WILLIAN JEFFERSON FREITAS DA SIVA

UM MÉTODO PARA RECONSTRUÇÃO DE OBJETOS 3D COMBINANDO COLORAÇÃO DE VOXELS E ESTRUTURA BASEADA EM MOVIMENTO

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Computação Visual.

Orientador: Prof. Dr. Anselmo Antunes Montenegro

Niterói

2013

WILLIAN JEFFERSON FREITAS DA SILVA

UM MÉTODO PARA RECONSTRUÇÃO DE OBJETOS 3D COMBINANDO COLORAÇÃO DE VOXELS E ESTRUTURA BASEADA EM MOVIMENTO

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Computação Visual.

Aprovada em Julho de 2013.

BANCA EXAMINADORA

Prof. Dr. Anselmo Antunes Montenegro – Orientador
UFF

Prof. Dr. Ricardo Guerra Marroquim
COPPE-UFRJ

Prof. Dr. Marcos de Oliveira Lage Ferreira
UFF

Niterói
2013

Para minha família e amigos.

AGRADECIMENTOS

Em Primeiro lugar agradeço a Deus por te me ajudado e guiado durante todo o caminho. Sem Deus eu não poderia finalizar este trabalho.

Aos meus pais Eulalia de Freitas da Silva e Wilson da Silva, por todo amor e apoio dado. Vocês sempre estiveram presentes quando precisei e me ensinaram como não desistir. Mostraram-me que as dificuldades fazem parte de uma vida vitoriosa e que lutar por ideal nunca é em vão.

Ao Anselmo Antunes Montenegro por toda paciência e dedicação dispensada na arte do ensinar. Por sua boa orientação e disponibilidade sempre que precisei.

Para equipe do CASNAV pela grande ajuda e tempo disponibilizado para me ajudar a terminar este trabalho.

Para todos os amigos que se lembraram de mim em suas orações.

Para a Universidade Federal Fluminense, por prover um ambiente de trabalho.

Para todos que me ajudaram, obrigado.

RESUMO

Reconstrução de cenas é uma tarefa desafiadora na Computação Visual. Basicamente, o problema de reconstrução consiste em gerar modelos tridimensionais de uma cena real a partir de múltiplas fotografias. Muitas são as áreas onde a reconstrução de objetos 3D pode ser aplicada. Um exemplo são os jogos eletrônicos, robótica, realidade virtual e medicina. Geralmente, é um processo que requer a calibração de câmeras que capturaram as imagens antes da etapa de reconstrução.

Este trabalho apresenta um método para realizar a reconstrução tridimensional de cenas que combina a técnica de extração de estrutura baseada em movimento (*Structure from Motion* - SFM) com reconstrução volumétrica por *Coloração de Voxels* (*Voxel Coloring*). A abordagem utiliza as imagens adquiridas da cena a ser reconstruída para gerar uma nuvem de pontos através do SFM. A nuvem de pontos obtida é então processada para gerar uma versão inicial do volume de ocupação, na forma de uma octree. Esta estrutura, após reamostrada uniformemente gera o dado de entrada para o algoritmo de coloração de voxels o qual, por sua vez, produz a versão final do modelo em representação volumétrica.

Em comparação aos métodos existentes, a abordagem proposta não requer o uso de padrões de calibração e torna a fase de calibração de câmeras transparente. A utilização da nuvem de pontos permite uma nova forma de isolar o modelo a ser reconstruído do resto da cena, além de reduzir a quantidade de dados que o método de coloração de voxels deve processar.

Palavras-chave: Reconstrução, voxel coloring, structure from motion

ABSTRACT

Scene reconstruction is a challenging task in *Visual Computing*. It consists of generating three-dimensional models of a real scene from a set of multiple photographs. Many are the areas where it can be applied. As examples we cite games, robotics, virtual reality and medicine. It usually involves calibrating the cameras before the model can be finally reconstructed.

This dissertation presents a method for 3d object reconstruction from a set of images based on *Structure from Motion* (SFM) and *Voxel Coloring*. The approach uses the acquired images of the scene to be reconstructed, to generate a point cloud through the SFM method. The point cloud is processed to yield a preliminary version of the volume of occupancy of the scene represented as an octree. Then, the octree data structure is uniformly resampled to produce the input data for the voxel coloring algorithm which produces the final result of the reconstruction.

Compared to existing methods, the proposed approach requires no calibration fiducials and makes the step of camera calibration transparent. The use of the point cloud offers a new way to segment the model to be reconstructed from the rest of the scene and to decrease the amount of data the Voxel Coloring method has to process.

Keywords: Reconstruction, voxel coloring, structure from motion.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

Capítulo 1 – **INTRODUÇÃO**

Uma área de pesquisa fundamental na Computação Visual é a reconstrução da forma e aparência (função de atributos) de uma cena tridimensional complexa, a partir de múltiplas fotografias. Este é um problema antigo, desafiador e possui as mais variadas aplicações em áreas como jogos, realidade virtual, cartografia, arquitetura, navegação de robôs, filmes, medicina e algumas outras áreas, onde existe a preocupação em gerar novos pontos de vista virtuais de uma cena.

A pesquisa em reconstrução de cenas, possivelmente tem sua origem relacionada aos primeiros esforços para compreender o processo de percepção no sistema visual de animais superiores e seres humanos. Nesta fase inicial, entre os anos 60 e 80, pesquisadores das áreas de inteligência artificial, neurologia e psicologia, estudavam um meio de explicar, através de modelos, como o cérebro humano consegue reconstruir e interpretar as informações sobre a forma de uma cena codificada através de um par de imagens [1, 2]. Grande parte dos primeiros modelos propostos buscava descrever uma abstração da arquitetura existente nos sistemas visuais biológicos [3]. Era comum até então, modelar estes sistemas através de redes neuronais ou através de sistemas cooperativos, os quais podem ser entendidos como um caso especial de redes de autômatos celulares [4].

Com o tempo, houve uma mudança gradual do enfoque para a investigação de técnicas, as quais poderiam ser empregadas em tarefas que necessitam de algum conhecimento referente às formas existentes no espaço de trabalho.

Com o avanço tecnológico relacionado à aquisição de imagens e o surgimento de câmeras digitais de baixo custo, surge o interesse em reconstrução de cenas a partir de fotos tomadas por câmeras calibradas [3].

Os trabalhos em Computação Visual que investigam meios de reconstruir uma cena real, geralmente incluem um passo de calibração de câmera entre alguma das várias etapas que compõem os métodos associados.

A calibração de câmera é o processo pelo qual são obtidas informações sobre a os parâmetros intrínsecos e extrínsecos da câmera, isto é, parâmetros relacionados a transformação projetiva e a distorção causada pelo sistema de lentes [5] e a orientação e a posição da câmera, respectivamente. Normalmente, padrões de calibração, também conhecidos como marcas fiduciais, são utilizados com o objetivo de auxiliar nesta tarefa (veja Figura 1.1). Este é um processo que consome tempo, é normalmente tedioso e que em alguns casos se baseia na correspondência entre pontos da cena (3D) e pontos da imagem (2D). Felizmente, existem técnicas que

permitem a realização da calibração das câmeras sem a necessidade de se introduzir marcas fiduciais na cena, considerando apenas correspondências entre pontos bidimensionais em dois ou mais quadros obtidos da cena em questão. No Capítulo 3 é feito um pequeno estudo sobre um desses métodos.

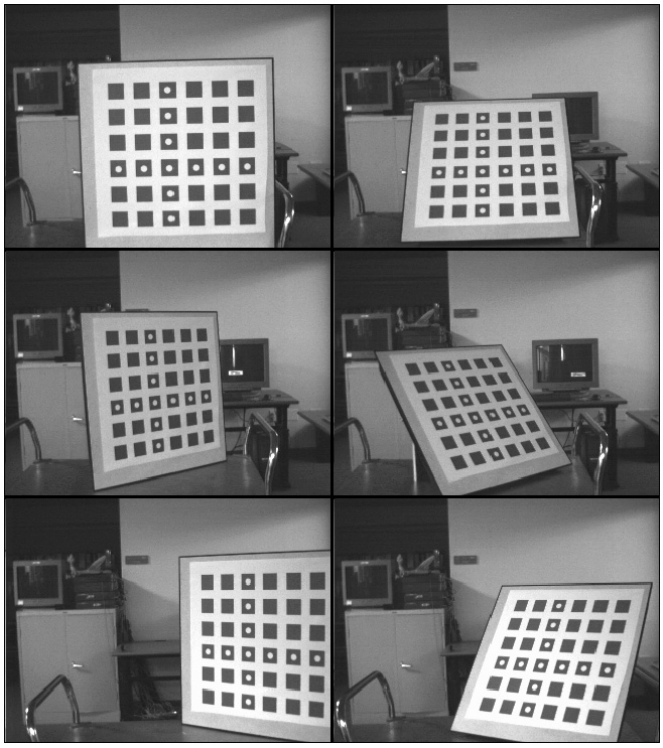


Figura 1.1- Exemplo de um padrão de calibração usado para calibrar câmeras.

Na literatura, existem formas diferentes pela qual uma cena é reconstruída. Neste trabalho a reconstrução é principalmente feita através da escultura do espaço [5,6,7,8].

A escultura do espaço (também conhecida como *carving*) consiste em um processo de manipulação do volume de ocupação da cena, de forma que ao final deste processo, um modelo volumétrico que se assemelha a cena é gerado. A manipulação do volume de ocupação é possível graças a representação da cena através de elementos conhecidos como voxels (veja Figura 1.2).

Segundo Paiva et al. [9], voxels são estruturas de dados em forma de paralelepípedos fortemente agrupados. São formados pela divisão do espaço do volume de ocupação, através de um conjunto de planos paralelos aos eixos desse espaço. Os voxels não se interceptam e tem tamanho suficientemente pequeno se comparado às características do volume de ocupação. Desta forma, neste trabalho um voxel é frequentemente tratado como uma unidade de volume.

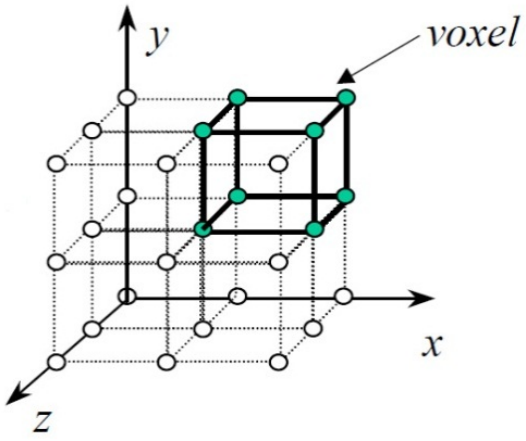


Figura 1.2- Exemplo de um voxel. Fonte: Paiva et al. [9]

Este trabalho apresenta um método simples e elegante para reconstrução de cenas a partir de uma sequência de imagens. A abordagem utiliza informações extraídas de uma sequência de quadros calibrados de um trecho de vídeo para estimar um conjunto de pontos tridimensionais pertencentes à cena. Posteriormente esse conjunto de pontos é tratado de forma que possa ser utilizado no processo de manipulação do volume de ocupação. O resultado obtido no término deste processo é a cena reconstruída.

1.1 OBJETIVO

Em métodos de reconstrução tridimensional de cenas [5,18,8,17,6], a calibração de câmera é geralmente um passo necessário, pois a partir dele, obtém-se a estimativa da pose, isto é, da orientação e posição da câmera no espaço para cada imagem obtida. Adicionalmente também são estimados dados que são utilizados para ajustes de escalas e possíveis distorções. Para que essa estimação possa ser feita, normalmente são utilizados padrões de calibração. Os dados gerados na fase de calibração são normalmente utilizados juntamente com múltiplas imagens obtidas da cena de forma a concluir a reconstrução.

Dado um conjunto de imagens de uma cena, o objetivo deste trabalho é obter uma reconstrução tridimensional dos objetos de interesse, representada de forma volumétrica, sem a utilização de padrões de calibração.

1.2 METODOLOGIA

Uma técnica muito utilizada para reconstrução de cenas a partir de imagens é a que se baseia em reconstrução volumétrica. Um exemplo típico é o método denominado *Voxel Coloring* (VC)[5]. O uso desta técnica possibilita uma disposição geral de câmeras, onde todas podem encontrar-se afastadas uma das outras e distribuídas esparsamente, diferentemente de métodos baseados em estereo convencionais, onde a *baseline*, isto é a distância entre os centros de projeção das câmeras é pequena.

Uma das limitações do *Voxel Coloring* é a de que as imagens de entrada precisam ser calibradas. A calibração pode ser obtida através da inserção de padrões de calibração, denominados *fiduciais*, que podem ser calibrados, por exemplo, através de métodos como o de Tsai e Zhang [12]. Entretanto, em alguns casos, a cena não favorece ou até mesmo impede a inserção de tais padrões. Para eliminar este problema é necessário usar métodos de calibração que não requerem a presença de fiduciais na cena.

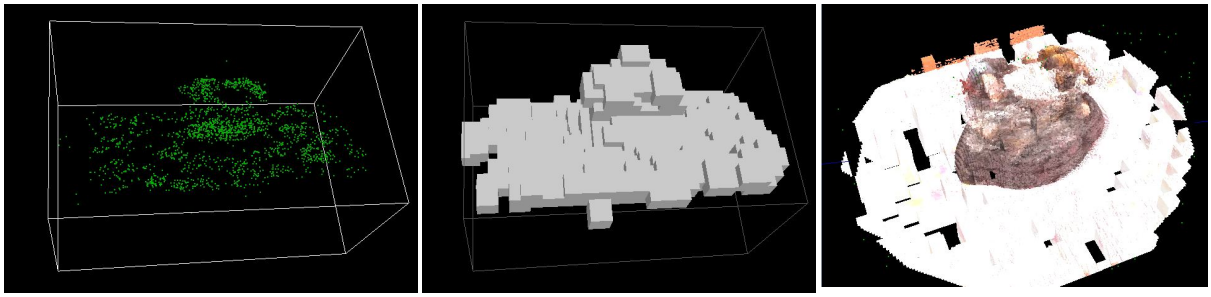
Neste trabalho a calibração de câmera é realizada pelo método denominado *Structure from Motion* (SFM)[13], que é um conjunto de técnicas utilizadas em uma vasta gama de aplicações incluindo levantamento fotogramétrico [14], reconstrução automática de modelos de realidade virtual a partir de sequências de vídeo [15] e a determinação do movimento de uma câmera [16].

A tarefa de calibração, quando realizada por SFM, não faz uso de marcas *fiduciais*, pois os parâmetros intrínsecos e extrínsecos da câmera são estimados sem a necessidade de um prévio conhecimento da geometria da cena. Estes parâmetros, intrínsecos e extrínsecos, são também conhecidos respectivamente como parâmetros internos e externos, onde o primeiro descreve características ópticas e o último, o posicionamento da câmera no espaço. Adicionalmente, o SFM gera um conjunto de amostras de pontos tridimensionais do modelo a ser reconstruído, formando uma nuvem de pontos.

Um interessante desafio da pesquisa foi encontrar um meio de combinar as qualidades de ambos os métodos, o que requer tornar a estrutura de dados gerada pelo SFM compatível com a estrutura de dados esperada como entrada pelo VC.

Para resolver o problema de compatibilidade, uma octree é utilizada para representar o espaço que contém a nuvem de pontos através de uma representação por decomposição espacial adaptativa. A estrutura adaptativa permite determinar, de forma automatizada, as células que melhor se adaptam a distribuição da nuvem de pontos, gerando um volume de ocupação o

mais conexo o possível. A estrutura determinada pela octree é então reamostrada uniformemente para gerar o dado de entrada esperado pelo algoritmo de coloração de voxels, que requer , por sua vez, uma grade regular de elementos volumétricos (Figura 1.3). É este volume preliminar que é enviado para o Voxel *Coloring* para finalizar a reconstrução.



(a) (b) (c)

Figura 1.3 (a) Estrutura de dados de entrada (nuvem de pontos). (b) Reconstrução grosseira utilizando octree (c) Reconstrução Final

1.3 ORGANIZAÇÃO DO TRABALHO

O trabalho está organizado da seguinte maneira: no Capítulo 2 é discutido, de forma bem resumida, como alguns trabalhos resolvem o problema de reconstrução de cenas; os capítulos 3 e 4 são capítulos de descrevem fundamentos do trabalho, onde são discutidas as técnicas de *Structure from Motion* (SFM) e *Voxel Coloring* (VC), respectivamente; no capítulo 5 o método de reconstrução proposto é detalhado; no capítulo 6 os resultados obtidos são discutidos e finalmente no capítulo 7, as conclusões finais sobre o trabalho são apresentadas, assim como algumas sugestões para trabalhos futuros.

Capítulo 2 – **TRABALHOS RELACIONADOS**

Formalmente uma imagem pode ser definida como uma função $f(x, y)$, onde o valor de f para qualquer par (x, y) é chamado de nível de intensidade da imagem naquele ponto [17].

Analisando de forma geral, a essência de uma imagem é a projeção de uma cena tridimensional em um plano bidimensional, onde neste processo a profundidade é descartada. A projeção de um ponto tridimensional em um ponto específico de uma imagem bidimensional cria uma relação de correspondência entre os dois. A Figura 2.1, elucida tal relação onde um raio r liga dois pontos, um ponto tridimensional X e um ponto bidimensional $\tilde{x}$, no plano bidimensional da imagem.

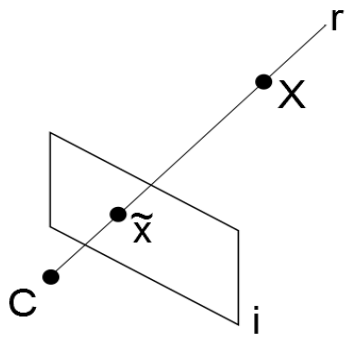


Figura 2.1- Projeção $\tilde{x}$ de um ponto tridimensional X na imagem i. r é o raio de projeção e C representa uma câmera.

O problema é que, a partir de uma única imagem, não é possível determinar qual ponto no espaço tridimensional, ao longo do raio de projeção, corresponde ao ponto da imagem (Figura 2.2). Se duas imagens estão disponíveis, então é possível, através de um processo conhecido como triangulação (ver Capítulo 3), encontrar as coordenadas de um ponto tridimensional no espaço, desde que se conheça a correspondência entre pontos nas duas imagens. Dentro de um ambiente adequado, a aplicação deste método em um conjunto de pontos característicos das imagens, pode ser suficiente para gerar uma reconstrução. No entanto, geralmente o resultado desta reconstrução é um conjunto de pontos esparsos e alterados por ruído.

Gerar modelos de cenas reais tem sido um dos objetivos centrais na Computação Visual. Isto é normalmente feito, pela identificação de varias partes da cena e, a partir dessas

partes, uma representação do todo é obtida. No entanto, há na literatura diversas técnicas para reconstrução de cenas e cada uma apresenta sua própria solução para o problema.

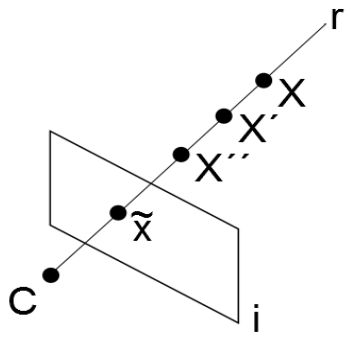


Figura 2.2 – Exemplo de três pontos tridimensionais distintos que estão restritos ao mesmo raio de projeção.

Em [18], Burrus aproveita o surgimento de câmeras capazes de capturar informações de cor e profundidade em tempo real, conhecidas como RGB-D e mostra como é possível tirar proveito desses tipos de informação para estimativa de pose e aquisição de modelos tridimensionais em um contexto de manipulação robótica. Adicionalmente o trabalho faz uso das informações de profundidade para obter uma reconstrução volumétrica e grosseira da cena através da técnica de *space carving* [6] com baixo custo de processamento.

Burrus faz uso de uma mesa giratória para a aquisição de imagens e consegue reconstruir uma cena sem a necessidade de um plano de fundo especial e, além disso, mostra como se torna fácil discriminar um objeto de interesse do plano de fundo e obter silhuetas através de uma simples limitação de profundidade. É importante esclarecer que a câmera utilizada por Burrus, necessita ser previamente calibrada e o uso de um padrão de calibração é necessário. Neste trabalho, utilizamos o método de *Voxel Coloring* em uma configuração de cena semelhante. Entretanto, não necessitamos de uma calibração que demande o uso de marcas fiduciais, o que nos permite utilizar o método em configurações mais livres, onde a câmera se move ao longo da cena. Além disso, não nos baseamos na existência de uma câmera que capture profundidade.

Newcombe e Davison[19], estudam o problema de reconstrução de cenas por superfícies densas e apresentam um método capaz de reconstruir, de forma rápida, uma cena real percorrida por uma única câmera em tempo real. Para isso, o método utiliza a técnica de SFM como ponto de partida para obtenção das poses da câmera e uma nuvem de pontos esparsa. A

nuvem de pontos gerada pelo SFM é usada para estimar inicialmente uma superfície de cena continua que é usada para formar a base para um refinamento denso.

Em [20], Nakazawa, M. et al propõem, um método para reconstruir cenas dinâmicas através da integração das informações de profundidade obtidas por múltiplos Kinects. Os Kinects são posicionados de forma a cobrirem totalmente a superfície de um objeto a ser reconstruído. Todos os sensores necessitam estar devidamente calibrados e para esta tarefa o método de Zhang [12] é utilizado. As estimações das poses de cada kinect são obtidas através da tecnica de *bundle adjustment* que é brevemente elucidada no Capítulo 3. É um exemplo de outra abordagem que necessita de um fiducial para calibração de uma câmera.

Capítulo 3 – *STRUCTURE FROM MOTION*

Uma parte fundamental deste trabalho está ligada ao processo de calibração de câmera que é, resumidamente, a estimação dos parâmetros intrínsecos, extrínsecos e de distorção de lentes de uma câmera [12,21,22]. O processo de calibração usado neste trabalho é feito por *Structure From Motion* [13] que, além de estimar as poses da câmera, também pode, simultaneamente, calcular a posição tridimensional de alguns pontos da cena, resolvendo um problema de correspondência, que nada mais é que, a importante tarefa de determinar quais partes de uma determinada imagem correspondem a quais partes de outra imagem, onde as diferenças são devido ao movimento da câmera e possivelmente o movimento de objetos da cena.

Obter informações sobre uma estrutura geométrica a partir de imagens bidimensionais é uma tarefa desafiadora, pois em geral, não é possível reverter o processo de formação da imagem, isto é, irá existir uma ambiguidade no processo de reconstrução da cena, uma vez que, com somente uma imagem, não é possível determinar a distância do ponto da cena observada ao centro da câmera. Sendo assim, para resolver o problema de reconstrução, mais informações são necessárias.

Uma forma de conseguir mais informações sobre a cena é usar a correspondência de pontos de imagens obtidas a partir de vários pontos de vista. Isso quer dizer que um ponto tridimensional para o qual seja conhecida a sua projeção em dois ou mais pontos de vista, pode ser reconstruído por um método chamado triangulação, que será discutido de forma breve posteriormente neste capitulo.

SFM é um conjunto de técnicas usadas para, como já dito anteriormente, realizar a recuperação de pontos da cena e estimação dos parâmetros das câmeras, usando a correspondência dos pontos de imagens de uma cena em vários pontos de vista. Seus métodos são utilizados em varias aplicações como a determinação do movimento de uma câmera (muito utilizado em filmes, já que permite que um objeto gerado por computador possa ser inserido em uma filmagem de uma cena real) ou a reconstrução automática de modelos de realidade virtual a partir de sequencias de vídeo [15].

Neste capitulo, serão mostradas as etapas do SFM, levando em consideração o modelo de projeção da câmera *pinhole*, que é o mais normalmente usado na literatura e que será brevemente discutido na seção 3.2. Em seguida será elucidado na seção 3.3 o problema de correspondência de pontos da imagem para dois pontos de vista, que será tratado na seção 3.4 e que é um pré-requisito para que se possa calcular a *geometria epipolar* dos mesmos. Na seção 3.5, é mostrado como se pode obter a matriz fundamental, que é a matriz que relaciona pontos

em diferentes imagens em coordenadas de câmera, e também elucida que a mesma pode ser decomposta para recuperação do movimento da câmera, assim como sua matriz de projeção. O processo de triangulação é brevemente descrito na seção 3.6, a adaptação do problema de correspondência para múltiplos pontos de vista é discutida na seção 3.7 e finalmente, na seção 3.8, e feita uma breve descrição do método de *bundle adjustment*, utilizado para refinar as informações reconstruídas.

Na próxima seção, serão estabelecidas algumas notações para melhor compreensão do texto.

3.1 NOTAÇÕES

Neste capitulo serão consideradas coordenadas homogêneas e euclidianas. Um ponto no espaço homogêneo será representado por $\widetilde{\boldsymbol{X}} \sim \left[\tilde{X}\ \tilde{Y}\ \tilde{Z}\ \tilde{W}\right]^T$, onde $\sim$ significa uma igualdade a menos de uma escala. Um ponto do espaço euclidiano terá a forma $\mathbf{X} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$.

3.2 CÂMERA *PINHOLE*

A projeção *pinhole* (Figura 3.1), bastante conhecida na literatura, é um bom modelo de aproximação do comportamento da maioria das câmeras reais. Neste modelo, podemos identificar três componentes que permeiam a relação de pontos tridimensionais e seus correspondentes em uma imagem bidimensional. São as seguintes transformações:

- Corpo rígido;
- Coordenadas da câmera para coordenadas do plano de imagem da câmera;
- Coordenadas do plano de imagem para coordenadas de pixel.

3.2.1 CORPO RÍGIDO

A transformação de corpo rígido relaciona pontos $\tilde{X} \sim [X\ Y\ Z\ 1]^T$ no sistema de coordenadas do mundo com os pontos $\tilde{X}_c \sim [X_c\ Y_c\ Z_c\ 1]$ no sistema de coordenadas da câmera. Isso também pode ser escrito da seguinte forma:

$$\begin{bmatrix} X_C \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \sim \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.1}$$

Onde R é uma matriz de rotação 3x3, que representa a orientação da câmera e T ∈ $\mathbb{R}^3$ é um vetor que representa a translação da câmera. R e T definem a pose da câmera e são conhecidos como seus parâmetros extrínsecos.

3.2.2 COORDENADAS DE CÂMERA PARA COORDENADAS DO PLANO DE IMAGEM

Essa transformação correlaciona pontos $\tilde{X}_c \sim [X_c\ Y_c\ Z_c\ 1]$ no sistema da câmera (tridimensionais) para pontos $\tilde{x} \sim [x\ \ y\ \ 1]^T$ no plano de imagem da câmera (bidimensionais). Podemos expressar essa relação da seguinte maneira:

$$x = f\frac{X_c}{Z_x} \qquad y = f\frac{Y_C}{Z_c}, \tag{3.2}$$

onde f é a distância focal. Mudar o valor de f apenas muda a escala da imagem, dessa forma pode se fazer $f = 1$. Essa relação pode ser expressa pela seguinte equação matricial:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} 1000 \\ 0100 \\ 0010 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{3.3}$$

3.2.3 COORDENADAS DO PLANO DE IMAGEM PARA COORDENADAS DE PIXEL

Essa transformação relaciona pontos do plano de imagem da câmera em coordenadas de pixel $\tilde{u} \sim [u \quad v \quad 1]^T$ (ambos bidimensionais). Essa relação pode ser expressa como:

$$\tilde{u} \sim K\tilde{x} \tag{3.4}$$

K matriz de calibração de câmera na forma:

$$K = \begin{bmatrix} \alpha_u\ s\ u_0 \\ 0\ \alpha_v v_0 \\ 0\ \ 0\ \ 1 \end{bmatrix} \tag{3.5}$$

sendo α_u e α_v fatores de escala e s é o cisalhamento. Os parâmetros u_0 e v_0 são as coordenadas do ponto principal. Esses são conhecidos como parâmetros intrínsecos de uma câmera. Ponto principal é o ponto onde o eixo óptico intercepta o plano de imagem da câmera.

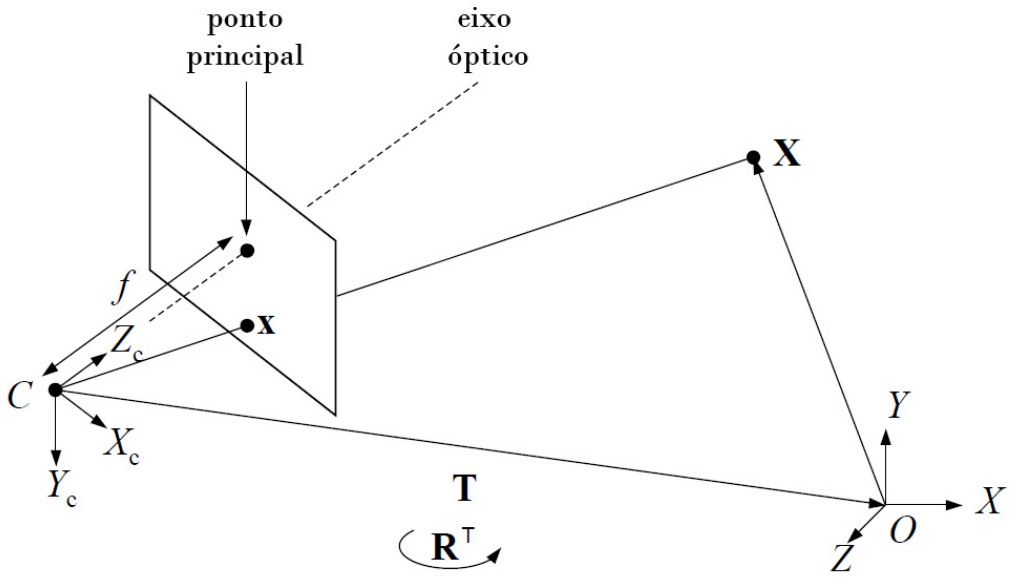


Figura 3.1- Modelo de projeção *pinhole*. *O* representa a origem do sistema de coordenadas do mundo e C a origem do sistema de coordenadas da câmera. R e T são os parâmetros extrínsecos da câmera, isto é, representam a transformação de corpo rígido entre pontos em coordenadas do mundo e da câmera.

Observando as equações (3.1), (3.3) e (3.4), é possível afirmar que um ponto tridimensional $\tilde{X}$ está relacionado à sua coordenada de pixel $\tilde{u}$ pela seguinte equação:

$$\tilde{u} \sim P\tilde{X}, \tag{3.6}$$

onde $P \sim K[R \quad T]$ é a matriz de projeção.

3.3 CORRESPONDÊNCIA ENTRE PONTOS

O problema de correspondência consiste, resumidamente, em identificar pontos em duas ou mais vistas que sejam as projeções do mesmo ponto no espaço. A teoria geométrica do *SFM* assume que esse problema pode ser resolvido [13]. Uma interessante e eficiente forma de solucionar este problema é usar as características, em escala local, das imagens.

A correspondência de pontos (ou feições) funciona através da detecção de pontos nas imagens. Por exemplo, as feições detectadas pelo método detector de cantos Harris [23], estão localizados em valores máximos da função de autocorrelação local da imagem. A vizinhança local desses pontos contém grande variação de intensidade e assim podem ser comparativamente mais fácieis de diferenciar.

Sendo possível encontrar correspondências em duas ou mais imagens através da detecção de pontos de interesse, podemos caracterizar a aparência da imagem na vizinhança local (dos pontos de interesse) usando um descritor adequado. Quanto mais descritores similares duas feições da imagem possuírem, maior será a probabilidade de correspondência entre os pontos associados a tais feições.

3.4 A MATRIZ ESSENCIAL

Segundo Longuet-Higgins (1981)[24] uma matriz essencial relaciona um par de vistas calibradas e pode ser estimada a partir de oito ou mais correspondências de pontos[25, 26]. Essa matriz pode ser decomposta e assim é possível obter a orientação e posição relativa da câmera. Essa ideia se assemelha a de Kruppa [27], onde dados cinco pontos tridimensionais distintos em dois pontos de vista, é possível recuperar a posição e orientação relativa da câmera, assim como posição desses pontos a menos de um fator de escala global desconhecido.

Analisando a Figura 3.2, vemos a linha epipolar $\widetilde{l'}$, que pode ser considerada como a projeção do raio que sai do centro óptico C e intercepta o primeiro plano de imagem em $\tilde{x}$. Perceba que $\widetilde{l'}$ intercepta o plano formado pelos centros ópticos C, C$'$ e o ponto de imagem $\tilde{x}$. Esse plano é chamado de plano epipolar. Note também que todas as linhas epipolares em uma imagem tem um ponto em comum, que é a projeção de C$'$ (o segundo eixo óptico). Esse ponto é denotado por $\tilde{e}$ e $\widetilde{e'}$ para a primeira e segunda câmera respectivamente e é chamado de epipolo. Desta forma é possível perceber que a projeção de um ponto tridimensional na imagem um (1) (plano de imagem à esquerda na Figura 3.2), tem a sua projeção correspondente na imagem dois (2) (plano de imagem à direita na Figura 3.2**Error! Reference source not found.**) restrita a sua linha epipolar correspondente. Esse fato é chamado de restrição epipolar.

A restrição epipolar pode ser representada algebricamente usando-se a matriz essencial E que relaciona pontos correspondentes em duas imagens em coordenadas dos sistemas de câmera [28].

$$X = RX' + T \tag{3.7}$$

$$X^T[T]_X X = X^T[T]_X RX' + X^T[T]_X T$$

$$\therefore$$

$$X^T[T]_X RX' = X^T EX' = 0 \tag{3.8}$$

A matriz essencial (3x3) é representada por $E \sim [T]_X R$ e $[T]_X$ é uma matriz que representa a operação de produto vetorial. Para $T = [t_x\ t_y\ t_z]^T$, temos:

$$[T]_X = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \tag{3.9}$$

onde X' é um ponto euclidiano no sistema de coordenadas da câmera C′, X é o ponto euclidiano no sistema de coordenadas da câmera C. Finalmente, R e T são respectivamente uma matriz(3x3) de rotação e um vetor tridimensional.

A Equação 3.8 também se mantém para pontos de imagem $\tilde{x}$ e $\tilde{x}'$. Assim tem-se a equação da restrição epipolar (Figura 3.2):

$$\tilde{x}^T E \tilde{x}' = 0 \tag{3.10}$$

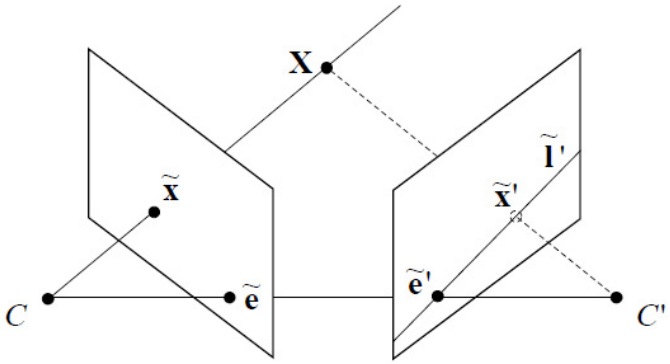


Figura 3.2- ilustração da Geometria epipolar para duas câmeras.

Observe que a matriz E depende somente de R e T e está definida a menos de um fator de escala arbitrário.

3.5 A MATRIZ FUNDAMENTAL

Da Equação 3.4 é possível chegar à equação:

$$\tilde{x} \sim K^{-1}\tilde{u} \tag{3.11}$$

Sendo assim, pode-se reescrever a equação 3.9 em termos de coordenadas medidas em unidades de pixel:

$$
\begin{aligned}
(K^{-1}\tilde{u})^T E\left(K'^{-1}\tilde{u}'\right) &= 0 \\
\tilde{u}^T\left(K^{-1^T} E K'^{-1}\right)\tilde{u}' &= 0 \\
\tilde{u}^T F \tilde{u}' &= 0
\end{aligned}
\tag{3.12}
$$

onde $F \sim K^{-1^T} E K'^{-1}$ é a matriz (3x3) fundamental e pode ser linearmente estimada desde que haja oito ou mais pontos correspondentes. Caso o leitor deseje saber como estimar a matriz, recomenda-se a leitura de [13].

A matriz fundamental pode ser decomposta a fim de se recuperar o movimento da câmera e consequentemente a sua matriz de projeção. Se as matrizes de calibração são conhecidas é possível recuperar um par de matrizes de projeção compatíveis, a menos de um parâmetro de ambiguidade. Este parâmetro corresponde a um fator de escala desconhecido para a translação da câmera. Sejam K e K' duas matrizes de calibração de câmera conhecidas. Levando em consideração a Equação 3.11, a matriz fundamental pode ser transformada em essencial:

$$E \sim K'^T F K \tag{3.13}$$

Dessa forma é possível decompor esta matriz em duas outras: Uma de distorção simétrica correspondente a uma translação e outra, ortonormal, que correspondente a uma rotação entre os pontos de vista. [13].

$$E \sim [T]_X R \tag{3.14}$$

Caso o Leitor se interesse por saber como essa decomposição pode ser feita, novamente recomenda-se [13].

3.6 TRIANGULAÇÃO

Através da triangulação, pontos tridimensionais podem ser encontrados a partir de suas posições na imagem em dois ou mais pontos de vista, desde que as matrizes de projeção sejam dadas. De forma ideal, pontos tridimensionais deveriam ser encontrados no local de interseção dos raios retroprojetados. Entretanto, isso geralmente não acontece devido ao ruído de

medição. Desta forma, pontos tridimensionais devem ser escolhidos de forma a minimizar uma métrica de erro apropriada.

O algoritmo de reconstrução padrão minimiza a soma dos quadrados dos erros entre as posições de imagem medidas e as previstas do ponto tridimensional em todas as vistas que o mesmo é visível. Ou seja:

$$X = \arg\min_{X} \sum_i ||u_i - \hat{u}_i (P_i, X)||^2$$

Onde $u_i \; e \; \hat{u}_i(P_i, X)$ são, respectivamente as posições medidas e previstas da imagem na vista i (Figura 3.3). Sob a hipótese que a medição de ruído na coordenada da imagem é uma distribuição Gaussiana, esta abordagem fornece a máxima verossimilhança de X. Uma solução não interativa para duas vistas é descrita em [29]. Para casos onde há mais que duas vistas, a minimização da métrica de erro pode ser alcançada interativamente por uma otimização não linear.

Uma importante observação é que esta abordagem pode falhar em encontrar uma solução de custo mínimo local caso não haja uma inicialização suficientemente boa. Uma estratégia bastante conhecida é empregada explorando a equação 3.6. Os vetores tridimensionais $\tilde{u}_i$ e $P_i\tilde{X}$ são paralelos e por isso podemos escrever:

$$[\tilde{u}_i]\times P_i\tilde{X} = 0.$$

Essa equação provê somente duas restrições em $\tilde{X}$ que podem ser organizadas em uma equação matricial da forma:

$$A\tilde{X} = 0$$

onde A é uma matriz $3n \times 4$ e n é o número de vistas nas quais o ponto reconstruído é visível. A solução necessária para o ponto homogêneo $\tilde{X}$ minimiza $\left\|A\tilde{X}\right\|$ sujeito a $\left\|\tilde{X}\right\| = 1$ e é dada pelo autovetor de $A^{\mathrm{T}}A$ correspondente ao menor autovalor.

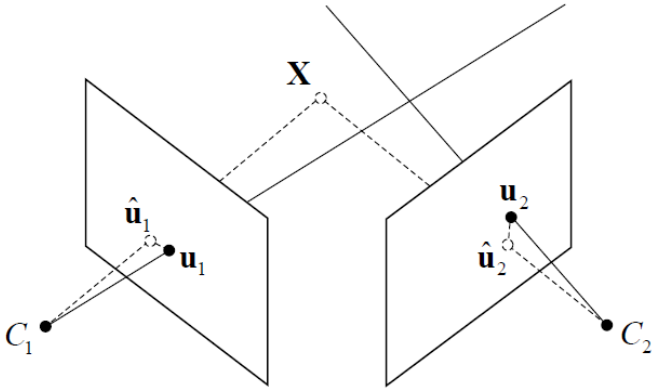


Figura 3.3- Exemplo de triangulação. Conhecendo-se as matrizes de projeção, um ponto tridimensional X pode ser calculado a partir das suas coordenadas de pixel (u_1,u_2, ...) em dois ou mais pontos de vista (C_1, C_2, ...)

3.7 *STRUCTURE FROM MOTION* EM MÚLTIPLOS PONTOS DE VISTA

Até agora foi somente visto a aplicação de *SFM* para apenas dois pontos de vista. Isso porque tanto a matriz essencial quando a fundamental, encapsulam a restrição geométrica relacionada a apenas dois pontos de vista.

Uma forma de contornar essa restrição é fazer uso de uma classe de algoritmos chamada algoritmos sequenciais. Esses algoritmos incorporam sucessivos pontos de vista, um de cada vez (veja Figura 3.4), e conforme cada vista é registrada, uma reconstrução parcial é incrementada pelo cálculo de todos os pontos tridimensionais que são visíveis em dois ou mais pontos de vista. Esse cálculo é feito por meio da triangulação.

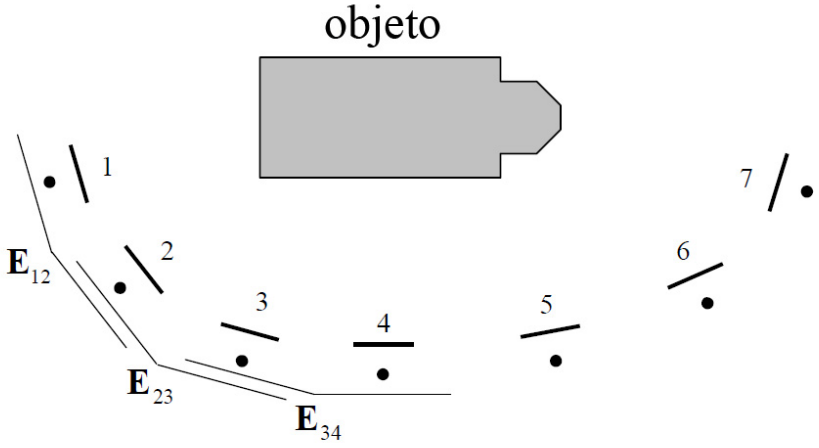


Figura 3.4- Exemplo de registro sequencial. Os pontos de vista de 1 a 7 são incorporados um de cada vez pelo cálculo das matrizes essenciais E_{12}, E_{23}, E_{34}, etc.

3.8 *BUNDLE ADJUSTMENT*

Bundle adjustmente (BA) é a etapa final onde os parâmetros de estrutura e movimento são refinados iterativamente, pela minimização de uma função de custo adequada. Esta etapa depende criticamente de uma inicialização adequada.

Considere a situação na qual um conjunto de pontos tridimensionais X_j é visto por um conjunto de câmeras com matrizes de projeção P_i. Denote por u_{ij} as coordenadas do j-ésimo ponto como visto pela i-ésima câmera. O SFM fornece uma estimativa inicial de P_i e X_j.

O BA trabalha através da minimização de uma função de custo que está relacionada a uma soma ponderada dos quadrados dos erros de reprojeção.

Dado um conjunto de observações de ruidosas, o objetivo do *Bundle Adjustment* é determinar uma estimativa ótima de um conjunto de parâmetros θ. É importante deixar claro que a maioria desses parâmetros não pode ser observada diretamente. É possível citar como exemplo as matrizes de projeção e as coordenadas de pontos tridimensionais. Na verdade esses parâmetros nos possibilitam fazer predições de quantidades que podem ser observadas diretamente, por exemplo, a coordenada de pixel calculada a partir da projeção de pontos tridimensionais.

Seja o conjunto de predições $Z(\theta)$ e o conjunto de observações correspondentes ser $\overline{Z}$. Sendo assim o erro de predição residual ΔZ é dado por:

$$\Delta Z = \overline{Z} - Z(\theta). \tag{3.15}$$

A função de custo adequada , a qual o *bundle adjustment* minimiza, deve refletir a verossimilhança do residual ΔZ para obter uma estimativa de parâmetro de máxima verossimilhança. Sob a hipótese que a medição de ruído é uma distribuição gaussiana, a função de custo apropriada é uma soma quadrada dos erros, que nada mais é que a soma negativa da verossimilhança logarítmica:

$$f(\theta) = \frac{1}{2}\sum_i \Delta z_i(\theta)^T W_i \, \Delta z_i(\theta) \qquad \Delta z_i\,(\theta) = \bar{z}_i - z_i(\theta) \tag{3.16}$$

onde $\Delta z_i\,(\theta)$ é erro de previsão de uma feição (u_{ij}) e W_i é uma matriz de ponderação simétrica definida positiva, que é escolhida para aproximar a covariância inversa do ruído de medição associado a medição de $\overline{z}_i$.

Como já dito, o BA minimiza o erro de reprojeção ΔZ, que é expresso como as soma dos quadrados de um grande número de funções reais não lineares. Desta forma a minimização é obtida usando algoritmos de mínimos quadrados não lineares. Um exemplo destes algoritmos é o Levenberg-Marquardt [30], que é conhecido devido sua fácil implementação e porque também faz uso de uma estratégia eficaz de atenuação, que lhe confere a capacidade de convergir de forma rápida a partir de uma vasta gama de valores inicias.

Ao final deste processo o conjunto de pontos obtido após o processo de triangulação encontra-se refinado. Consequentemente, o conjunto de pontos obtidos está pronto para ser usado em outras aplicações.

Capítulo 4 - **VOXEL COLORING**

Introduzido por S.Seitz e C.Dyer [5] a Coloração de Voxels ou Voxel *Coloring* é um método que propõe a reconstrução volumétrica de uma cena fazendo uso de imagens obtidas de ambientes reais para recuperar a informação de "cor" (radiação luminosa) [5] da cena no espaço volumétrico. Esta técnica possui intrinsicamente a habilidade de contemplar grandes mudanças de visibilidade e evita problemas de correspondência de imagens, pois trabalha em um espaço de cena discretizado por voxels, que são percorridos em uma ordem de visibilidade fixa. Desta forma esta técnica consegue lidar com o problema de oclusão e permite ao mesmo tempo uma disposição de câmeras de entrada, onde todas estão afastadas uma das outras e amplamente distribuídas no ambiente da cena.

A aquisição de modelos tridimensionais fotorrealísticos de cena reais a partir de pontos de vista amplamente distribuídos é o problema considerado em S.Seitz e C.Dyer [5], onde o termo fotorrealismo é usado para descrever reconstruções tridimensionais de cenas reais, cujas reprojeções possuam informações de cor e textura suficientes para reproduzir de forma precisa as imagens da cena. Para garantir isso são propostos dois critérios que devem ser atendidos por uma técnica de reconstrução fotorrealistica: foto fidelidade e ampla cobertura de ponto de vista.

A foto fidelidade garante que o modelo reprojetado reproduza de forma precisa as imagens de entrada, onde a cor, textura e resolução de pixel são preservadas. Já a ampla cobertura de ponto de vista assegura que as reprojeções sejam precisas sobre uma grande variedade de pontos de vista, requerendo que as imagens de entrada estejam amplamente distribuídas no ambiente.

O resto do capitulo está organizado da seguinte maneira: Na seção 4.1 será definido o problema. Na seção 4.2 serão discutidas as condições sob as quais o problema admite solução. Em 4.3 será discutida a invariância à coloração de um voxel e sua unicidade e na seção 4.4 o algoritmo que soluciona o problema será especificado.

4.1 DEFINIÇÃO DO PROBLEMA DA COLORAÇÃO DE VOXELS

Colorir os voxels de um volume tridimensional de forma a garantir a foto fidelidade do mesmo em relação a um conjunto de imagens de entrada, é o problema que o Voxel Coloring contempla. É importante especificar que o escopo do problema em S.Seitz e C.Dyer [5] contém somente cenas e iluminação estáticas e que são apenas consideradas superfícies aproximadamente Lambertianas.

Seja S uma cena tridimensional que será reconstruída e representada como um conjunto fechado de voxels. O símbolo V representa o conjunto de todos os voxels. Cada elemento de volume $v \in V$ ocupa um volume homogêneo do espaço e possui uma cor fixa. A notação utilizada para representar a cor de um voxel v em uma cena S é $cor(v, S)$. O conjunto de todas as imagens é dado por $I = \{I_1, I_2, I_3, \dots, I_n\}$. Um pixel de uma imagem é representado por $p \in I$ e sua cor é denotada por $cor(p, I_i)$. De forma 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.

Duas definições a respeito de S são claramente mencionadas em [5]:

1. *S é dita completa em relação a um conjunto I se, para cada imagem I_i e cada pixel $p \in I_i$, existe um* voxel $v \in S$ *tal que* $v = S(p)$.
2. *Uma cena completa é considerada consistente em relação a um conjunto de imagens I se, para toda imagem I_i e todo pixel $p \in I_i$ a relação abaixo é verdadeira:*

$$cor(p, I_i) = cor(S(p), S)$$

Levando em consideração as definições anteriores podemos formalizar o problema:

Definição do problema de coloração de voxels: Seja V o conjunto de voxels que corresponde à discretização do espaço $U \subset \mathbb{R}^3$ no qual S está contida. Considere também, uma coleção de imagens I obtidas a partir de C, um conjunto de câmeras calibradas cujos centros de projeção se encontram em pontos cp_i tais que $\text{cp}_\text{i} \in \mathbb{R}^3 - \text{U}$. O problema de coloração do conjunto de voxels V se resume em obter para todo elemento $v \in V$ uma atribuição de cor de tal forma que o conjunto V, quando renderizado a partir de cada uma das câmeras C_i, reproduza as imagens I_i.

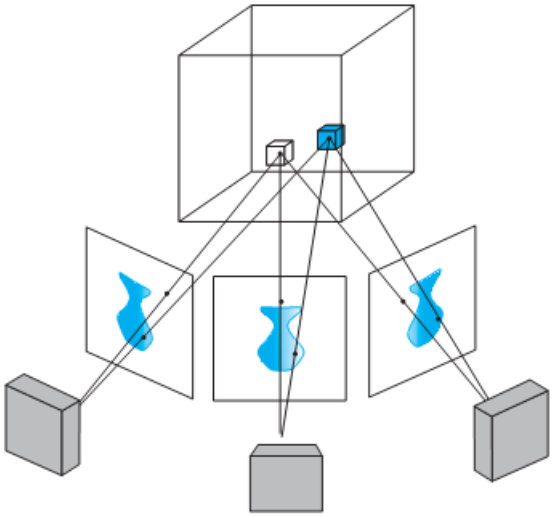


Figura 4.1- Ilustração da coloração de voxels. Dois voxels são projetados em três planos diferentes. Em cada um desses planos a cor de ambos os voxels se mantém.

4.2 RESTRIÇÕES

Para que seja possível chegar na solução deste problema é preciso antes analisar dois fatores: unicidade dos voxels e o processamento do *Voxel Coloring*.

A unicidade está relacionada ao fato de que a coloração de múltiplos voxels pode ser consistente com um dado conjunto de imagens. Para este caso o problema está bem definido?

O processamento está relacionado à forma como um voxel pode ser calculado a partir de um conjunto de imagens de entrada.

Com o intuito de atender a esses dois fatores uma restrição de visibilidade foi estabelecida. Essa restrição garante a identificação de algumas unidades de volume invariantes, aquelas para as quais as cores estão unicamente definidas em todos os pontos de vista. Além disso, a restrição também define uma ordem de profundidade dos voxels pela qual o processo de colorir pode ser feito em um único passo atravessando todo o volume que encapsula a cena.

Essa restrição tem a função de simplificar a tarefa de determinar as relações de visibilidade entre os voxels de um volume. Conhecendo a relação de ordem existente entre eles e o conjunto de câmeras do contexto da reconstrução, evita-se o tratamento de determinação de visibilidade por métodos mais sofisticados.

Considere P e Q como dois pontos de uma cena e I uma imagem de uma câmera posicionada em C. P somente irá ocultar Q se P pertencer ao segmento de reta $\overline{CQ}$. Essa restrição de ordenação de visibilidade pode ser formalizada da seguinte maneira:

- *Existe uma norma* $\|\cdot\|$ *de forma que para todos os pontos* P *e* Q *de uma cena e imagem de entrada I,* P *oculta* Q *em I somente se* $\|P\| < \|Q\|$.

Essa norma não é compatível com todas as configurações de câmeras possíveis. No entanto, em [5] é afirmado que a restrição de ordenação de visibilidade será satisfeita sempre que nenhum ponto da cena esteja contido dentro do fecho convexo determinado pelos centros de projeção das câmeras. (veja Figura 4.2) Em S.Seitz e C.Dyer [5] é usada a norma de compatibilidade de oclusão $\|P\|_C$, que é definida como sendo a distancia Euclidiana de P até C, onde C é referenciado como o volume da câmera.

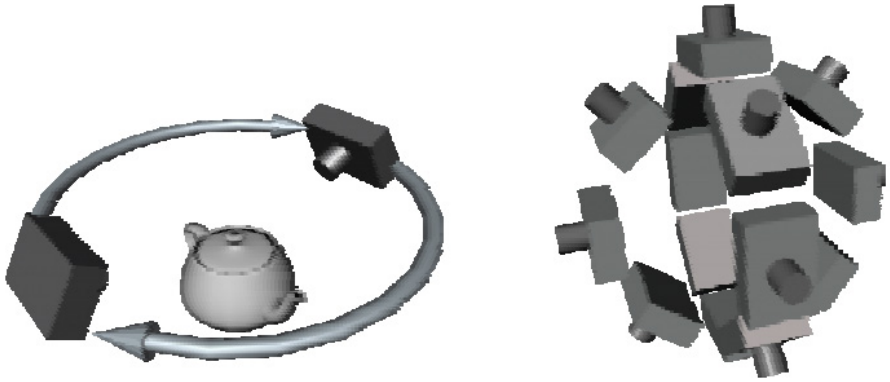


Figura 4.2 – Configurações de câmeras compatíveis. Fonte: S.Seitz e C.Dyer [5]

4.3 INVARIANTES À COLARAÇÃO E UNICIDADE

Agora é possível discutir como a ordenação dos voxels, segundo a restrição de ordenação de visibilidade, pode ser usada na reconstrução da cena. Infelizmente, existem varias soluções possíveis para o problema de reconstrução por fotoconsistência. Isso é possível porque vários conjuntos diferentes de voxels, contidos no espaço da cena, podem reproduzir as imagens de entradas se as cores certas forem atribuídas. Devido a esta possibilidade de múltiplas soluções para o problema, é necessário obter uma forma de garantir a unicidade destas unidades de volume. Isto é possível através do conceito de voxels invariantes, já mencionado na seção anterior. Formalizando tem-se:

- Um voxel v é invariante à coloração em relação a um conjunto de imagens, se para cada par de cenas S e S' ambas consistentes com as imagens, $v \in S \cap S'$ implica que $cor(v, S) = cor(v, S')$

É interessante ressaltar que um voxel invariante à coloração não precisa estar presente em todas as cenas consistentes, porém sua cor deve ser igual em todas em que o mesmo estiver

presente. De fato, qualquer conjunto de imagens que obedeça a restrição de ordenação de visibilidade, possui invariantes à coloração suficientes para formar uma reconstrução de cena completa.

Seja $I_1, \dots, I_m$ um conjunto de imagens. Se p é um ponto de uma imagem qualquer do conjunto e v_p é definido como sendo o voxel em $\{S(p) | \, S\ cena\ é\ consistente\}$ que está mais próximo ao conjunto de câmeras, Pode-se afimar que v_p é um invariante à coloração. A prova dessa afirmação pode se encontrada no artigo original [5].

A coloração de voxels de um conjunto de imagens $I_1, \dots, I_m$ é definida como:

$$\bar{S} = \{v_p | p \in I_i, 1 \leq i \leq m\} \tag{4.2}$$

4.4 PROCESSANDO A COLORAÇÃO DE VOXEL

Sempre que a configuração das câmeras obedecer as restrições de ordenação de visibilidade é possível afirmar que o volume que encapsula a cena, o conjunto de voxels V, pode ser dividido em um conjunto de camadas de voxels que se distanciam uniformemente das câmeras. Isso pode ser expresso da seguinte forma:

$$V_C^d = \{v | \|v\|_C = d\} \tag{4.3}$$

$$V = \bigcup_{i=1}^{r} V_C^{d_i} \tag{4.4}$$

onde $d_1, \dots, d_r$ uma sequência crescente de números naturais.

Resumidamente, o algoritmo se baseia simplesmente em percorrer cada unidade de volume em todas as camadas, seguindo uma ordem crescente da numeração das camadas. Esses voxels são projetados nas imagens nos quais os mesmos estão visíveis e avaliados em relação a sua consistência. Caso um desses voxels seja considerado consistente, ele é colorido

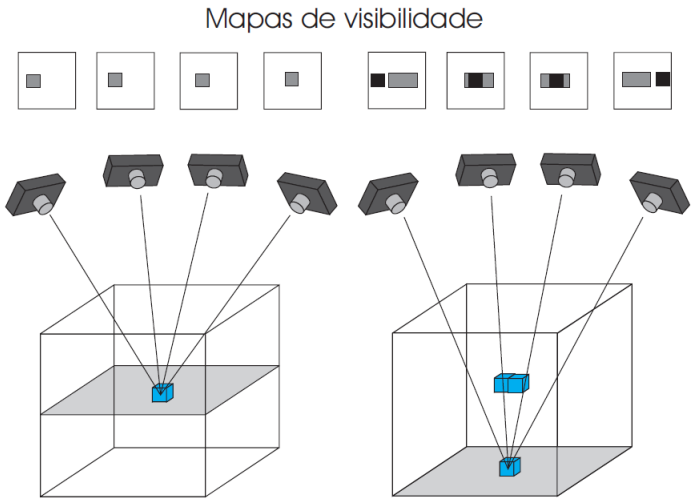


Figura 4.3 – Exemplo de determinação de visibilidade

segundo uma função das cores tomadas das imagens nas quais o mesmo é visível. Se este não for consistente, uma cor transparente é dada representando a remoção do voxel do modelo (Figura 4.1).

Um mapa de visibilidade Mv é utilizado para facilitar a tarefa de determinação da visibilidade do voxel. Para cada imagem I_i haverá um mapa de visibilidade Mv_i (Figura 4.3).

No início do algoritmo todos os elementos nos mapas de visibilidades tem o valor zero em cada uma de suas entradas. Ter valor zero significa que o elemento é visível e que nenhuma estrutura que pode causar oclusão ainda foi detectada. Cada camada $V_C^{d_i}$ tem um numero fixo igual de voxels. Tem-se que cada voxel $v_j \in V_C^{d_i}$, quando avaliado, tem seu centroide projetado em uma imagem I_i e se o valor da posição obtida na projeção em Mv_i for igual a zero então o voxel é visível, caso contrario, o mesmo está ocluso. Sempre que uma unidade de volume for considerada consistente, os mapas de visibilidade são atualizados. Isso é feito de maneira simples, bastando apenas atribuir o valor um à posição associada à projeção do centroide do voxel em cada uma das imagens onde o mesmo é visível.

As imagens que são obtidas das cenas reais não representam uma cena totalmente Lambertiana e por isso medidas estatísticas são utilizadas para mensurar a verossimilhança da consistência de um voxel. No artigo original [5] é usada a seguinte estatística:

$$\lambda_k(v) = \frac{(n-1)s^2}{\sigma_0^2} \tag{4.5}$$

onde k é o numero de imagens em que o voxel está visível, s o desvio padrão calculado sobre o conjunto de cores na projeção de v_j em I_i, n a cardinalidade deste conjunto de cores e σ_0 o desvio padrão de uma distribuição normal que representa os erros introduzidos pelo sensor da câmera.

O teste de foto-consistência é baseado em um teste estatístico que considera que as cores observadas são variáveis aleatórias, com média desconhecida e desvios padrão conhecidos. Os desvios representam o nível de ruído de cada sensor que registrou as imagens e pode ser estimado através do desvio padrão das cores observadas em várias amostras de imagem. O teste de foto-consistência para um voxel considera como hipótese nula a igualdade das médias das variáveis aleatórias X_i, que modelam as cores observadas nas projeções de um voxel. A estatística usada para estimar a igualdade das medias é:

$$\sum_{i=1}^{n} \left(\frac{X_i - \mu}{\sigma_i}\right)^2$$

Tal estatística tem distribuição X^2 com grau de liberdade $n - 1$, onde n é o número de cores observadas. O limiar é baseado no valor associado à região crítica com nível de significância escolhido pelo experimentador.

Na nossa implementação não modelamos o ruído associado aos sensores da câmera e utilizamos um teste mais simples, que compara o desvio padrão das cores observadas na projeção de um voxel com um limiar determinado de forma empírica. Para uma explicação mais completa sobre a medida estatística aqui discutida, recomenda-se ao leitor ler [5] e [3]. Abaixo é apresentado o pseudocódigo do algoritmo de coloração de voxels.

Algoritmo 1 Coloração de voxels

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

Capítulo 5 – ESCULTURA DO ESPAÇO DE CENA POR COLORAÇÃO DE VOXELS E STRUCTURE FROM MOTION

O SFM e a escultura do espaço de cena por *Voxel Coloring* podem, a principio, parecer não ter relação um com outro. Isso porque, embora sejam técnicas normalmente usadas na reconstrução de cenas, manipulam e geram dados de naturezas completamente diferentes. Um dos desafios desse trabalho, foi justamente encontrar uma forma eficaz de compatibilizar esses dados.

5.1 ETAPAS DO MÉTODO

A abordagem proposta pode ser dividida em quatro passos principais:

- Aquisição da sequência de imagens;
- Aplicação do método *Structure from Motion;*
- Reconstrução preliminar do volume de ocupação utilizando uma octree;
- Aplicação do método *Voxel Coloring;*

A seguir cada etapa será brevemente detalhada, no entanto a Figura 5.3 apresenta um-fluxograma que resume de forma clara as etapas desta abordagem.

5.1.1 AQUISIÇÃO DA SEQUÊNCIA DE IMAGES

Esta é uma etapa simples e consiste em preparar o ambiente da cena para capturar uma sequência de imagens. O objeto modelo é posicionado sobre uma base giratória. A partir do momento que essa base inicia a rotação, um vídeo deste movimento é gravado. Este vídeo é dividido em quadros, os quais são enviados para a próxima etapa.

5.1.2 APLICAÇÃO DO MÉTODO DE STRUCTURE FROM MOTION

Nesta etapa o SFM, já discutido no Capítulo 3, é aplicado e duas subetapas ocorrem: estimação dos parâmetros da câmera e obtenção de uma nuvem de pontos.

É importante destacar que o *Structure From Motion* não foi implementado neste trabalho. No entanto, foi usado o *software Voodoo Camera Tracker* que já possui uma robusta solução embutida. Trata-se de uma ferramenta usada para integração de cenas virtuais e reais. A sequência de imagens capturada na etapa anterior é usada neste programa para estimativa dos parâmetros da câmera.

Esta é a etapa onde os parâmetros intrínsecos e extrínsecos da câmera são obtidos. A maneira como essa estimativa é feita é explicada no capitulo sobre *Structure From Motion.* No entanto é válido observar, de forma breve, como essa etapa é tratada pelo *Voodoo Camera Tracker.* A estimação consiste em cinco passos de processamento:

- Detecção automática de pontos característicos;
- Análise automática de correspondências;
- Eliminação de correspondências incorretas;
- Estimação robusta dos parâmetros da câmera;
- Refinamento final dos parâmetros da câmera;

Detecção automática de pontos característicos

Os pontos característicos são detectados com precisão subpixel usando um detector de cantos. Neste trabalho utilizamos o detector de cantos *Harris*[23] devido a sua forte invariância a rotação, escala e iluminação.

Análise automática de correspondências

Os pontos podem ser correspondidos a partir de uma imagem para a seguinte através da escolha de correspondências que tem a mais alta correlação cruzada de intensidade de imagem para regiões em torno dos pontos. Nesta fase configuramos o *Voodoo Camera Tracker* para utilizar o método *Cross Correlation* para encontrar correspondências entre as imagens.

Eliminação de correspondências incorretas

Objetos em movimento ou distorções na cena podem fazer com que algumas das correspondências sejam incorretas. A fim de evitar que esse tipo de problema ocorra, um algoritmo de amostragem randômica é empregado para detectar correspondências ruins. Neste trabalho configuramos o *software* para trabalhar com o Algoritmo *RANSAC* (*RANdom SAmple Consensus*) [31]. *RANSAC* é um método iterativo usado para estimar parâmetros de um modelo matemático a partir de um conjunto de dados que contem *outliers.* O Algoritmo RANSAC assume que o conjunto de dados que está sendo analisado é composto por *inliers* e *outliers*. *Inliers* podem ser explicados por um modelo de valores de parâmetros, enquanto *outliers* não se encaixam neste referido modelo sob nenhuma circunstância. Neste contexto, correspondências ruins também podem ser consideradas *outliers*. A vantagem no uso do

RANSAC é a habilidade estimar de forma robusta o modelo de parâmetros, isto é, este método é capaz de estimar os parâmetros com um alto grau de acurácia, mesmo diante de um número significante de correspondências ruins.

Estimação robusta dos parâmetros da câmera

É de forma incremental que o os parâmetros da câmera são estimados. Isso ocorre através da utilização de uma técnica de otimização aplicada nas correspondências boas, chamadas *inliers*.

Refinamento final dos parâmetros da câmera

Nesta etapa, um passo de refinamento é aplicado em todos os parâmetros de câmera da sequência. É uma tentativa de distribuir uniformemente os erros de estimativa ao longo da sequência.

Vale ressaltar que os parâmetros intrínsecos só precisam ser estimados uma única vez, pois são os mesmos desde que a câmera utilizada no método não mude. Já os parâmetros extrínsecos são estimados para cada quadro da sequência obtida na primeira etapa. Parâmetros extrínsecos fornecem a pose da câmera em cada quadro da sequência. Estes parâmetros serão aproveitados na última etapa do método.

5.1.2.1 OBTENÇÃO DE UMA NUVEM DE PONTOS

Nesta fase, pontos tridimensionais são estimados através do processo de triangulação (ver Capítulo 3). Para cada imagem obtida na etapa um, uma nuvem de pontos é obtida. Ao final desta etapa teremos um conjunto de nuvem de pontos que reconstrói de modo esparso a cena. Esta nuvem é então enviada para etapa quatro.

5.1.3 RECONSTRUÇÃO PRELIMINAR POR OCTREE

Nesta importante etapa, a nuvem de pontos obtida na etapa três, é usada para gerar uma reconstrução preliminar grosseira do modelo real. Basicamente nesta etapa é construída uma representação da nuvem de pontos através de uma octree, onde cada nó ramo define uma *bounding box* dos nós subjacentes e o nó raiz descreve uma caixa delimitadora que encapsula todos os pontos. Cada nó da octree tem oito ou nenhum filhos.

O objetivo da octree é subdividir o espaço de forma adaptativa, em células que contenham pelo menos um ponto ou nenhum ponto. É escolhido um nível máximo da árvore de

forma que as células consideradas ocupadas, isto é, que contenham pelo menos um ponto da nuvem formem um conjunto de células o mais conexo o possível. O nível escolhido depende da distribuição dos pontos da nuvem, Quando mais uniforme for a distribuição mais refinada pode ser a octree chegando até mesmo no nível em que cada célula contém somente um ponto da nuvem. Por outro lado, na maioria dos casos, a distribuição dos pontos da nuvem é bastante não uniforme o que faz com que uma octree muito refinada gere um conjunto de células desconectadas o que não caracteriza a estrutura do modelo a ser reconstruído, conexo em sua maior parte. Neste trabalho o nível máximo foi selecionado de modo empírico para os exemplos investigados nos testes. A Figura 5.1 mostra um exemplo de volume de ocupação gerado a partir de uma nuvem de pontos obtida pelo método SFM.

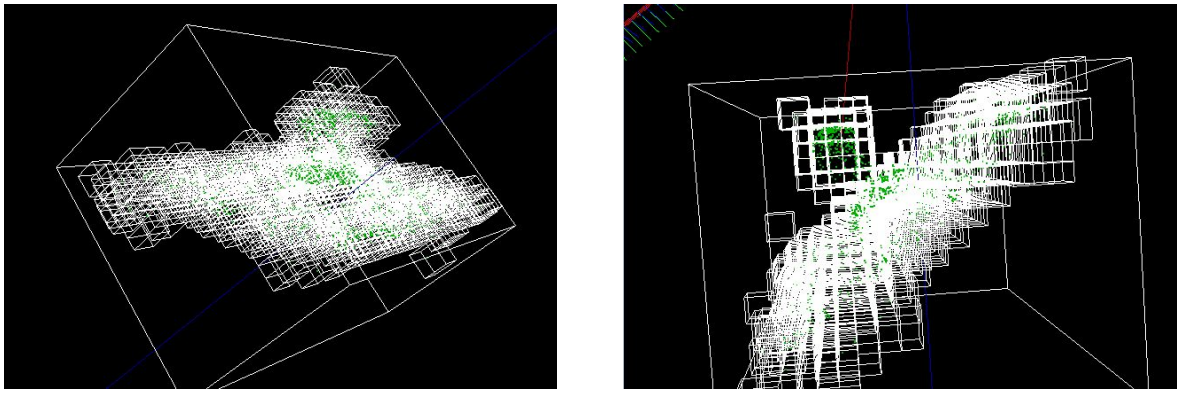


Figura 5.1 – Demonstração da voxelização da nuvem de pontos.

O objetivo da subdivisão do espaço contendo a nuvem de pontos é gerar uma voxelização de tal nuvem, produzindo um volume de ocupação inicial grosseiro. Antes de prosseguir para a próxima etapa, cada célula da octree, que contém pelo menos um ponto da nuvem é reamostrada, sendo subdivida regularmente em voxels de tamanho idêntico ao desejado para o volume final a ser refinado pelo Voxel Coloring..Após a nuvem de pontos ser convertida em voxels, fica configurada a versão inicial do volume de ocupação. A Figura 5.2 apresenta um bom exemplo do que acontece ao final desta etapa.

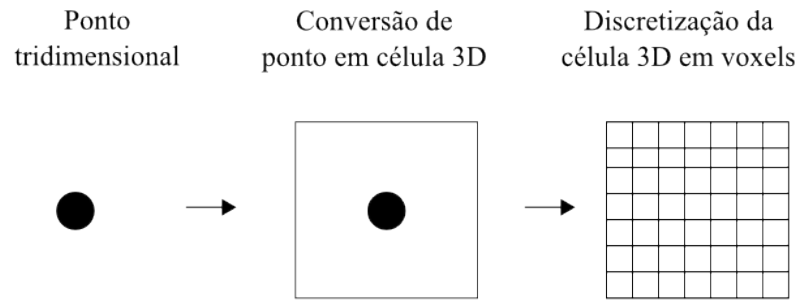


Figura 5.2- Exemplo da transformação de um ponto tridimensional em voxels.

5.1.4 APLICAÇÃO DO MÉTODO *VOXEL COLORING*

Está é a ultima etapa da abordagem adotada neste trabalho. Aqui o volume de ocupação, já devidamente discretizado em voxels, é recebido e a reconstrução é finalizada. Nesta etapa a câmera já está propriamente calibrada e por isso, é possível projetar cada voxel do volume de ocupação da cena no plano de imagem da câmera. Isso é feito para cada pose estimada na segunda etapa. É interessante ressaltar que, embora as poses da câmera sejam estimadas para cada quadro obtido na primeira etapa, não há a necessidade de usar Voxel Coloring para todas. É possível obter bons resultados com somente alguns dos pontos de vista estimados.

Na Figura 5.3 é apresentado um fluxograma que descreve cada um dos passo do método proposto.

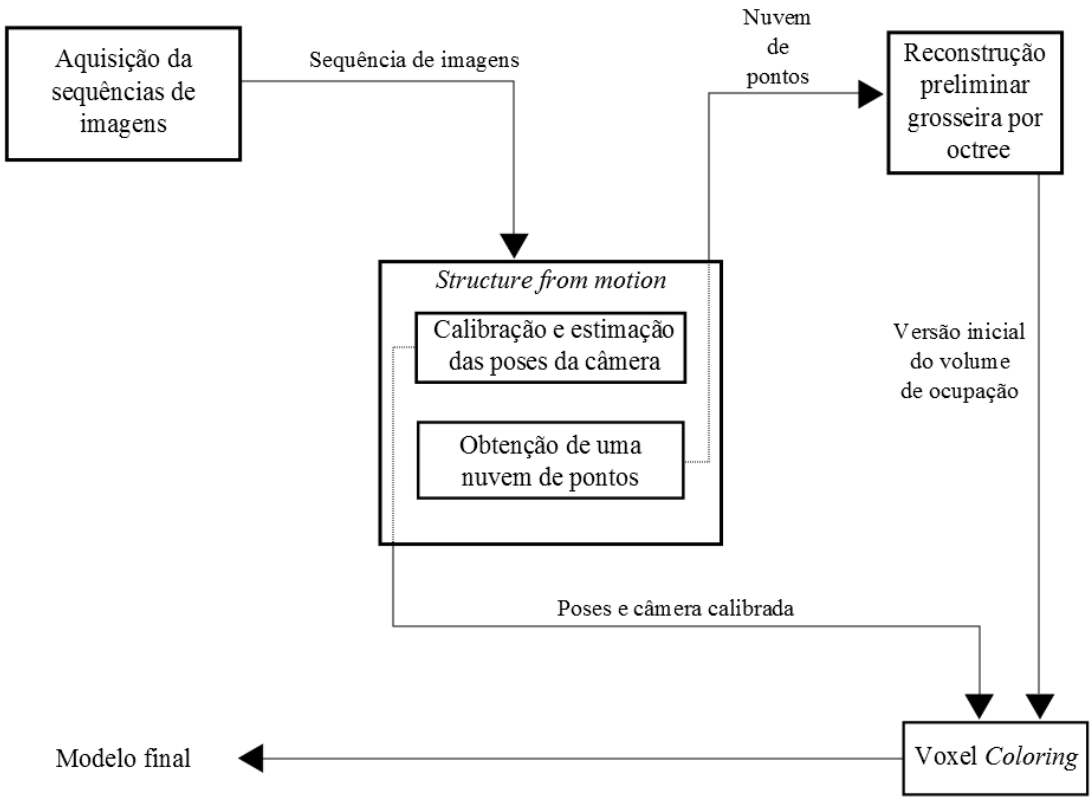


Figura 5.3- Fluxograma das etapas do método proposto.

Capítulo 6 – **EXPERIMENTOS E RESULTADOS**

A fim de avaliar a metodologia proposta neste trabalho, foi necessário produzir experimentos. Todo o equipamento e programas necessários para cumprir esta tarefa são apresentados na seção 6.1.

Basicamente os experimentos consistem em registrar uma sequência de vídeo do objeto que se deseja reconstruir, levando sempre em consideração as características discutidas na seção 6.2. Embora não seja um fator obrigatório, a resolução dos quadros das sequências de vídeo gravadas neste experimento é de 640 x 480 pixels. A sequência de vídeo é longa o suficiente para registrar uma volta completa da base giratória.

Os resultados de cada etapa são apresentados na seção 6.3. É importante ressaltar que em contraste ao trabalho de Burrus [18], a abordagem adotada nesta pesquisa também funcionaria em uma situação na qual a câmera não é fixa, isto é, a câmera se move em relação a uma cena estática.

6.1 *HARDWARE* E *SOFTWARE* UTILIZADOS

A técnica apresentada neste trabalho foi implementada em C++. O código fonte foi compilado na IDE Microsoft® Visual Studio 2010. Para desenvolver o *software*, foi utilizado um *notebook* HP Pavilion dv6 com as seguintes especificações: Intel® Core™ I7-2720QM CPU @ 2.20GHz 2.20GHz, 8.00GB de memória RAM, placa gráfica Radeon™ HD 6770M e sistema operacional Microsoft® Windows™ 7 64-bit.

O equipamento utilizado para obtenção das imagens foi a câmera digital DSRL SONY® α37 STL-A37K com resolução de 16.1 megapixels.

Na realidade a câmera captura um vídeo que tem seus quadros extraídos e transformados em imagens JPEG. Para essa tarefa, foi utilizado o programa *Free Video to* JPG *Converter* v.5.0.22 *build* 128.

A estimativa dos parâmetros da câmera foi feita utilizando o programa *Voodoo Camera Tracker* 1.1.0 *for Windows.*

Além da reconstrução de um modelo real, foi realizado um teste com um modelo sintético. Para a modelagem de uma cena sintética, a qual pudesse ser utilizada pelo *Voodoo Camera Tracker,* utilizamos o programa *Blender* 2.65.

6.2 CARACTERISTICAS DO EXPERIMENTO

No experimento com o modelo real a câmera é fixada em um tripé de modo que, aponte para o modelo que se deseja reconstruir e fique posicionada acima dele, satisfazendo a restrição necessária a utilização do método *Voxel Coloring*. Este modelo fica posicionado em cima de uma base giratória em movimento.

Deve se ter o cuidado de realizar o experimento em um ambiente adequado para produzir imagens com poucos ruídos. A sequência de vídeo é tomada sob uma fonte de luz estática e não são utilizados padrões de calibração. Apenas um subconjunto de quadros da sequência de vídeo é utilizado no experimento.

6.3 RESULTADOS

Nesta seção serão apresentados os resultados obtidos através dos experimentos. Primeiro cada um dos dados utilizados nos testes comparativos é brevemente descrito e caso seja possível, apresentamos a reconstrução por SFM, que gera estruturas de pontos esparsos. Em seguida a análise qualitativa é feita considerando os métodos de *Voxel Coloring* e o método proposto neste trabalho. Depois analisamos o desempenho desses métodos e avaliamos o impacto do uso da octree na abordagem proposta.

6.3.1 DADOS UTLIZADOS NOS TESTES

Os testes foram realizados sobre dados sintéticos e reais. Dados sintéticos consistem em imagens geradas através da renderização de objetos poligonais por câmeras sintéticas. Esses dados são descritos no formato *Wavefront* e foram obtidos diretamente da Internet. O modelo sintético utilizado é denominado Tyrant e antes de ser utilizados nos testes, foi manipulado através da ferramenta *Blender* para compor uma cena sintética que pudesse ser processada pelo *Voodoo Camera Tracker*. Denominamos este dado por TyrantScene (Figura 6.1).

Os dados reais denominados Rivals (Figura 6.2), é uma cena composta por um plano texturizado e um objeto de plástico composto por duas formas humanas, ricas em detalhes estruturais e texturas.

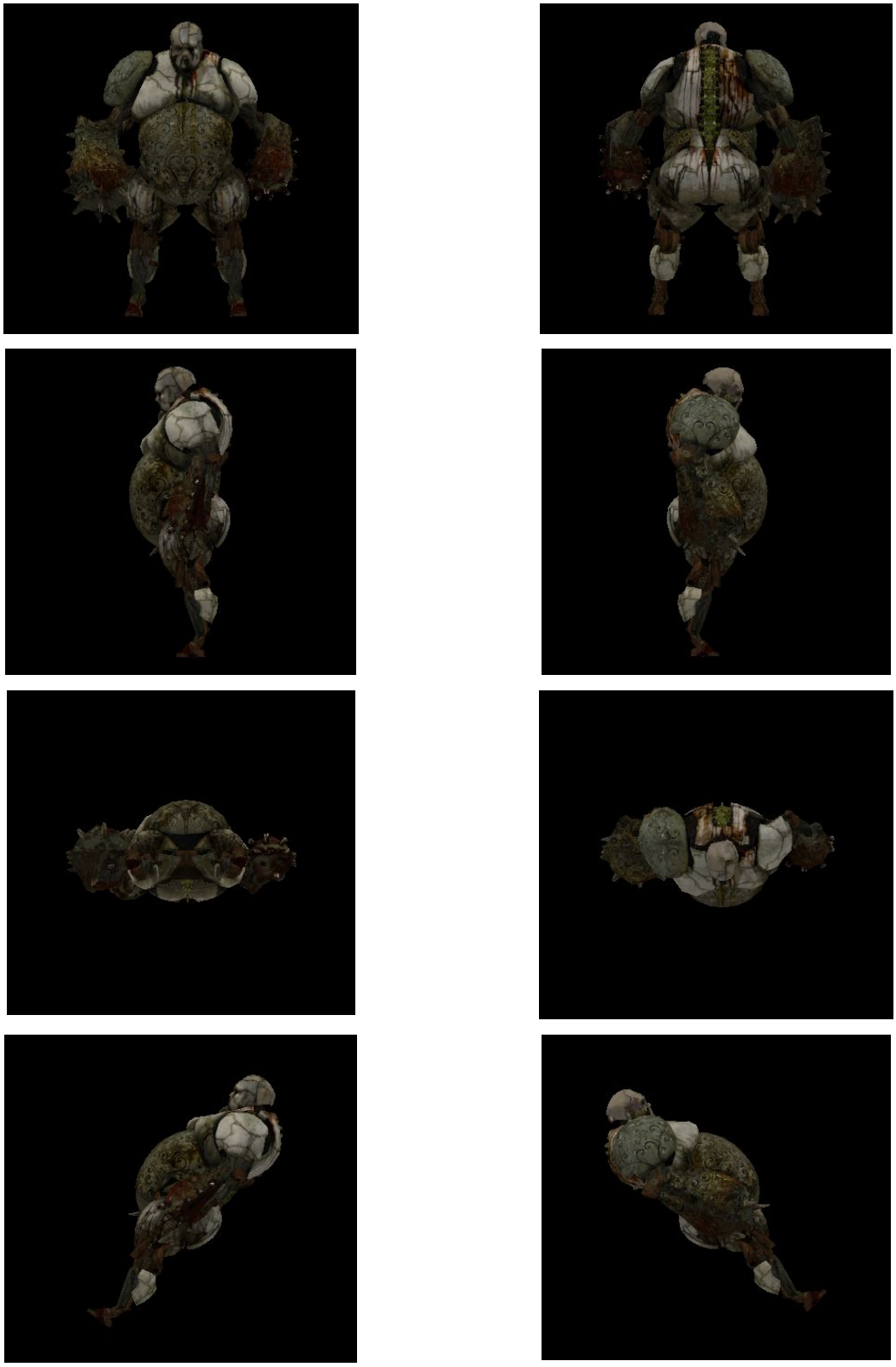


Figura 6.1- Dado Tyrant. Imagem de entrada

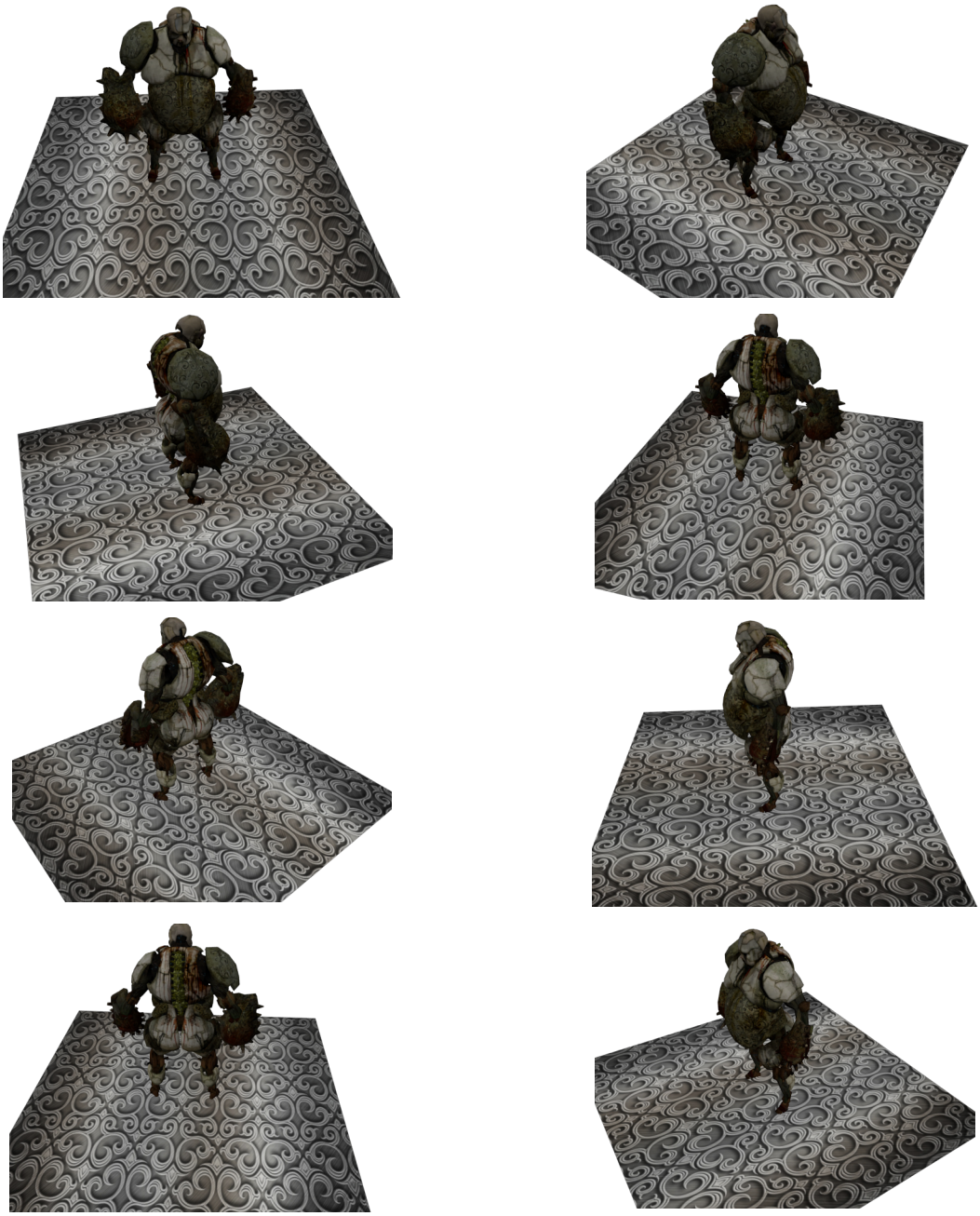


Figura 6.2 Dado TyrantScene. Cena de entrada sintética para o SFM

Figura 6.3 – Dado Rivals. Imagem de entrada real para o SFM

6.3.2 RECONSTRUÇÃO POR *STRUCTURE FROM MOTION*

Nesta seção somente iremos mostrar a reconstrução de nuvens de pontos geradas pelo SFM para os dados TyrantScene e Rivals, pois estes são os dados utilizados no método proposto neste trabalho. Com o intuito de facilitar a detecção de pontos de interesse na cena, utilizamos planos texturizados, tanto na cena real como na cena sintética. Os dados sintéticos necessitam criticamente desses planos pois, em geral, não apresentam texturas. A ausência de textura no caso de malhas geométricas sem muitos detalhes, pode prejudicar a reconstrução de pontos 3D por SFM. A Figura 6.4 e Figura 6.5 mostram respectivamente nuvens de pontos reconstruídas a partir dos dados TyrantScene e Rivals.

Pode-se observar que no caso do modelo TyrantScene, a nuvem de pontos gerada pelo SFM é relativamente uniforme o que favorece a geração de um volume de ocupação mais bem definido. O mesmo não pode ser afirmado para o dado Rivals onde a distribuição dos pontos é bem menos uniforme. Neste caso, as células geradas pela estrutura da octree são mais grosseiras, de forma a gerar um volume de ocupação inicial relativamente conexo.

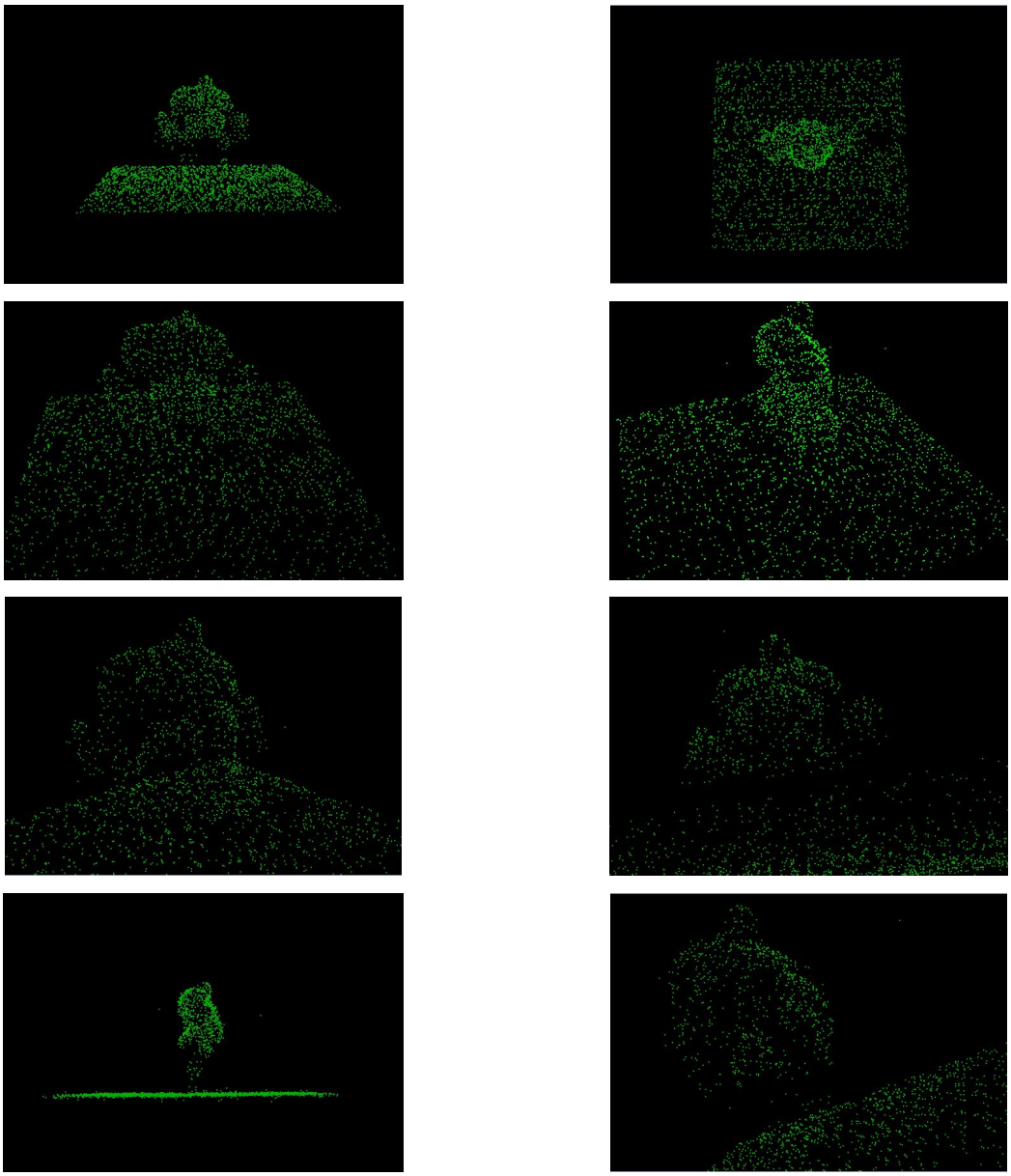


Figura 6.4 – Dado TyrantScene. Reconstruído por *Structure from Motion*

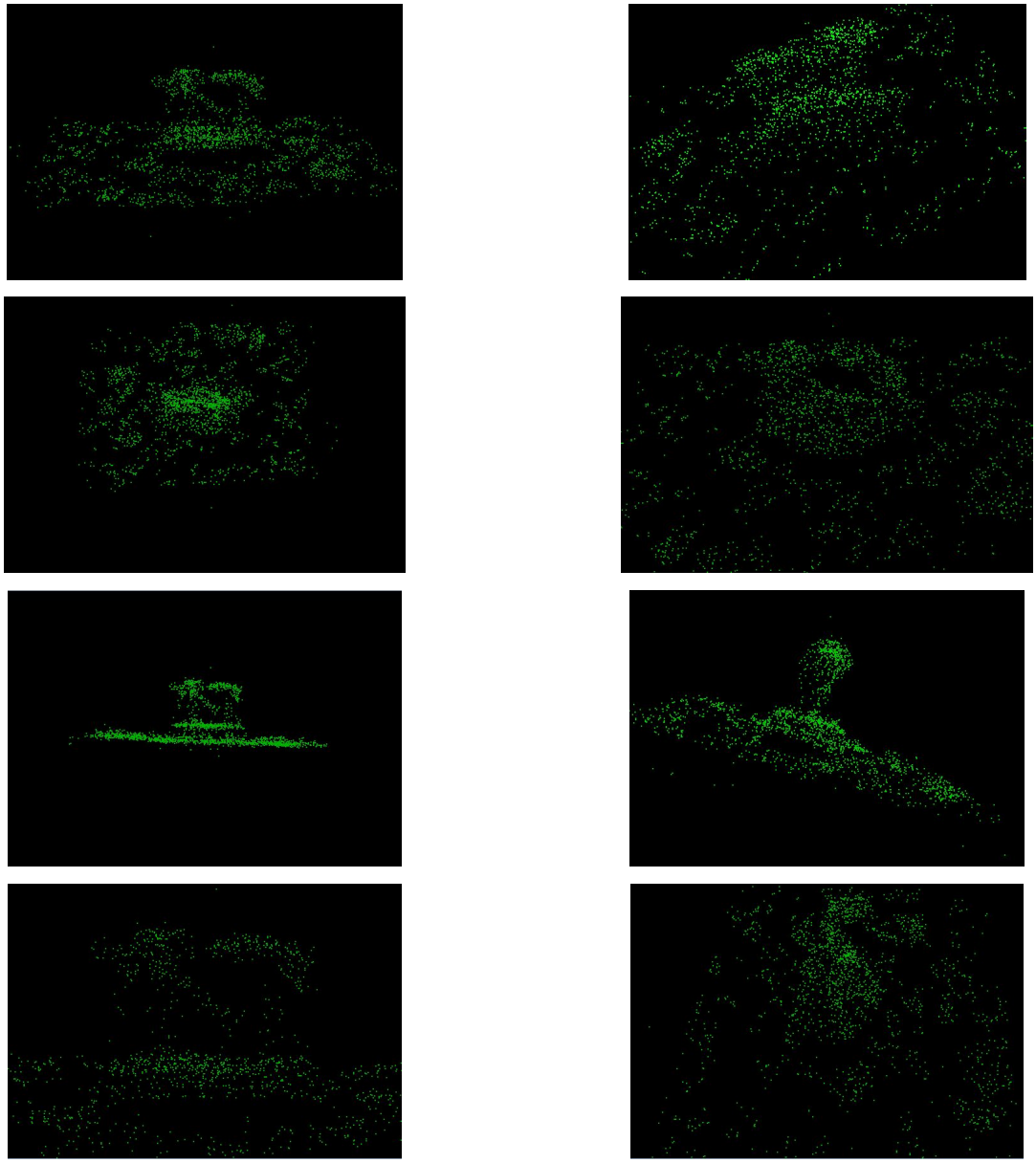


Figura 6.5 - Dado Rivals. Reconstruído por *Structure from Motion*

6.3.3 ANÁLISE QUALITATIVA

Apresentamos agora algumas imagens obtidas através da reconstrução realizada por *Voxel Coloring.* Em seguida resultados obtidos usando o método proposto serão mostrados.

A Figura 6.6 apresenta os resultados obtidos ao final da reconstrução por *Voxel Coloring*, onde a primeira imagem é uma reconstrução a partir de um ponto de vista original e as demais são novos pontos de vista. Repare que a imagem à direita na terceira fileira quase não tem informação de cor. A explicação para isso se dá pelo fato que nenhuma das câmeras sintéticas tem vista para essa região do modelo sintético.

Na Figura 6.7, temos uma comparação entre o dado original e o dado reconstruído. A reconstrução feita apresenta vários voxels na superfície que não receberam informação de cor por terem sido classificados erroneamente como não visíveis. Os dados originais estão na coluna à esquerda e os reconstruídos à direita.

A Figura 6.8 nos mostra resultados obtidos utilizando o conjunto de dados TyrantScene e o método proposto neste trabalho. Perceba que o resultado apresentado sofre grande influência de *aliasing,* que é um efeito indesejável e geralmente ocorre quando um mapa de textura apresenta regiões com frequências muito elevadas se comparadas à taxa de amostragem. O algoritmo de *Voxel Coloring* implementado não contempla uma solução para o problema de *aliasing*, o que explica a menor fidelidade ao conjunto de dados nas reconstruções feitas pelo método proposto. Na Figura 6.9, mostramos uma comparação entre o dado original e o dado reconstruído em algumas vistas diferentes.

A Figura 6.10 compara o dado Tyrant, reconstruído por Voxel Coloring e o dado TyrantScene, reconstruído por coloração de voxels e *Structure from Motion*. Apesar de serem dados diferentes, é possível entender com essa comparação que a abordagem adotada por este trabalho não apresenta um resultado tão fiel à imagem original quanto o método *Voxel Coloring.*

Na Figura 6.11 mostramos os resultados da reconstrução utilizando o método de coloração de voxels e SFM do conjunto de dados Rivals. Novamente, as reconstruções são prejudicadas por causa do *aliasing,* no entanto, devido à riqueza de detalhes deste conjunto, o teste resultou em uma reconstrução que se aproxima da imagem real . A Figura 6.12 mostra uma comparação com a cena real e a cena reconstruída em algumas vistas diferentes. Finalmente, na figura 6.13 e 6.14 mostramos respectivamente os dados TyrantScene e Rivals, reconstruídos, em resolução menor, pelo método proposto neste trabalho, com e sem o uso da octree. É possível perceber que quando a octree não é utilizada a reconstrução é infiel ao dado de entra-

da. Quando impedimos o uso da octree, passamos para uma abordagem de reconstrução somente por *Voxel Coloring,* que é um método que depende fortemente - principalmente na ausência de uma riqueza de texturas - da segmentação do plano de fundo das imagens usadas como dados de entrada para a reconstrução, segementação esta que não é considerada neste trabalho como dado de entrada para os algoritmos

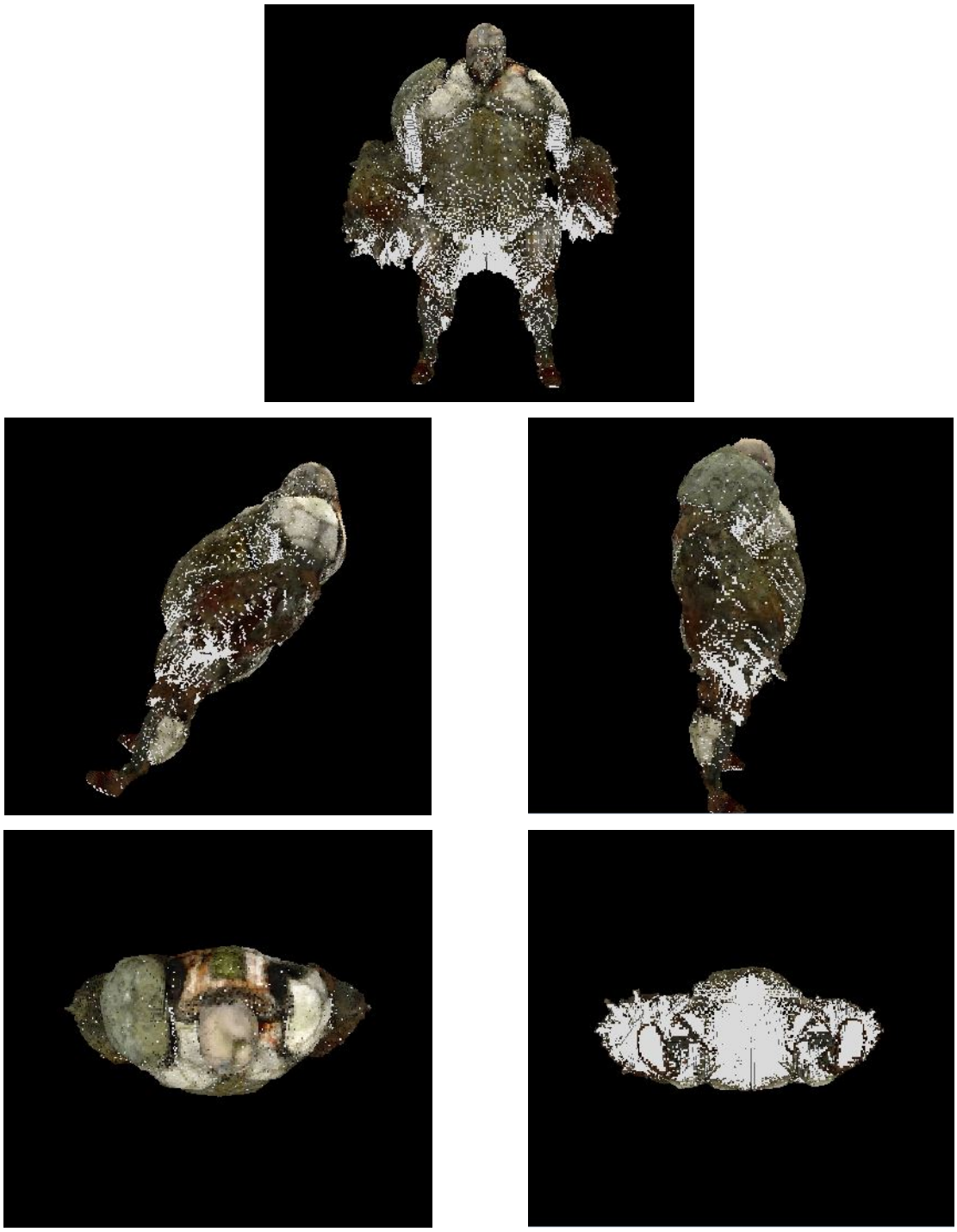


Figura 6.6 Reconstrução do conjunto de dados Tyrant por *Voxel Coloring*

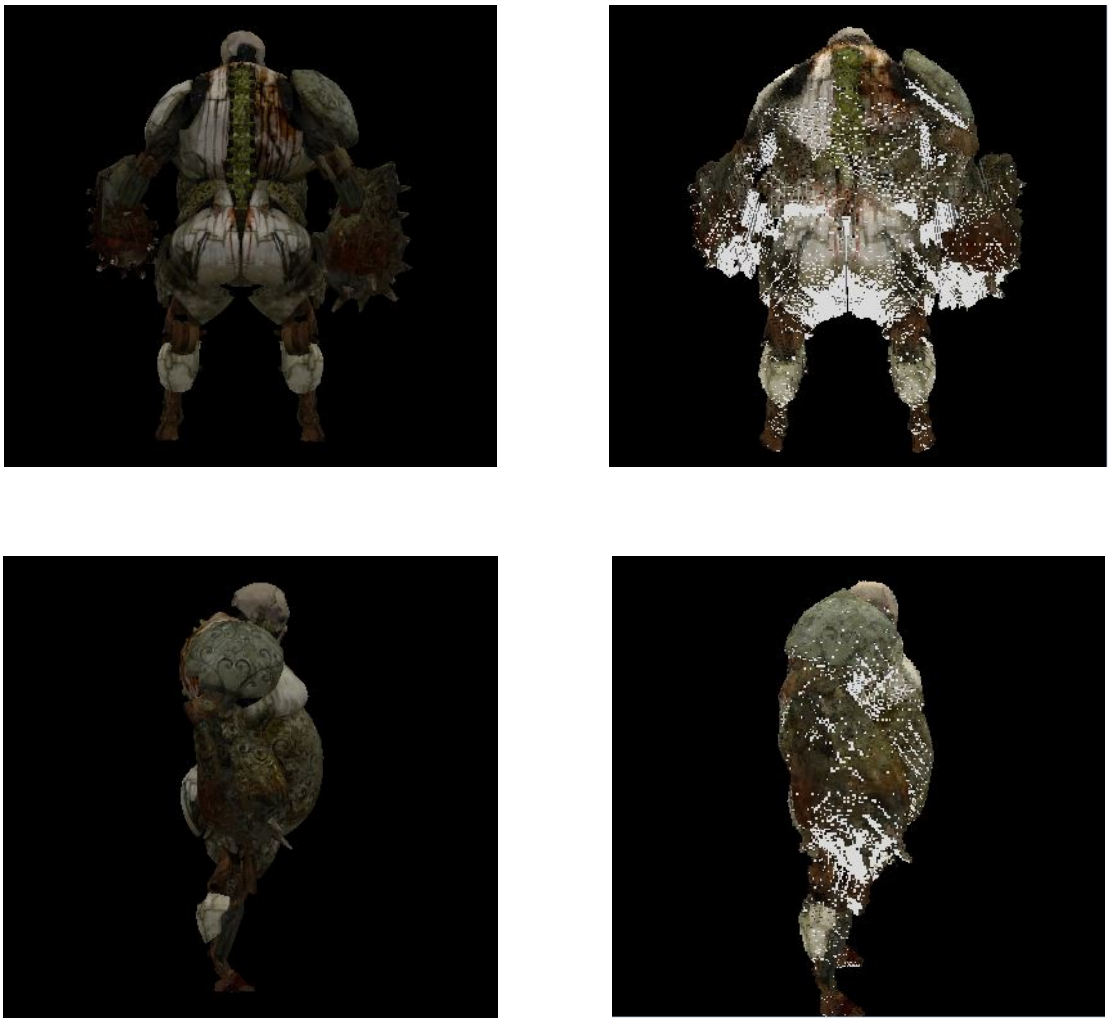


Figura 6.7- Comparação entre o dado de entrada (esquerda) e a cena reconstruída somente por *voxel coloring* (direita)

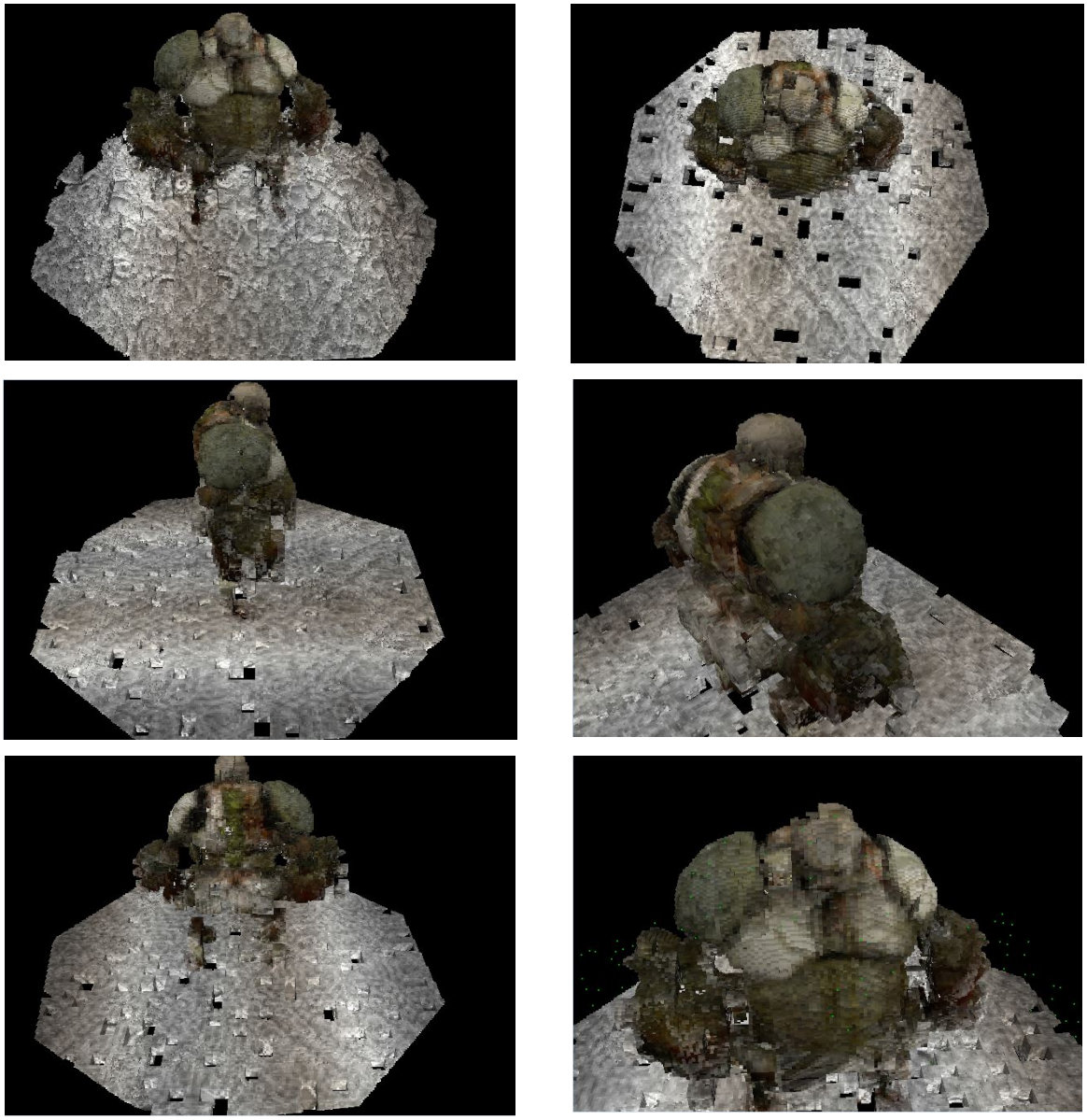


Figura 6.8 – Dado Tyrant. Reconstrução feita por coloração de voxels *e structure from motion*

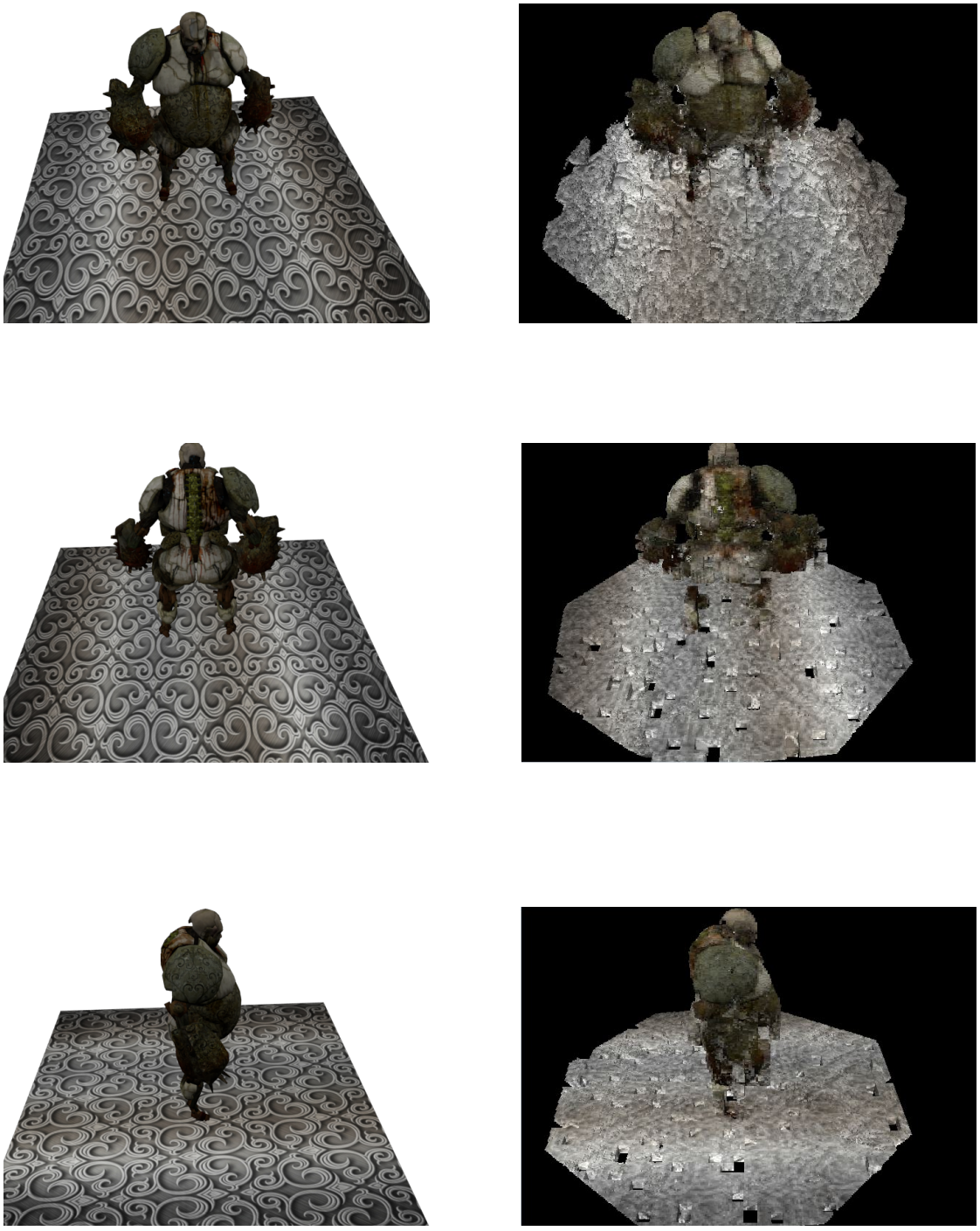


Figura 6.9 – Dado TyrantScene. Comparação entre o dado original (esquerda) e o dado reconstruído (direita)

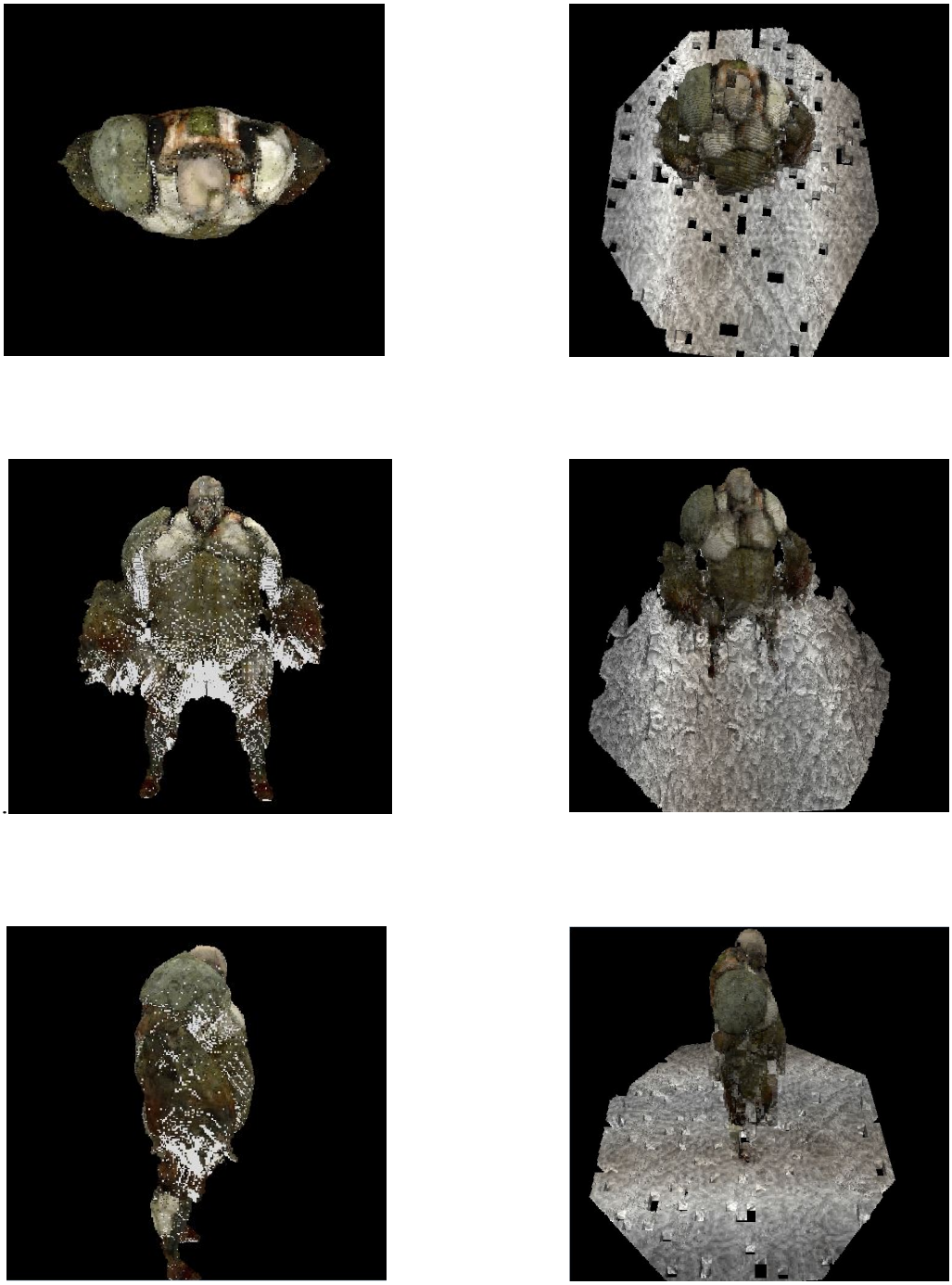


Figura 6.10 – Dado Tyrant (esquerda) reconstruído por *Voxel Coloring* e dado TyrantScene (direita) reconstruído por coloração de voxels e *Structure from Motion*

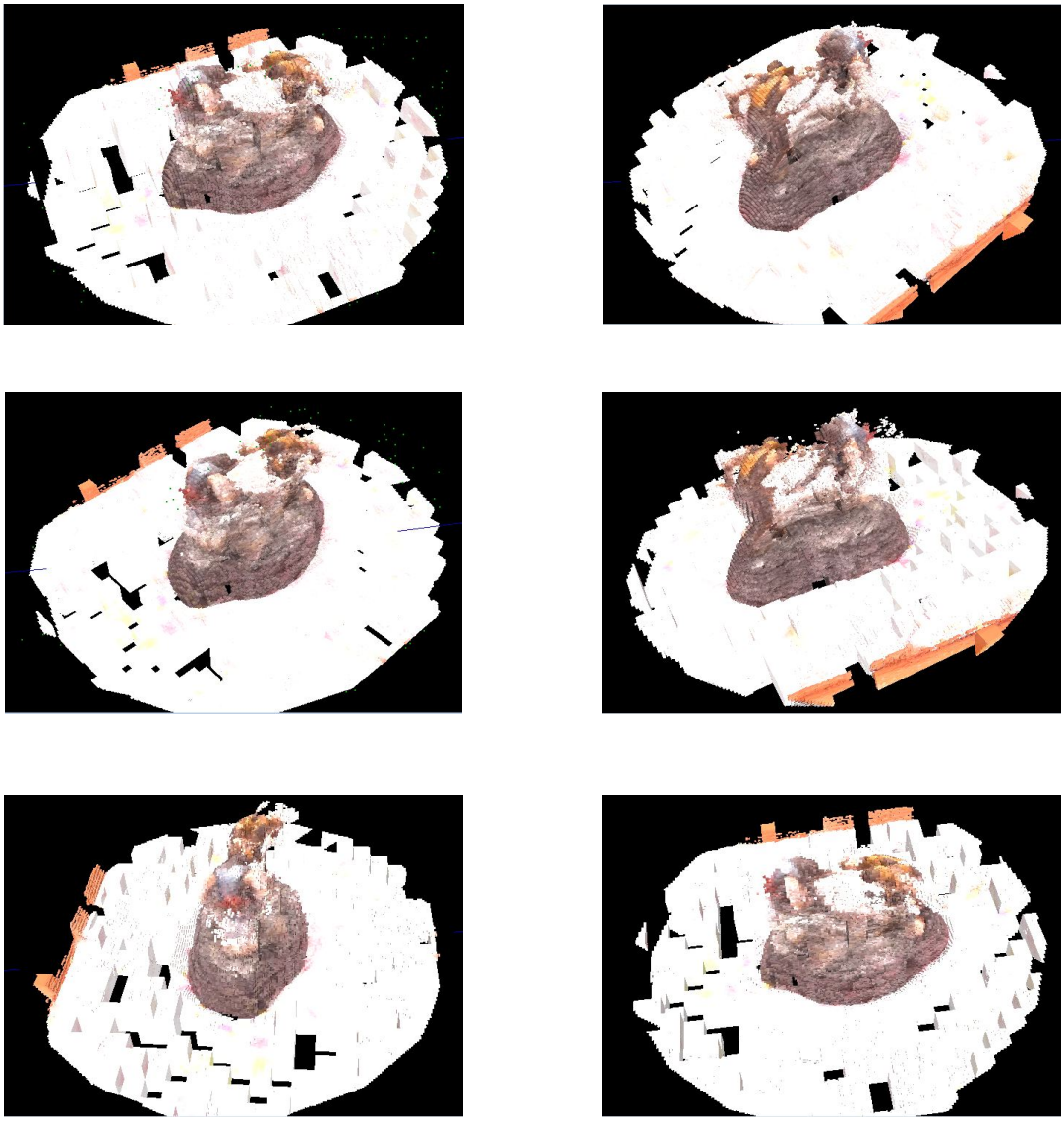


Figura 6.11 Dado Rivals. Reconstruído por coloração de voxels e *Structure from Motion*

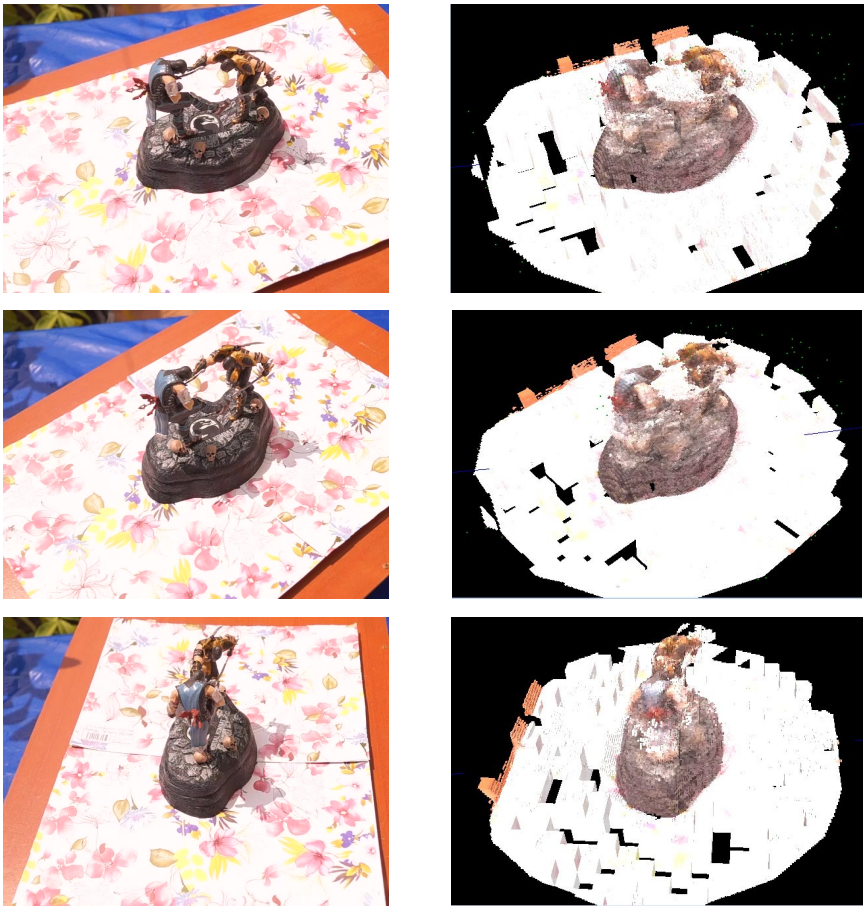


Figura 6.12 - Dado Rivals. Comparação entre o dado original (esquerda) e o dado reconstruído (direita)

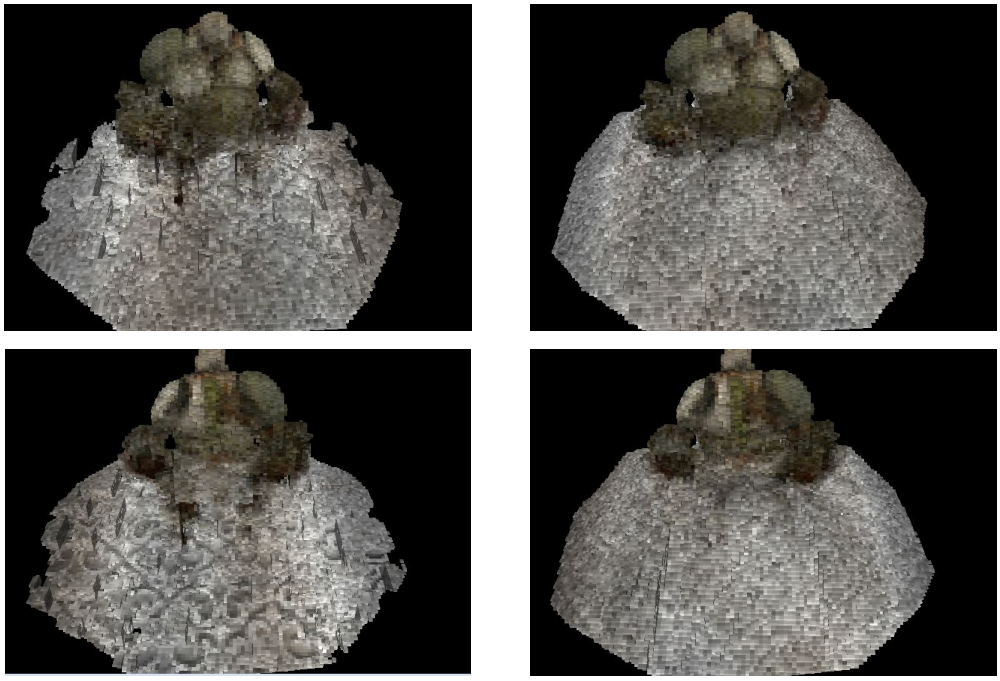


Figura 6.13 - Dado TyrantScene. Reconstrução por coloração de voxels e *Structure from Motion* com (esquerda) e sem (direita) octree

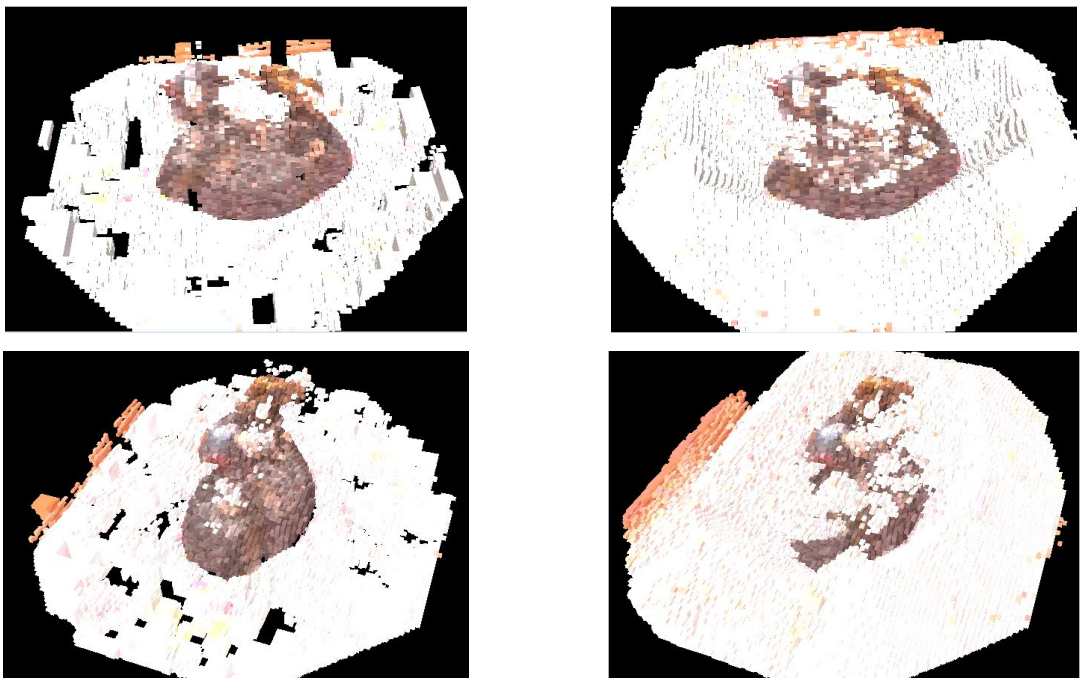


Figura 6.14 – Dado Rivals. Reconstrução por coloração de voxels e *Structure from Motion* com (esquerda) e sem (direita) octree

6.4 ANÁLISE DE DESEMPENHO

As tabelas de 6.1 a 6.5 mostram resultados obtidos individualmente pelo método proposto neste trabalho e por *Voxel Coloring* no conjunto de dados utilizados.

Analisando as tabelas, podemos perceber que o método proposto, combinando coloração de voxels e *Structure from Motion*, é mais rápido que o método *Voxel Coloring* usado isoladamente. Isto se deve ao fato de que a abordagem proposta usa uma octree de modo a permitir uma reconstrução preliminar da cena. Dessa forma, quando a reconstrução final por *voxel coloring* é iniciada, o volume de ocupação já não possui o mesmo número de voxels que teria caso a octree não fosse utilizada. Analisando as tabelas 6.4 e 6.5 podemos evidenciar isso. Nestas tabelas, a primeira linha se refere a uma reconstrução utilizando método proposto. A última linha reflete uma reconstrução que usa o mesmo método, mas o uso da octree é vetado, isto é, o volume de ocupação não é esculpido de forma preliminar.

Deve-se deixar claro entretanto que o tempo utilizado pelo SFM não é considerado, já que aqui é considerado como um etapa de pré-processamento.

Resolução	total de voxels	voxels removidos	voxels consistentes	tempo
256x256x256	16777216	15599083	1178133	63s

Tabela 6.1- *Voxel Coloring* aplicado ao conjunto de dados Tyrant

Resolução	total de voxels	voxels removidos	voxels consistentes	tempo
256x256x256	2101870	978648	1123222	59s

Tabela 6.2 – Coloração de voxels e *Strucutre from Motion* aplicado ao conjunto de dados TyrantScene

Resolução	total de voxels	voxels removidos	voxels consistentes	tempo
256x256x256	1533744	735323	798421	70s

Tabela 6.3 – Coloração de voxels e *Structure from Motion* aplicado ao conjunto de dados Rivals

Resolução	total de voxels	voxels removidos	voxels consistentes	tempo	Octree
128x128x128	194464	126953	67511	5s	sim
128x128x128	2097152	1786313	310839	30s	não

Tabela 6.4 - Dado Rivals. Reconstrução por coloração de voxels e *Structure from Motion.*

Resolução	total de voxels	voxels removidos	voxels consistentes	tempo	Octree
128x128x128	255875	114995	140880	7s	sim
128x128x128	2097152	1595069	502083	54s	não

Tabela 6.5 - Dado TyrantScene. Reconstrução por coloração de voxels e *Structure from Motion*

Capítulo 7 – **CONCLUSÃO E TRABALHOS FUTUROS**

São muitas as áreas onde a reconstrução volumétrica de cenas pode ser utilizada. É importante que qualquer método utilizado nesta tarefa de reconstrução possa alcançar o objetivo de reproduzir o modelo real da forma mais fiel possível. No entanto, também é importante que não seja uma tarefa complicada de ser cumprida, isto é, quanto mais praticidade houver para reconstruir um modelo melhor.

Neste trabalho foi apresentado um método para reconstruir cenas de forma volumétrica utilizando SFM e *Voxel Coloring*. O *Structure From Motion* é usado para calibração e estimação das poses da câmera e geração de uma nuvem de pontos que fornece uma aproximação esparsa do modelo real. Através da octree, é possível utilizar a nuvem de pontos para tornar possível a obtenção de uma reconstrução preliminar e grosseira do modelo. Isto é, uma versão inicial do volume de ocupação que é usada como dado de entrada para o algoritmo de *Voxel Coloring*. Foi também discutido que, devido ao uso do SFM, recuperar os parâmetros intrínsecos e extrínsecos da câmera se torna uma tarefa transparente e evita o uso de padrões de calibração. As estruturas estimadas pelo SFM caracterizam uma nuvem de pontos que dá origem a uma aproximação do volume de ocupação via uma octree. Esse volume é discretizado em voxels e dessa forma está pronto para ser refinado pelo *Voxel Coloring*.

Também foi mostrada uma nova forma de isolar o modelo a ser reconstruído do resto da cena em configurações com câmeras fixas e com uma cena com objetos de interesse em movimento. Pela natureza do SFM, somente pontos que apresentam uma deslocação de uma imagem para outra são considerados como pontos característicos. Dessa maneira todo o fundo da cena, que é estático não é contemplado. Quando a qualidade das imagens de entrada é ideal, o SFM gera um conjunto de pontos tridimensionais cuja disposições no espaço se assemelham ao modelo que se deseja reconstruir.

A calibração da câmera é uma etapa transparente, pois está embutida no SFM. Deste modo, usar padrões de calibração se torna desnecessário neste trabalho. Isso faz com que a utilização deste método em alguma aplicação se torne mais simples e aceitável. Consideramos este fato um ponto forte do trabalho.

Outro ponto forte, se deve ao uso de uma octree, que permite uma reconstrução preliminar grosseira do volume de ocupação, o que torna o processo final de reconstrução mais rápido.

A qualidade das reconstruções, no entanto, se tornou um ponto negativo devido ao problema de *aliasing* não tratado nesta abordagem.

7.1 TRABALHOS FUTUROS

Um problema evidente que pode ser investigado em um trabalho futuro é o problema de *aliasing*. Um investigação sobre técnicas de *anti-aliasing* é um bom ponto de partida.

Uma forma de melhorar o método utilizado é através da utilização de câmeras RGB-D como o Microsoft Kinect. Câmeras RGB-D são sensores que combinam a informação de cor com a informação de profundidade por pixel. Usando um equipamento desse tipo pode ser proveitoso, pois a obtenção das imagens e geração dos pontos de nuvens seriam feitas em uma só etapa por um mesmo equipamento. O interessante neste ponto é encontrar um método robusto para obter as poses da câmera relativas a cada imagem obtida pelo sensor.

Outra possível investigação futura é reconstruir um campo escalar via método variacional usando, por exemplo, funções de base radial para estimar o volume de ocupação inicial para reconstrução via *Voxel Coloring*.

REFERÊNCIAS

[1] B. Julesz, "Binocular depth perception of computer-generated patterns.," *Bell System Technical Journal*, 1960.

[2] D. Marr and T. Poggio, "A theory of human stereo vision," 1977.

[3] A. A. Montenegro, "Reconstrução de cenas a partir de imagens através de escultura do espaço por refinamento adaptativo," Tese de Doutorado, Universidade Federal Fluminense, Rio de Janeiro, 2003.

[4] D. Marr and T. Poggio, "Cooperative computation of stereo disparity," DTIC Document, 1976.

[5] S. Seitz and C. Dyer, *Photorealistic Scene Reconstruction by Voxel Coloring*. 1997.

[6] K. N. Kutulakos and S. M. Seitz, "What Do N Photographs Tell Us about 3D Shape?," 1998.

[7] W. B. Culbertson, T. Malzbender, and G. Slabaugh, "Generalized voxel coloring," *Lecture notes in computer science*, pp. 100–115, 1999.

[8] L. Massone, P. Morasso, and R. Zaccaria, "Shape From Occluding Contours," pp. 114–120, Jan. 1985.

[9] A. PAIVA, R. SEIXAS, and M. GATTASS, "Introdução à Visualização Volumétrica." Jan-1999.

[10] C. H. Chien and J. K. Aggarwal, "Volume/surface octrees for the representation of three-dimensional objects," *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 1, pp. 100–113, Outubro 1986.

[11] R. Szeliski, "Rapid octree construction from image sequences," *CVGIP: Image Underst.*, vol. 58, no. 1, pp. 23–32, Jul. 1993.

[12] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330 – 1334, Nov. 2000.

[13] "CHAPTER 13 Structure from motion." [Online]. Available: http://mi.eng.cam.ac.uk/~cipolla/publications/publications/contributionToEditedBook/2008-SFM-chapters.pdf. [Accessed: 18-Jan-2013].

[14] K. Kraus, *Photogrammetry*. Ferdinand Dummlers Verlag, 1993.

[15] A. Zisserman, A. Fitzgibbon, and G. Cross, "VHS to VRML: 3D graphical models from video sequences," in *IEEE International Conference on Multimedia Computing and Systems, 1999*, 1999, vol. 1, pp. 51 –57 vol.1.

[16] K. N. Kutulakos and J. R. Vallino, "Calibration-Free Augmented Reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 1, pp. 1–20, Jan. 1998.

[17] R. C. Gonzales and P. Wintz, *Digital image processing (2nd ed.)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1987.

[18] N. Burrus, M. Abderrahim, J. G. Bueno, and L. Moreno, "Object Reconstruction and Recognition leveraging an RGB-D camera," in *Proceedings of the 12th IAPR Conference on Machine Vision Applications*, 2011.

[19] R. A. Newcombe and A. J. Davison, "Live dense reconstruction with a single moving camera," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 1498–1505.

[20] M. Nakazawa, I. Mitsugami, Y. Makihara, H. Nakajima, H. Habe, H. Yamazoe, and Y. Yagi, "Dynamic scene reconstruction using asynchronous multiple Kinects," in *2012 21st International Conference on Pattern Recognition (ICPR)*, 2012, pp. 469–472.

[21] "Camera Calibration - Computer Vision System Toolbox for MATLAB & Simulink." [Online]. Available: http://www.mathworks.com/products/computer-vision/description6.html. [Accessed: 29-Sep-2013].

[22] M. Campos and V. Neto, “DCC884 – Visão Computacional Calibração de Câmeras,” Minas Gerais, Maio de-2007.
[23] C. Harris and M. Stephens, “A combined corner and edge detector,” in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.
[24] H. C. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections,” *Nature*, vol. 293, no. 5828, pp. 133–135, print Setembro 1981.
[25] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.
[26] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.
[27] E. Kruppa, *Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung*. Hölder, 1913.
[28] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*. Cambridge, MA, USA: MIT Press, 1993.
[29] R. I. Hartley and P. Sturm, “Triangulation,” *Computer vision and image understanding*, vol. 68, no. 2, pp. 146–157, 1997.
[30] “LM_Teoria.pdf.” [Online]. Available: http://www.tecgraf.puc-rio.br/~mgattass/LM_Fabiola/LM_Teoria.pdf. [Accessed: 05-Oct-2013].
[31] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.