

## 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  $\pi_k$ , no espaço de cena, é dado por um conjunto de imagens projetadas  $Ip$  onde cada imagem projetada  $Ip_i$  corresponde ao mapeamento projetivo de uma imagem  $I_i$  sobre o plano  $\pi_k$ .

Primeiramente, identificamos cada imagem  $I_i \in Ip$ , e sua respectiva câmera  $C_i$ , com um projetor 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  $Po_i$ , 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  $Ip_i$ , para cada imagem de entrada  $I_i$ , desenhando os polígonos  $Po_i$  apropriadamente texturizados, sobre o plano de referência  $\pi_k$  (Figura 4.1).

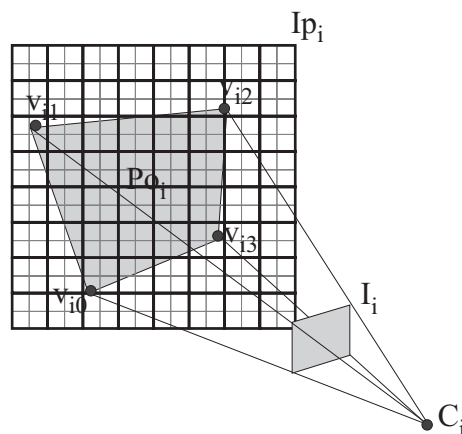


Figura 4.1: Projeção de uma imagem em um plano de referência

O desenho de  $Po_i$  sobre o plano  $\pi_k$ , 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  $Ip_i$  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,  $256 \times 256$ . 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.

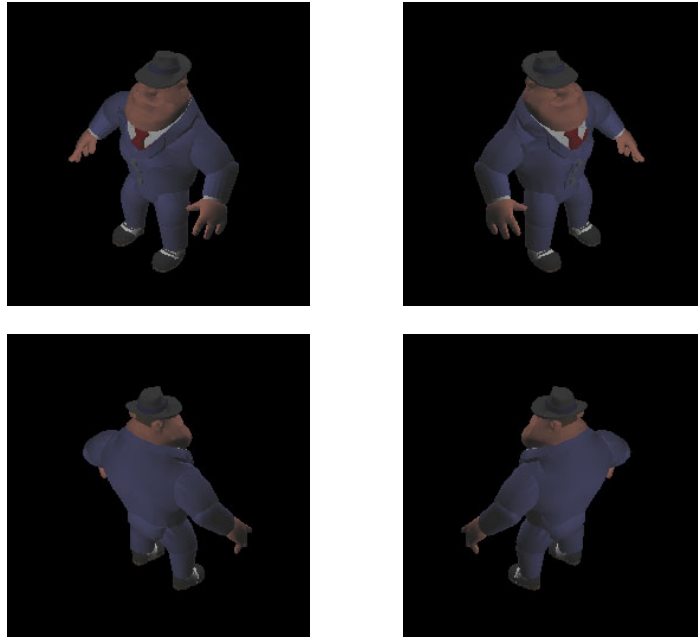


Figura 4.2: Imagens tomadas de um modelo sintético

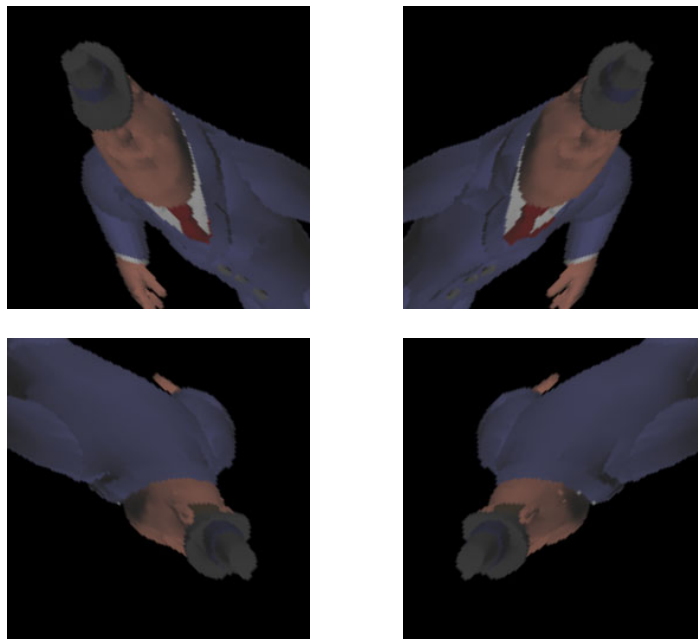


Figura 4.3: Imagens projetadas sobre um plano de referência no espaço da cena

### 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,



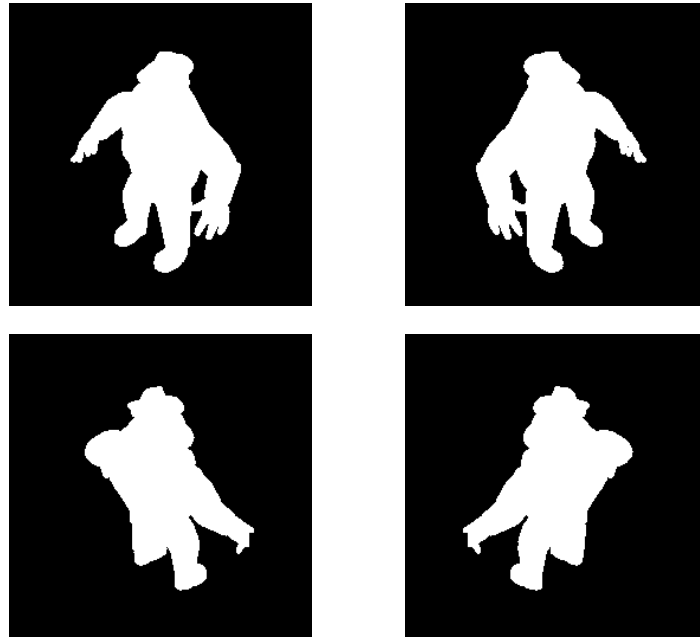


Figura 4.4: Segmentação do objeto nas imagens

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

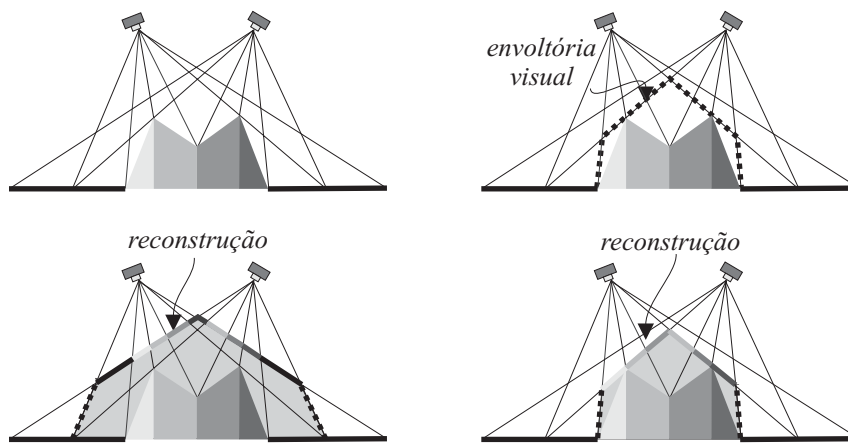


Figura 4.5: Complementariedade entre a envoltória fotométrica (photo hull) e envoltória visual (visual hull).

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.

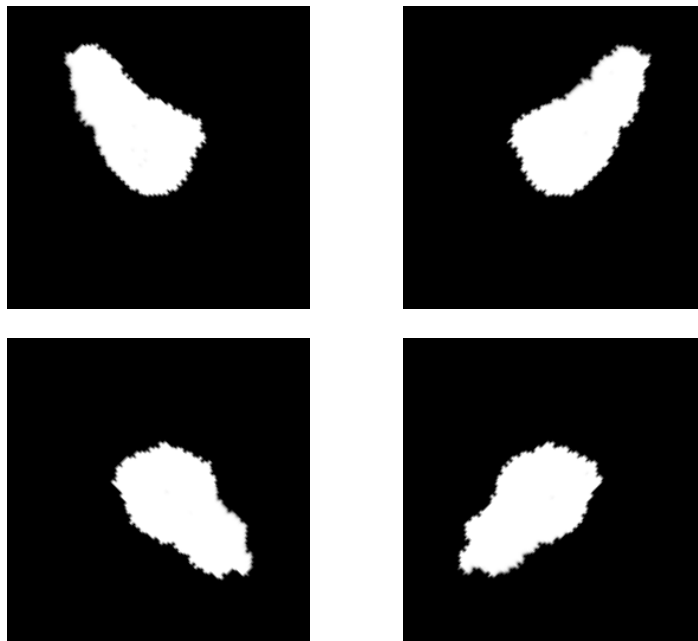


Figura 4.6: Mapas de visibilidade projetados sobre um plano de referência no espaço da cena

#### 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  $Mvp_i$ . 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 possuem 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  $\mu$  igual à cor do objeto e variância  $\sigma_i^2$ , 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  $\mu$  é considerado desconhecido, mas os valores de  $\sigma_i^2$ , 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  $\mu$  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  $\mu$  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  $\alpha$  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  $\chi^2$  é uma outra soma de distribuições  $\chi^2$ , 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**  $ev_{st} \in Ev$  **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 imagem [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( $n, I, C, \text{limiar}$ )

---

Determine um conjunto de voxels  $V$  que contenha a cena.

Limpe os mapas de visibilidade  $Mv_i$ .

**para cada** plano de referência  $\pi^k$ , que corta uma camada de voxels  $V_k$  a uma distância  $\text{dist}(k)$  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  $rp_{ijk} = v_{ijk} \cap \pi_k$ .

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

**se**  $v.fconsist = \text{CONSISTENTE}$  **ent\^ao**

Atribua a m\^edia das componentes de cor das amostras de  $rp_{ijk}$  \^as respectivas componentes de cor de  $v_{ijk}$ .

**sen\^ao**

Atribua o valor zero \^a opacidade de  $v_{ijk}$ .

**fim se**

**fim para**

Atualize cada um dos mapas de visibilidade  $Mv_i$ .

**fim para**

---

## 4.8

### An\^alise

O processamento do espa\^co da cena, fundamentado na t\^cnica de mapeamento projetivo de textura, permite um tratamento unificado e eficiente das opera\^oes de registro, reamostragem e manipula\^ao das informa\^oes necess\^arias \^a avalia\^ao da foto-consist\^encia dos elementos que comp\^oem o volume de reconstru\^ao inicial. Apesar das vantagens introduzidas por esta t\^cnica e de sua relativa efici\^encia, podemos perceber, que o mecanismo b\^sico ainda \^e o mesmo do m\^todo original, isto \^e, est\^a fundamentado em uma seq\^u\^encia de consultas efetuadas de forma exaustiva sobre um espa\^co uniformemente particionado.

O ponto chave, nesta afirma\^ao, \^e o de que os m\^todos de escultura do espa\^co que conhecemos, \^a exce\^ao do m\^todo de Prock, n\^o s\^o capazes de aproveitar, em sua totalidade, as informa\^oes sobre a organiza\^ao espacial da cena que podem ser inferidas a partir das imagens de entrada, as quais podem tornar o processo bem mais eficiente.

No cap\^itulo seguinte, iremos propor um m\^todo de escultura do espa\^co por refinamento adaptativo com base em uma subdivis\^ao hier\^arquica do espa\^co atrav\^es de uma octree e multiresolu\^ao das imagens de entrada.