

Mapa de Saliência para Nuvem de Pontos Usando Projeções

Victor F. Figueiredo e Ricardo L. de Queiroz

Resumo— Algoritmos para a criação de mapas de saliência são bem estabelecidos para imagens, porém não há literatura sobre tais algoritmos para nuvens de pontos. Utilizamos então projeções ortográficas em planos bidimensionais para que possam ser aplicados os algoritmos já existentes na literatura para a criação de um mapa de saliência tridimensional. Os resultados de cada mapa de saliência são projetados para os voxels 3D e os resultados das múltiplas projeções são utilizados para gerar um mapa de saliência 3D. Resultados apresentam a região do rosto das nuvens de pontos como sendo a mais saliente.

Palavras-Chave— Mapa de saliência, nuvem de pontos.

Abstract— Algorithms for creating saliency maps are well established for images, even though there is no literature on such methods for point clouds. We use orthographic projections in 2D planes which are subject to well established saliency detection algorithms to create a 3D saliency map. The results of each saliency map are projected to the 3D voxels and the results of the many projections are used to generate a 3D saliency map. Results have shown the region of the face of the point clouds as being the most salient.

Keywords— Saliency map, point cloud.

I. INTRODUÇÃO

A proliferação de imagens computacionais voltadas para a detecção 3D e o aumento de aplicações 3D como direção autônoma e realidade aumentada tornou as nuvens de pontos (do inglês, *point clouds* ou PC) mais importantes [1]. Uma nuvem de pontos é composta por um conjunto de pontos no espaço representados em um sistema de coordenadas tridimensional (X, Y, Z) e comumente tem o objetivo de representar a superfície externa de um objeto ou cena. Ela é representada por sua geometria e seus atributos [2]. A geometria de uma nuvem de pontos é descrita por um conjunto V com as coordenadas de todos os pontos:

$$V = \{v_1, v_2, \dots, v_n\} = \left\{ \begin{array}{c} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \vdots \\ (x_n, y_n, z_n) \end{array} \right\} \quad (1)$$

$n = 1, \dots, N$, onde N é o número de pontos da nuvem de pontos. Os atributos podem ser representados de maneira similar por um conjunto C , em que cada entrada desse

conjunto possui D atributos:

$$C = \{c_1, c_2, \dots, c_n\} = \left\{ \begin{array}{c} (a_{11}, \dots, a_{1D}) \\ (a_{21}, \dots, a_{2D}) \\ \vdots \\ (a_{n1}, \dots, a_{nD}) \end{array} \right\} \quad (2)$$

$n = 1, \dots, N$. Comumente, os atributos incluem componentes de cor, mas podem também incluir transparência, normais, vetores de movimento e mais.

As nuvens de pontos podem apresentar regiões de interesse que têm significado ou relevância especiais [3]. Estas regiões podem ser utilizadas para manter uma maior fidelidade durante a compressão, como é bem estudado para imagens e vídeos [4].

Apesar da vasta literatura existente para a determinação de mapas de saliência e regiões de interesse em imagens e vídeos (veja, por exemplo, [5] e [6]), há pouca literatura disponível sobre a criação de mapas de saliência em nuvens de pontos [7] e o trabalho em segmentação de nuvens de pontos ainda está em desenvolvimento [8], [9], [10]. Neste artigo, propomos a criação de mapas de saliência para nuvens de pontos utilizando projeções bidimensionais.

Mapas de saliência são estudados há muitos anos [11], [12], [13], [14], [15], incluindo tecnologias modernas como redes neurais e outras. Eles foram desenvolvidos com o objetivo de identificar, em imagens, regiões que recebem uma maior atenção na visualização por humanos, tentando mimetizar a composição e o processamento da visão humana.

De acordo com [16] e [17], o propósito de um mapa de saliência é representar a visibilidade, ou saliência, em todos os locais do campo visual por uma quantidade escalar e orientar a seleção dos locais observados, com base na distribuição espacial da saliência. A saliência é definida como, primeiramente, o quão diferente um local é da sua vizinhança em termos de cor, orientação, movimentação e profundidade [11]. Dessa forma, um mapa de saliência é um mapa topograficamente organizado, que indica a localização de objetos salientes no campo visual e, não, o que são tais objetos.

II. CRIAÇÃO DO MAPA DE SALIÊNCIA DE NUVEM DE PONTOS BASEADO EM PROJEÇÃO

Muitos algoritmos de visão computacional foram desenvolvidos e estudados para imagens 2D, inclusive para criação de mapas de saliência. No entanto, não há literatura para tais algoritmos em nuvem de pontos. Reconhecemos o nível de dificuldade para desenvolver soluções que atuem diretamente no espaço 3D esparsa, mas conhecemos diversas soluções para

Victor F. Figueiredo, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília-DF, e-mail: fabre@ieee.org; Ricardo L. de Queiroz, Departamento de Ciência da Computação, Universidade de Brasília, Brasília-DF, e-mail: queiroz@ieee.org. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal Superior - Brasil (CAPES) - Código de Financiamento 001. Este trabalho foi parcialmente financiado por CNPq (301647/2018-6).

a resolução do problema no espaço 2D. A solução apresentada nesse artigo é a utilização de projeções 2D, para que assim possam ser gerados os mapas de saliência em um espaço bidimensional e, então, o mapeamento dos pixels das imagens nos voxels 3D correspondentes.

Em outras palavras, tarefas de visão computacional 3D são executadas em projeções 2D com assistência da re-projeção dos pixels 2D nos voxels 3D. As informações de várias projeções são agregadas para que sejam obtidas informações sobre toda a nuvem de pontos. A ideia é ilustrada na Fig. 1.

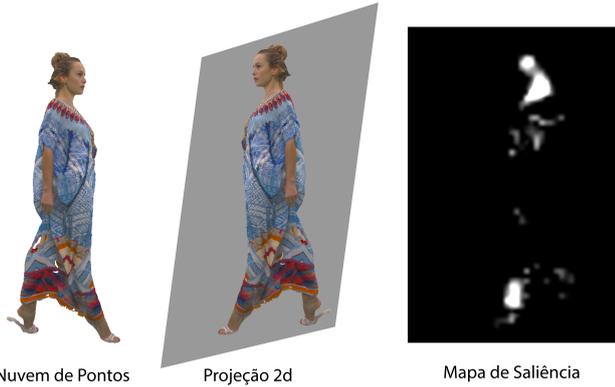


Fig. 1. Criação de mapa de saliência utilizando projeção de nuvem de pontos em duas dimensões.

Começamos projetando ortogonalmente a nuvem de pontos P no plano 2D I . Se imaginarmos um voxel como um cubo 3D e um pixel como um quadrado de tamanho da face do cubo e, se projetarmos ortogonalmente qualquer uma das seis faces do cubo, é possível mapear de maneira única a face do voxel para um pixel no plano 2D projetado. Consequentemente, o mapeamento $P \rightarrow I$ seria reversível. Caso seja projetado em uma direção oblíqua, a projeção do cubo não será um quadrado e, sim, um polígono mais complexo. Tal projeção não se encaixaria em um pixel quadrado e seria parcialmente projetada em pixels adjacentes. Para lidar com tal situação existem diversas soluções que variam em grau de precisão e complexidade. Em $P \rightarrow I$ e $I \rightarrow P$, uma solução é computar a cor do voxel ou pixel por uma combinação linear de várias projeções.

Uma alternativa é aumentar a resolução da nuvem de pontos ao replicar voxels e pixels e simplesmente atribuir a cor do voxel ao pixel com a maior área de projeção correspondente. Na re-projeção $I \rightarrow P$ é possível atribuir o voxel cujo centro está mais próximo da linha de projeção de um pixel no plano 2D a este pixel. Após todos os voxels serem atribuídos a um pixel correspondente, é necessário reduzir a nuvem de pontos à sua resolução original. Problemas de interpolação semelhantes surgem mesmo que não se assumam voxels cúbicos e pixels quadrados. É essencial ser capaz de mapear voxels para pixels e depois remapear pixels para voxels.

O algoritmo de criação de mapa de saliência baseado em projeção funciona da seguinte forma:

- Mapeie os voxels 3D em um plano na direção (θ, ϕ) , onde $-90^\circ \leq \theta \leq +90^\circ$ é a elevação e $0^\circ \leq \phi \leq 360^\circ$ é o azimute, ver Fig. 2.

- Gere o mapa de saliência 2D, atribuindo o peso a cada pixel que corresponde a um voxel.
- Mapeie os pixels novamente para os voxels 3D. Como um pixel pode ser mapeado para múltiplos voxels, pode-se optar por um arredondamento ou outros métodos de decisão.

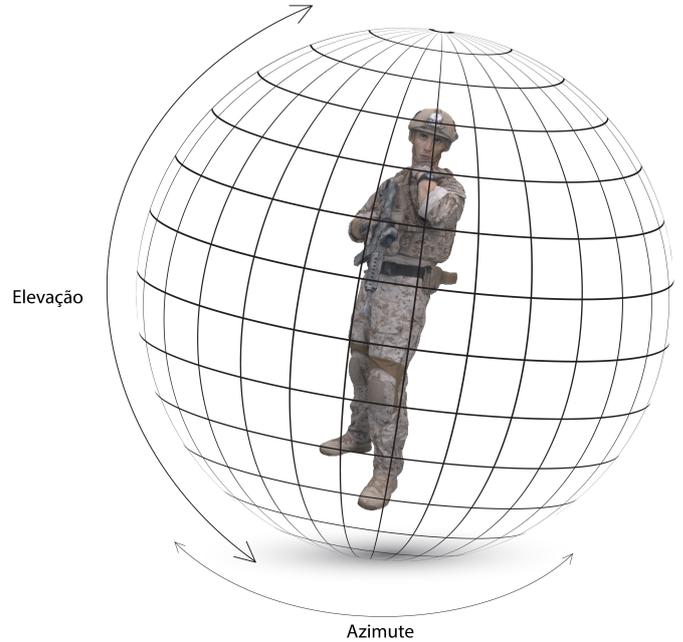


Fig. 2. Representação da nuvem de pontos e ângulos de elevação e azimute.

Com o algoritmo acima, dada uma nuvem de pontos voxelizada e o par (θ, ϕ) , obtemos o valor de saliência de um conjunto de voxels. Contudo, várias direções são testadas uma vez que não sabemos qual vista é a mais relevante para a visualização do usuário. Escaneamos o espaço (θ, ϕ) ao variar θ de θ_{min} a θ_{max} em passos de $\Delta\theta$ e ϕ de ϕ_{min} a ϕ_{max} em passos de $\Delta\phi$.

O algoritmo é genérico por natureza e, para construir os mapas de saliência, utilizamos o algoritmo desenvolvido por Walther e Koch [5]. Os parâmetros utilizados nos testes foram $\theta_{min} = -70^\circ$, $\theta_{max} = 90^\circ$, $\phi_{min} = 0^\circ$, $\phi_{max} = 359^\circ$, $\Delta\theta = \Delta\phi = 10^\circ$. Dessa forma, $N_\theta = 17$ e $N_\phi = 36$, são feitas $N_a = 612$ projeções para cada nuvem de pontos. Após as 612 projeções, o valor de saliência de um voxel é a soma do valor de saliência do pixel correspondente em cada projeção. Este método para fusão dos atributos é semelhante ao apresentado em [17]. No fim de todas as projeções, os valores de saliência dos voxels são normalizados para um intervalo de contínuo de 0,0 a 1,0. Com o intuito de suavizar a transição da região saliente para a região não saliente, é proposto um filtro espacial passa-baixas com um *kernel* cúbico de tamanho $9 \times 9 \times 9$.

Em resumo, o algoritmo pode ser descrito como:

- Para $\theta = \theta_{min} : \Delta\theta : \theta_{max}$
Para $\phi = \phi_{min} : \Delta\phi : \phi_{max}$
-nuvem de ponto é projetada na direção (θ, ϕ) ;
-rodar o algoritmo de criação de mapa de saliência;
-re-projete o valor de saliência dos pixels nos voxels;

- Normalize os valores de saliência para um intervalo de 0, 0 a 1, 0;
- Filtre o mapa de saliência 3D com um filtro passa-baixas.

III. RESULTADOS EXPERIMENTAIS

Para testar a criação dos mapas de saliência utilizamos 5 nuvens de pontos: Boxer, David, Longdress, Loot e Soldier, todas voxelizadas com profundidade 10 (i.e. $1024 \times 1024 \times 1024$ voxels) [18], [19], [20].

Os resultados são apresentados nas Figs. 3 a 8 na forma de um mapa de saliência clássico (i.e. em níveis de cinza) e de um mapa de saliência com uma coloração por temperatura, onde cores mais próximas do vermelho representam um maior valor de saliência e cores mais próximas do azul um menor valor de saliência.

É possível notar que em todos os exemplos a região mais saliente contém o rosto, ou uma parte dele, e em alguns casos (como nas Figs. 6 e 8) uma região próxima ao rosto também é considerada como sendo mais saliente que o resto.

IV. TRABALHOS FUTUROS

Por conter uma grande quantidade de dados, é necessário realizar a compressão das nuvens de pontos para qualquer aplicação prática. Diversos algoritmos de compressão de nuvem de pontos existem na literatura [21], [22]. Atualmente, o MPEG possui o objetivo de padronizar tais tecnologias [23]. No entanto, não há na literatura nenhuma proposta que contenha a compressão de nuvem de pontos utilizando mapas de saliência. Portanto, trabalhos futuros incluirão um algoritmo que utilize mapas de saliência para a determinação de regiões de interesse com pesos variáveis para a compressão de nuvens de pontos, baseando-se em [3].

V. CONCLUSÃO

Introduzimos a criação de mapas de saliência para nuvens de pontos, fazendo o uso de algoritmos e conceitos já existentes na literatura para a criação de tais mapas em imagens 2D. Pela grande dificuldade da determinação dos atributos diretamente em um espaço tridimensional esparsa, foram utilizadas projeções bidimensionais de diferentes vistas da nuvem de pontos e os atributos foram determinados nestas projeções e re-projetados na nuvem de pontos com a utilização de algoritmos já estabelecidos na literatura. Os resultados de cada mapa de saliência são projetados para os voxels 3D e os resultados das múltiplas projeções são utilizados para gerar um mapa de saliência 3D. Em todos os testes realizados a região mais saliente contém o rosto ou uma parte dele.

Trabalhos futuros incluem a implementação da codificação de nuvem de pontos, utilizando mapas de saliência para a determinação de regiões de interesse com pesos multi-nível.

REFERÊNCIAS

- [1] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko, "Emerging MPEG standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2019.



Fig. 3. Uma vista da nuvem de pontos "David" e seus respectivos mapas de saliência.



Fig. 4. Uma vista da nuvem de pontos "Boxer" e seus respectivos mapas de saliência.



Fig. 5. Uma vista da nuvem de pontos "Loot" e seus respectivos mapas de saliência.



Fig. 6. Uma vista da nuvem de pontos "Longdress" e seus respectivos mapas de saliência.



Fig. 7. Uma vista da nuvem de pontos "Soldier"(quadro 537) e seus respectivos mapas de saliência.

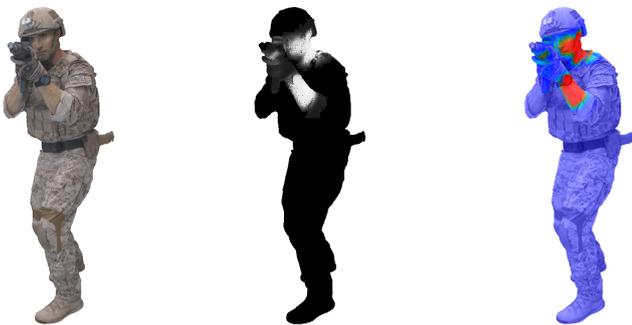


Fig. 8. Uma vista da nuvem de pontos "Soldier"(quadro 695) e seus respectivos mapas de saliência.

- [2] G. Sandri, R. L. de Queiroz, and P. A. Chou, "Comments on "Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform","" *ArXiv e-prints*, May 2018.
- [3] G. Sandri, V. F. Figueiredo, P. A. Chou, and R. de Queiroz, "Point cloud compression incorporating region of interest coding," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 4370–4374.
- [4] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 19–33, 2014.
- [5] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, pp. 1395–1407, 2006.
- [6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, Dec 2001, pp. I–I.
- [7] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "Pointcloud saliency maps," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1598–1606.
- [8] Z. Wen, Y. Yan, and H. Cui, "Study on segmentation of 3d human body based on point cloud data," in *2012 Second International Conference on Intelligent System Design and Engineering Application*, 2012, pp. 657–660.
- [9] M. Qiao, J. Cheng, W. Bian, and D. Tao, "Biview learning for human posture segmentation from 3d points cloud," *Plos one*, vol. 9, p. e85811, Jan. 2014.
- [10] P. Mandikal, N. K. L., and R. V. Babu, "3d-psrnet: Part segmented 3d point cloud reconstruction from a single image," *CoRR*, vol. abs/1810.00461, 2018. [Online]. Available: <http://arxiv.org/abs/1810.00461>
- [11] C. Koch and S. Ullman, "Shifts in selective visual attention: Towards the underlying neural circuitry," *Matters of Intelligence: Conceptual Structures in Cognitive Neuroscience*, pp. 115–141, 1987.
- [12] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *ArXiv e-prints*, Dec. 2013.
- [13] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1265–1274.

- [14] A. Maity, "Improvised salient object detection and manipulation," *International Journal of Image, Graphics and Signal Processing*, vol. 8, pp. 53–60, 11 2015.
- [15] E. Mendi and M. Milanova, "Image segmentation with active contours based on selective visual attention," in *8th WSEAS International Conference on Signal Processing*, May 2009.
- [16] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [17] E. Niebur and C. Koch, "Control of selective visual attention: Modeling the 'where' pathway," *Neural Information Processing Systems*, vol. 8, pp. 802–808, 1996.
- [18] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "8i voxelized full bodies — a voxelized point cloud dataset," ISO/IEC JTC1/SC29/WG1 & WG11 JPEG & MPEG, input documents M74006 & m40059, Jan. 2017.
- [19] M. Krivokuća, P. A. Chou, and P. Savill, "8i voxelized surface light field (8iVSLF) dataset," ISO/IEC JTC1/SC29/WG11 MPEG, input document m42914, Jul. 2018.
- [20] C. Loop, Q. Cai, S. O. Escolano, and P. A. Chou, "Microsoft voxelized upper bodies - a voxelized point cloud dataset," 2017, provided by Microsoft <https://jpeg.org/plenodb/pc/microsoft/>.
- [21] R. L. de Queiroz and P. A. Chou, "Motion-compensated compression of dynamic voxelized point clouds," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3886–3895, 2017.
- [22] G. Sandri, R. L. de Queiroz, and P. A. Chou, "Compression of plenoptic point clouds," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1419–1427, 2019.
- [23] 3D Graphics Subgroup, "Call for proposals for point cloud compression v2," Abril 2017.