

## 2

### Rendering baseado em imagens e as plataformas celulares

Nos computadores pessoais de hoje, com o advento dos processadores gráficos programáveis (GPU) e evolução das técnicas de renderização, as aplicações e gráficos de alta resolução e de grande realismo são cada vez mais comuns.

Já em celulares, devido às restrições de processamento e memória e a inexistência de GPU ou processadores com algum tipo de aceleração gráfica na maioria dos celulares fabricados em grande escala, uma saída é a utilização de técnicas para representar um cenário completo ou objetos que façam parte dele a partir de imagens, ao invés de elementos geométricos. Tais técnicas pertencem à linha de pesquisa chamada de *Image-Based Rendering*.

Na reprodução de ambientes e objetos utilizando técnicas de Computação Gráfica, a complexidade depende fortemente dos modelos utilizados. Em contrapartida, várias técnicas de IBR têm sua complexidade dependente da resolução das imagens utilizadas, o que pode ser uma vantagem para pequenos dispositivos.

Porém, algumas dessas técnicas, mesmo em plataformas desktop e estações de alto desempenho, apresentam-se como problemas complexos. O desafio na plataforma de celulares é, portanto, muito grande e apenas os aspectos básicos podem ser tratados.

A área de *Image-Based Rendering* tem sua base na combinação entre as técnicas de Computação Gráfica 3D e Visão Computacional, e tem sua definição mais formal a partir da função plenóptica.

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 da luz.

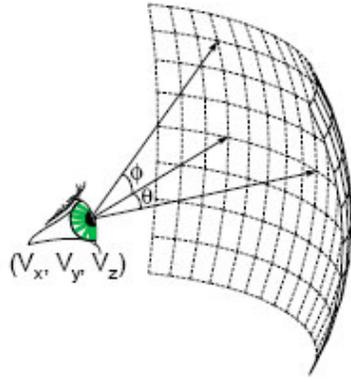


Figura 1 Ilustração representando algumas das variáveis da função plenóptica

Abaixo são apresentadas as principais sub-áreas e suas aplicações. O tópico 2.4 apresenta os trabalhos que utilizam as técnicas de rendering baseado em imagens para a visualização de ambientes e objetos em plataformas de baixa performance, como os telefones celulares.

## 2.1. Modelagem de ambientes completos:

Os trabalhos mais próximos da definição da função plenóptica e que procuram representar um cenário completo são o *Ligth Field* (Gortler, 1996) e o *Lumigraph* (Levoy, 1996).

Ambos tentam descrever cada raio de luz do ambiente através das duas coordenadas de interseção de cada raio com dois planos de suporte. Porém, para representar todos os raios possíveis, é necessário um grande número de planos.

Esse tipo de modelagem, até mesmo para *desktops*, requer muito processamento e uma grande quantidade de amostras do ambiente, podendo, em alguns casos, tornar-se impraticável.

Um trabalho semelhante aos outros dois citados é O *Unstructural Lumigraph Rendering* (Buheler, 2001). Sua modelagem é mais simples, entretanto, o sistema requer que seja feita uma estimativa da posição da câmera, bem como uma aproximação geométrica da cena. Quanto melhor é esta aproximação, menos imagens são necessárias.

Já Debevec (1996) apresenta uma abordagem onde procura extrair informações geométricas da cena a partir de imagens do mundo real. Este tipo de trabalho é muitas vezes chamado de modelagem baseado em imagens (*Image Based Modeling*).

Após obter um modelo aproximado do ambiente, as próprias imagens são aplicadas como textura. Quanto maior o grau de liberdade que queira se dar ao usuário, é maior a complexidade dos modelos a serem gerados e é maior o número de imagens fontes necessárias. Esta modelagem também é dependente da arquitetura real dos prédios presentes no ambiente.

A modelagem completa de cenários apresenta grandes possibilidades de aplicações para dispositivos móveis, como a navegação em ambientes virtuais. Estas aplicações poderiam aproveitar-se da localização do usuário e a até mesmo da captura dos movimentos feitos com o aparelho, a partir das diferenças das imagens capturadas pela câmera em intervalos de tempo.

Apesar disso, devido à grande quantidade de informação que necessitam para gerar as vistas do ambiente, estes tipos de modelagem acabam sendo inadequados para a utilização em dispositivos móveis que ainda contam com pouco espaço de armazenamento comparativamente aos *desktops*.

## **2.2. Modelagem de Panoramas:**

Dentre as técnicas que se propõe à representação de cenas, a modelagem de panoramas é uma abordagem mais simples.

Esta técnica consiste no uso de um suporte geométrico para a projeção de imagem panorâmica representando o ambiente que se quer reproduzir. Ela foi apresentada inicialmente em (Chen, 1995), onde foi utilizado um suporte cilíndrico, e se tornou conhecida devido à aplicação *Quick Time VR*.

Atualmente há trabalhos que também utilizam outras figuras geométricas como suporte, como cubos e esferas, onde o observador encontra-se sempre no centro, podendo variar os ângulos de azimute e de elevação definidos na função plenóptica, com algumas limitações dependendo do suporte utilizado.

Esse tipo de modelagem, muito utilizado para representar objetos e paisagens que estejam a uma grande distância do observador, quando utilizado para a navegação em ambientes virtuais, apresenta uma navegação discreta em as cenas.

Para tentar fazer uma navegação de forma contínua, (Aliaga,2001) apresenta uma proposta chamada de *Plenoptic Sticking*. Porém, além de um processo de autoria complexo, a quantidade de dados necessária pode tornar esta modelagem impraticável para ambientes extensos.

A modelagem de panoramas, por sua simplicidade de autoria e pela pequena quantidade de dados necessários para a representação de uma cena, é factível de implementação em ambientes celulares, e é a técnica explorada na aplicação proposta por esta dissertação.

### **2.3. Image Warping e Morphing**

O *warping* de uma imagem consiste numa função que transforma a posição de um ponto pertencente a uma imagem fonte, produzindo uma nova imagem.

Transições suaves de uma imagem fonte para uma segunda imagem gerada são chamadas de *morphing*.

A partir de um conjunto de figuras, com diferentes vistas de um objeto, a técnica do *view morphing* (Seitz, 1996) é capaz de gerar vistas deste objeto para ângulos não fornecidos pelo conjunto inicial. Isto é feito através de uma interpolação suave entre duas imagens, a partir da correspondência entre pontos de uma imagem e outra. A correspondência para alguns destes pontos deve ser dada pelo usuário, sendo que os demais pontos são calculados automaticamente .

Esta correspondência de pontos, que torna a autoria um pouco mais complicada já não é necessária na técnica de *3D Image Warping* (McMillan,1997).

Esta técnica possibilita a geração de vistas de um ambiente para alterações do ponto de observação, seja pelo posicionamento ou alteração dos ângulos da câmera virtual. Sua grande vantagem é que necessita apenas das profundidades que cada pixel de uma imagem representa em uma cena para conseguir gerar novas vistas. Essa informação de profundidade é trivialmente gerada para imagens sintetizadas, e possível de ser estimada para fotos reais, com a ajuda de equipamentos auxiliares.

Para uma visualização completa de um objeto ou um ambiente pode-se utilizar um conjunto finito de imagens para se gerar todas as vistas necessárias.

## 2.4.

### Visualização de Objetos e Ambientes virtuais em plataformas de baixa performance

A visualização, seja de ambientes ou de objetos com grande realismo e complexidade, é uma necessidade em algumas aplicações. Porém, nem sempre os dispositivos são capazes de elementos muito complexos.

Uma alternativa para a representação desses modelos em dispositivos de baixa performance ou sem suporte à visualização de gráficos 3D, é a utilização de técnicas que se baseiem na representação desses modelos por modelos menos complexos ou por imagens.

A substituição de um modelo por imagens é comumente denominada de impostores; onde *sprites* são impostores representados por pequenos planos onde se mapeia uma textura, com a imagem do objeto; os *billboards* são *sprites* que têm sua normal sempre apontando para a câmera; e outros mais complexos como os *billboard clouds* (Decoret,2003) são uma composição de *billboards* com transparência.

Giacomo et al.(2005) apresentou uma abordagem onde substituiu modelos de rostos em 3D, utilizados para animação em telefones celulares, por representações utilizando gráficos vetoriais, e depois aplicando *Gourad-Shading* já no dispositivo final.

Alguns trabalhos utilizam-se de uma arquitetura onde há a presença de um servidor capaz de criar e visualizar objetos complexos. Até mesmo dispositivos que possuam capacidade de gerar gráficos 3D podem não conseguir gerar modelos com grande número de polígonos, e tais modelos podem ser muito grande para serem enviados ao cliente através de redes como a Internet. Hoppe (1996) apresentou os chamados *progressive meshes* onde modelos simplificados são enviados ao cliente seguidos de conjuntos de instruções, dizendo como criar representações mais complexas. Apesar de diminuir o tráfego de dados, ainda é necessário que o dispositivo cliente seja capaz de criar e renderizar modelos 3D.

Um outro exemplo onde os *meshes* são simplificados foi apresentado por Jehaes et al. (2005). Neste caso, além da simplificação dos *meshes* para o envio dos modelos pela rede, ainda são transmitidas imagens representando vistas dos objetos. Assim, a renderização da cena usa uma abordagem híbrida, onde objetos que necessitem de um maior nível de detalhe, são renderizados a partir dos *meshes*, para os objetos com menor nível de detalhe, técnicas de

renderização baseada em imagens são aplicadas às vistas recuperadas do servidor, gerando uma nova representação.

Em (Rodrigues,2006) é apresentada uma arquitetura genérica para visualização e navegação em ambientes virtuais. Neste caso, o ambiente é separado em células e os modelos necessários para a renderização em cada célula são enviados, sob demanda, de um servidor para a aplicação cliente. A prioridade no envio dos modelos é baseada na navegação do usuário para que o cliente seja capaz de renderizar cenas de qualidade o mais rápido possível. Porém, sua comunicação se baseia em redes locais sem fio, o que limita sua aplicação a uma gama menor de dispositivos móveis.

Baseando-se em clientes sem suporte à renderização de gráficos 3D, muitos autores propuseram sistemas onde um servidor de alta capacidade gerasse esses gráficos e os transmitisse para o cliente, como uma simples imagem ou até mesmo fluxo de vídeo. Um exemplo de trabalho nessa linha é apresentado por Engel et al (1999 e 2000), onde um serviço baseado em CORBA era capaz de renderizar imagens em três dimensões e enviá-las a clientes Java e *Web*.

Porém, estas arquiteturas, além da latência da rede, ainda apresentam as taxas de reprodução das cenas pelos clientes finais dependentes das taxas de renderização do servidor.

Trabalhos como o de Reagan & Pose(1994) e Mark(1999) apresentam uma arquitetura para modificar o *pipeline* gráfico convencional, separando a taxa de quadros gerada pelo renderizador, da taxa de quadros que do *display* final. A figura 2 mostra a diferença entre as duas arquiteturas.

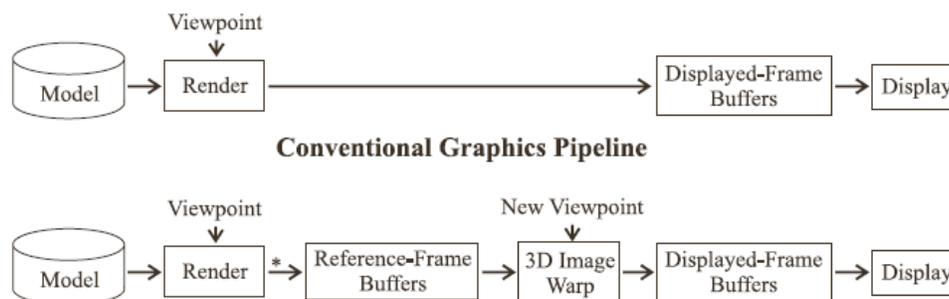


Figura 2 -Diferença entre o pipeline gráfico convencional e o proposto por Mark(1999) e Reagan & Pose(1994) . Imagem retirada de (Mark 1999)

Ambos os trabalhos incluem um componente que cria visualizações, utilizando técnicas de warping, a partir de quadros de referência renderizados a partir de modelos 3D.

Um trabalho onde técnicas de IBR são empregadas em uma arquitetura cliente servidor foi apresentado por Li et al. (2001) . Neste caso panoramas e mosaicos concêntricos capturados e previamente processados são persistidos no servidor e posteriormente recuperados pelo cliente através da internet.

Já Bao & Gourlay (2003, 2004) apresentaram um sistema para a navegação em ambientes virtuais utilizando-se dispositivos móveis e de baixa capacidade de processamento.

A partir de uma primeira visualização gerada e enviada ao cliente, toda vez que o usuário solicita, uma nova visualização é gerada como descrito abaixo.

Quando o usuário solicita uma mudança no ponto de observação da cena, o dispositivo cliente utiliza o *3D Image Warping* para gerar uma nova vista aproximada. Enquanto isso, com base neste mesmo pedido, o dispositivo cliente solicita uma nova visualização ao servidor. Este, por sua vez, gera duas visualizações, uma a partir do modelo real, e outra a partir do mesmo algoritmo de *warping* utilizado no cliente. Com base nestas duas imagens é gerado um resíduo, enviado ao cliente, e utilizado para corrigir a visualização corrente.

Em (Thomas et. al, 2005) um cliente é capaz de gerar novas vistas de um ambiente a partir de imagens enviadas de um servidor. Para otimizar a renderização e evitar artefatos indesejáveis como buracos nas cenas, este trabalho propõe uma configuração de câmeras capaz de capturar as imagens-chave necessárias, porém, o estudo se baseia apenas no um meio urbano.

Boukerche & Pazzi (2006 a, 2006 b, 2006 c) criaram uma série de otimizações e mecanismos com o intuito de tornar possível a visualização e navegação de ambientes virtuais 3D em plataformas móveis. Em seus trabalhos eles apresentam uma estratégia para *streaming* de imagens geradas por um servidor, e uma estratégia para bufferização dessas imagens no cliente, que aplica técnicas de IBR na geração da vista corrente. Nestes trabalhos o servidor também possui um algoritmo de predição de movimento, e assim sempre enviando as imagens que serão necessárias para renderização da cena nas próximas interações do usuário.

Um revés de vários dos sistemas apresentados acima, é tráfego contínuo de dados, que pode ser problemático em redes celulares, além de custoso para usuários finais.

Mesmo assim, eles mostram que outras técnicas, em conjunto com as pertencentes à área de IBR, podem apresentar-se como boas alternativas para a apresentação de ambientes e objetos com grande qualidade gráfica em dispositivos com pouca capacidade de processamento.