

## 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.

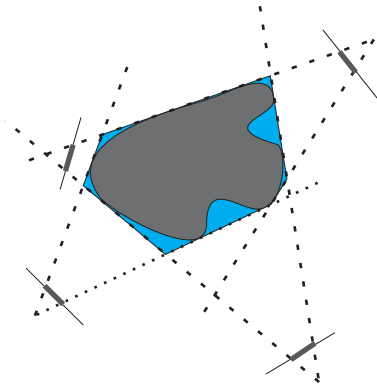


Figura 2.1: Envoltória visual inferida

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, obtêm-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 consequência 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,  $C = \{C_i, \dots, C_n\}$  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  $I_i$ .*



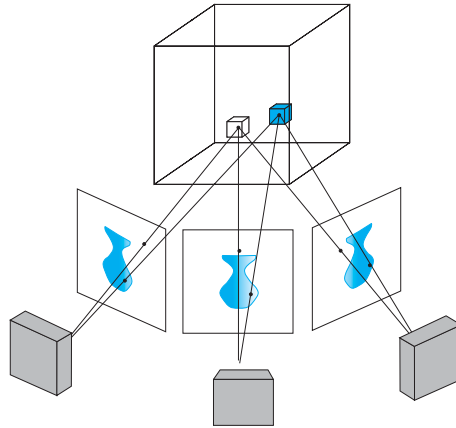


Figura 2.2: Coloração de voxels

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 reprojeção 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 invariante 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  $cor(v_p, S) = cor(S(p), S) = cor(p, I_i)$ .*

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:

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

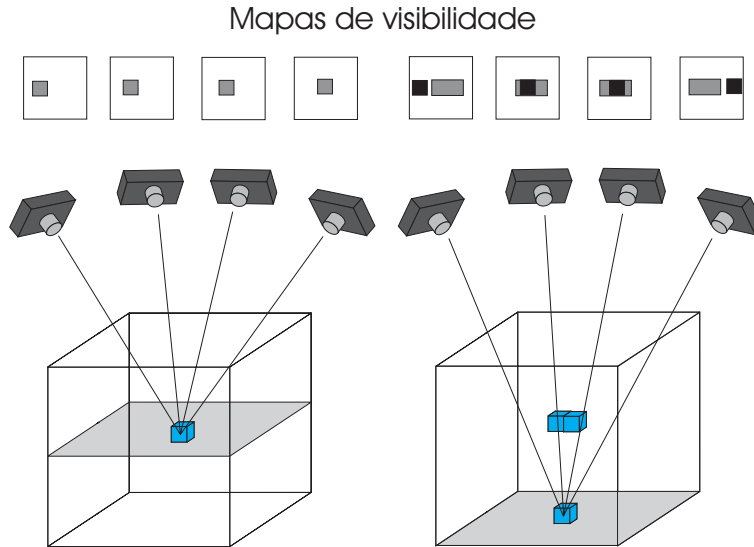


Figura 2.3: Determinação da visibilidade

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  $\sigma_0$  o desvio padrão de uma distribuição normal representando os erros introduzidos pelo sensor. Sob hipótese de consistência,  $\lambda_k(V)$  tem distribuição  $\chi^2_{n-1}$ . Se  $\lambda_k \geq T$ , onde  $T$  é um limite global calculado na distribuição  $\chi^2$  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.

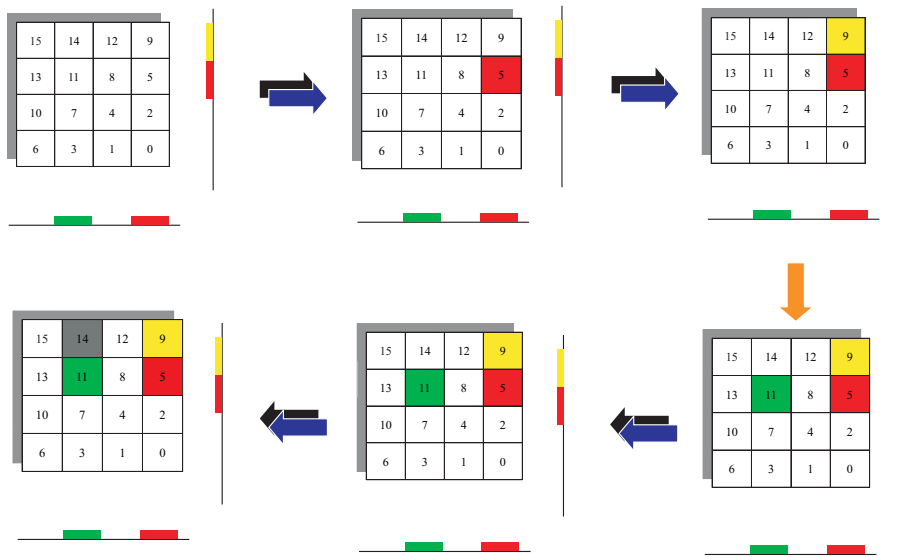


Figura 2.4: Exemplo de aplicação do método de coloração de voxels

## 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  $\xi$  a cor da luz refletida por  $p$  ao longo de  $\xi$ . 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  $rad_p(\vec{p}\vec{c})$ . 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  $\xi_1, \dots, \xi_k$  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  $\xi_i$  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  $p$ .*

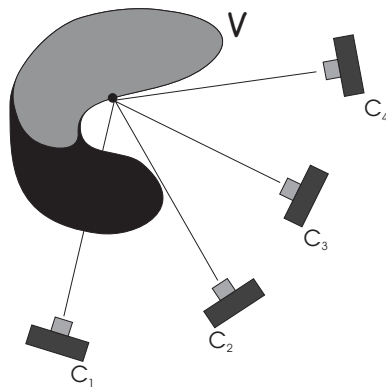


Figura 2.5: Teorema do subconjunto

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 contém  $p$  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  $\Pi$  posicionado totalmente à frente de  $\mathcal{V}$ . Move-se  $\Pi$  em direção a  $\mathcal{V}$  até que  $\Pi$  corte um subconjunto de voxels de  $\mathcal{V}$ .

Em seguida, determina-se o conjunto de voxels na interseção de  $\Pi$  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  $\Pi$  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).

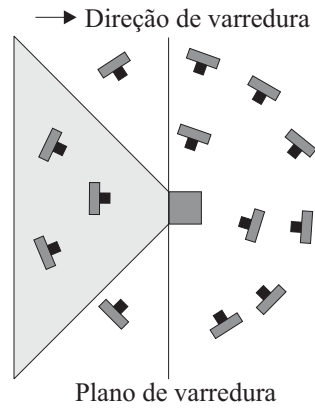


Figura 2.6: Câmeras consideradas em uma varredura para um determinado voxel.

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  $\Pi$  à frente do volume inicial  $\mathcal{V}$ .

Mova  $\Pi$  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  $\Pi$  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  $\Pi$

---

---

**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.

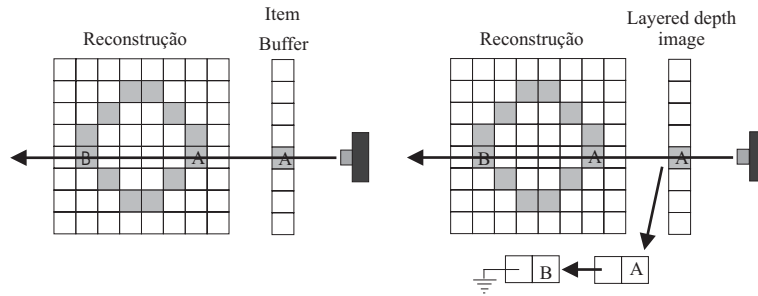


Figura 2.7: GVC-IB e GVC-LDI

### 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.

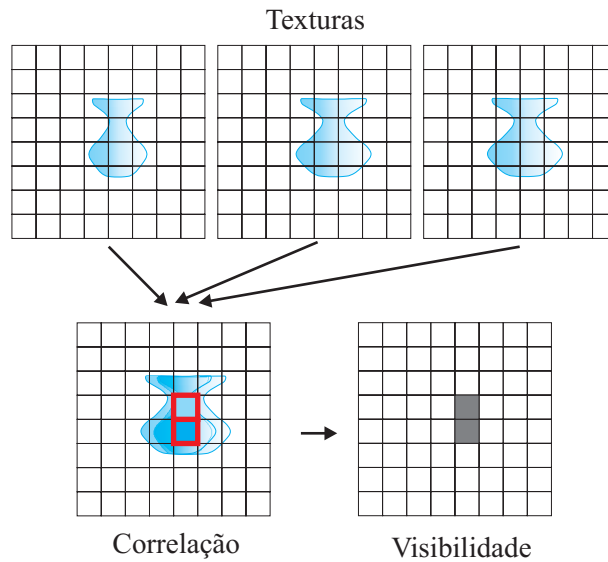


Figura 2.8: Projeção das texturas no espaço da cena

### 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.

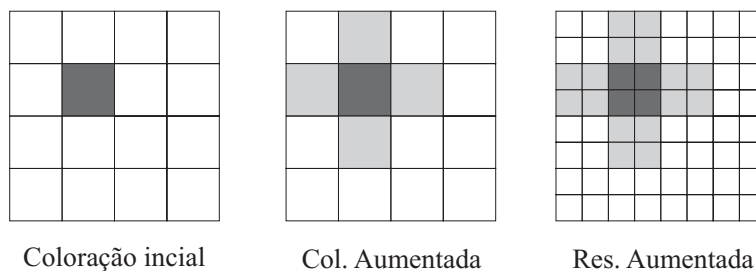


Figura 2.9: Refinamento

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 uma raio de vizinhança limitado. Além de representar transformações paramétricas, uma *shuffle transform* também é capaz de representar transformações envolvendo descarte e

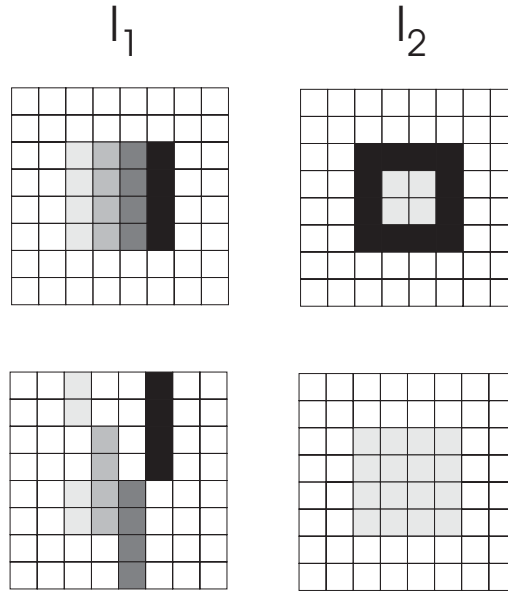


Figura 2.10: Exemplos de shuffle transforms

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