confiança desejado identificamo-lo como pixel de fundo, caso contrário, identificamo-lo com parte de uma região correspondente aos objetos de interesse nas imagens.

Quando aplicamos tal teste para segmentar as imagens, verificamos que as regiões de sombra e penumbra pertencentes ao fundo da cena acabavam sendo classificadas como parte dos objetos de interesse. Para evitar tais problemas criamos novas matrizes de intervalos de confiança, desta vez para componentes de cromaticidade, e adicionamos um novo teste ao processo de segmentação.

Desta forma, para cada pixel classificado como objeto de interesse, segundo o teste baseado em intervalos de confiança para as componentes de cor, verificamos as suas componentes de cromaticidade. Se estas componentes estiverem dentro dos respectivos intervalos de confiança, classificamos o pixel em questão como de fundo.

Após a classificação, aplicamos um filtro morfológico nos mapas de segmentação resultantes com o objetivo de remover pequenos elementos espúrios e fechar pequenos buracos no interior das regiões correspondentes aos objetos.

6.5 Reconstrução

A etapa de reconstrução é realizada através dos algoritmos de escultura do espaço que foram descritos neste trabalho. É necessário apenas a especificação da resolução desejada (resolução máxima no caso do algoritmo de escultura do espaço adaptativo) e a região do espaço que determina o volume de reconstrução inicial.

7

Resultados

7.1

Introdução

Neste capítulo, apresentamos e comparamos os resultados obtidos através da aplicação, na reconstrução de objetos sintéticos e reais, do método de escultura do espaço em suas três variações: (a) no espaço das imagens, (b) no espaço da cena e (c) por refinamento adaptativo.

Em primeiro lugar, apresentamos cada um dos dados utilizados nos testes comparativos, juntamente com as respectivas imagens de segmentação dos objetos de interesse. Em seguida, apresentamos imagens dos objetos reconstruídos por cada um dos métodos avaliados e efetuamos uma análise comparativa da qualidade das reconstruções. Além disso, analisamos os resultados produzidos pelos diferentes métodos em função da resolução do espaço de reconstrução. Mostramos também os resultados produzidos pelo método de escultura do espaço por refinamento adaptativo em cada um dos níveis de refinamento e finalmente, analisamos o desempenho de cada um dos métodos e apresentamos algumas estatísticas em forma de tabelas.

Para facilitar a exposição utilizaremos as seguintes notações para cada um dos métodos avaliados:

- **EEEI** - Escultura do espaço no espaço das imagens.
- **EEEC** - Escultura do espaço no espaço da cena.
- **EERA** - Escultura do espaço através de refinamento adaptativo.

7.1.1

Dados utilizados nos testes

Os testes foram realizados sobre dois conjuntos de dados sintéticos e dois conjuntos de dados reais. Os primeiros consistem de imagens geradas através da renderização de objetos poligonais através de câmeras sintéticas.

Tais dados, denominados respectivamente Al e Dolphins, são descritos no formato *Wavefront*, e foram obtidos diretamente da Internet. Os últimos consistem em imagens calibradas obtidas de objetos infláveis reais através do sistema de aquisição descrito no capítulo anterior. Utilizamos um tecido de cor roxa para cobrir todo o aparato de forma a facilitar a segmentação. O primeiro conjunto de dados reais denominamos Tigre e o segundo denominamos Dino. Para cada conjunto de dados apresentamos primeiramente suas imagens de entrada e seus respectivos mapas de segmentação.

O primeiro dado sintético consiste em uma forma humana estilizada que apresenta algumas estruturas bastante interessantes. Basicamente é uma forma suave mas que apresenta estruturas delicadas como as das mãos. Sua textura é simples já que consiste em regiões de cor homogênea. O segundo dado é formado por um grupo de três golfinhos. O ponto interessante deste dado é que ele é composto por três diferentes componentes conexas o que nos permite demonstrar o poder dos métodos de escultura do espaço. Uma outra característica é a presença de estruturas bastante finas como as caudas. O terceiro e o quarto dados são imagens de objetos plásticos infláveis reais que diferem principalmente na riqueza de textura.

7.1.2 Resultados da segmentação

Nos dados sintéticos a segmentação foi trivialmente obtida através de uma simples subtração do fundo preto da cena renderizada como ilustram as figuras 7.1, 7.2, 7.3 e 7.4. Nos casos reais a simples utilização de um fundo de cor diferente não foi suficiente para que pudéssemos obter uma boa segmentação já que as imagens do fundo, que visualmente parecem ser homogêneas, são extremamente ruidosas e além disso bastante reflexivas. Um outro problema também observado é que o fundo é levemente modificado no momento em que o objeto é colocado no aparato. Estes problemas nos levou a desenvolver o algoritmo de segmentação baseada em intervalos de confiança apresentado no capítulo 6.

Podemos observar, nas figuras 7.5 e 7.6 que os resultados produzidos pela segmentação dos dados reais são bastante satisfatórios se considerarmos a dificuldade intrínseca do problema. Os contornos dos objetos estão razoavelmente bem delineados e praticamente não se encontram pontos espúrios, com exceção de alguns pequenos elementos no dado Dino. Um outro ponto a ser notado é que foram criados pouquíssimos buracos no interior dos objetos, como aparecem por exemplo no dado Tigre, graças à

aplicação de filtros morfológicos sobre o resultado da classificação com base nos intervalos de confiança. Os pequenos buracos que ainda restaram foram gerados por erros de segmentação causados pela grande reflectividade existente na superfície branca do objeto que praticamente se torna da cor do fundo. Esta dificuldade deve ser esperada em objetos de plástico brilhante como os utilizados.

É interessante notar que em uma versão anterior do algoritmo de segmentação, em que o teste de cromaticidade não era feito, as sombras dos objetos projetadas no fundo eram erroneamente classificadas. Graças aos testes de cromaticidade realizados e a escolha de um fundo de cor distinta das cores presentes nos objetos de interesse a segmentação foi realizada com sucesso.



Figura 7.1: Dado I - AI - Imagens de entrada

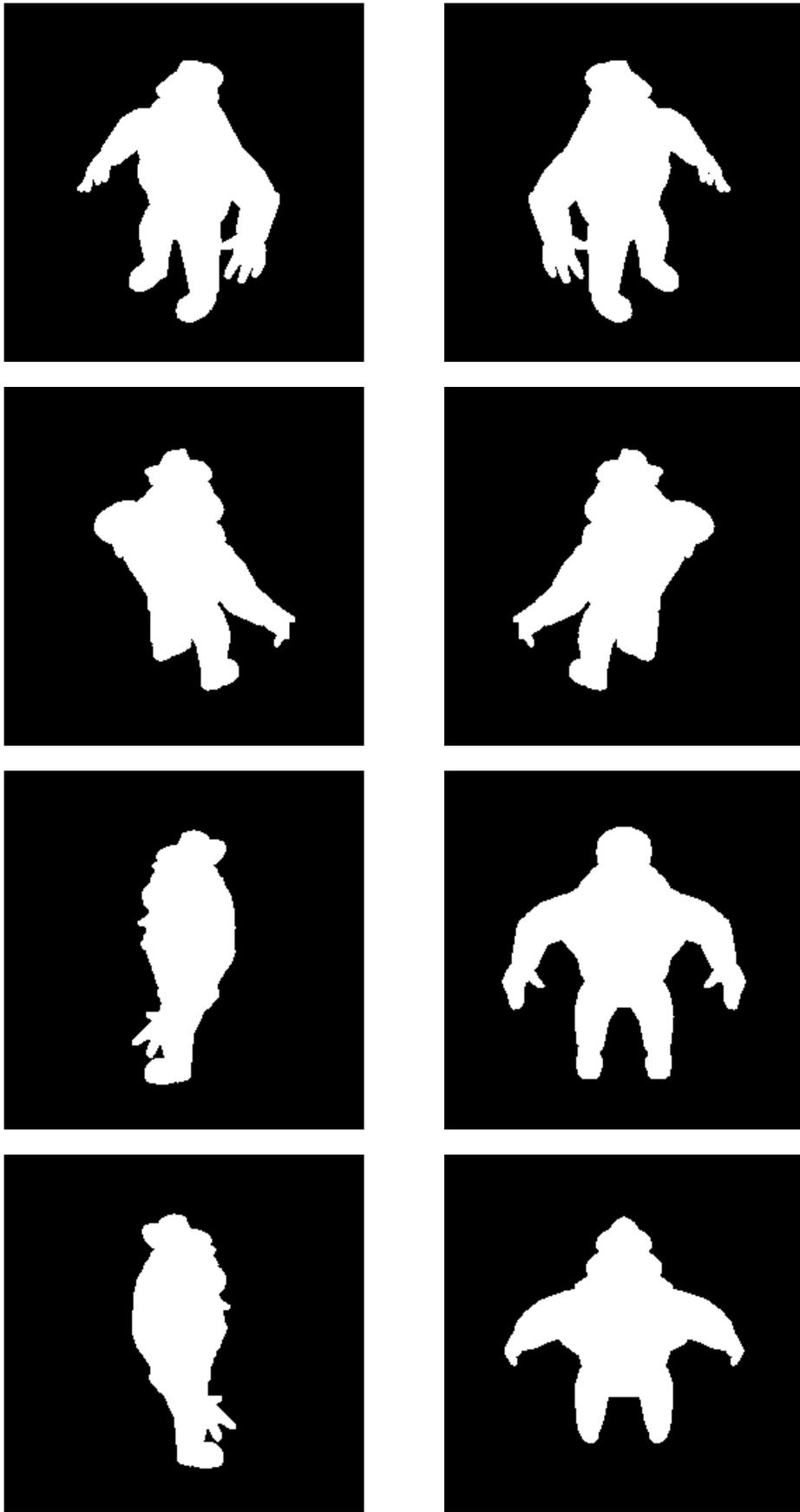


Figura 7.2: Dado I - Al - Imagens segmentadas

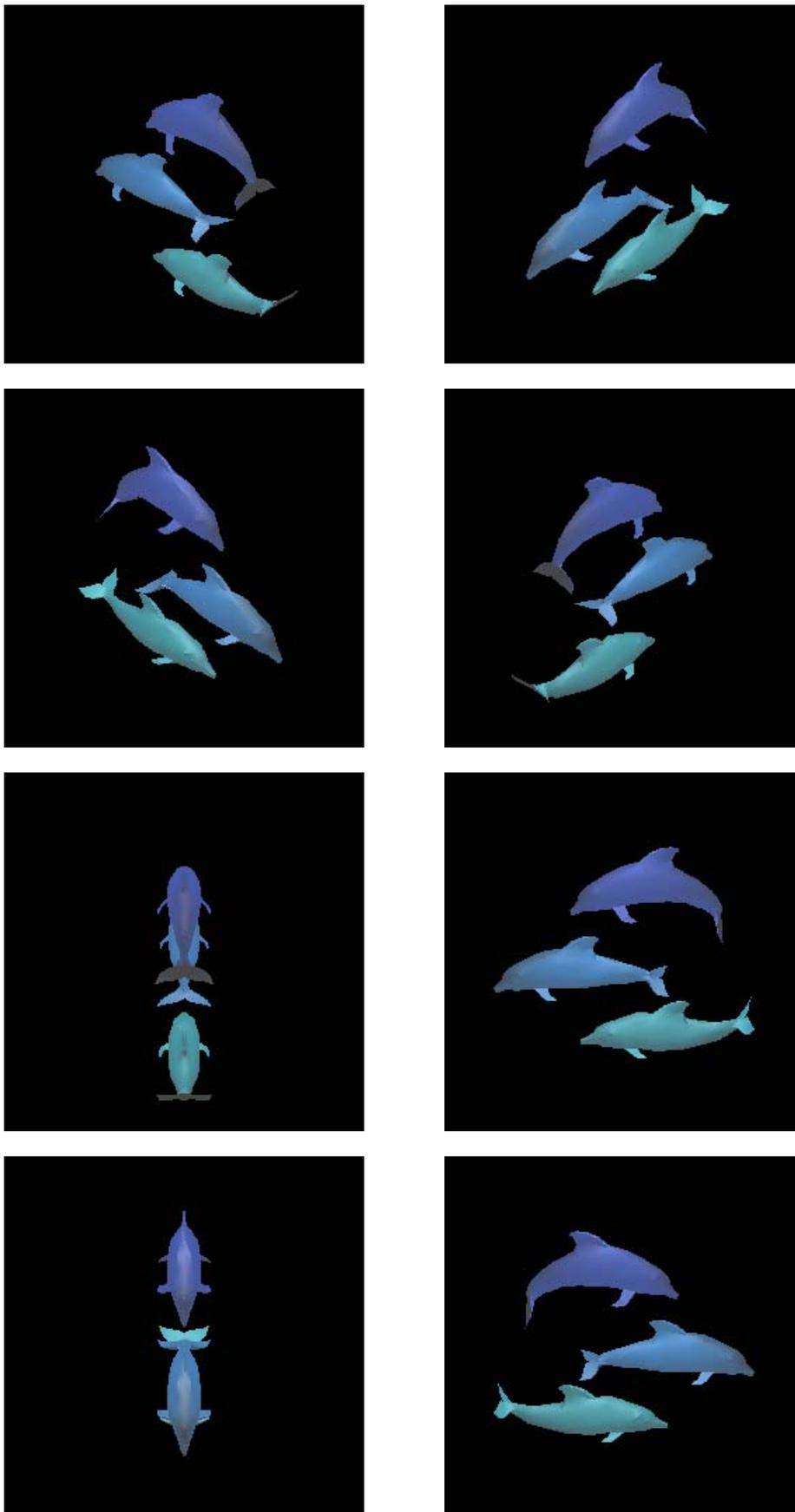


Figura 7.3: Dado II - Dolphins - Imagens de entrada

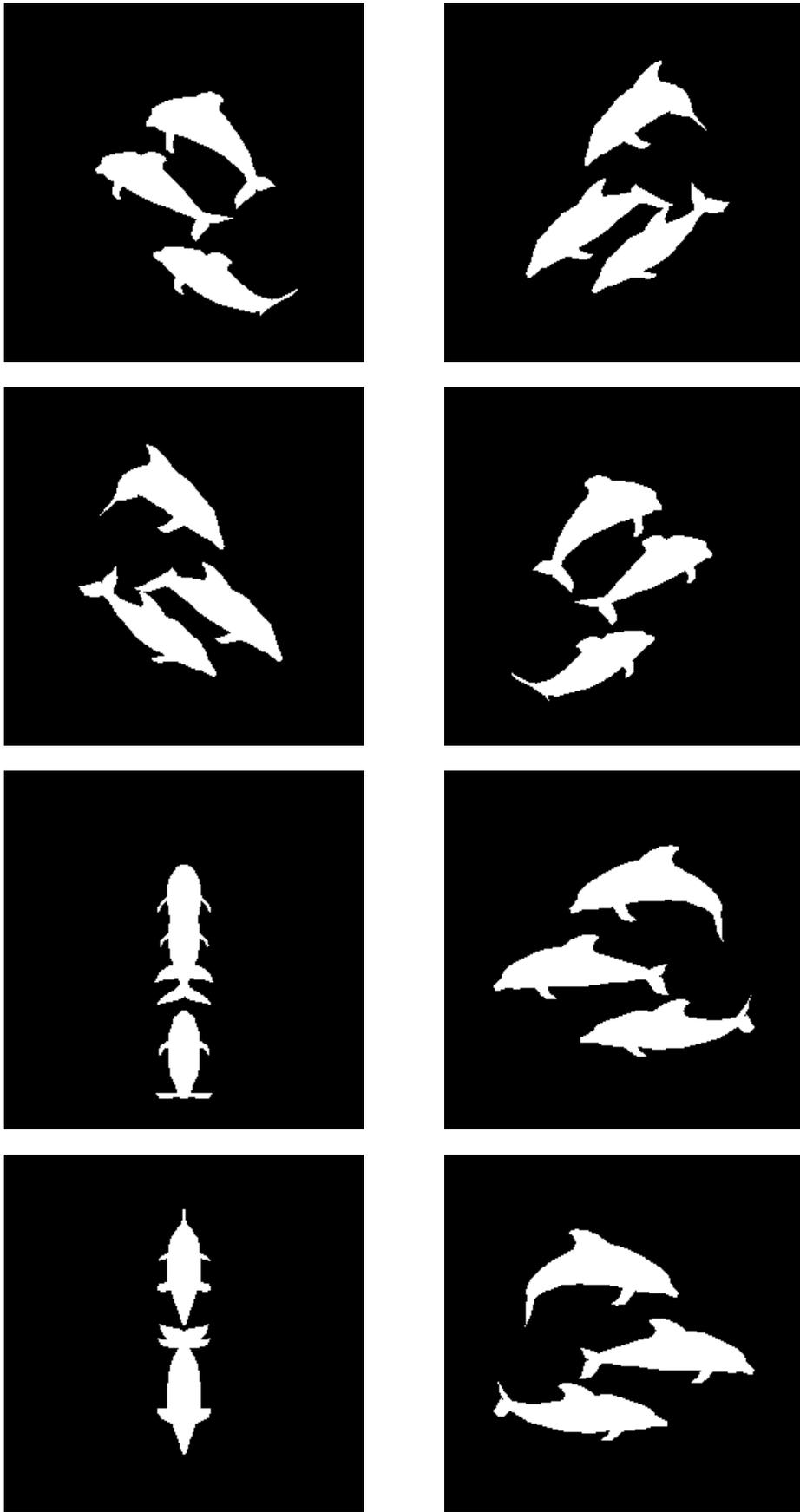


Figura 7.4: Dado II - Dolphins - Imagens de segmentação

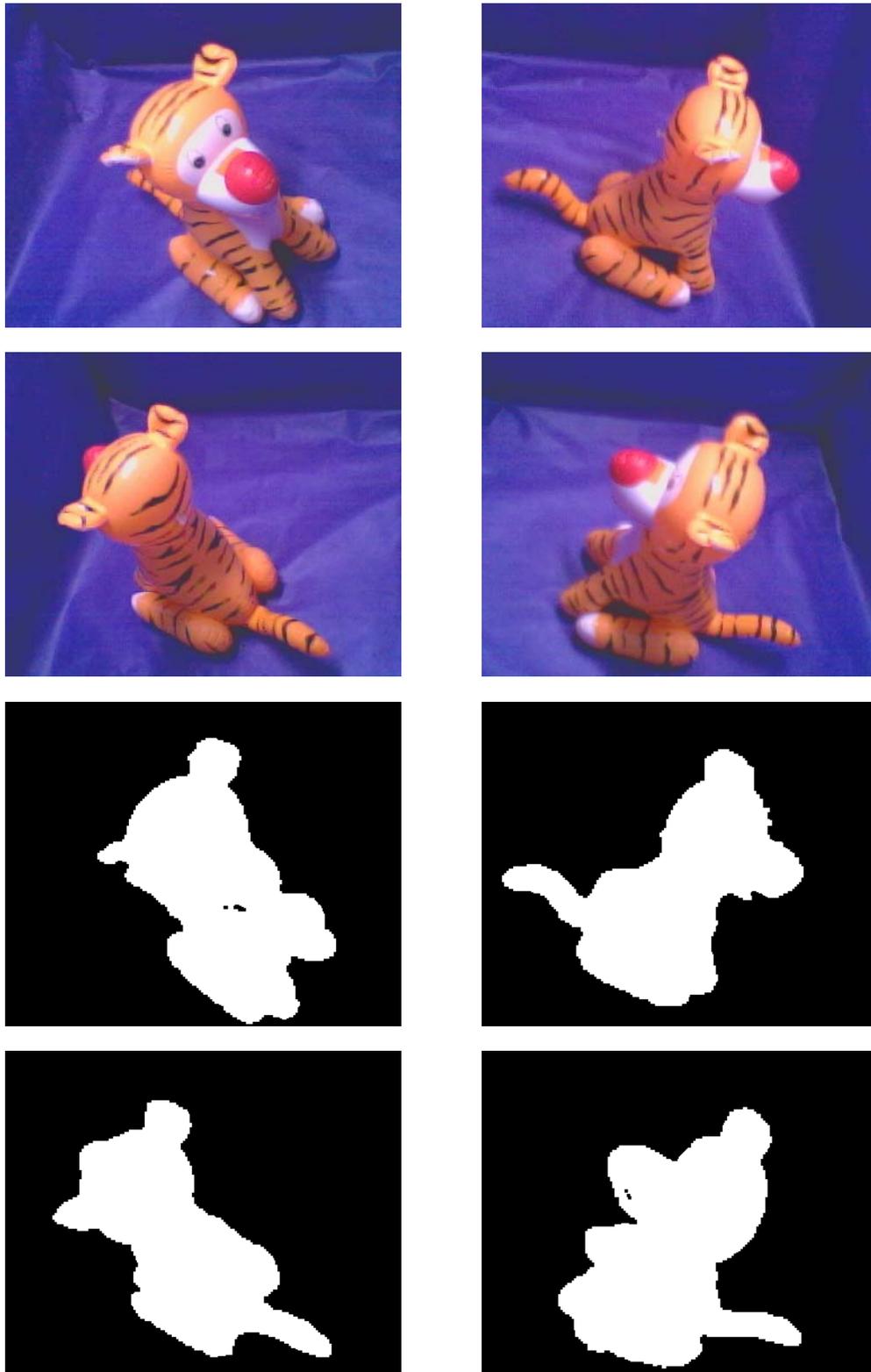


Figura 7.5: Dado III - Tigre - Imagens do objeto e da segmentação

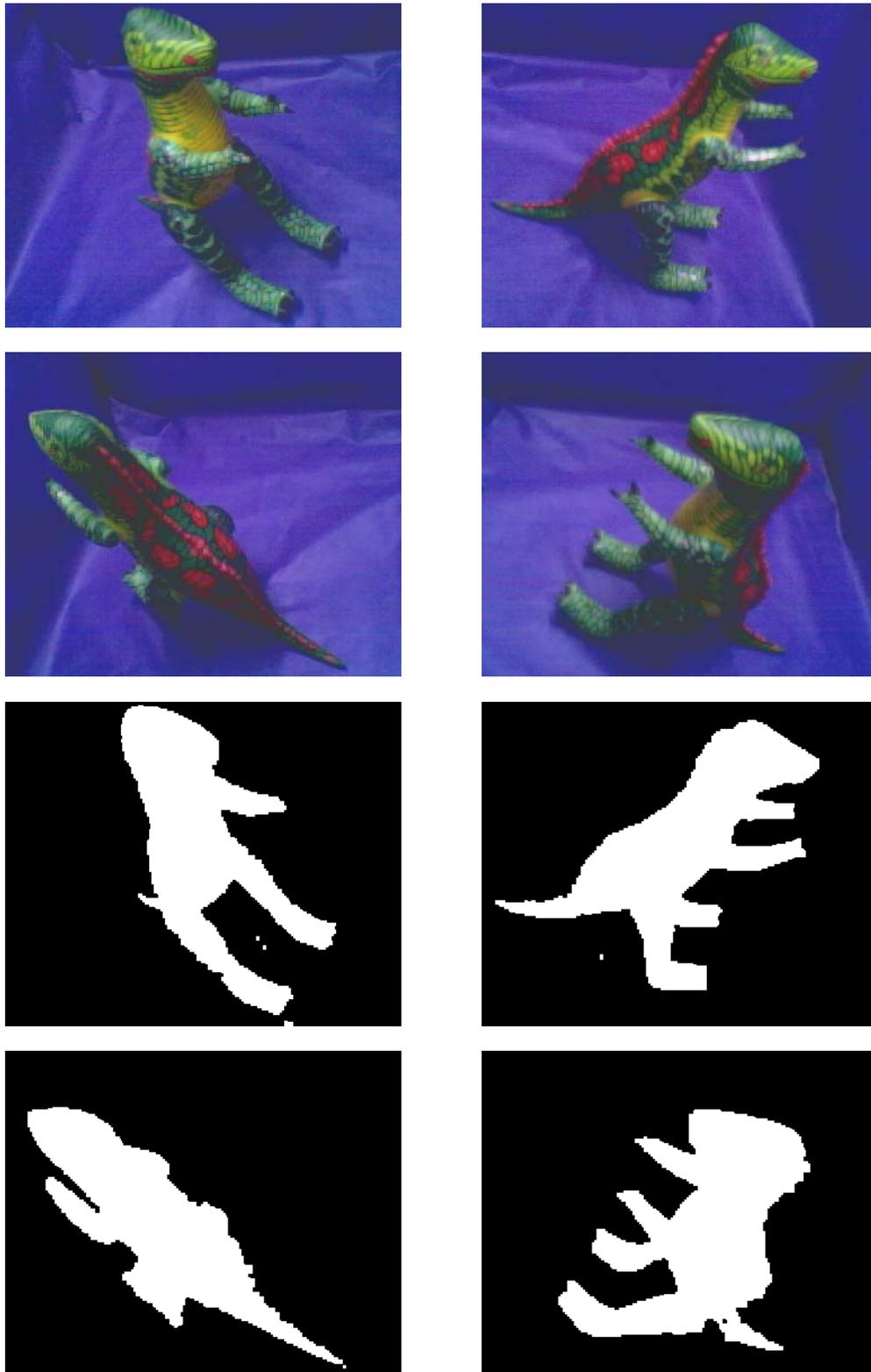


Figura 7.6: Dado IV - Dino - Imagens do objeto e da segmentação

7.2

Análise qualitativa

Apresentamos agora algumas imagens obtidas através da reconstrução realizada por cada um dos métodos de escultura do espaço na resolução máxima onde praticamente temos um voxel para cada pixel. As figuras 7.7, 7.8, 7.9, 7.10, 7.11 e 7.12 mostram os resultados da aplicação das três variações do método de escultura do espaço para os dois casos sintéticos. As figuras 7.13, 7.14, 7.15, 7.16, 7.17 e 7.18 são as correspondentes para os casos reais. As quatro primeiras imagens são imagens do objeto reconstruído a partir dos pontos de vista originais e as quatro últimas correspondem a imagens obtidas a partir de novos pontos de vista.

Em termos qualitativos, os resultados produzidos pelo método no espaço da cena (EEEC) e o método por refinamento adaptativo (EERA) (que também trabalha no espaço da cena) são praticamente equivalentes. Já os resultados produzidos pelo método no espaço das imagens (EEEI), apresentam vários voxels na superfície que não receberam informação de cor por terem sido classificados erroneamente como não visíveis durante a etapa de reconstrução. A figura 7.19 mostra uma ampliação da reconstrução do dado Tigre na qual as falhas estão mais visíveis.

Apesar das falhas apresentadas, podemos perceber que os resultados produzidos pelo método EEEI são ligeiramente mais nítidos que os anteriores. Uma explicação para a maior nitidez das imagens da reconstrução obtida pelo método EEEI está no fato de que este método não utiliza procedimento de filtragem algum, ao contrário dos métodos EEEC e EERA. Como os métodos EEEC e EERA filtram as imagens registradas no espaço da cena, a reconstrução naturalmente terá suas informações fotométricas mais suavizadas que no método EEEI. Logo, é de se esperar que os resultados produzidos pelos métodos que trabalham no espaço da cena apresentem um aspecto um pouco mais "borrado". Nos exemplos desta seção, o método EEEI reproduziu as informações fotométricas de forma razoável.

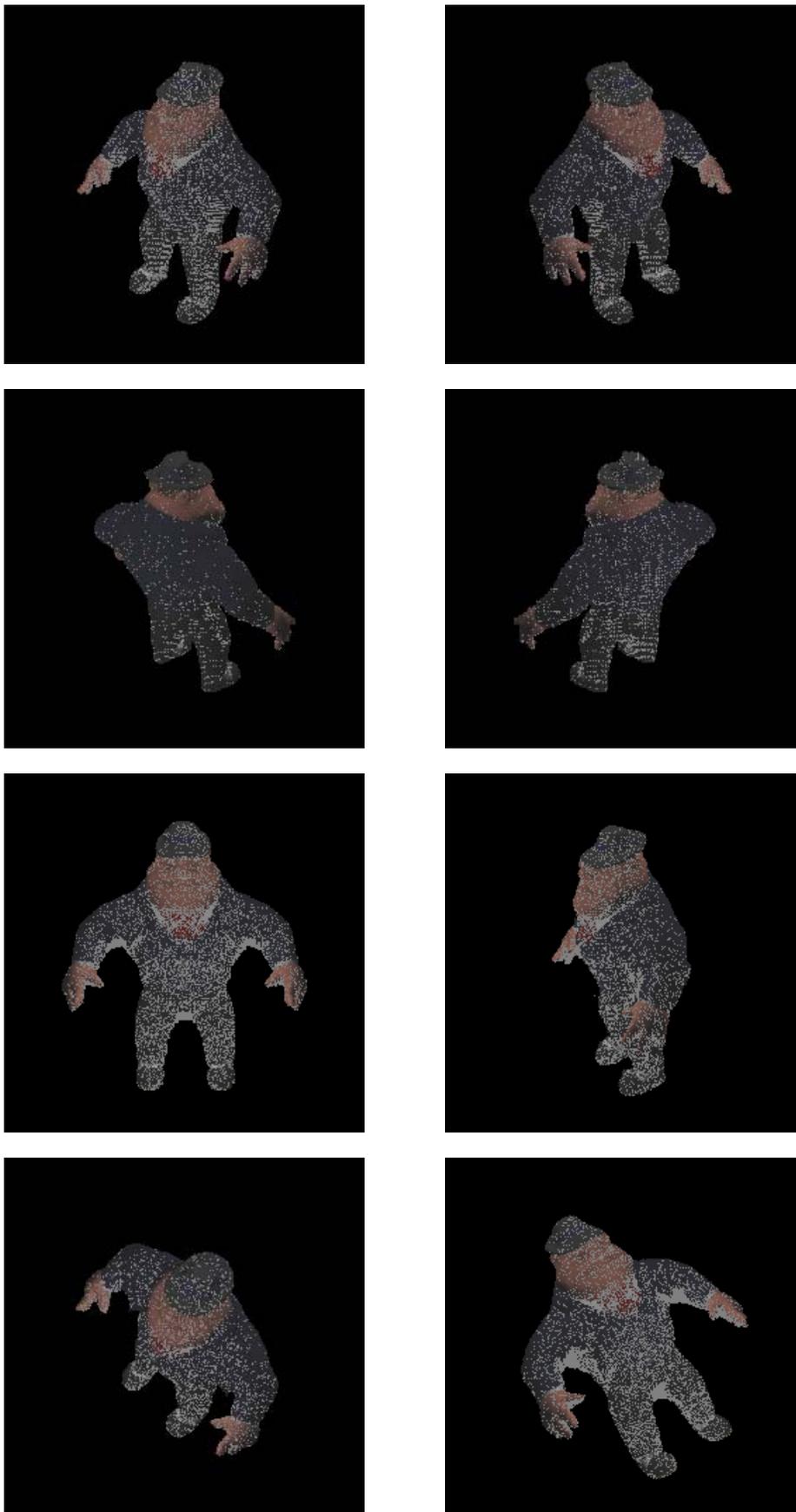


Figura 7.7: Reconstrução do dado I através do algoritmo EEI



Figura 7.8: Reconstrução do dado I através do algoritmo EEEEC



Figura 7.9: Reconstrução do dado I através do algoritmo EERA



Figura 7.10: Reconstrução do dado II através do algoritmo EEEI

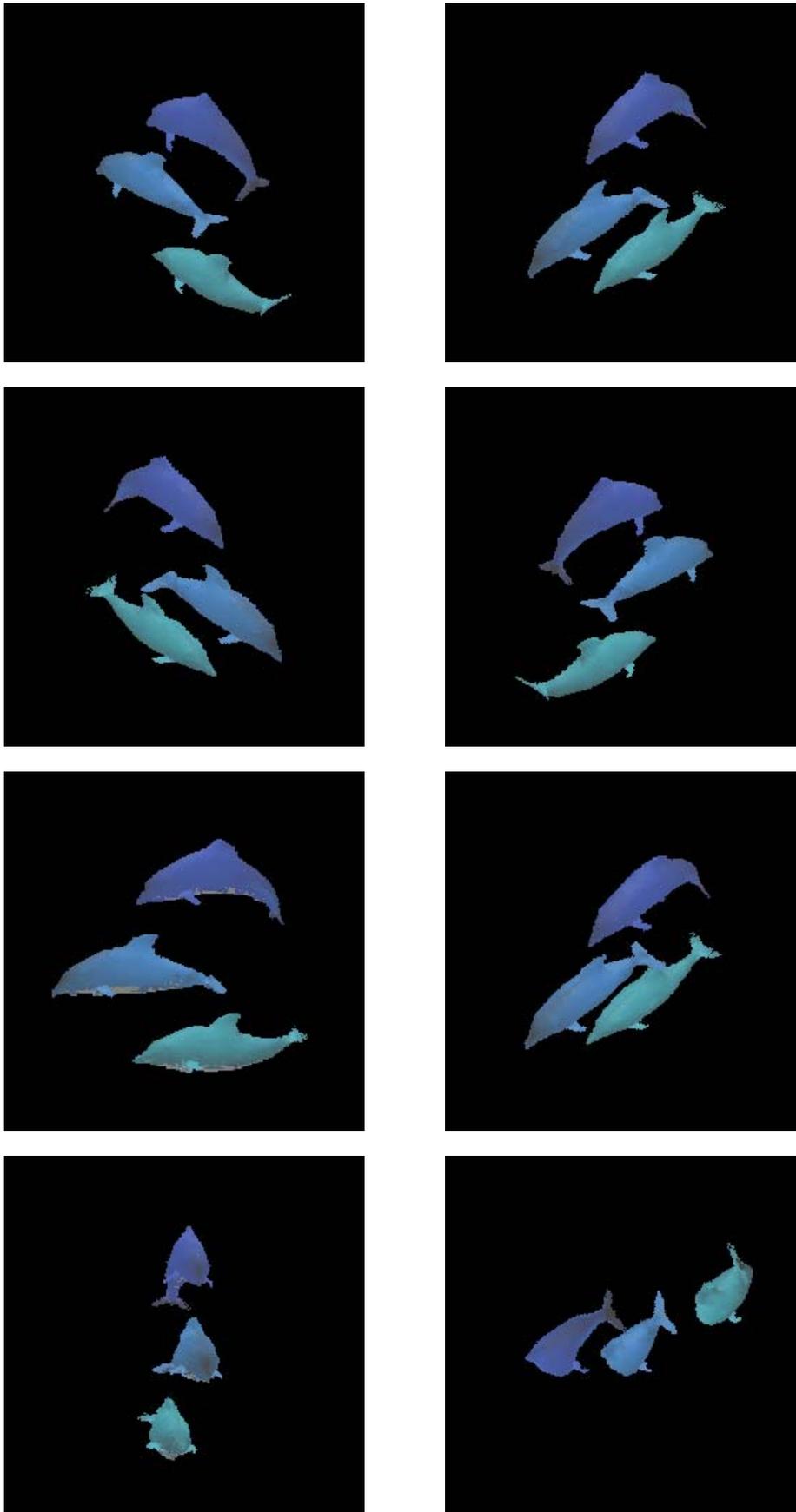


Figura 7.11: Reconstrução do dado II através do algoritmo EEEEC

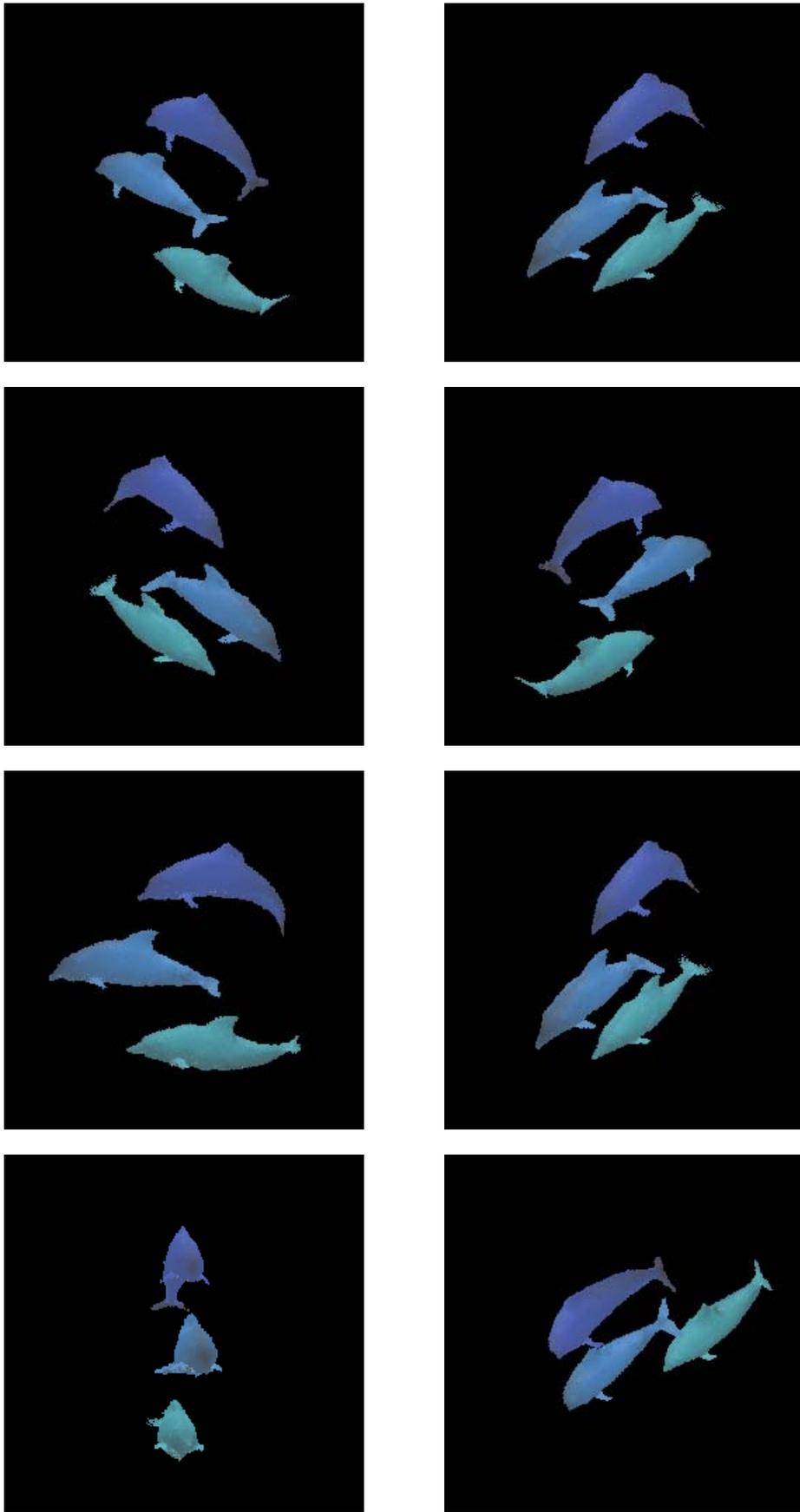


Figura 7.12: Reconstrução do dado II através do algoritmo EERA

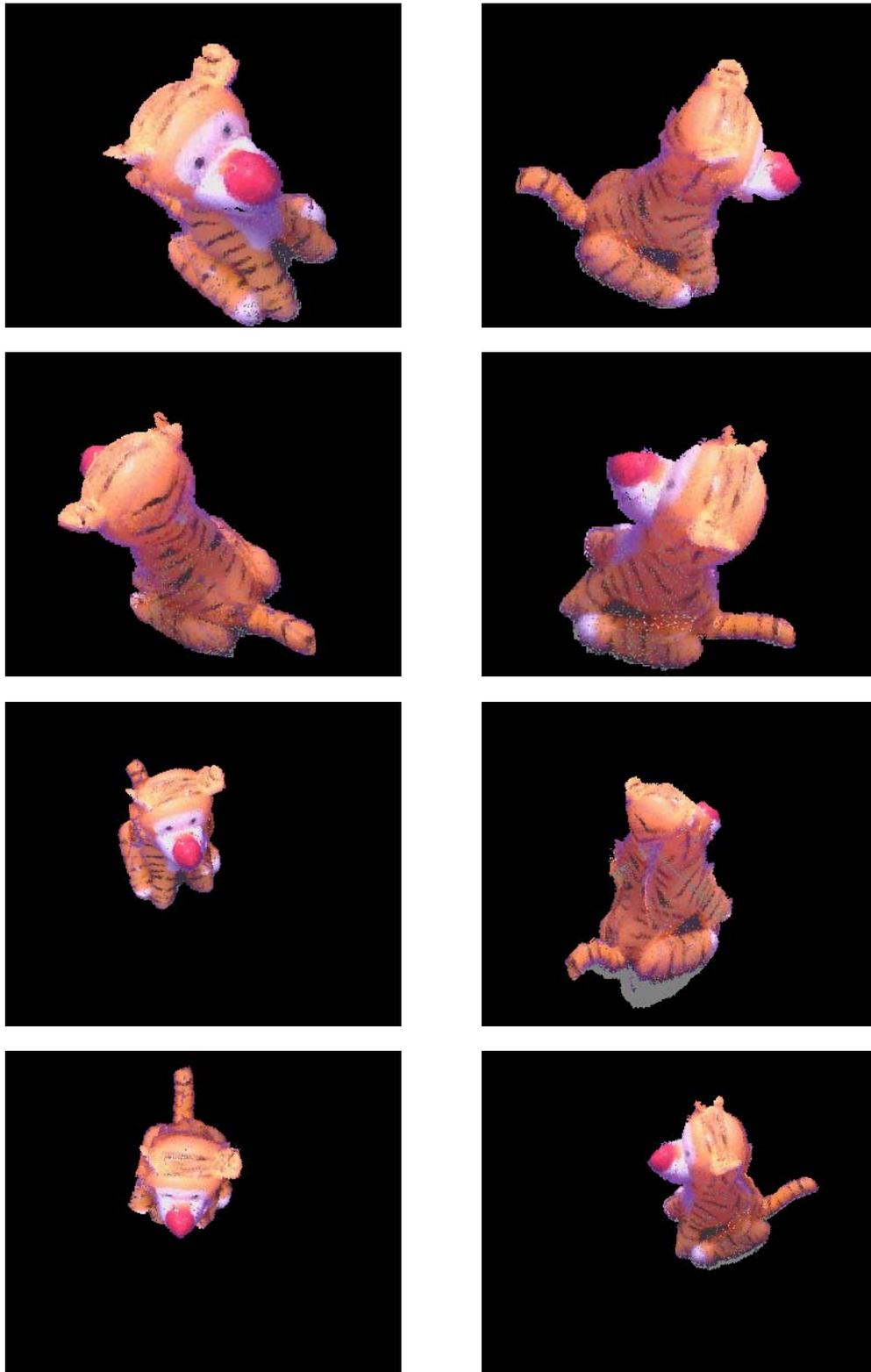


Figura 7.13: Reconstrução do dado III através do algoritmo EEI



Figura 7.14: Reconstrução do dado III através do algoritmo EEEC



Figura 7.15: Reconstrução do dado III através do algoritmo EERA

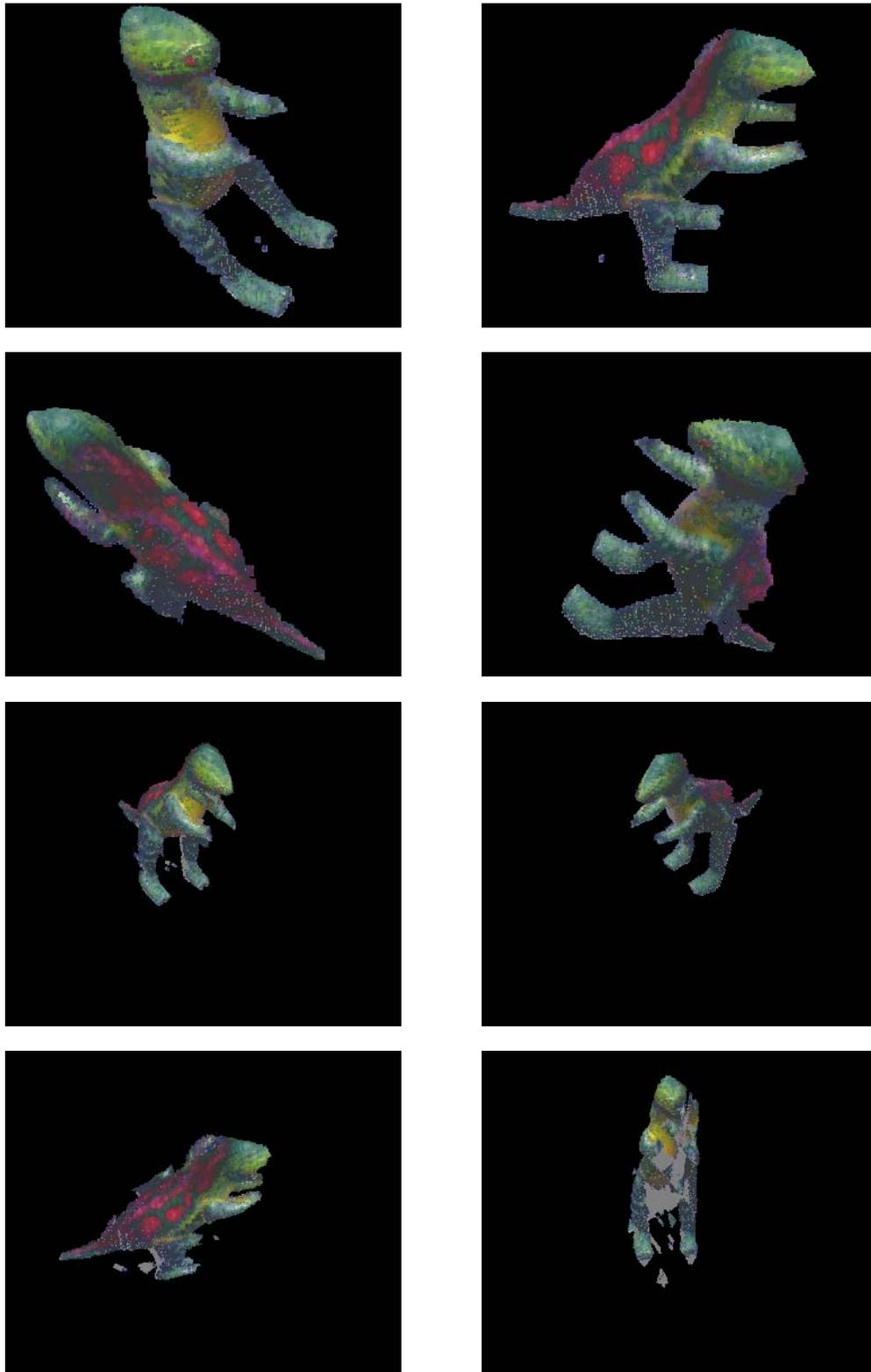


Figura 7.16: Reconstrução do dado IV através do algoritmo EEI

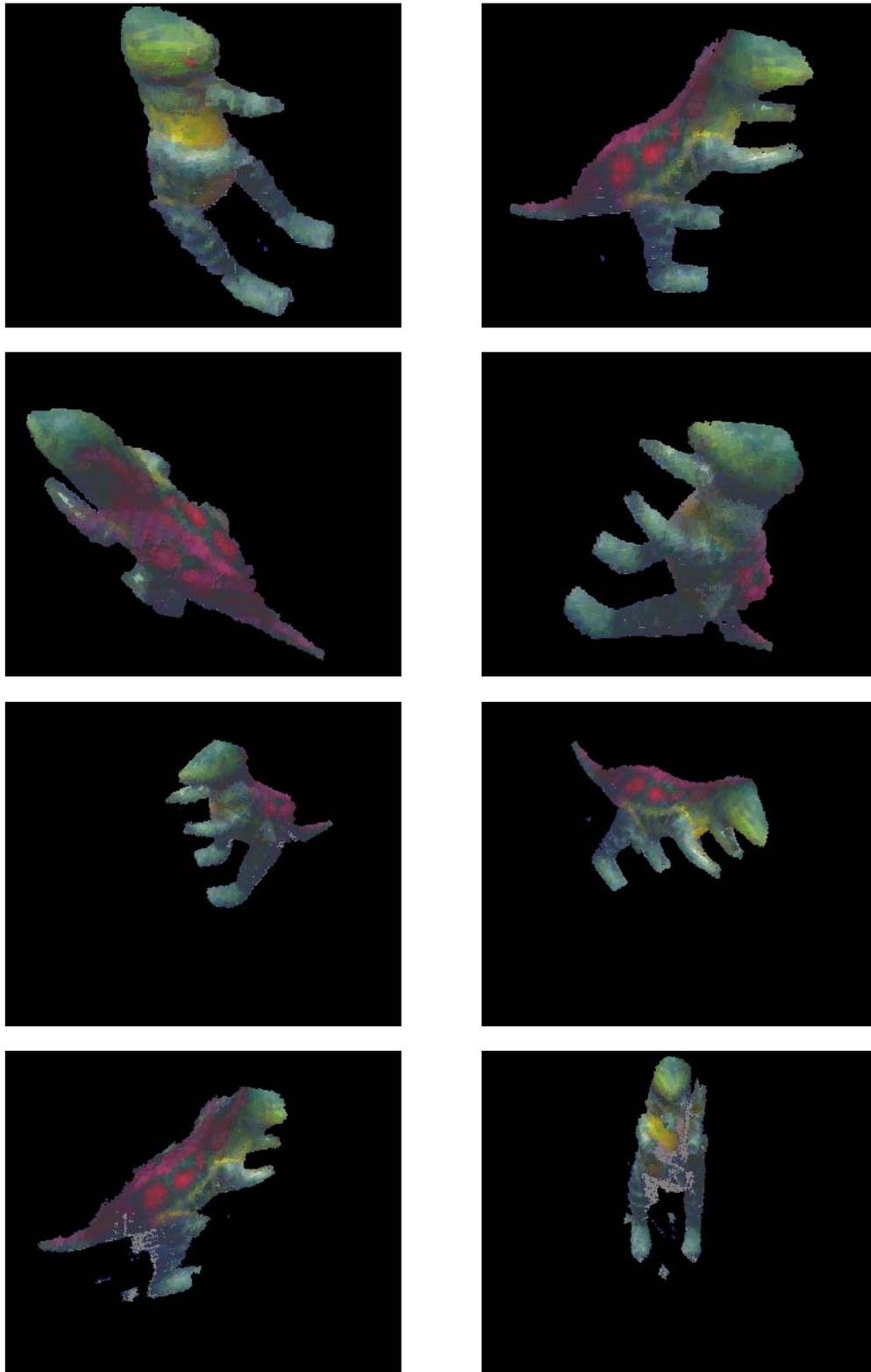


Figura 7.17: Reconstrução do dado IV através do algoritmo EEEEC

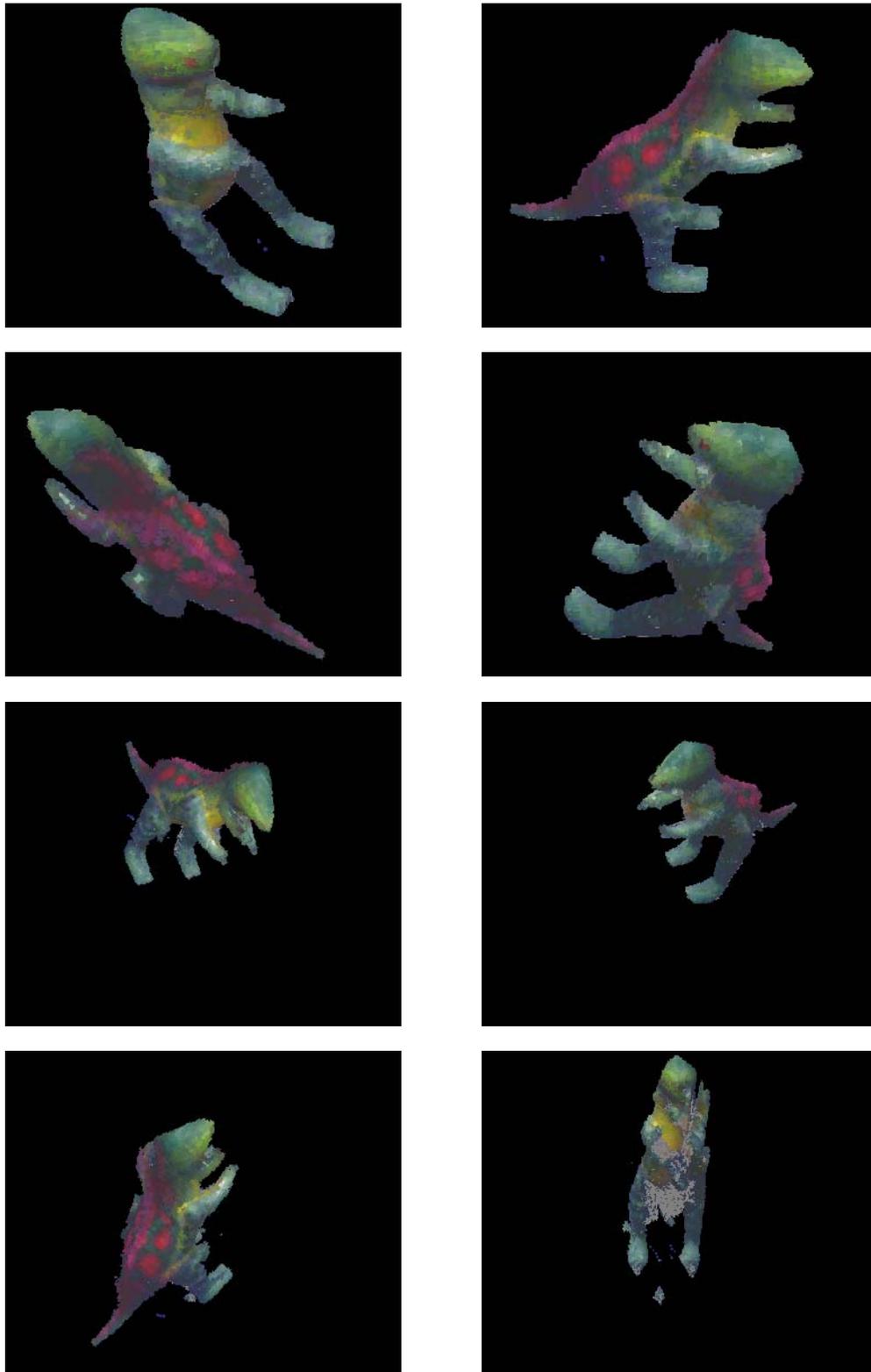


Figura 7.18: Reconstrução do dado IV através do algoritmo EERA

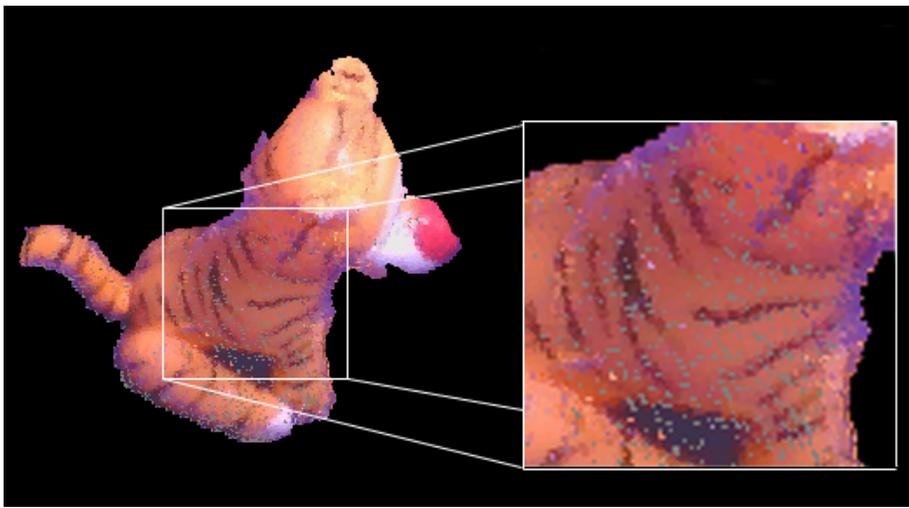


Figura 7.19: Voxels classificados como não visíveis

7.3

Análise em função da resolução

Na análise anterior todas as reconstruções foram geradas no nível de resolução máximo dificultando uma avaliação correta do comportamento de cada uma das variantes do método em relação aos problemas de amostragem.

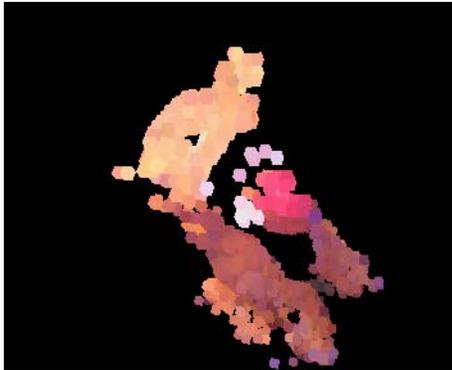
As figuras 7.20 e 7.21 mostram resultados do dado Tigre e do dado Dino, respectivamente para as três variantes do método de escultura do espaço para as resoluções 64×64 e 128×128 . Os resultados dos dados sintéticos apresentaram comportamentos semelhantes porém menos evidentes por isto não são mostrados.

Através da análise dos resultados em resoluções mais baixas fica evidente a importância de se amostrar as informações fotométricas de modo correto. O método de escultura do espaço no espaço das imagens claramente não consegue reconstruir os objetos reais de forma correta quando utilizamos um espaço de reconstrução de baixa resolução. Como podemos observar através das figuras 7.20(a) e 7.21(a), uma boa parte dos voxels na superfície do objeto não aparecem na reconstrução final, justamente por terem sido erroneamente classificados como não foto-consistentes. Isto ocorre porque a reamostragem inadequada faz com que alguns voxels pertencentes à superfície do objeto sejam projetados sobre pixels cujas cores não são consistentes quando na verdade deveriam ser, o que faz com sejam eliminados da reconstrução final. Podemos observar também, que os atributos de cor dos objetos reconstruídos pelo método que trabalha no espaço das imagens são nitidamente afetados por aliasing (Figuras 7.20(b) e 7.21(b)).

Os métodos que trabalham no espaço da cena, por outro lado, fornecem resultados corretos apesar de terem sua qualidade comprometida quando a resolução é muito baixa. Além disso os efeitos de aliasing são bastante atenuados quando comparados aos que ocorrem nas reconstruções obtidas através dos métodos que trabalham no espaço das imagens. Nas resoluções intermediárias, os resultados são bastante satisfatórios com qualidade próxima a da resolução máxima obtida no caso do dado Tigre. A complexidade da textura do dado Dino fez com que nestes casos a solução ainda apresentasse algumas falhas.

Os resultados obtidos através de reconstruções em espaços de baixa resolução demonstram definitivamente que é necessário um tratamento correto do problema de reamostragem para que possamos realizar o processo de escultura do espaço por refinamento adaptativo pois, inevitavelmente, tais métodos precisam lidar de modo apropriado com a avaliação da foto-

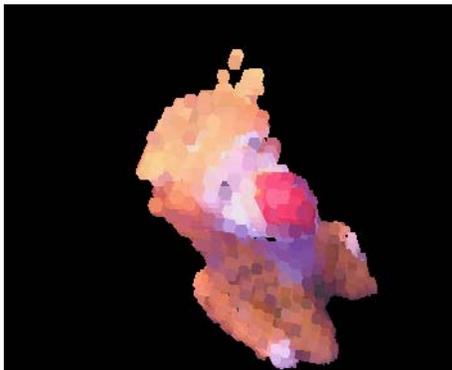
consistência de elementos em baixa resolução. A forma mais eficaz e também eficiente de conseguirmos amostrar as informações de forma correta nos parece ser através do registro das informações no espaço da cena através de mapeamento projetivo com filtragem baseada em *mipmapping*, a qual seleciona automaticamente um nível da pirâmide em função do nível corrente da octree.



7.20(a): EEEI - Tigre - 64x64



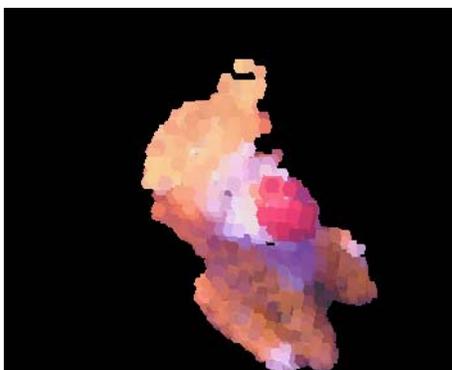
7.20(b): EEEI - Tigre - 128x128



7.20(c): EEEC - Tigre - 64x64



7.20(d): EEEC - Tigre - 128x128

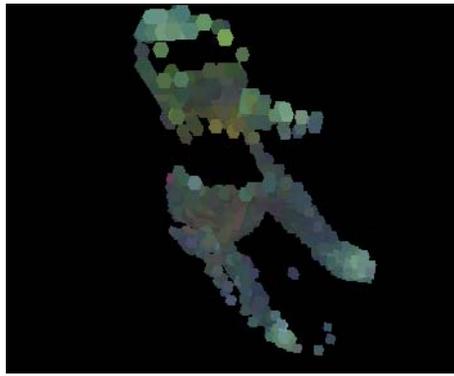


7.20(e): EERA - Tigre - 64x64

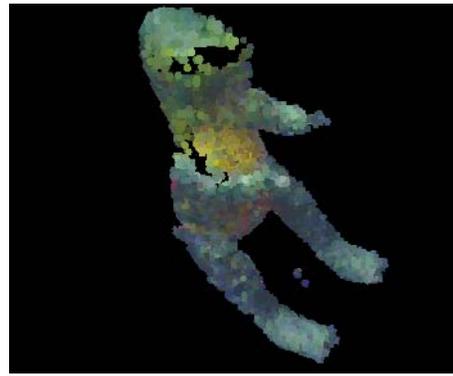


7.20(f): EERA - Tigre - 128x128

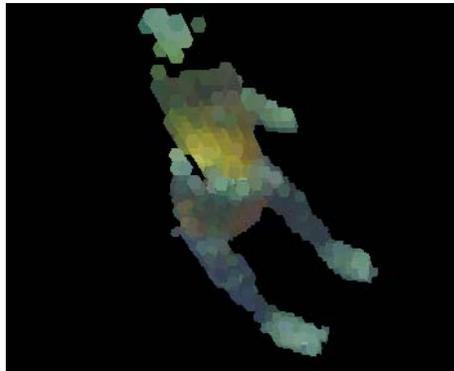
Figura 7.20: Comparação entre os diversos métodos no níveis de resolução 6x64 e 128x128.



7.21(a): EEEI - Dino - 64x64



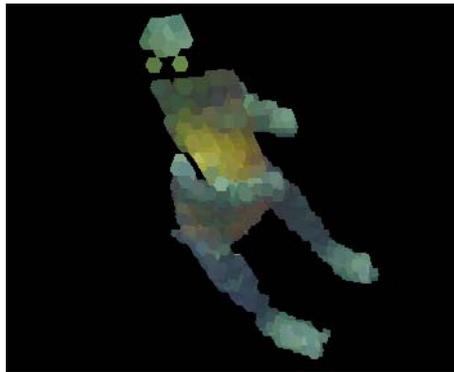
7.21(b): EEEI - Dino - 128x128



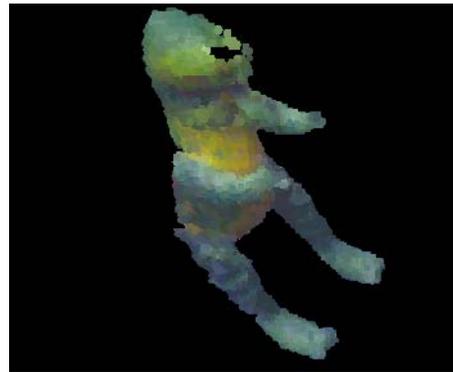
7.21(c): EEEEC - Dino - 64x64



7.21(d): EEEEC - Dino - 128x128



7.21(e): EERA - Dino - 64x64



7.21(f): EERA - Dino - 128x128

Figura 7.21: Comparação entre os diversos métodos no níveis de resolução 6x64 e 128x128.

7.4

Análise do algoritmo EERA nos diferentes níveis de refinamento

Os métodos de escultura do espaço por refinamento adaptativo determinam partes do objeto a ser reconstruído progressivamente à medida em que o processo é executado. As figuras 7.22, 7.23, 7.24 e 7.25 ilustram os resultados parciais produzidos pelo método em cada nível de refinamento.

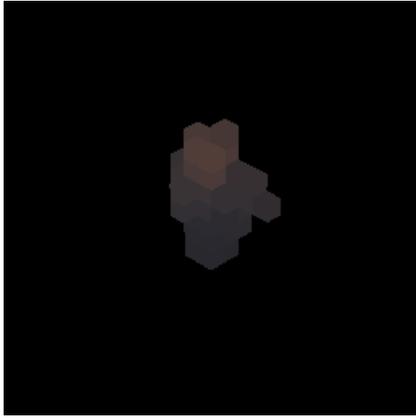
No dado Al não foram detectadas células foto-consistentes nos níveis de 0 a 3, e no dado Dolphins nos níveis de 0 a 4. Por este motivo, as imagens correspondentes a estes níveis foram omitidas por não trazerem nenhuma informação. O mesmo ocorre para os dados reais nos níveis de 0 a 4.

O fato dos níveis iniciais classificarem a maioria das células como indefinidas é coerente com a forma através da qual os limiares são determinados na avaliação da foto-consistência das células da octree. Como mostra o capítulo 6, tais limiares são bastante rígidos no início do processo para que possamos compensar a redução nos desvios observados causados pela filtragem. Nos dados sintéticos, onde as texturas são menos ricas, o método consegue recuperar uma estimativa grosseira da forma já nos estágios iniciais como mostram as figuras 7.22(b) e 7.23(a). Já nos dados reais, isto é um pouco mais difícil devido às texturas serem mais complexas, o que impede a determinação de foto-consistência nas etapas iniciais do processo.

Uma outra característica que pode ser observada no método proposto é o fato de ser conservativo, atendendo o requisito básico do método de escultura do espaço. Apesar de nos estágios iniciais nenhuma parte do objeto ter sido detectada, temos a garantia, no final do processamento, de que nenhum fragmento da superfície é perdido devido à estratégia de refinamento. Entretanto, persistem os problemas de perdas causadas pela imprecisão da segmentação e pela não adequabilidade dos limiares para certas regiões da cena. O grande problema seria se perdêssemos partes do objeto, caso o processo não fosse conservativo, como ocorre no método de Prock.

Finalmente, podemos observar que o método produz, nos estágios de refinamento finais, reconstruções cujos atributos fotométricos são coerentes com as imagens de entrada. Como explicamos antes, isto se deve principalmente ao fato de que as informações de segmentação projetadas sobre os planos de referência são filtradas de forma a ficarem compatíveis com a resolução do dado volumétrico. Nos níveis de resolução mais baixos o resultado da filtragem faz com que as células próximas à superfície do objeto sejam normalmente classificadas como indefinidas. Nestes casos, as células

na fronteira só são consideradas foto-consistentes nos níveis de resolução finais onde a taxa de amostragem é semelhante a da imagem original, o que permite uma correta atribuição de cores. Cabe reafirmar que os resultados obtidos através da variante por refinamento adaptativo são visualmente semelhantes aos obtidos através da variante no espaço da cena não adaptativa, como pode ser visto nas figuras da seção sobre análise qualitativa.



7.22(a): Nível de refinamento 4



7.22(b): Nível de refinamento 5



7.22(c): Nível de refinamento 6

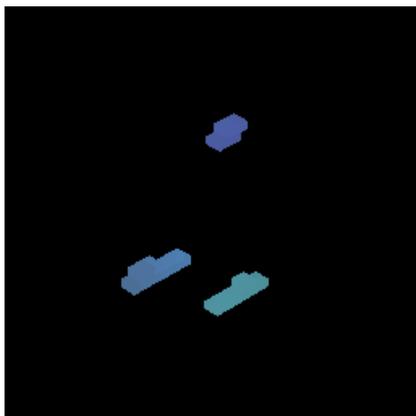


7.22(d): Nível de refinamento 7

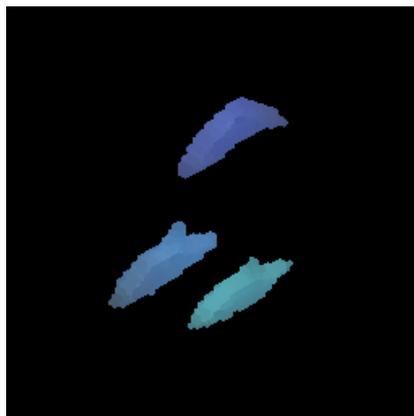


7.22(e): Nível de refinamento 8

Figura 7.22: Estágios de refinamento de 4 a 8 da reconstrução do dado I através do algoritmo EERA.



7.23(a): Nível de refinamento 5



7.23(b): Nível de refinamento 6

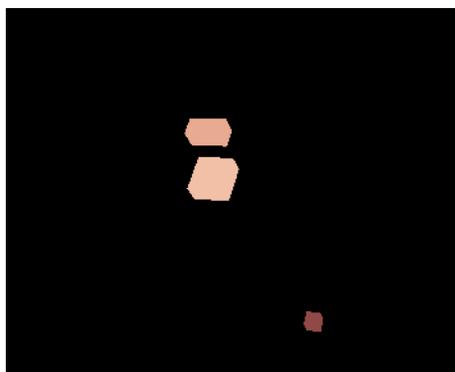


7.23(c): Nível de refinamento 7

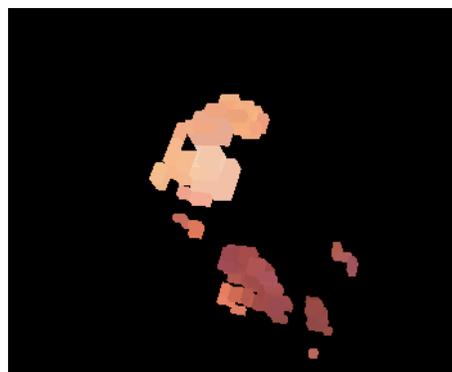


7.23(d): Nível de refinamento 8

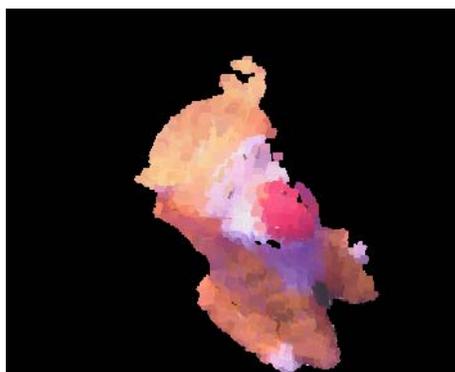
Figura 7.23: Estágios de refinamento de 5 a 8 da reconstrução do dado II através do algoritmo EERA.



7.24(a): Nível de refinamento 5



7.24(b): Nível de refinamento 6

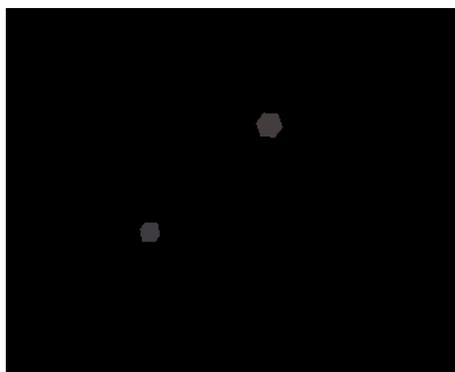


7.24(c): Nível de refinamento 7

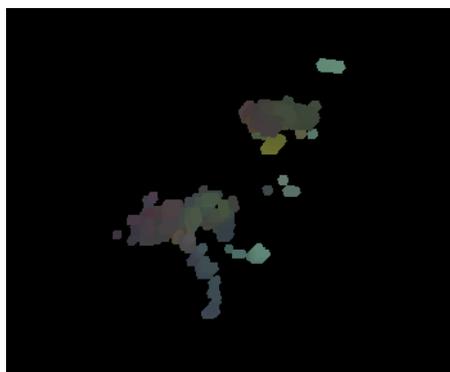


7.24(d): Nível de refinamento 8

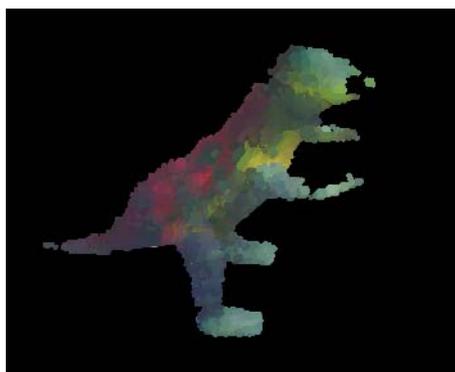
Figura 7.24: Estágios de refinamento de 5 a 8 da reconstrução do dado III através do algoritmo EERA.



7.25(a): Nível de refinamento 5



7.25(b): Nível de refinamento 6



7.25(c): Nível de refinamento 7



7.25(d): Nível de refinamento 8

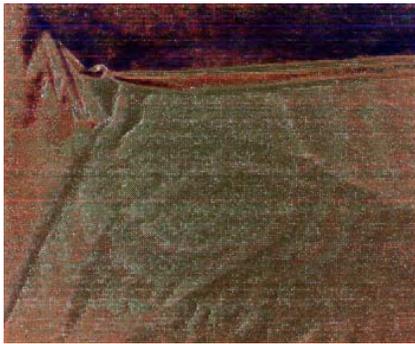
Figura 7.25: Estágios de refinamento de 5 a 8 da reconstrução do dado IV através do algoritmo EERA.

7.5

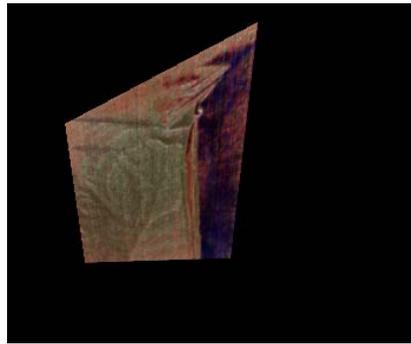
Análise do teste estatístico baseado em limiares locais

Os resultados apresentados anteriormente foram obtidos utilizando-se um critério de foto-consistência que compara os valores de uma estatística calculada sobre as observações com um limiar global estabelecido empiricamente. Nesta seção, apresentamos alguns resultados obtidos através da utilização de um critério de foto-consistência baseado no teste estatístico proposto no capítulo 5, o qual modela diferenças nas observações geradas pelo ruído produzido por cada um dos sensores de cada uma das câmeras.

Para que este teste pudesse ser realizado associamos um mapa de desvios para cada câmera a partir dos desvios calculados para os mapas de intervalos de confiança utilizados na segmentação do objeto de interesse (ver capítulo 6, seção 6.3). A figura 7.26(a) ilustra um dos mapas de desvio de uma das câmeras e a figura 7.26(b) o seu registro sobre um plano no espaço da cena.



7.26(a): Mapa de desvios



7.26(b): Mapeamento de desvios registrado

As figuras 7.26(c), 7.26(d), 7.26(e), 7.26(f), 7.26(g) 7.26(h), 7.26(i), 7.26(j), 7.26(k) mostram os resultados obtidos pela aplicação dos algoritmos EEI, EEC e EERA sobre o dado Tigre para três diferentes níveis de significância, enquanto que as figuras 7.27(a), 7.27(b), 7.27(c), 7.27(d), 7.27(e) 7.27(f), 7.27(g), 7.27(h), 7.27(i), representam os resultados obtidos através do mesmo tipo de teste aplicado ao dado Dino. Os dados sintéticos não foram avaliados já que o modelo proposto não permite avaliar adequadamente o tipo de erro existente no processo de geração de imagens correspondente.

Podemos observar que os resultados obtidos através da aplicação dos algoritmos EEAC e EEAR aos dados Tigre e Dino, como mostram respectivamente as figuras 7.26(h), 7.26(k) e 7.27(f), 7.27(i) são um pouco mais nítidos que os obtidos através do critério anterior. Podemos perceber

um maior detalhe por exemplo nas texturas de ambos os objetos. Isto se deve ao fato de que o uso de limiares locais permite que o critério seja mais rígido descartando mais elementos que não fazem realmente parte da superfície. Por outro lado, fica evidente que foi necessário aumentar muito o nível de significância, em torno de 99,99999999%, para que não houvesse um excesso de perdas de partes da cena a ser reconstruída, como ocorre, por exemplo, nas figuras 7.26(f), 7.26(g) e 7.27(g), 7.26(j), correspondentes ao dado Tigre, e nas figuras 7.27(d), 7.27(e) e 7.27(g), 7.27(h), referentes ao dado Dino. A maioria das perdas ocorre devido ao fato de que os limiares utilizados são determinados somente com base em um modelo que descreve probabilisticamente as diferenças observadas nas imagens de uma região da cena de acordo com o nível de ruído introduzido pelos sensores. Logo, tais limiares não são capazes de levar em consideração outras fontes de erro como, por exemplo, o fato das superfícies da cena não serem perfeitamente lambertianas. Neste caso, eles se tornam excessivamente rígidos quando utilizados na avaliação de regiões da imagem correspondentes a regiões da cena cuja diferença de cor observada nas imagens de entrada se deve predominantemente a fenômenos não lambertianos, como, por exemplo, reflectância.

O algoritmo EEEI, com o critério de foto-consistência baseado em limiares locais, não produziu resultados satisfatórios, produzindo muitas perdas na reconstrução como podemos observar através das figuras 7.26(c), 7.26(d), e 7.26(e), correspondentes ao dado Tigre, e através das figuras 7.27(a), 7.27(b) e 7.27(c), correspondentes ao dado Dino. Este resultado nos fornece mais indícios de que uma amostragem não adequada contribui para uma aumento do desvio dos valores de cor observados nas diferentes imagens de uma certa região da cena, impedindo assim o uso de limiares mais rígidos os quais poderiam possibilitar a obtenção de resultados mais precisos.

Finalmente, concluímos que o teste baseado em limiares locais permite uma avaliação em termos probabilísticos da foto-consistência de um dado elemento do espaço em relação às imagens de entrada, além de possibilitar a realização de uma checagem mais flexível, que se adapta aos diferentes níveis de erro local que podem existir nas diversas regiões da superfície da cena. Ainda assim, verificamos que é preciso adicionar ao modelo alguma componente capaz de contemplar erros não causados pelo ruído dos sensores, o que pode ser feito inicialmente, através da adição de alguma constante de erro global aos limiares de comparação. O simples aumento do nível de significância, estratégia aqui adotada, não é o mais adequado devido

aos limites determinados pela precisão numérica. De fato, a estratégia ideal envolve a utilização de um modelo probabilístico mais completo o que requer estudos mais profundos que devem ser realizados em trabalhos futuros.



7.26(c): EEEI - Nível de significância 99,99%



7.26(d): EEEI - Nível de significância 99,9999%



7.26(e): EEEI - Nível de significância 99,99999999%



7.26(f): EEEI - Nível de significância 99,99%



7.26(g): EEEI - Nível de significância 99,9999%



7.26(h): EEEI - Nível de significância 99,99999999%



7.26(i): EEEI - Nível de significância 99,99%



7.26(j): EEEI - Nível de significância 99,9999%

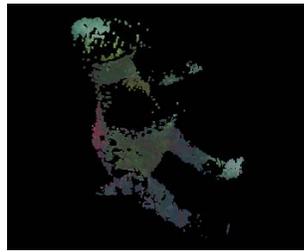


7.26(k): EEEI - Nível de significância 99,99999999%

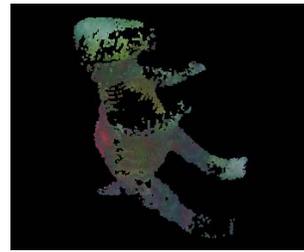
Figura 7.26: Resultado da aplicação dos algoritmos para diferentes níveis de significância do teste estatístico proposto no capítulo 5



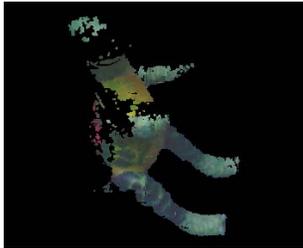
7.27(a): EEEI - Nível de significância 99,99%



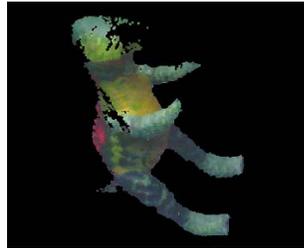
7.27(b): EEEI - Nível de significância 99,9999%



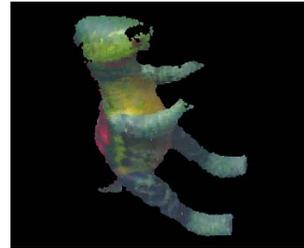
7.27(c): EEEI - Nível de significância 99,99999999%



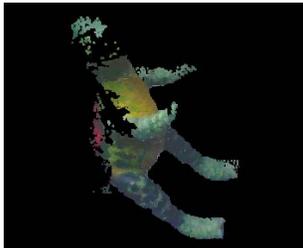
7.27(d): EEEC - Nível de significância 99,99%



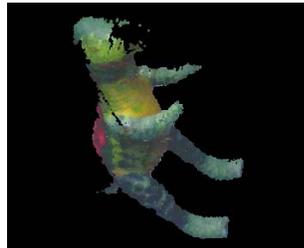
7.27(e): EEEC - Nível de significância 99,9999%



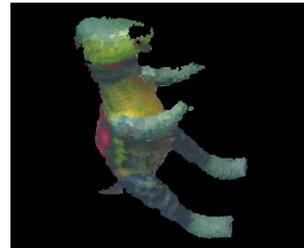
7.27(f): EEEC - Nível de significância 99,99999999%



7.27(g): EERA - Nível de significância 99,99%



7.27(h): EERA - Nível de significância 99,9999%



7.27(i): EERA - Nível de significância 99,99999999%

Figura 7.27: Resultado da aplicação dos algoritmos para diferentes níveis de significância do teste estatístico proposto no capítulo 5

7.6

Análise de desempenho

As tabelas de 7.1 a 7.12 mostram resultados obtidos pelas três variantes EEI,EEEC e EERA, para o conjunto de dados utilizados. Os tempos foram medidos em um computador Pentium IV 1.6 Mhz, com 512Mb de RAM e placa gráfica GeForce II 32Mb.

Analisando as tabelas, podemos perceber que o método que trabalha no espaço da imagem é bem mais rápido que o método no espaço da cena. Isto se deve ao fato, já mencionado anteriormente, que as operações de cópia do framebuffer para memória convencional introduzem um retardo bastante grande tornando o tempo de processamento do método EEEEC bem maior. O método EERA, que se baseia na estratégia de refinamento adaptativo, é mais lento nas reconstruções de baixa resolução, no entanto torna-se o mais rápido de todos à medida em que aumentamos a resolução, mesmo com todo o retardo introduzido pelas operações de cópia do framebuffer.

A maior eficiência do método de escultura do espaço por refinamento adaptativo depende, entretanto, da relação existente entre o volume inicial de reconstrução e o volume de ocupação do objeto. A maior eficiência no caso do dado Dolphins comprova esta afirmação. Neste dado a razão entre o volume ocupado pelo objeto e o volume inicial de reconstrução é pequena. Nos dados reais utilizamos um volume de reconstrução que é bem próximo do volume ocupado do objeto o que explica o menor do ganho em eficiência.

7.6.1

Desempenho dos métodos aplicados à reconstrução do dado AI

tamanho das imagens	total de voxels	voxels removidos	voxels consistentes	tempo
64x64	262144	242683	19461	1s
128x128	2097152	1941614	155538	6s
256x256	16777216	15532893	1244323	49s

Tabela 7.1: Resultado da aplicação do algoritmo EEI na reconstrução do dado I

tamanho das imagens	total de voxels	voxels removidos	voxels consistentes	tempo
64x64	262144	244540	17604	3s
128x128	2097152	1948049	149103	12s
256x256	16777216	15530995	1246221	70s

Tabela 7.2: Resultado da aplicação do algoritmo EEI na reconstrução do dado I

tamanho das imagens	tempo
64x64	4s
128x128	8s
256x256	18s

Tabela 7.3: Resultado da aplicação do algoritmo EERA na reconstrução do dado I

7.6.2

Desempenho dos métodos aplicados à reconstrução do dado Dolphins

tamanho das imagens	total de voxels	voxels removidos	voxels consistentes	tempo
64x64	262144	257438	4706	1s
128x128	2097152	2059575	37577	6s
256x256	16777216	15532893	299199	45s

Tabela 7.4: Resultado da aplicação do algoritmo EEI na reconstrução do dado II

tamanho das imagens	total de voxels	voxels removidos	voxels consistentes	tempo
64x64	262144	258286	3858	3s
128x128	2097152	2062666	34486	12s
256x256	16777216	16476192	301024	71s

Tabela 7.5: Resultado da aplicação do algoritmo EEI na reconstrução do dado II

tamanho das imagens	tempo
64x64	3s
128x128	6s
256x256	13s

Tabela 7.6: Resultado da aplicação do algoritmo EERA na reconstrução do dado II

7.6.3

Desempenho dos métodos aplicados à reconstrução do dado Tigre

tamanho das imagens	total de voxels	voxels removidos	voxels consistentes	tempo
64x64	262144	260314	1803	1s
128x128	2097152	2033893	63259	2s
256x256	16777216	16010084	767132	12s

Tabela 7.7: Resultado da aplicação do algoritmo EEI na reconstrução do dado III

tamanho das imagens	total de voxels	voxels removidos	voxels consistentes	tempo
64x64	262144	252885	9259	4s
128x128	2097152	2009781	80371	9s
256x256	16777216	16058761	718455	31s

Tabela 7.8: Resultado da aplicação do algoritmo EEI na reconstrução do dado III

tamanho das imagens	tempo
64x64	6s
128x128	11s
256x256	22s

Tabela 7.9: Resultado da aplicação do algoritmo EEI na reconstrução do dado III

7.6.4

Desempenho dos métodos aplicados à reconstrução do dado Dino

tamanho das imagens	total de voxels	voxels removidos	voxels consistentes	tempo
64x64	262144	259663	2481	1s
128x128	2097152	2033893	33985	2s
256x256	16777216	16777216	342698	15s

Tabela 7.10: Resultado da aplicação do algoritmo EEI na reconstrução do dado IV

tamanho das imagens	total de voxels	voxels removidos	voxels consistentes	tempo
64x64	262144	252885	3542	4s
128x128	2097152	2061380	35772	9s
256x256	16777216	16474304	302912	34s

Tabela 7.11: Resultado da aplicação do algoritmo EEI na reconstrução do dado IV

tamanho das imagens	tempo
64x64	5s
128x128	10s
256x256	21s

Tabela 7.12: Resultado da aplicação do algoritmo EERA na reconstrução do dado IV

8

Conclusões e trabalhos futuros

8.1

Conclusões e comentários finais

Neste trabalho, investigamos alguns aspectos relacionados à reconstrução de cenas a partir de imagens através de métodos de escultura do espaço. De certa forma, o trabalho aqui apresentado, se aproxima dos trabalhos de Sainz e Prock, uma vez que se baseia na realização do processo de escultura diretamente no espaço da cena com base no registro de informações através da técnica de mapeamento projetivo de texturas. Entretanto, diferentemente destes trabalhos, justificamos o uso de tal técnica não somente para simplificação da etapa de registro e otimização do processo de escultura, mas também como uma forma de lidar apropriadamente com os efeitos causados por aliasing durante a amostragem das informações necessárias a avaliação da foto-consistência dos voxels que compõem o espaço de reconstrução.

Além disso, procuramos tratar todos os subproblemas pertinentes à escultura do espaço no espaço da cena exclusivamente através de mapeamento de textura projetivo, incluindo tanto o tratamento da visibilidade quanto o da oclusão parcial causada pelos elementos determinados como foto-consistente nos estágios intermediários do processo de reconstrução. Sainz por exemplo, trata do problema de oclusão através de uma técnica bastante distinta, baseando-se na determinação da sombra gerada pelos elementos detectados como foto-consistentes sobre o restante do volume a ser esculpido).

Procuramos também explorar o uso do mapeamento de textura projetivo como forma de registro de informações relevantes em um nível não encontrado nos trabalhos anteriores. Por exemplo, propusemos a avaliação da foto-consistência dos voxels através de um critério estatístico que permite a adoção de limiares específicos para cada pixel das imagens de entrada, os quais, por sua vez são estimados com base no ruído introduzido por cada

um dos sensores das câmeras. Tais limiares são inseridos em uma textura, juntamente com as informações de oclusão para que posteriormente possam ser registrados e utilizados durante a avaliação da foto-consistência.

A principal inovação deste trabalho está, entretanto, no desenvolvimento de um método de escultura do espaço capaz de recuperar a forma e os atributos fotométricos de uma cena através de um processo de refinamento adaptativo. É verdade que Prock também propôs um método capaz de reconstruir uma cena de forma progressiva. Contudo, ele não o faz de forma adaptativa. Além disso, o método por ele proposto pode implicar em perdas de partes da cena caso o volume de reconstrução inicial não tenha resolução suficiente para capturar a maior parte das feições importantes. Através da adoção de uma estrutura espacial adaptativa e de critérios de subdivisão apropriados, conseguimos não só efetuar o processo de forma adaptativa como também garantimos que não haverá perdas causadas pelo processo por refinamento independentemente da resolução do volume de reconstrução inicial.

Finalmente, como resultado final dos esforços, desenvolvemos um sistema de reconstrução de objetos a partir de imagens que utiliza as técnicas de reconstrução aqui apresentadas.

Nas seções abaixo, efetuamos uma pequena digressão sobre cada um dos aspectos abordados no presente trabalho, enumerando as principais conclusões.

8.1.1

Processamento no espaço da cena

O processamento no espaço da cena, baseado na técnica de mapeamento projetivo de textura, não só possibilitou o tratamento correto e eficiente do problema de registro das informações provenientes de cada uma das câmeras, com também foi crucial para que o método pudesse ser aplicado sob condições adversas, em que as imagens de entrada eram de baixa resolução e foram bastante corrompidas por ruído proveniente dos sensores. Graças às operações de filtragem existentes nos procedimentos de mapeamento projetivo das placas gráficas atuais, foi possível realizar apropriadamente a amostragem das informações necessárias à etapa de determinação da foto-consistência, reduzindo consideravelmente os problemas de aliasing existentes nos métodos que efetuam o processamento no espaço das imagens. Além disso, através da técnica de mapeamento projetivo de texturas, nos foi possível lidar de forma adequada com o problema de gerenciamento das

informações sobre a visibilidade dos elementos do espaço de reconstrução, quer sejam voxels, quer sejam células de uma estrutura espacial hierárquica.

Apesar das vantagens proporcionadas, o processamento no espaço da cena também introduz algumas dificuldades inexistentes no processamento no espaço das imagens. Primeiramente, toda e qualquer informação necessária à realização do processo de escultura do espaço deve ser registrada sobre planos de referência através de mapeamento projetivo, o que nos obriga a codificá-las através de mapas de textura. Em nosso caso, o uso de mais informações codificadas através de mapas de textura implica em um maior número de cópias do framebuffer para memória convencional, o que pode chegar a ser proibitivo. Entretanto, cremos que através do cálculo da foto-consistência diretamente na placa gráfica tal problema será solucionado, pois não precisaremos efetuar tais cópias.

Um outro problema é o de que a filtragem das imagens, apesar de necessária, altera os valores dos pixels nas regiões próximas às transições entre regiões que representam algum tipo de classificação como, por exemplo, no caso dos mapas de segmentação e de visibilidade. Este efeito faz com que as fronteiras entre tais regiões se tornem imprecisas o que pode levar à introdução de artefatos indesejáveis na cena reconstruída. A simples desabilitação da operação de filtragem não resolve o problema pois nestes casos, as fronteiras podem ser afetadas significativamente por aliasing o que introduz artefatos muito piores.

Por este motivo, é fundamental que tais dados, antes de serem utilizados, sejam reclassificados através de funções capazes de remapear os novos valores nos valores de classificação originais, o que por sua vez, requer a escolha de funções de classificação adequadas. Estas funções normalmente são bastante simples; no entanto, requerem a escolha de limiares de separação apropriados, caso contrário a classificação original pode ser corrompida. Não conhecemos no momento uma estratégia melhor para lidar com este problema, já que não é possível, sem o auxílio de programação em placa, realizar uma filtragem seletiva, isto é, que não afete determinado grupos de pixels com certas características, sem alterar os demais.

8.1.2

Uso das informações sobre segmentação dos objetos de interesse nas imagens

O uso das informações de segmentação foi fundamental para que obtivéssemos sucesso nas reconstruções a partir das imagens obtidas por

nosso sistema de aquisição. Em primeiro lugar, através desta estratégia nos foi possível obter resultados razoáveis através de pouquíssimas imagens o que seria muito difícil se nos baseássemos somente em informações fotométricas.

Por outro lado, o uso de informações de segmentação nos obrigou a desenvolver estratégias capazes de lidar com o problema de segmentação de modo eficaz, pois a segmentação errônea dos objetos de interesse pode levar à obtenção de reconstruções incorretas. Por exemplo, podemos perder partes do objeto devido à segmentação não apropriada ou então podemos incluir parte do fundo da cena como objeto de interesse o que nos obriga a usar limiares para o teste de foto-consistência extremamente rígidos para que tais partes sejam removidas, o qual pode ser problemático principalmente quando o fundo da cena não é muito distinto das cores do objeto que desejamos reconstruir.

8.1.3

Estratégias de escultura do espaço por refinamento adaptativo

Concluimos também que os métodos de escultura do espaço convencionais realizam um número de operações de avaliação de foto-consistência desnecessário, principalmente quando o objeto de interesse ocupa uma porção restrita do espaço de reconstrução. Por este motivo, um dos focos deste trabalho foi desenvolver um método adaptativo que fosse capaz de concentrar esforços nas regiões do espaço próximas à região que contém a cena.

A solução que foi apresentada é baseada em um processo de refinamento no qual a estrutura da cena é descoberta progressivamente à medida em que o processo de reconstrução é realizado. Para que pudéssemos realizar o processo de reconstrução através de refinamento adaptativo foi fundamental a utilização das informações sobre a segmentação dos objetos nas imagens já que estas são as únicas informações que nos garantem de forma simples e segura a remoção de espaços vazios de grandes dimensões. Obviamente é possível também descartar grandes espaços extensos somente através das informações fotométricas, entretanto isto requer uma análise mais profunda e complexa sobre a natureza da variabilidade existente nas regiões das imagens registradas sobre o espaço da cena.

Uma questão crítica, relacionada à estratégia por refinamento adaptativo, é a forma como as informações provenientes das câmeras são amostradas durante a etapa de registro e avaliação da foto-consistência. Nestes casos, como as imagens são projetadas no espaço da cena em níveis de resolução de-

pendentes do grau de refinamento foi fundamental a utilização de operações de filtragem baseadas em mipmaps.

Os resultados demonstraram que a estratégia proposta é bastante eficiente quando comparada à estratégia convencional que se baseia em um processamento exaustivo sobre o espaço de reconstrução. Além disso, acreditamos que tal abordagem ainda pode se tornar muito mais eficiente, caso o teste para determinação da foto-consistência seja implementado diretamente em hardware gráfico. Desta forma, poderemos não só efetuar as operações de maneira muito mais rápida, como também evitaremos o atraso causado pelas cópias das informações da memória de vídeo para a memória convencional, operação está que corresponde à maior parte do tempo computacional gasto durante o processo de escultura do espaço.

8.1.4

Construção de sistemas de reconstrução de baixo custo

Finalmente, concluímos que através das técnicas apresentadas neste trabalho, é possível construir sistemas de reconstrução 3D de baixo custo, capazes de produzir resultados aceitáveis nos casos em que não é necessário uma grande precisão. Além disso, por serem de baixo custo, são facilmente escaláveis, como, por exemplo, através do acréscimo de um maior número de câmeras. Nada impede também que câmeras de alta resolução sejam usadas, o que aumentaria significativamente a qualidade das reconstruções obtidas. Resumindo, acreditamos que os resultados obtidos indicam que os métodos de escultura do espaço são bastante promissores e que podem futuramente ser utilizados extensivamente em aplicações que envolvem a aquisição de forma de cenas com qualidade fotométrica.

8.2

Trabalhos futuros

Neste trabalho apresentamos algumas contribuições aos métodos de reconstrução de cenas a partir de imagens através de escultura do espaço. Entretanto, é evidente que esta linha de pesquisa está longe de ser esgotada pois restam ainda muitos tópicos a serem investigados. Abaixo citamos alguns possíveis temas para investigação que podem ser seguidos com base no presente trabalho.

8.2.1

Programação em placas gráficas

Um dos primeiros tópicos a ser investigado futuramente é a verificação do ganho em eficiência que pode ser obtido através do cálculo da estimativa utilizada na etapa de determinação da foto-consistência, diretamente nas placas gráficas através de programação sobre fragmentos. O uso desta técnica elimina a necessidade de cópia do conteúdo do framebuffer para memória convencional o que, por sua vez, pode tornar os algoritmos que trabalham no espaço da cena muito mais eficientes.

Com o surgimento das novas placas gráficas acreditamos que a programação das estatísticas seja bem mais simples que a que pode ser realizada atualmente através da customização de registradores e combinadores. Acreditamos também que as linguagens utilizadas para a programação sobre fragmentos tenham alcançado um grau de abstração razoável tornando simples o processo de implementação.

8.2.2

Investigação de outras estruturas de dados espaciais

Em nosso trabalho, descrevemos o espaço de reconstrução da cena através de um esquema de subdivisão espacial, representado através de uma octree. Apesar dos resultados satisfatórios, acreditamos que outras estruturas de dados espaciais possam permitir uma maior adaptabilidade às características da cena, reduzindo ainda mais o número de elementos individuais a serem analisados.

Em especial, acreditamos que as árvores de partição binária (*bintrees*) sejam capazes de fornecer resultados superiores aos obtidos atualmente. Uma vantagem que pode ser obtida através desta estrutura é a de que os planos de subdivisão podem ser selecionados de forma arbitrária permitindo assim uma maior adaptabilidade à geometria da cena.

8.2.3

Uso de câmeras não-fixas

Nosso sistema de reconstrução é composto por um conjunto de câmeras fixas montadas sobre um cubo, as quais são calibradas com base em um padrão de calibração fixo no espaço da cena. Este tipo de configuração impõe sérias restrições ao número de imagens que podem ser utilizadas no

processo de reconstrução, pois o acréscimo de imagens requer a adição de novas câmeras no sistema de aquisição.

Um trabalho futuro consistiria em adaptar o sistema existente para que possa lidar com câmeras reposicionáveis, o que é viável já que o processo de calibração utilizado é capaz de calibrar câmeras que se movem suavemente e funciona mesmo quando o padrão está apenas parcialmente visível.

8.2.4

Comparação entre métodos para o registro das informações

O registro das informações no espaço da cena, necessários para a determinação de foto-consistência, também pode ser realizado em memória convencional através de técnicas de *warping projetivo* em vários passos.

Um trabalho futuro consiste em comparar a eficiência entre o processo de registro através do mapeamento projetivo de textura por hardware e o processo que envolve uma operação de *warping* por software.

8.2.5

Reconstrução em tempo real

Em algumas aplicações é essencial que o processo de reconstrução seja realizado em tempo real como, por exemplo, em aplicações de telepresença e imersão em ambientes virtuais. Nestes casos é fundamental que os algoritmos utilizados sejam os mais rápidos possíveis.

Para alcançar este objetivo é necessário a utilização de diversas técnicas em conjunto, algumas delas abordadas neste trabalho, e outras que foram apenas mencionadas nesta seção sobre trabalhos futuros.

Primeiramente, é fundamental que o processo de registro e determinação de foto-consistência seja realizado diretamente em hardware gráfico. Isto eliminará retardos introduzidos pela cópia de framebuffer além de liberar a CPU para a realização de outras tarefas. Um segundo aspecto que talvez seja essencial é a utilização de paralelismo. Podemos facilmente observar que os métodos de escultura do espaço são essencialmente paralelizáveis, pois várias etapas podem ser efetuadas de forma independente para cada uma das imagens, como, por exemplo a etapa de segmentação e a etapa de registro das informações no plano de referência. Um terceiro aspecto que deve ser levado em consideração, e que de certa forma já está sendo explorado, é o uso de estruturas de subdivisão espacial que nos permita aproveitar

não só a coerência espacial existente na cena, mas também a coerência temporal. A primeira, na verdade já é utilizada no método que apresentamos. O uso da coerência temporal, por sua vez, é fundamental em qualquer método que se proponha a realizar tarefas em tempo real.

8.2.6

Refinamento do teste probabilístico e busca por novos modelos

No capítulo 5 deste trabalho, descrevemos um teste probabilístico para determinação da foto-consistência dos elementos do espaço de reconstrução que leva em consideração erros locais nas observações provenientes de ruído produzido pelas câmeras. Vimos também, através dos resultados, que tal teste não se apresentou totalmente satisfatório, pois, a princípio, não leva em consideração outras fontes de erro, que estão presentes na reconstrução de cenas reais. A extensão deste teste para que possa lidar com fontes de erro mais gerais é naturalmente um tema que deve ser pesquisado em trabalhos de pesquisa futuros. Ademais, possíveis interações com os trabalhos de de Broadhurst [71, 72] e Bhotika [73], os quais procuram modelar o processo de reconstrução através de formalismos probabilísticos, devem ser cuidadosamente investigadas.

8.2.7

Uso de estimativa de superfície na determinação da foto-consistência

Neste trabalho, abordamos métodos que avaliam a foto-consistência de regiões do espaço, quer sejam voxels ou células de uma octree, através de amostras em uma região planar. Apesar de produzir bons resultados, esta abordagem limita consideravelmente a ação dos métodos de escultura do espaço.

Um aspecto que pretendemos investigar é a possibilidade de avaliarmos a foto-consistência em superfícies de referência mais genéricas, as quais seriam provenientes de uma estimativa da superfície da cena. De fato, o método que pretendemos investigar receberia como entrada uma subdivisão do espaço representando a envoltória visual, isto é, uma estimativa da superfície da cena a partir das suas silhuetas nas imagens. Tal subdivisão espacial seria triangulada em seus diversos níveis de resolução através de um processo conhecido como triangulação dual [78]. Desta forma, para cada célula, teríamos uma estimativa do fragmento de superfície da cena que passa em seu interior.

Com base nesta estrutura calcularíamos a envoltória fotométrica (*photo hull*) através da avaliação da foto-consistência de cada um de seus elementos com base nas superfícies estimadas. O processo seria aplicado primeiramente sobre as células dos níveis de menor resolução e posteriormente nas de maior resolução. Células inconsistentes seriam removidas do modelo, e em seguida, faríamos um ajuste na estimativa da superfície da cena. As células consideradas indefinidas seriam subdividas, tendo também suas superfícies locais reajustadas. Já as células foto-consistentes teriam seus fragmentos de superfície colorizados de acordo com uma combinação das porções das imagens sobre eles projetados. O resultado da aplicação deste método seria uma reconstrução poligonal texturizada do objeto em múltiplas resoluções.

Bibliografia

- [1] [Http://www.opengl.org](http://www.opengl.org).
- [2] [Http://www.nvidia.com](http://www.nvidia.com).
- [3] [Http://www.nvidia.com](http://www.nvidia.com).
- [4] [Http://www.lua.org](http://www.lua.org).
- [5] [Http://www.intel.com/research/mrl/research/opencv/index.htm](http://www.intel.com/research/mrl/research/opencv/index.htm).
- [6] JULESZ, B.. **Binocular depth perception of computer-generated patterns**. The Bell Systems Technical Journal 39, p. 1125–1162, 1961.
- [7] MARR, D.; POGGIO, T.. **Cooperative computation of stereo disparity**. Science, 194(4262):283– 287, October 1976.
- [8] MARR, D.; POGGIO, T.. **A theory of human stereo vision**. Technical Report 451, Massachusetts Institute of Technology. Artificial Intelligence Laboratory, November 1977. AI Memo.
- [9] BAKER, H.; BINFORD, T. O.. **Depth from edge and intensity based stereo**. In: INT. JOINT CONF. ON ARTIFICIAL INTELLIGENCE, p. 832–835, Milan, Italy, 1981.
- [10] GRIMSON, W. E. L.. **From Images to Surfaces**. MIT Press, Cambridge, Mass., 1981.
- [11] MARTIN, W.; AGGARWAL, J. K.. **Volumetric descriptions of objects from multiple views**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 5(2):150–158, March 1983.
- [12] CHIEN, C. H.; AGGARWAL, J. K.. **A Volume/Surface representation**. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, p. 817–820, Montreal, Canada, July 30 Aug. 2 1984.

- [13] CROW, F. C.. **Summed-area tables for texture mapping.** In: SIGGRAPH'84 PROCEEDINGS, VOL. 18, NO.3, p. 207–212. Computer Graphics, 1984.
- [14] POLLARD, S.; MAYHEW, J. E. W. ; FRISBY, J.. **PMF: A stereo correspondence algorithm using disparity gradient limit.** Perception, 14:449–470, 1985.
- [15] OHTA, Y.; KANADE, T.. **Stereo by intra- and inter-scanline using dynamic programming.** IEEE Trans. Pattern Analysis and Machine Intelligence, 7(2):139–154, 1985.
- [16] MASSONE, L.; MORASSO, P. ; ZACCARIA, R.. **Shape from occluding contours.** In: SPIE CONFERENCE ON INTELLIGENT ROBOTS AND COMPUTER VISION, SPIE VOL. 521, p. 114–120, November 1985.
- [17] BARNARD, S. T.. **A stochastic approach to stereo vision.** In: 5TH NATIONAL CONF. ON AI, p. 676–680, Philadelphia, 1986.
- [18] CHIEN, C. H.; AGGARWAL, J. K.. **Volume/Surface octrees for the representation of three-dimensional objects.** In: COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING, VOL. 36, NO., p. 100–113, October 1986.
- [19] BESL, P. J.. **Surfaces in Early Range Image Understanding.** PhD thesis, University of Michigan, 1986.
- [20] VEENSTRA, J.; AHUJA, N.. **Efficient octree generation from silhouettes.** In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 537–542, Miami Beach, Florida, June 1986.
- [21] HECKBERT, P. S.. **Survey of texture mapping.** In: IEEE COMPUTER GRAPHICS AND APPLICATIONS 6, 11, p. 56–67, November 1986.
- [22] TSAI, R.. **An efficient and accurate camera calibration technique for 3d machine vision.** In: PROCEEDINGS OF IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 364–374, Miami Beach, FL, 1986.

- [23] POTMESIL, M.. **Generating octree models of 3d objects from their silhouettes in a sequence of images.** , Computer Vision, Graphics and Image Processing, 40(1):1–29, October 1987.
- [24] BESL, P. J.. **Surfaces in Range Image Understanding.** Springer-Verlag, 1988.
- [25] OSHER, S.; SETHIAN, J.. **Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations.** Journal of Computational Physics, 79:12– 49, 1988.
- [26] HECKBERT, P. S.. **Fundamentals of texture mapping and image warping.** Master’s thesis, Dept. of Eletrical Engineering and Computer Science. University of California, Berkeley, CA 94720, June 17 1989.
- [27] SRIVASTAVA, S.; AHUJA, N.. **Octree generation from object silhouettes in perspective views.** Computer Vision, Graphics and Image Processing, 49(1):68 – 84, January 1990.
- [28] SEGAL, M.; KOROBKIN, C.; WIDENFLET, R. V.; FORAN, J. ; HAEBERLI, P.. **Fast shadows and lightning effects using texture mapping.** In: PROCEEDINGS OF SIGGRAPH’92, Chicago, USA., July 26-31 1992.
- [29] SZELISKI, R.. **Rapid octree construction from image sequences.** Computer Vision, Graphics and Image Processing: Image Understanding, 58(1):23–32, July 1993.
- [30] LARSON, R. D.; SHAH, M. S.. **Method for generating addresses to textured graphics primitives stored in RIP maps,** 1993. US Patent 05222205.
- [31] LAURENTINI, A.. **The visual hull concept for silhouette-based image understanding.** IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(2), February 1994.
- [32] FROMHERZ, T.; BICHSEL, M.. **Shape from contours as initial step in shape from multiple cues.** In: ISPRS COMMISSION III SYMPOSIUM ON SPATIAL INFORMATION FROM DIGITAL PHOTOGRAMMETRY AND COMPUTER VISION, p. 240–256, Munich, Germany, 1994.

- [33] KANG, S. B.; WEBB, J. A.; ZITNICK, C. L. ; KANADE, T.. **A multibaseline stereo system with active illumination and real-time image acquisition.** In: IEEE INT. CONF. ON COMPUTER VISION (ICCV), p. 88–93, June 1995.
- [34] FROMHERZ, T.; BICHSEL, M.. **Shape from multiple cues: Integrating local brightness information.** In: FOURTH INTERNATIONAL CONFERENCE FOR YOUNG COMPUTER SCIENTIST, ICYCS 95, p. 855–862, Beijing, P. R. China, 1995.
- [35] FAUGERAS, O.. **Three-Dimensional Computer Vision, A Geometrical Viewpoint.** MIT Press, 1996.
- [36] LANE, R. A.; N.A.THACKER. **Stereo vision research: An algorithm survey.** January 29 1996.
- [37] CURLESS, B.; LEVOY, M.. **A volumetric method for building complex models from range images.** In: SIGGRAPH 96 CONFERENCE PROCEEDINGS, p. 303–312. ACM SIGGRAPH, Addison Wesley, August 1996.
- [38] MOEZZI, S.; KATKERE, A.; KURAMURA, D. ; JAIN, R.. **Reality modeling and visualization from multiple video sequences.** IEEE Computer Graphics and Applications, 16(6):58–63, 1996.
- [39] CURLESS, B. L.. **New Methods for Surface Reconstruction from Range Images.** PhD thesis, Department of Electrical Engineering, Stanford University, 1997. Phd dissertation.
- [40] PULLI, K.; COHEN, M.; DUCHAMP, T.; HOPPE, H.; MCDONALD, J.; SHAPIRO, L. ; STUETZLE, W.. **surface modeling and display from range and color data.** In: KEYNOTE ADDRESS AT INTERNATIONAL CONFERENCE ON IMAGE ANALYSIS AND PROCESSING '97, p. 385–397. Lecture Notes in Computer Science 1310 - Springer-Verlag, 1997.
- [41] PULLI, K.; COHEN, M.; DUCHAMP, T.; HOPPE, H.; SHAPIRO, L. ; STUETZLE, W.. **View-based rendering: Visualizing real objects from scanned range and color data.** In: 8TH EUROGRAPHICS WORKSHOP ON RENDERING, June 1997.
- [42] PULLI, K.; DUCHAMP, T.; HOPPE, H.; MCDONALD, J.; SHAPIRO, L. ; STUETZLE, W.. **Robust meshes from multiple**

- range maps.** In: INT. CONF. ON RECENT ADVANCES IN 3-D DIGITAL IMAGING AND MODELING, p. 205–211, May 1997.
- [43] PULLI, K.. **Surface Reconstruction and Display from Range and Color Data.** PhD thesis, University of Washington, 1997. A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.
- [44] SEITZ, S.; DYER, C.. **Photorealistic scene reconstruction by voxel coloring.** In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 1067 – 1073, June 1997.
- [45] MOEZZI, S.; TAI, L. ; GERARD, P.. **Virtual view generation for 3d digital video.** IEEE Multimedia, 4(1):18–26, January - March 1997.
- [46] GOMES, J.; VELHO, L.. **Image Processing for Computer Graphics.** Springer-Verlag, New York, 1997.
- [47] ROY, S.; COX, I.. **A maximum-flow formulation of the n-camera stereo correspondence problem.** In: INT. CONF. ON COMPUTER VISION, ICCV'98, Bombay, India, 1998.
- [48] ISHIKAWA, H.; GEIGER, D.. **Oclusions, discontinuities, and epipolar lines in stereo.** In: FIFTH EUROPEAN CONFERENCE ON COMPUTER VISION, (ECCV'98), Freiburg, Germany, 2-6 June 1998.
- [49] ISHIKAWA, H.; GEIGER, D.. **Segmentation by grouping junctions.** In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 1998.
- [50] KUTULAKOS, K. N.; SEITZ, S. M.. **What do n photographs tell us about 3d shape?** Technical Report 680, Computer Science Dept. U. Rochester, January 1998.
- [51] SHADE, J.; GORTLER, S.; HE, L. ; SZELISKI, R.. **Layered depth images.** In: PROCEEDINGS OF A.C.M. SIGGRAPH, p. 231– 242, 1998.
- [52] PROCK, A.; DYER, C.. **Towards real-time voxel coloring.** In: DARPA IMAGE UNDERSTANDING WORKSHOP, p. 315–321, 1998.

- [53] FAUGERAS, O.; KERIVEN, R.. **Variational principles, surface evolution, PDE's, level set methods, and the stereo problem.** In: IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 7, NO., p. 336–344., March 1998.
- [54] MALLAT, S.. **A Wavelet Tour of Signal Processing.** Academic Press, 1998.
- [55] EISERT, P.; STEINBACH, E. ; GIROD, B.. **Multi-hypothesis, volumetric reconstruction of 3-d objects from multiple calibrated camera views.** In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, p. 3509–3512., 1999.
- [56] SEITZ, S.; DYER, C.. **Photorealistic scene reconstruction by voxel coloring.** International Journal of Computer Vision, 35(2):151–173, 1999.
- [57] CULBERTSON, W. B.; MALZBENDER, T. ; SLABAUGH, G.. **Generalized voxel coloring.** In: ICCV WORKSHOP, VISION ALGORITHMS THEORY AND PRACTICE, p. 100–115. Springer-Verlag Lecture Notes in Computer Science 1883, September 1999.
- [58] BONET, J. D.; VIOLA, P.. **Roxels: Responsibility weighted 3d volume reconstruction.** In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, VOL1, p. 415 – 425. IEEE, 1999.
- [59] SAITO, H.; KANADE, T.. **Shape reconstruction in projective grid space from large number of images.** In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION - VOLUME2, p. 49–54, June 23-25 1999.
- [60] KIMURA, M.; SAITO, H. ; KANADE, T.. **3d voxel construction based on epipolar geometry.** In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, p. 135–139, 1999.
- [61] SETHIAN, J.. , **Level Set Methods and Fast Marching Methods.** Cambridge University Press, second edition, 1999.
- [62] HARTLEY, R.; ZISSERMAN, A.. **Multiple View Geometry.** Cambridge University Press, 2000.
- [63] KUTULAKOS, K. N.; SEITZ, S. M.. **A theory of shape by space carving.** International Journal of Computer Vision, 38(3):199–218, July 2000.

- [64] SLABAUGH, G.; MALZBENDER, T. ; CULBERTSON, W. B. , **volumetric warping for voxel coloring on an infinite domain**. In: PROCEEDINGS OF THE WORKSHOP ON 3D STRUCTURE FROM MULTIPLE IMAGES FOR LARGE-SCALE ENVIRONMENTS (SMILE), p. 41–50, July 2000.
- [65] KUTULAKOS, K. N.. **Approximate n-view stereo**. In: EUROPEAN CONFERENCE ON COMPUTER VISION, p. 67–83. Springer Lecture Notes in Computer Science 1842, June/July 2000. Volume1.
- [66] BROADHURST, A.; CIPOLLA, R.. **A statistical consistency check for the space carving algorithm**. In: Mirmehdi, M.; B.Thomas, editors, 11TH BRITISH MACHINE VISION CONFERENCE, VOLUME1, p. 282–291, Bristol, September 2000.
- [67] SLABAUGH, G.; CULBERTSON, W. B.; MALZBENDER, T. ; SCHAFFER, R.. **Improved voxel coloring via volumetric optimization**. Technical Report 3, Center for Signal and Image Processing, Georgia Institute of Technology, 2000.
- [68] VEDULA, S.; BAKER, S.; SEITZ, S. ; KANADE, T.. **Shape and motion carving in 6d**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 592– 598, 2000. Volume2.
- [69] STEINBACH, E.; GIROD, B.; EISERT, P. ; BETZ, A.. **3-d object reconstruction using spatially extended voxels and multi-hypothesis voxel coloring**. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, p. 774–777, 2000. Volume 1.
- [70] STEINBACH, E.; GIROD, B.; EISERT, P. ; BETZ, A.. **3-d reconstruction of real-world objects using extended voxels**. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, p. 138–141, 2000. Volume 3.
- [71] BROADHURST, A.; DRUMMOND, T. ; CIPOLLA, R.. **A probabilistic framework for the space carving algorithm**. In: 8TH INTERNATIONAL CONFERENCE OF COMPUTER VISION, p. 388–393, Vancouver, Canada, July 2001. IEEE Computer Society Press.
- [72] BROADHURST, A.. **A Probabilistic Framework for Space Carving**. PhD thesis, University of Cambridge, Trinity College, September

2001. Dissertation submitted to the University of Cambridge for the degree of Doctor of Philosophy.

- [73] BHOTIKA, R.; FLEET, D. J. ; KUTULAKOS, K. N.. **A probabilistic theory of occupancy and emptiness**. Technical Report 753, University of Rochester, Department of Computer Science, 2001.
- [74] SLABAUGH, G.; CULBERTSON, W. B.; MALZBENDER, T. ; SCHAFFER, R.. **A survey of methods for volumetric scene reconstruction from photographs**. In: INTERNATIONAL WORKSHOP ON VOLUME GRAPHICS, 2001.
- [75] SZENBERG, F.; GATTASS, M. ; P.C.P.CARVALHO. **Automatic camera calibration for image sequences of a football match**. In: ICAPR 2001, p. 301–303, 2001.
- [76] KOLMOGOROV, V.; ZABIH, R.. **Multi-camera scene reconstruction via graph cuts**. In: EUROPEAN CONFERENCE ON COMPUTER VISION, May 2002.
- [77] SAINZ, M.; BAGHERZADEH, N. ; SUSIN, A.. **Hardware accelerated voxel carving**. In: 1ST IBERO-AMERICAN SYMPOSIUM ON COMPUTER GRAPHICS, p. 289–297, Guimarães, Portugal, July 1-5 2002.
- [78] JU, T.; LOSASSO, F.; SCHAEFER, S. ; WARREN, J.. **Dual contouring of hermite data**. In: ACM SIGGRAPH 2002, p. 339–346, 2002.
- [79] HART, E.; MITCHELL, J. L.. **Hardware shading with ext_vertex_shader and ati_fragment_shader**. Technical report, ATI Technologies, 2002.
- [80] BAKER, S.; SIM, T. ; KANADE, T.. **When is the shape of a scene unique given its light-field: A fundamental theorem of 3d vision?** IEEE Transaction on Pattern Analysis and Machine Intelligence, 25(1):100–109, January 2003.