

## Um método de Avaliação Genérico para Sistemas de Visualização Baseados em Imagens

Esteban W. Gonzalez Clua  
Computer Science Department, PUC-Rio  
[esteban@inf.puc-rio.br](mailto:esteban@inf.puc-rio.br)

Bruno Feijó  
Computer Science Department, PUC-Rio  
[bruno@inf.puc-rio.br](mailto:bruno@inf.puc-rio.br)

Marcelo Dreux  
Department of Mechanical Engineering, PUC - Rio  
[dreux@mec.puc-rio.br](mailto:dreux@mec.puc-rio.br)

PUC-RioInf.MCC29/03 August, 2003

**Abstract:** Much progress had been obtained in image-based rendering techniques and many others ideas are been developed and implemented in this area. Although, there isn't an ideal system that implements the plenoptic function in its totality in a viable aspect in relation to memory, CPU process and authority. The known and developed systems try to create a sub-set of the domain and image of the function, bringing in each implementation restrictions and limitations. These work is at the same time a survey of the main developed works and a tentative to create a set of criteria that allows a critic and comparative evaluation of these systems.

**Keywords:** Image-based rendering, plenoptic function.

**Resumo:** Têm-se obtido bastantes progressos em técnicas de image-based rendering e vêm surgido inúmeras abordagens para implementação de aplicações nesta área. Entretanto, não existe um sistema ideal, que implemente a função plenóptica na sua totalidade e que ao mesmo tempo seja viável em termos de memória, processamento e autoria. Assim, os diversos sistemas procuram criar um sub-conjunto do domínio e da imagem desta função, sendo que em cada abordagem sempre existem restrições e limitações. Este trabalho tem como objetivo fazer um resumo sobre as principais técnicas conhecidas, bem como criar um conjunto de critérios objetivos que permitam uma avaliação crítica e comparativa dos diversos trabalhos propostos.

**Palavras-chave:** Image-based rendering, função plenóptica

## 1. Introdução - A função plenóptica

A técnica de Image-based rendering surgiu a partir de uma combinação entre as técnicas de Computação Gráfica 3D e Visão Computacional, tendo em vista adquirir resultados foto-realistas e ao mesmo tempo rápidos na visualização de cenas. Atualmente, com o avanço do poder computacional das placas gráficas, esta técnica ganhou grande importância para aplicações de realidade virtual, simulações e jogos.

Para uma definição mais formal do que vem a ser ibr, deve-se recorrer ao conceito de função plenóptica (*plenus* = completa e *optic* = visão), definida inicialmente por [Adelson 91]. Esta é uma função parametrizada que descreve tudo o que é possível de ser visto a partir de qualquer parte do espaço, a qualquer momento e a qualquer comprimento de onda. Em [McMillan et al 95] a função é definida como:

$$p = P(\mathbf{q}, \mathbf{f}, \mathbf{l}, V_x, V_y, V_z, t)$$

Onde  $(V_x, V_y, V_z)$  é a posição onde o observador se encontra,  $\mathbf{q}$  e  $\mathbf{f}$  são os ângulos de azimute e elevação do vetor do observador e  $\mathbf{l}$  é a frequência da luz a ser observada. Para o caso de uma cena dinâmica, adiciona-se a variável  $t$  para descrever o tempo.

Pode-se entender esta função plenóptica como uma função capaz de descrever todas as possíveis vistas para uma dada cena. Diz-se que esta função está completa se para um dado ponto toda a esfera que está a sua volta pode ser vista e incompleta no caso de apenas uma parte desta abóbada poder ser visível. Vários autores afirmam que todos os algoritmos de image-based rendering se reduzem a uma implementação da função plenóptica.

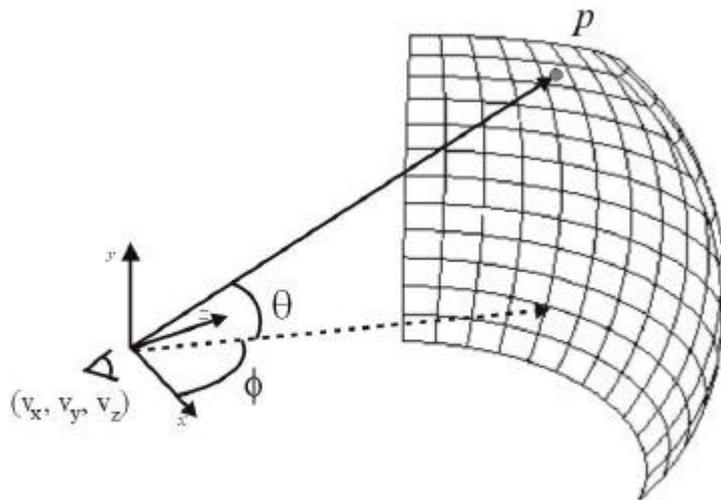


Figura 1 – Qualquer ponto visível para qualquer posição de um observador pode ser descrito pela função plenóptica.

Assim sendo, uma possível definição para a maioria dos problemas de ibr pode ser descrita da seguinte forma: “dado um conjunto de amostras ( completa ou incompleta ) da função plenóptica, o objetivo da ibr será gerar uma representação contínua desta função” [McMillan et al 95].

Devido a impossibilidade de implementação da função plenóptica no seu caso geral, os sistemas de ibr procuram criar amostras da mesma a partir de um conjunto finito e discreto de

dados. Desta maneira, toda abordagem irá possuir restrições inerentes à solução proposta, apesar dos grandes progressos que se vêm obtendo recentemente. Assim sendo, não existe - pelo menos até agora - uma solução ótima para todos os casos, o que obriga que sejam elaborados métodos comparativos, que permitam analisar e compará-los entre si, permitindo uma adequada escolha para a implementação de uma aplicação.

## 2 Trabalhos Relacionados

Em [Buehler 01 et al.] os autores pretendem dar uma lista de propriedades que um algoritmo de ibr deveria seguir e classificam posteriormente os principais trabalhos já desenvolvidos nesta área de acordo com estes tópicos. As propriedades indicadas pelos autores são:

- 1 Utilização de uma referência geométrica: Sempre que for possível de se utilizar alguma referência geométrica a partir do conjunto de imagens, isto deve ser feito, de maneira a reconstruir de forma mais perfeita o raio de visão desejado;
- 2 Entrada não estruturada: Deve-se dar preferência a algoritmos que tenham como dados de entrada imagens de cameras posicionadas em locais genéricos e que não devam seguir alguma regra de posicionamento;
- 3 Consistência Epipolar: Quando um raio desejado passa pelo centro de projeção de uma camera este raio pode ser reconstruído trivialmente. Neste caso, um algoritmo ideal deveria utilizar um raio da imagem fonte. A consistência epipolar irá reconstruir este raio corretamente, sem deformação da geometria;
- 4 Desvio angular mínimo: O raio a ser escolhido é aquele que tem um ângulo menor entre a posição do observador e do conjunto de todas as imagens do banco de dados disponível. Esta medida nem sempre pode ser feita da forma mais adequada e correta;
- 5 Continuidade: Ao variar alguns dos parâmetros da função plenóptica, deve haver uma continuidade entre as sequências de imagens formadas;
- 6 Sensibilidade à resolução: As diversas imagens de um banco de dados possuem em muitos casos regiões comuns. Na reconstrução deve-se procurar escolher sempre aquela que apresente maior resolução possível para uma determinada área a ser reconstruída;
- 7 Consistência de raios equivalentes: Havendo diversas formas possíveis de reconstrução para uma imagem, deve-se manter uma consistência entre as diversas soluções;
- 8 Tempo Real: A função plenóptica deve permitir mudar algumas de suas variáveis, garantindo que esta mudança ocorra em tempo real.

A tabela 1 mostra uma comparação, com base nos tópicos indicados, dos principais trabalhos publicados em ibr. [Buehler 01 et al.]

	A	B	C	D	E	F	G	H	I
1- Referência Geométrica		Ok							
2- Entrada não Estruturada	Ok	Ok	Ok				Ok	Ok	Ok
3- Consistência Epipolar									Ok
4- Desvio angular mínimo		Ok							
5- Continuidade	Ok	Ok	Ok	Ok	Ok	Ok		Ok	Ok
6- Sensibilidade à resolução	Ok	Ok	Ok	Ok		Ok	Ok	Ok	Ok
7- Consistência de raios equivalentes			Ok		Ok	Ok		Ok	Ok
8- Tempo Real	Ok	Ok		Ok	Ok	Ok	Ok	Ok	Ok

Tabela 1 – Comparação sugerida por [Buehler 01 et al.] entre diversos trabalhos desenvolvidos em image-based rendering. A = [Levoy et al. 96], B = [Gortler et al. 96], C = [Debevec et al. 96], D = [Pulli et al. 97], E = [Debevec et al 98], F = [Pighin et al. 98], G = [Heigi et al. 99], H = [Wood et al. 00], I = [Buehlere 01 et al].

Entretanto, estas propriedades estão restritas a aspectos da correteza na visualização - a violação de alguma destas propriedades poderá acarretar num resultado não esperado e não satisfatório na imagem final - não ajudando na avaliação de um sistema de ibr em sua totalidade, tal como suas restrições impostas, eventuais dificuldades na fase de autoria ou dimensão do sub-conjunto da função plenóptica.

Em [Aliaga 01] os autores fazem uma comparação de alguns dos principais métodos de ibr baseados na dimensão do sub-conjunto da função plenóptica criada pelo sistema.

### **3 Critérios necessários para uma avaliação crítica de um sistema de IBR**

Sugere-se neste artigo uma metodologia crítico-comparativa, através de um conjunto de itens propostos, capaz de descrever detalhadamente qualquer sistema de ibr, tendo em vista não apenas aspectos de visualização, mas também limitações impostas.

#### **3.1 Processo de autoria**

A viabilidade de um sistema pode ser dada em muitos casos pela dificuldade ou facilidade que o processo de autoria requer. Um sistema ideal seria aquele onde qualquer usuário poderia construir o ambiente, sem a necessidade de equipamentos especiais ou sofisticados.

Algumas aplicações requerem, por exemplo, a profundidade de cada pixel. Esta informação pode ser obtida em alguns casos na fase de captura da imagem, sendo necessário para tal a utilização de equipamentos especiais. No caso de imagens sintéticas, esta informação é fácil de ser obtida, bastando armazenar o seu Z-buffer correspondente. Por outro lado, a aplicação pode apresentar ferramentas para extrair estes dados através de uma aproximação da geometria da imagem, como em [Oh et al. 01].

Caso o sistema utilize um panorama para construir um environment map, é importante distinguir qual a geometria do objeto que servirá para tal. Para aplicações de panoramas cilíndricos será necessária uma ferramenta de stitcher, como apresentado em [Szeliski 96] para compor uma sequência de fotos. Já para panoramas esféricos ou cúbicos serão necessários aparelhos de captura ou métodos de composição mais sofisticados como em [Aliaga 01].

Pode ser interessante indicar qual o método utilizado para estimar a posição de câmera, caso isto seja necessário para o algoritmo. Em [Takahashi et al. 00], por exemplo, o método indica a utilização de um sistema de Global Positioning – System (GPS), o que inviabiliza sua implementação para muitos casos.

Neste item cabe indicar se a entrada de dados é estruturada ou não [Buehler 01 et al]. Caso a entrada seja estruturada, a fase de autoria será mais complexa.

#### **3.2 Número de variáveis da função plenóptica que podem ser manipuladas continuamente e em Tempo Real (graus de liberdade)**

Na prática, todos os métodos de visualização baseados em ibr possibilitarão a manipulação de um sub-conjunto de variáveis (menor ou igual a 7) da função plenóptica. Quanto maior o número de variáveis livres, maior será o grau de liberdade permitido na interatividade do sistema implementado.

Assim por exemplo, restringindo o problema para cenas estáticas e para um comprimento de onda fixo, reduz-se a função plenóptica para 5 dimensões. [McMillan et al 95] usa imagens com valores de profundidade para os pixels de forma a reconstruir uma função de 5 dimensões. Outros exemplos de trabalhos que reduzem a função plenóptica para esta dimensão são [Chen 93] e [Kang 96].

Para espaços sem obstrução pode-se reduzir a função para 4 dimensões, fazendo com que a cena ou o observador estejam presos a uma caixa ou um cilindro. São exemplos desta redução os

trabalhos de [Levoy et al. 96] e [Gortler et al. 96].

No trabalho de [Shum et al 99] os autores capturam imagens e fazem com que o movimento da câmera esteja preso a círculos concêntricos e paralelos ao chão, sendo um exemplo de redução da função para 3 dimensões.

Finalmente, para o caso de se fixar o observador e se permitir alterar apenas a direção do observador e o fator de zoom, pode-se chegar a uma função plenóptica bidimensional. Exemplos típicos para estes casos são os panoramas esféricos e cilíndricos, onde se destacam os trabalhos de [Chen 95], [Szeliski 96] e [Szeliski et al. 97].

### **3.3 Continuidade e constraints impostos na mudança do valor dos parâmetros $V_x$ , $V_y$ , $V_z$ da função plenóptica**

Estes parâmetros descrevem a posição onde se encontra o observador. Alterar o valor de alguma destas variáveis corresponde a caminhar com o observador numa cena. Alguns sistemas apenas permitem alterar estes valores de forma discreta, o que faz o observador dar saltos de um local para outro, tais como [Chen 95], [Szeliski et al. 97]. A vantagem destes sistemas é que na maioria das vezes, o processo de autoria é mais fácil e não é necessário obter informações de profundidade dos pixels. São exemplo de sistemas onde esta mudança é contínua [Buehler et al 01] e [Aliaga et al. 01].

Para fazer esta mudança de observador ser contínua, será necessário deformar e interpolar um conjunto de imagens, para pontos de vistas distintos. Estas interpolações em geral trarão alguns problemas, tais como image fold (mais de um pixel da imagem de referência passa são mapeados para um mesmo pixel da imagem resultante) e surgimento de buracos (informações que estavam oclusas na imagem inicial passam a ser necessárias para a imagem gerada).

### **3.4 Densidade de amostras para situações ideais**

Cada sistema tem sua base de dados ideal. Em muitos casos, para que a interatividade e imersão dentro de um ambiente seja completa, serão requeridos um grande número de amostras de imagens. Em geral, sistemas que requerem depth information, terão a vantagem de necessitarem amostras mais esparsas de imagens [Debevec et al. 96]. Para se fazer uma medida objetiva deste valor, pode-se criar uma relação entre o tamanho da região que se permite navegar com o número de imagens necessárias para montar este cenário. Em [Chai et al. 00] realiza-se um estudo detalhado da quantidade mínima de imagens necessárias para métodos baseados no light-field.

### **3.5 Extração de dados geométricos a partir das imagens**

Costuma-se dividir os sistemas de ibr em dois conjuntos distintos: aqueles que requerem informações da geometria da cena, tendo como um extremo algoritmos de View Dependent Texture Mapping e aqueles que não requerem nenhuma informação da geometria, tendo apenas como entrada conjuntos de imagens. Alguns autores preferem inclusive separar os conjuntos, chamando o primeiro de image-based modeling [Debevec et al. 96] e o segundo de image-based rendering. Para os casos onde se requer informações de geometria devem ser desenvolvidas ferramentas especiais de autoria, que permitam esta extração manual ou automática. Possuem em geral a vantagem de necessitarem menos imagens de entrada. Entretanto, por ser necessário realizar a estimativa da geometria da cena, o sistema dependerá da complexidade do local sendo construído.

### **3.6 Variável $t$ da função plenóptica pode ser manipulada pelo sistema**

Poucos trabalhos foram feitos até agora neste sentido. Grande parte das aplicações de ibr consistem em walk-through dentro de cenários ou visualização de objetos, mas não prevêm que o tempo transcorra linearmente, permitindo que diversos objetos estejam movendo-se individualmente. Isto porque a densidade de amostras se torna extremamente grande ao armazenar sub-conjunto de imagens animadas.

O sistema de avaliação se tornará completo, acrescentando-se os seguintes itens propostos por [Buehlere 01 et al]: Consistência Epipolar, Sensibilidade à resolução e Consistência de raios equivalentes. Estes critérios, entretanto, apenas fazem sentido quando se prevê uma mudança contínua dos valores dos parâmetros  $V_x$ ,  $V_y$ ,  $V_z$ .

Vale a pena destacar também se o sistema é capaz de representar objetos que estão entre o observador e o plano de fundo (em geral, um panorama).

#### **4. Avaliação dos principais trabalhos já desenvolvidos em ibr utilizando os critérios propostos**

Segue-se uma avaliação de alguns dos principais sistemas que utilizam ibr, utilizando-se os critérios sugeridos no capítulo 3.

##### **4.1 Movie-Map [Lippman 80]**

O sistema apenas possui destaque por ser um dos primeiros trabalhos na área com uma implementação bem sucedida. O processo de autoria consiste em adquirir e armazenar um grande conjunto de imagens, associando para cada uma quais são seus valores correspondentes para  $\mathbf{q}$ ,  $\mathbf{f}$ ,  $V_x$ ,  $V_y$ ,  $V_z$  e  $t$ . O processo de visualização consiste apenas numa consulta a uma tabela: procura-se no banco de imagens aquela que mais se aproxima ao valor das variáveis da função plenóptica do observador. Este sistema, apesar de ter 6 dimensões, requer uma densidade de amostras enorme para poder haver um mínimo de continuidade nos movimentos. Apesar da simplicidade do sistema, nele já se prevê a possibilidade de utilizar a variável  $t$ .

##### **4.2 QuickTime VR [Chen 95]**

Neste trabalho o autor apresenta uma técnica bem sucedida para navegação em cenários criados a partir de panoramas, utilizando a idéia básica de environment maps [Blinn et al 76]. Este sistema divide os movimentos possíveis para o observador em 4 tipos:

- Rotação da câmera: Pode ser facilmente obtido através de uma reprojeção do environment-map. (Este movimento será muito restrito no caso de panoramas cilíndricos)
- Rotação de objetos colocados na cena: Caso o objeto não pertença ao panorama, ao realizar um movimento da câmera, ocorrerá a perda de consistência da visualização do mesmo. Para tanto, o autor sugere que sejam tiradas fotos para todos os ângulos possíveis de se ver o objeto. Dependendo do ângulo entre o observador e o objeto, escolhe-se uma das imagens do repertório. O inconveniente deste método consiste em que será necessário um grande número de fotos para cada objeto. Utilizando-se o algoritmo de View Morphing [Seitz et al. 96] isto pode ser otimizado.
- Movimento do observador: Cria-se à partir de vários panoramas distintos, para um conjunto finito de posições  $V_x$ ,  $V_y$ ,  $V_z$ .
- Zoom de câmera: Pode ser feito simplesmente dando zoom no segmento da imagem que está sendo mostrada. Para evitar problemas de aliasing, pode-se utilizar técnicas de mip-maps no armazenamento do panorama.

A implementação deste trabalho consiste no sistema QuickTime VR, que utiliza um panorama cilíndrico para tal, tendo em vista a facilidade no processo de autoria. Em relação aos graus de liberdade, este método é classificado como bidimensional, pois o movimento da câmera não é contínuo e está restrito a um conjunto finito de pontos do espaço, sendo as únicas variáveis que podem variar continuamente  $\mathbf{q}$  e  $\mathbf{f}$  - ângulos de azimute e elevação do observador, ainda que o ângulo de elevação esteja restrito pelo fato do panorama ser cilíndrico. Isto pode ser resolvido através de panoramas cúbicos ou esféricos, embora a autoria para estes casos se torne mais complexa. Não se prevê um tratamento para a variável tempo, o que restringe a navegação a ambientes estáticos.

A aplicação QuickTime VR tornou-se uma referência para trabalhos de ibr. Os principais fatores são um processo de autoria relativamente simples e a densidade de amostras pequena, embora no caso de prever movimentos grandes do observador, assim como utilizar bastantes objetos no interior do environment map esta densidade pode tornar-se enorme e inviabilizar uma aplicação.

### **4.3 Light Field [Gortler et al. 96] e Lumigraph [Levoy et al. 96]**

Estes métodos poderiam ser chamados de "força bruta", no sentido de que criarão um environment map para cada ponto do espaço que se deseja modelar. Por um lado, não será necessário armazenamento algum de profundidade, mas por outro lado, para que haja continuidade da mudança das variáveis da função plenóptica, especialmente de  $V_x$ ,  $V_y$ ,  $V_z$ , será necessário uma grande quantidade de amostras, o que irá requerer muita memória do sistema. Este é o principal fator que limita a utilização deste método, com a tecnologia que se possui hoje em dia, embora os autores cheguem a implementar um método de compressão adequado ao problema, capaz de atingir um fator de até 200:1 de compressão.

Para descrever qualquer raio para qualquer ponto do interior de uma região, [Gortler et al. 96] sugere descrever cada raio de luz através da interseção do mesmo com dois planos distintos, sendo  $(u, v)$  e  $(s, t)$  as coordenadas da interseção em cada um dos planos. Pode-se demonstrar que desta forma descreve-se um sub-conjunto da função plenóptica com 4 dimensões. Deve-se ressaltar que apenas 2 planos não serão capazes de representar todos os raios do interior de uma cena. Assim sendo será necessário definir vários pares de planos distintos e não paralelos.

Em relação à autoria, o processo requer um trabalho custoso - pois será necessário um conjunto grande de imagens. No processo de aquisição das imagens pode-se imaginar que ambos os planos serão grids com um espaçamento de tamanho fixo entre os nós. A camera será colocada numa coordenada  $(s,t)$  e serão tiradas  $M \times N$  fotos, onde  $M$  e  $N$  é a resolução do outro plano (alvo da camera). Para cada posição da camera o seu alvo irá percorrer cada uma das coordenadas  $(u, v)$  do outro plano. Embora a rigor, não seja necessário nenhum equipamento especial, [Gortler et al. 96] sugere a montagem de uma plataforma baseada em padrões de cores para determinar a posição da camera durante a captura de amostras. Já [Levoy et al. 96] utiliza um aparelho controlado por computador para posicionar uma câmera corretamente.

Embora pouco tenha se estudado para determinar o número mínimo de amostras de imagens necessárias [Oh et al. 01] tenham sido feitos, [Chai et al. 00] apresenta um eficiente método para determinar a quantidade mínima de amostras necessárias para modelagem de cenas para aplicações que utilizem o algoritmo de Light Field.

### **4.4 Modeling and Rendering Architecture From Photographs: A Hybrid Geometry - and Image-Based Approach [Debevec et al. 96]**

Neste trabalho o autor propõem que a partir de um conjunto de imagens arquitetônicas de um ambiente real, seja gerado um modelo aproximado e a seguir aplicadas estas imagens sobre os modelos, com técnicas tradicionais de projeção de texturas. Numa primeira etapa o usuário cria um modelo básico da arquitetura utilizando primitivas simples. Durante este processo serão estabelecidas correspondências entre as arestas das primitivas e as arestas dos objetos presentes no conjunto de imagens fornecidas.

Num segundo passo é feito um refinamento do casamento entre estas fotos e os modelos, para que haja uma correspondência mais precisa nas dimensões. Também será feito um refinamento para a correspondência das posições das cameras, utilizando para esta estimativa uma posição calculada pela restrição de linhas horizontais e verticais na imagem. Esta técnica prevê ainda que detalhes da modelagem sejam estimados utilizando métodos de visão estéreo para calcular um mapa de profundidade de cada imagem original.

Há autores que procuram separar esta técnica da ibr padrão, tal como o próprio [Debevec 96 et al.]. Outros pretendem unificá-los, tratando-os como apenas um caso particular da função

plenóptica, como é o caso de [Buehler 01 et al.], que sugere que neste caso, como nos outros, o que se está desejando fazer é apenas uma interpolação do valor de cores para um raio desejado, dada uma combinação de raios de entrada, embora neste caso particular a interpolação é feita utilizando um modelo geométrico para determinar qual é o pixel de cada imagem que corresponde ao raio da imagem de saída.

Apesar do método possuir um grau de liberdade alto (5 dimensões:  $\mathbf{q}$ ,  $\mathbf{f}$ ,  $V_x$ ,  $V_y$  e  $V_z$ ) o espaço de navegação é pequeno e restrito. Quanto mais se deseje ampliar esta região, maior será o trabalho de autoria, tornando-se em alguns casos impraticável. Além disso, sua ferramenta de autoria prevê que os objetos das imagens sejam do tipo arquitetônico (linhas retas), o que limita a utilização do sistema para cenas particulares.

#### **4.5 Plenoptic Stitching: A Scalable Method for Reconstructing 3D Interactive Walkthroughs [Aliaga 01]**

Neste trabalho os autores descrevem um sistema de ibr com 4 graus de liberdade, onde se prevê uma mudança contínua dos valores de  $V_x$ ,  $V_y$  e  $V_z$ , (o observador pode caminhar livremente dentro de uma limitação de espaço), assim como alterar o valor de azimute  $\mathbf{f}$ , podendo-se no entanto representar apenas cenas panorâmicas, sendo impossível a presença de objetos flutuantes na cena.

O método consiste inicialmente em capturar imagens de um ambiente com um pequeno carro rádio-controlado que percorre caminhos dentro de um determinado ambiente. Para o bom funcionamento do sistema será necessário que sejam adquiridas em torno de 30 imagens por metro, utilizando uma camera omnidirecional (capaz de adquirir fotos de 360° de latitude por 180° de longitude). Estes caminhos percorridos devem cruzar-se, de maneira a formar um grid irregular, sendo cada célula do grid denominado de image loop. Pode-se entender estes image loops como diversos panoramas justapostos, não havendo necessidade de uma forma pré-definida para cada panorama. Dada a esta grande quantidade de imagens necessárias, o sistema se torna inviável para um walkthrough extenso numa área grande (um parque, por exemplo).

Feita a captura dos panoramas, o sistema exige que se calcule a estimativa da posição da câmera para cada image. Sugere-se para isto a *bacon-based pose estimation algorithm for omnidirectional cameras*, que realizará um tracking partindo de uma imagem inicial (os autores sugerem que se coloquem duas lâmpadas no ambiente, indicando-se suas posições na imagem inicial, de maneira a poder realizar o tracking).

A interpolação de imagens será necessária sempre que o observador estiver no interior de uma célula. Neste caso, a composição será feita utilizando-se colunas de pixels das imagens que estiverem nas bordas dos image-loops mais próximos da camera.

Para que o sistema possa ser tempo real, é necessário criar uma fase de pré-processamento, onde sobretudo se cria uma data structure para que possa haver acesso rápido às linhas radiais e às suas funções de mapeamento requeridas na reconstrução. Este pré-processamento pode consumir bastante tempo para ambientes extensos.

#### **4.6 Unstructural Lumigraph Rendering [Buehler et al. 01]**

O método foi elaborado, tendo em vista as propriedades de visualização apresentadas no capítulo 2 e permite movimentos contínuos em 5 graus de liberdade. Uma das vantagens que possui em relação a maioria dos sistemas é o de não precisar uma sequência de imagens de entrada que sigam uma ordem pré-estabelecida, como ocorre no Lumigraph ou no Light-field. Isto possibilita que a autoria seja simples, bastando capturar imagens do local a ser modelado, através de aparelhos convencionais. Entretanto, o sistema requer que seja feita uma estimativa da posição da câmera, assim como uma aproximação geométrica da cena. Quanto melhor for a aproximação, menos imagens serão necessárias.

Inicia-se o algoritmo através do cálculo do câmera blending field para um conjunto de vértices

do plano de projeção da camera, fazendo-se a seguir a interpolação para todo o plano. Este camera blending field será utilizado na reconstrução de imagens não existentes e consiste numa tabela de distribuição de pesos para cada ponto visto pela camera, indicando qual é a imagem que deve ser usada para gerá-lo e quanto é a sua contribuição (um mesmo ponto pode ser descrito por várias imagens, desde que a soma de todos os pesos seja igual a 1). Para construir esta distribuição de pesos, realiza-se o cálculo das imagens que possuem o menor ângulo formado entre o seu centro de projeção e o ponto que está sendo observado com a posição real da camera com o mesmo ponto.

Embora a aproximação da geometria das imagens possa ser um processo complexo, e não ser conveniente para representar ambientes extensos, o algoritmo demonstra ter bom comportamento quando se utilizam aproximações grosseiras da geometria, como por exemplo criando um grid de polígonos coincidente com o plano de projeção da camera e com alguns poucos vértices deslocados. A aproximação geométrica é possível somar a geometria de outros objetos, possibilitando que haja uma mistura de ibr com modelagem tradicional.

Os autores não descrevem nenhuma técnica particular para obter a estimativa da posição da camera, mas este dado é obrigatório para a visualização. Em relação às propriedades descritas no próprio [Buheler et al. 01], todas são obedecidas.

## **5 Conclusão e Trabalhos Futuros**

Os autores afirmam que não poderá haver um algoritmo que implemente a função plenóptica na sua totalidade e que portanto não haverá um sistema que seja ideal para qualquer aplicação. Tendo em vista este fato, a crescente importância que os sistemas de ibr vêm assumindo na computação gráfica e o grande número de implementações da função plenóptica, este trabalho demonstra a importância de se criar um sistema capaz de padronizar uma avaliação crítico-comparativa, uma vez que um algoritmo pode ser adequado para algumas aplicações, sendo que para outras pode ser totalmente inviável.

Neste artigo escolheram-se 7 trabalhos, dentre os que se consideram de maior relevância na área, para realizar uma análise sucinta, através dos critérios propostos. Cabe futuramente não apenas ampliar o conjunto de trabalhos analisados, mas também detalhar mais o estudo, criando-se tabelas comparativas em relação ao tempo de pré-processamento, frame-rate, volume de imagens necessárias para criar a cena, tempo gasto no processo de realização da aproximação geométrica e de cálculo de estimativa da posição da camera, aliasing obtido nas imagens geradas por interpolação, etc.

## **6 Agradecimentos**

O primeiro autor está agradecido ao CNPq pelo amparo dado à pesquisa, e ao ICAD – Igames, grupo tecnológico em Computação Gráfica da PUC-Rio, que deu suporte a esta pesquisa.

## **7 Bibliografia**

[Adelson 91] Adelson, E. H.; Bergsen J. The Plenoptic Function and the Elements of Early Vision. In Computational Models of Visual Processing, MIT Press, Cambridge, MA, 3-29, 1991.

[Aliaga et al. 01] Aliaga, D. G.; Carlbom, I. Plenoptic Stitching: A Scalable Method for Reconstructing 3D Interactive Walkthroughs. Siggraph'01 Computer Graphics Proceedings.

[Blinn et al 76] Blinn, J. F.; Newell, M. E. Texture and Reflection in Computer Generated Images. Communications of the ACM, 19(10), pp 542-547, October 1976.

[Buehlere 01 et al.] Buehler, C.; Bosse, M.; McMillan, L.; Gortler, S.; Cohen, M. Unstructural Lumigraph Rendering. Siggraph'01 Computer Graphics Proceedings.

- [Chai et al. 00] Chai, J. X.; Tong, X.; Chan, S. C.; Shum, H. Y. Plenoptic Sampling. Siggraph'00 Computer Graphics Proceedings.
- [Chen et al. 93] Chen, S. E.; Williams, L. View Interpolation for Image Synthesis". Siggraph'93 Computer Graphics Proceedings.
- [Chen 95] Chen, S., E. Quicktime VR - An Image-based Approach to Virtual Environment Navigation, Siggraph'95 Computer Graphics Proceedings.
- [Debevec et al. 96] Debevec, P. E.; Taylor, C. J.; Malik, J. Modeling and Rendering Architecture From Photographs: A Hybrid Geometry - and Image-Based Approach. Siggraph'96 Computer Graphics Proceedings.
- [Debevec et al. 98] Debevec, P. E.; Yu, Y.; Borshukov, G. D. Efficient view-dependent image-based rendering with projective texture-mapping. Eurographics Rendering Workshop 1998.
- [Gortler et al. 96] Gortler, S., J.; Grzeszczuk, R.; Szeliski, R.; Cohen, M., F. The Lumigraph. Siggraph'96 Computer Graphics Proceedings.
- [Greene 86] Greene, N. Environment Mapping and other applications of World projections. IEEE Computer Graphics and Applications, 6, pp 21 - 29. 1986.
- [Heigi et al. 99] Heigi, B.; Koch, R.; Pollefeys, M.; Denzler, J.; Gool, L. V. Plenoptic Modeling and rendering from image sequences taken by hand-held camera. Proc. DAGM 99, pp 94 - 101.
- [Kang et al. 96] Kang, S. B.; Szeliski, R. 3D Scene Data Recovery Using Omnidirectional Baseline Stereo. IEEE Computer Vision and Pattern Recognition (CVPR 96), 364-370, 1996.
- [Levoy et al. 96] Levoy, M.; Hanrahan, P. Light Field Rendering. Siggraph'96 Computer Graphics Proceedings.
- [Lippman 80] Lippman, A. Movie Maps: An Application on the Optical Videodisc to Computer Graphics. Siggraph'80 Computer Graphics Proceedings.
- [McMillan et al 95] McMillan, L.; Bishop, G. Plenoptic Modeling: An Image-Based Rendering System. Siggraph'95 Computer Graphics Proceedings.
- [McMillan 97] McMillan, L. An Image-Based Approach to Three Dimensional Computer Graphics. Ph. D. thesis, University of North Carolina at Chapel Hill. 1997.
- [Oh et al. 01] Oh, B. M.; Chen, M.; Dorsey, J.; Durand, F. Image-based Modeling and Photo Editing. Siggraph'01 Computer Graphics Proceedings.
- [Pighin et al. 98] Pighin, F.; Hecker, J.; Lischinski, D.; Szeliski, R.; Salesin, D. H. Synthesizing realistic facial expressions from photographs. Siggraph'98 Computer Graphics Proceedings.
- [Pulli et al. 97] Pulli, K.; Cohen, M.; Duchamp, T.; Hoppe, H.; Shapiro, L.; Stuetzle, W. View-based Rendering: Visualizing real objects from scanned range and color data. Eurographics Rendering Workshop 1997, pp 23 - 34.
- [Seitz et al. 96] Seitz, S. M. and Dyer, C. R. View Morphing. Proc. SIGGRAPH 96. *Computer Graphics* (1996), pp. 21-30.
- [Shum et al 99] Shum, H.; He, L. Rendering with concentric mosaics. Siggraph'99 Computer Graphics Proceedings.
- [Szeliski 96] Szeliski, R. Video mosaics for virtual environments. IEEE Computer Vision and Applications, pp 22-30, 1996.
- [Szeliski et al. 97] Szeliski, R.; Shum, H. Creating full view panoramic image mosaics and texture-mapped models. Siggraph'97 Computer Graphics Proceedings.

[Takahashi et al. 00] Takahashi, T.; Kawasaki, H.; Ikeuchi, K.; Sakauchi, M. Arbitrary View Position and Direction Rendering for Large-Scale Scenes. *IEEE Computer Vision and Pattern Recognition (CVPR 00)*, pp 296-303, 2000.

[Wood et al. 00] Wood, D. N.; Azuma, D. I.; Aldinger, K.; Curless, B.; Duchamp, T.; Salesin, D. H., Stuetzle, W. Surface Light Fields for 3D photography. *Siggraph'00 Computer Graphics Proceedings*.