

Panoramas RGBD Omnidirecionais com Múltiplas Camadas e suas Aplicações

Aldo René Zang, Luiz Velho
VISGRAF Laboratory, IMPA
Rio de Janeiro, Brasil
Email: zang@impa.br

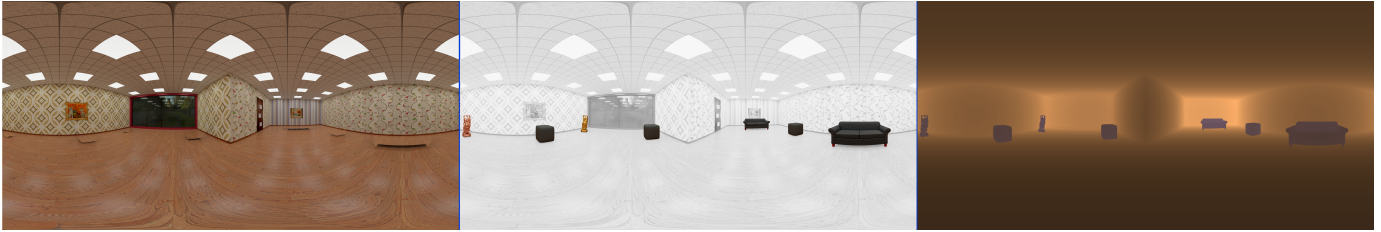


Figure 1. Panorama RGBD em camadas. Camada de fundo (esquerda), e camada de objetos da frente (centro). A imagem da direita contém as camadas de profundidade do fundo (laranja) e dos objetos (azul).

Abstract—We introduce a novel omnidirectional *RGBD* panorama with multiple layers (*PML*) concept. This new representation opens a new range of applications for interactive display area, such as stereo viewing, disparity handling and parallax effects. This representation also offers new solutions for image based rendering problems.

Resumo—Apresentamos o conceito de panoramas *RGBD* omnidirecionais com múltiplas camadas, *PMC*. Esta representação dá lugar a novas aplicações na área de visualização interativa, como por exemplo visualização estéreo, manipulação de disparidade e efeitos de paralaxe, ao tempo que proporciona novas soluções para problemas de renderização baseada em imagens.

Keywords—Panorama; depth field; IBL rendering; visualization;

I. INTRODUÇÃO

Tendo em vista os avanços recentes das tecnologias de captura de imagem e profundidade, estamos propondo uma nova classe de panoramas, os panoramas *RGBD* omnidirecionais com múltiplas camadas (*PMC*), que abrem uma nova gama de aplicações na área de visualização interativa, tais como: visualização estéreo calculada a partir de uma única *PMC*, manipulação de disparidade e efeitos de paralaxe. Esta representação também proporciona novas soluções para problemas de renderização com iluminação baseada em imagem, *IBL* (*image-based lighting*), como os tratados em Felinto et al. [1].

A proposta de *PMC*, é uma evolução do panorama com profundidade proposto nos trabalhos de Felinto et al. [1], [2], para resolver problemas de movimentação de câmara e posicionamento de objetos sintéticos em cenas reais capturadas. Em Felinto et al. [1] alguns problemas tem que ser resolvidos com ajuda do usuário, sendo necessário copiar e colar texturas de uma região da cena para outra, a fim de preencher a informação de regiões ocluídas do panorama que são refletidas

pelos objetos sintéticos. Estes problemas em particular podem ser diminuídos e até eliminados com o uso de um *PMC*. Além disso veremos que o *PMC* apresenta outras propriedades e usabilidades inerentes a suas características especiais.

A. Trabalhos relacionados

O uso de imagens omnidirecionais em computação gráfica não é recente. Um trabalho pioneiro é o de Blinn [3], sobre mapeamento de reflexão, que logo tornou-se uma ferramenta da industria de efeitos especiais. Em 1998, Paul Debevec [4] utilizou um imagem omnidirecional *HDR* como fonte de iluminação em renderizações. Em 2012, Felinto et al. [2] exploraram o uso de panoramas omnidirecionais com profundidade, para renderização com *IBL*. Na área de visualização baseada em imagens um passo importante foi dado por McMillan [5], onde é apresentada uma técnica baseada em warping de imagens para produzir novos pontos de vista a partir de uma imagem de referência, e apresenta alguns resultados com remapeamento de cenas usando panoramas.

Adelson e Bergen [6], definem como *função plenóptica* o feixe de raios visível a partir de qualquer ponto do espaço, a qualquer momento, em todos os comprimentos de onda. A *função plenóptica* descreve toda a energia que chega num ponto do espaço desde todas as direções. Adelson e Bergen [6] também apresentaram uma descrição funcional, dada por

$$\rho = P(\phi, \theta, \lambda, V_x, V_y, V_z, t), \quad (1)$$

onde (V_x, V_y, V_z) é a posição do observador, ϕ e θ são os ângulos de azimute e elevação da direção de observação, λ é o comprimento de onda da luz e a variável t descreve o tempo.

Pode-se entender a *função plenóptica* como sendo uma função capaz de descrever todas as possíveis vistas de uma determinada cena. Dizemos que temos uma amostra completa

função plenóptica se, para um ponto coberto pela função, toda a abóboda que está à sua volta pode ser vista, e uma amostra incompleta quando apenas uma parte desta abóboda é visível.

II. PANORAMAS OMNIDIRECIONAIS COM PROFUNDIDADE

Se a função *plenóptica* é construída para um único ponto de vista e um instante de tempo, sua dimensionalidade é reduzida de 7 para 2. Este é o princípio usado em mapeamento de reflexão, [3], onde a vista do ambiente desde uma posição fixa é representada por um mapa 2D.

Um *panorama omnidirecional* é um mapa esférico completo capturado num ponto (x, y, z) do espaço, i.e., uma amostra completa da *função plenóptica* no ponto (x, y, z) . Em Felinto et al. [1], [2], um *Panorama Omnidirecional com Profundidade* é definido como um mapa esférico completo onde para cada direção (ϕ, θ) , além da radiância, também é conhecida a distância até a primeira interseção com um objeto da cena.

A. Representação de um panorama

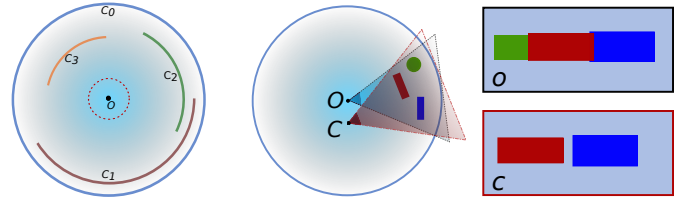
Dada uma amostra completa da *função plenóptica* no ponto (x, y, z) , i.e. um panorama omnidirecional, ela precisa ser armazenada num arquivo de imagem utilizando um mapeamento de imagem omnidirecional, obtendo assim uma representação digital de toda a luz da cena real numa imagem. Existem na literatura diversos mapeamentos da esfera S^2 sobre um suporte planar, a partir as quais podemos obter diferentes representações para armazenar um panorama omnidirecional. Algumas representações permitem acessar a informação de maneira intuitiva, mas podem ter redundância de informação devido às distorções inerentes ao mapeamento de planificação da esfera. Outros podem ter boa distribuição de amostras, mas são difíceis de ser armazenados pela forma irregular do suporte planar U . Neste trabalho utilizamos a representação equirectangular, que relaciona uma direção $(x, y, z) \in S^2$ com um ponto $(u, v) \in [0, 2\pi] \times [0, \pi]$ a traves da equação

$$(\phi, \theta) = \left(\arctan \frac{y}{x}, \arccos z \right). \quad (2)$$

III. PANORAMAS RGBD OMNIDIRECIONAIS EM CAMADAS

Dado um *panorama omnidirecional com profundidade* de uma cena real, é natural que existam descontinuidades entre os valores de profundidade, dado que a cena geralmente contem objetos a distâncias diferentes desde o observador (Figura 2). Em geral os pontos de descontinuidade na função de distância encontram-se na borda dos objetos. Usando as descontinuidades da função distância como guia, podemos dividir a esfera S^2 em regiões R_k , tais que todos os pontos de descontinuidade da função de distância estejam na fronteira de alguma região R_k e nunca no interior.

Até o momento falamos em *panoramas omnidirecionais com profundidade*. Eles tem a propriedade de que para o ponto de captura do mesmo, além da radiância incidente desde todas as direções, também conhecemos a profundidade da cena em todas as direções. Se renderizamos uma cena cuja câmara está posicionada na origem do panorama, não temos problemas pois dispomos de suficiente informação para gerar



(a) Estrutura da PMC. (b) Ilustração de uma cena armazenada num PMC.

Figura 2. (a) As camadas C_3, C_2, C_1, C_0 estão ordenadas segundo a profundidade crescente desde o centro O do panorama. Em linhas pontilhadas o raio de cobertura do PMC, dentro do qual a *função plenóptica* pode ser reconstruída com o PMC. (b) Ilustração de duas vistas da cena, uma desde o centro do panorama O , e outra desde uma posição diferente C . Objetos podem ser ocluídos ou aparecer ao mudar o ponto de vista.

uma imagem de saída que nada mais é do que uma porção limitada do panorama (Figura 3 (a)). Se desejarmos renderizar uma vista desde uma posição de câmera diferente à origem do panorama, podemos ter sérios problemas, mesmo contando com a profundidade, dado que alguma parte da cena que estava ocluída pode ficar visível, mas um *panorama omnidirecional com profundidade* não tem armazenada esta informação.

Como os problemas de movimentar a câmera estão ligados à falta de informação adicional tanto de profundidade quanto de cor nas regiões circundantes aos pontos de descontinuidade da *função plenóptica* restrita à profundidade, pensamos em formular o panorama de forma tal que contenha informação para reconstruir a *função plenóptica* não somente no ponto de captura (x, y, z) mas numa vizinhança V deste ponto. Para isto definimos um *panorama omnidirecional com múltiplas camadas, PMC*. Um PMC será definido como um conjunto de funções P_i de $V_i \subset S^2$ para um espaço de atributos C (cor, profundidade, e outros), tais que o domínio de todas elas fazem uma cobertura da esfera unitária S^2 , e as funções P_i restritas ao atributo de profundidade, $P_i|_p$, são contínuas.

$$F = \{P_0, \dots, P_k\} \quad k \in N \quad \text{such that} \quad (3)$$

$$P_i : V_i \subset S^2 \rightarrow C_i \subset R^3 \times R, \quad 0 \leq i \leq k \quad \text{and}, \quad (4)$$

$$\cup_{i=0}^k V_i = S^2 \quad (5)$$



(a) Câmara na origem do PMC. (b) Câmera fora da origem do PMC.

Figura 3. Efeito do movimento de câmera. Um puff oclui um canto da sala na imagem (a), enquanto que o canto fica visível na imagem (b).

$$P_i|_p : V_i \subset S^2 \rightarrow C_i|_p \subset R, \quad 0 \leq i \leq k \text{ is } C^0. \quad (6)$$

IV. CONSTRUÇÃO DE PANORAMAS EM CAMADAS

Uma pergunta pertinente é como construir ou capturar um panorama em camadas. Esta pergunta por si só abre uma nova série de problemas de pesquisa interessantes. Os panoramas utilizados neste trabalho foram construídos num software de modelagem, porém acreditamos que a captura deste tipo de informação não será problema no futuro próximo se olharmos para os novos dispositivos que vem sendo desenvolvidos ultimamente (Kinect, Tango, etc). Nossa primeira proposta é construir um panorama em camadas a partir de um panorama *RGBD* capturado ou modelado por computador. Dado que as camadas são delimitadas parcial ou totalmente por regiões de descontinuidade na profundidade (Figura 4), propomos como estratégia para definir as camadas, detectar as curvas de descontinuidade de profundidade, para logo unir os extremos das mesmas a fim de formar regiões fechadas. Esta união pode ser mediante uma linha reta ou guiada por uma análise de descontinuidades nas normais na camada de profundidade, dado que geralmente a transição de um objeto para outro tem descontinuidades de profundidade ou de normais. Uma vez calculadas as camadas com esta estratégia, resta preencher as regiões de oclusão gerados pelas camadas da frente nas camadas mais profundas. Isto pode ser feito de maneira satisfatória utilizando síntese de textura, [7], [8], ou inpainting, [9]. Em muitos casos remover um objeto grande de uma imagem e fazer inpainting da região pode gerar resultados pouco convincentes, porém no nosso caso isto não é problemático pois estamos interessados somente em áreas próximas das bordas das regiões geradas pela separação de camadas, e nesse caso muitas técnicas de inpainting ou mesmo síntese de textura apresentam resultados aceitáveis e realistas. Finalmente, a profundidade nas regiões de desocclusão também podem ser estimadas com inpainting.

V. VISUALIZAÇÃO DE PANORAMAS EM CAMADAS

Dado que o *PMC* foi proposto para armazenar informação da *função plenóptica*, uma aplicação inerente ao *PMC* é a visualização de amostras da *função plenóptica* desde um ponto próximo ao centro do *PMC*, isto é, gerar novas vistas da cena capturada no panorama com camadas.

Pensamos em duas abordagens para realizar a visualização parcial de um *PMC*: renderização baseada em pontos e renderização baseada em malhas. Apresentaremos aqui a abordagem baseada em amostras de pontos. A renderização baseada em pontos pode ser descrita na seguinte sequencia de passos:

- 1) selecionar amostras em cada camada do *PMC*;
- 2) calcular a posição no mundo das amostras selecionadas;
- 3) projetar as amostras no plano da imagem da câmera;
- 4) descartar as amostras que não são visíveis;
- 5) realizar uma filtragem das amostras visíveis;

O processo de classificação das amostras visíveis e não visíveis é feita por etapas. Inicialmente, o suporte da imagem é dividido em blocos de 3×3 pixels. Para cada bloco I identificamos as amostras localizadas nele. Seguidamente ordenamos

as amostras do bloco I segundo a camada e segundo a profundidade. Cabe remarcar que estamos considerando dentro do raio de liberdade do panorama para qualquer posição de câmara, a ordem de visualização das camadas não é alterado. Se todas as amostras do bloco I pertencem à mesma camada, são marcadas como visíveis. Se o bloco I tem amostras em mais de uma camada, fazemos uma comparação com os blocos da sua vizinhança 8-conectada. Para cada bloco vizinho consideramos a amostra com menos profundidade. Se as 8 (ou menos) amostras com menor profundidade dos blocos vizinhos pertencem à mesma camada que a amostra com menor profundidade do bloco I , significa que estamos na região interior da camada superior e não na borda da mesma (blocos verdes na Figura 6). Assim, marcamos as amostras da camada superior de I como visíveis e as amostras das demais camadas como não visíveis.

Finalmente podemos ter uma série de blocos que não podem ser classificados porque cada um deles tem amostras em varias camadas e sua vizinhança 8-conectada não tem amostras na mesma camada superior (Figura 6). Estes blocos estão localizados nas bordas das camadas. Para resolver o problema de visibilidade refinamos a amostragem destes blocos e sua vizinhança 8-conectada e volvemos a processar todos blocos com amostragem refinada usando o critério do parágrafo anterior. Durante o refinamento alguns blocos não classificados podem ter sido resolvidos e blocos que pareciam classificados podem mudar seu status para não resolvidos. Como a densidade a amostras é maior, a seguir fazemos uma classificação das amostras pixel a pixel em cada bloco não resolvido para determinar as amostras visíveis e não visíveis. Finalmente é feita uma filtragem com as amostras visíveis resultantes do processo de seleção anterior.

A. Filtragem de pontos

Atualmente realizamos uma filtragem espacialmente invariante ao longo de todo o domínio da imagem. Isto não é recomendado, pois as amostras resultantes no processo de visualização se movem de maneira que existem regiões em expansão e outras em contração (Figura 7). Assim, um filtro invariante pode não cobrir pixels da imagem a ser reconstruída em regiões com poucas amostras, ou pode suavizar a imagem reconstruída em regiões com alta concentração de amostras. O uso de filtros espacialmente variantes já foi estudado na área de warping de imagens, [10], e no futuro pretendemos trazer estas ideias para nosso sistema de renderização.

VI. CONCLUSÕES

Apresentamos uma nova formulação para panoramas que permite estender suas aplicações em computação gráfica. Esta formulação introduz novos desafios, tais como: resolver o problema de captura de um *PMC*, ou construir um *PMC* a partir de uma coleção de panoramas *RGBD* tradicionais. Inclusive podemos pensar no problema de criar um panorama em camadas a partir de um único panorama *RGBD* fazendo inpainting para cobrir as áreas adjacentes às descontinuidades de profundidade. Dado um panorama em camadas, o problema

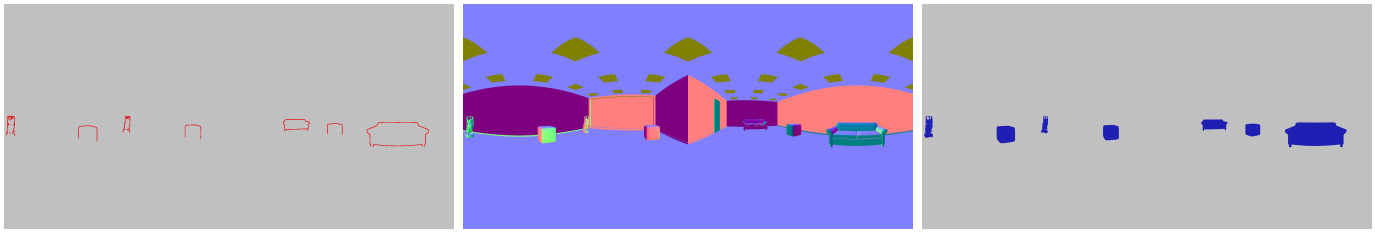


Figura 4. **Esq.:** Mapa das descontinuidades no atributo de profundidade. **Centro:** Mapa de normais. **Dir.:** Máscara da camada C_1 calculada usando o mapa de descontinuidade de profundidade e gradiente do mapa de normais.



Figura 5. Visualização de frustum limitado do panorama da figura 1, com posição de câmara fora do centro do panorama. A região azul de 15×15 pixels destacada na imagem é analisada na figura 6.

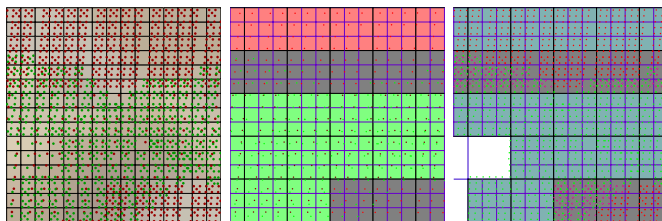


Figura 6. Filtragem da Figura 5. **Esq.:** Amostras da camada de fundo (vermelho), e da camada de objetos (azul). **Centro:** Classificação dos blocos 3×3 (magenta). Blocos com todas as amostras na camada de fundo (rosa), e blocos com vizinhança 8-conectada na mesma camada superior (verde) são resolvidos. Em cinza blocos que precisam refinamento da amostragem. **Dir.:** Refinamento da amostragem nos blocos cinzas e suas vizinhanças 8-conectada.

de visualização do mesmo exige a solução de problemas de oclusão como vimos anteriormente. Por outro lado um panorama em camadas pode fornecer uma solução ao problema de renderização composto de objetos sintéticos em cenários reais capturados. Neste caso o fundo pode ser o próprio panorama ou pode ser capturado separadamente. Possuir o panorama em camadas de uma cena permite explorar com mais liberdade a inserção de objetos virtuais, sem se preocupar se uma área oculta da cena está sendo refletida no mesmo, desde que aquela área esteja armazenada em alguma das camadas.

Outro problema que pretendemos estudar é a visualização em estéreo a partir de um panorama em camadas e, neste caso a manipulação de disparidade da visualização resultante. Além

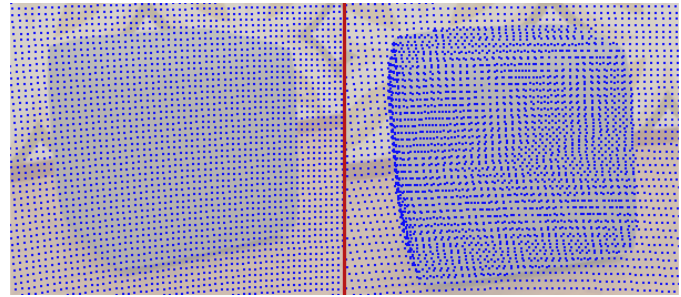


Figura 7. Distribuição de amostras na imagem final em uma visualização com amostragem regular do panorama. **Direita:** câmara localizada no centro do panorama. **Esquerda:** a câmara encontra-se deslocada do centro do panorama, fazendo com que algumas regiões de amostras se contraíam (lateral do puff).

da visualização baseada em amostras, estamos implementando um esquema de renderização baseada na geometria subjacente nas camadas do panorama, que pode ser calculada a partir da informação de profundidade e construída sob demanda.

REFERÊNCIAS

- [1] D. Q. Felinto, A. R. Zang, and L. Velho, "Production framework for full panoramic scenes with photo-realistic augmented reality," in *XXXVIII Latin American Conference of Informatics (CLEI)*, Medellin, 2012.
- [2] A. R. Zang, D. Felinto, and L. Velho, "Rendering synthetic objects into full panoramic scenes using light-depth maps," in *GRAPP & IVAPP 2013, Barcelona, Spain, 21-24 February, 2013*.
- [3] J. F. Blinn and M. E. Newell, "Texture and reflection in computer generated images," *Commun. ACM*, vol. 19, no. 10, Oct. 1976.
- [4] P. E. Debevec, "Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography." pp. 189-198: SIGGRAPH, 1998.
- [5] L. McMillan, Jr., "An image-based approach to three-dimensional computer graphics," Ph.D. dissertation, 1997.
- [6] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*. MIT Press, 1991, pp. 3-20.
- [7] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, "Real-time texture synthesis by patch-based sampling," *ACM Trans. Graph.*, vol. 20, no. 3, Jul. 2001.
- [8] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1999, pp. 1033-1038 vol.2.
- [9] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *CVPR Proceedings. IEEE Computer Society*, 2003.
- [10] P. S. Heckbert, "Fundamentals of texture mapping and image warping," EECS Dep., University of California, Berkeley, Tech. Rep., Jun 1989.