

Codificação Progressiva (*Embedded*) de Região de Interesse para Atributos de Nuvem de Pontos

Victor Fabre Figueiredo e Ricardo Lopes de Queiroz

Resumo— Este artigo apresenta uma nova abordagem para a codificação progressiva (*embedded*) de atributos de regiões de interesse (ROI, do inglês *Region of Interest*) em nuvens de pontos. O método proposto é um codificador progressivo de atributos de nuvens de pontos incorporando regiões de interesse. O método escolhido foi a intercalação de *bit-streams* para regiões ROI e não ROI, o que nos traz algumas vantagens. Em primeiro lugar, é independente de qualquer algoritmo específico de detecção de ROI. Em segundo lugar, não é acrescida qualquer complexidade ao decodificador. Os resultados demonstram uma melhoria na qualidade dos voxels reconstruídos dentro da ROI, acompanhada de alguma degradação dos voxels reconstruídos fora da ROI.

Palavras-Chave— Codificação progressiva, região de interesse, compressão de nuvens de pontos.

Abstract— This paper presents a novel approach to embedded Region-of-Interest (ROI) attribute coding for point clouds. The proposed method is an embedded point cloud attribute coder incorporating regions of interest. The method chosen was to interleave bit-streams for ROI and non-ROI regions, which brings us some advantages. Firstly, it is independent of any specific ROI detection algorithm. Secondly, no increased complexity is imposed on the decoder. The results demonstrate an improvement in the quality of reconstructed voxels within the ROI, accompanied by some degradation of reconstructed voxels outside the ROI.

Keywords— Embedded coding, region of interest, point cloud compression.

I. INTRODUÇÃO

Uma nuvem de pontos (PC, do inglês *Point Cloud*) é geralmente representada por um par de vetores, um para a geometria e outro para os atributos. O vetor de geometria descreve as coordenadas espaciais dos pontos e o vetor de atributos compreende outras informações, como cor, transparência, vetores normais, vetores de movimento e mais [1], [2]. Uma PC geralmente é composta por um grande número de pontos e, por causa disso, o Moving Pictures Expert Group (MPEG) desenvolveu algoritmos padrão para a compressão de PC [3], [4]. Neste trabalho, estamos interessados na compressão de atributos da nuvem de pontos e assumimos que a geometria da nuvem de pontos já foi codificada e transmitida ao decodificador.

Nuvens de pontos podem ter regiões de interesse (ROI, do inglês *Regions of Interest*) com significado ou relevância especial [5], [6]. Essas regiões podem ser usadas para aumentar

seletivamente a fidelidade durante a compressão, como ocorre na compressão de imagens e vídeos [7], [8], [9], [10], [11].

O algoritmo de Particionamento de Conjunto em Árvore Hierárquicas (SPIHT, do inglês *Set Partitioning in Hierarchical Trees*) [12] pode ser usado para codificar coeficientes derivados da decomposição por wavelet de uma imagem em um *embedded bit-stream*. Um *embedded bit-stream* é aquele que permite a decodificação de qualquer prefixo do *bit-stream*, onde prefixos mais longos proporcionam melhor reconstrução em taxas de bits mais altas e menores razões de compressão. O *bit-stream* completo pode ser decodificado para obter uma reconstrução sem perdas ou quase sem perdas. Em outras palavras, um único *bit-stream* é gerado para todas as aplicações e melhor qualidade é alcançada enviando mais bits do *bit-stream* único. Esta característica intrínseca permite a incorporação de *bit-stream* de qualidade inferior dentro de *bit-stream* de qualidade superior. Tal propriedade prova ser vantajosa, facilitando aplicações como transmissão *embedded*, *multicast*, degradação graciosa e proteção desigual contra erros [13], [14], [15]. Em [16], foi apresentado um método de codificação *embedded* de atributos de nuvem de pontos.

Inspirados por [5], [6], [16], propomos um método de combinar codificação *embedded* de atributos com ROI para nuvens de pontos. Nossa abordagem tem a vantagem de, em vez de adaptar cada codec de uma maneira específica para ajustar sua fidelidade, utilizarmos a ROI para modificar a medida de distorção. Existe um mapeamento simples da ROI para a medida de distorção, que pode ser quantificado (por exemplo, usando experimentos perceptuais) independentemente de qualquer codec progressivo em particular. Como nosso codec, escolhemos a codificação por transformação com Transformação Hierárquica Adaptativa à Região (RAHT, do inglês *Region-Adaptive Hierarchical Transform*) [17] porque é automaticamente otimizado para a medida de distorção em virtude de sua interpretação medida-teórica [18] e pode ser usado em um codificador progressivo. O método proposto também é independente de qualquer algoritmo de detecção de região de interesse ou saliência.

Embora exista literatura disponível sobre codificação *embedded* de ROI para imagens [19], [20], acreditamos que este é o primeiro artigo publicado sobre codificação de atributos de região de interesse *embedded* para nuvens de pontos.

II. CODIFICAÇÃO *Embedded* DE NUVEM DE PONTOS

O algoritmo SPIHT [12] é um codificador de *bit planes*. No codificador, é fornecido um passo Δ , um conjunto de coeficientes de valor real $\{c_t : t \in T\}$ indexados por um

Victor Fabre Figueiredo, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília-DF, e-mail: fabre@ieee.org; Ricardo Lopes de Queiroz, Departamento de Ciência da Computação, Universidade de Brasília, Brasília-DF, e-mail: queiroz@ieee.org. Este trabalho foi parcialmente financiado por Coordenação de Aperfeiçoamento de Pessoal Superior - Brasil (CAPES) - Finance Code 001 e CNPq (302957/2022-7).

conjunto T , e uma partição hierárquica de conjuntos de T . O codificador quantiza cada coeficiente c_t em um número inteiro $k_t = \text{round}(c_t/\Delta)$, e então transmite os *bit planes* de $|k_t|$ juntamente com $\text{sign}(k_t)$ se k_t for diferente de zero. Os *bit planes* são transmitidos do bit mais significativo $b = b_{\max} = \lfloor \log_2(\max_{t \in T} |k_t|) \rfloor$ para o bit menos significativo $b = 0$. Relativo ao *bit plane* b , um coeficiente c_t é *significativo* se $\lfloor \log_2(|k_t|) \rfloor \geq b$. Caso contrário, é *insignificante*.

Se um coeficiente c_t for insignificante, então é provável que um conjunto inteiro de coeficientes também seja insignificante. Se todos os coeficientes no conjunto forem insignificantes, então um único bit pode sinalizar isso. Caso contrário, bits adicionais podem sinalizar a significância de todos os coeficientes no conjunto sinalizando uma hierarquia predefinida de subconjuntos. Se um coeficiente c_t se torna significativo no *bit plane* b (enquanto não era significativo no *bit plane* $b+1$), então o sinal de c_t deve ser transmitido. Se um coeficiente c_t já se tornou significativo no *bit plane* $b+1$, então ele permanece significativo no *bit plane* b e, portanto, não precisa ser testado para significância. No entanto, o b -ésimo bit em $|c_t|$, chamado de bit de refinamento, deve ser transmitido. Para coeficientes RAHT, a hierarquia de conjuntos SPIHT é determinada pela árvore na qual os coeficientes RAHT são naturalmente organizados. Esta árvore depende apenas da geometria e é assumida como conhecida no decodificador.

III. CODIFICAÇÃO *Embedded* DE REGIÃO DE INTERESSE

Na compressão de nuvem de pontos, há um equilíbrio entre o número de bits gastos para codificar a nuvem de pontos e a qualidade da nuvem de pontos reconstruída no decodificador. Quanto maior a qualidade, mais bits são necessários. É suposto que regiões de interesse tenham uma significância semântica ou perceptual maior do que o restante da nuvem de pontos. Portanto, um codificador que prioriza a qualidade dessas regiões, enquanto degrada as outras, tende a produzir nuvens de pontos reconstruídas com uma melhor qualidade subjetiva, quando comparado a um codificador que trata todas as regiões igualmente, para o mesmo número de bits.

Neste trabalho, o rosto do sujeito na nuvem de pontos é escolhido como a ROI. Como o cérebro humano tende a ser mais sensível a artefatos introduzidos no rosto em nuvens de pontos reconstruídas, acredita-se que priorizar a qualidade do rosto do sujeito durante a compressão pode levar a uma melhor qualidade subjetiva [21].

O decodificador deve receber informações sobre a localização da ROI. Supondo que existam M voxels ocupados na nuvem de pontos, é necessário codificar um vetor binário $b = [b_0, b_1, \dots, b_{M-1}]$, representando a inclusão ou exclusão de cada voxel da ROI. Ao ordenar os voxels com base em seus códigos Morton [22] para manter os relacionamentos de vizinhança, os bits b_i podem ser empregados para construir um vetor diferencial $\bar{b} = [\bar{b}_0, \bar{b}_1, \dots, \bar{b}_{M-1}]$ onde

$$\bar{b}_i = \begin{cases} b_0 & i = 0 \\ 1 & b_{i-1} \neq b_i, i > 0 \\ 0 & b_{i-1} = b_i, i > 0 \end{cases} \quad (1)$$

O vetor \bar{b} exibe sequências prolongadas de zeros. Ele é codificado com um algoritmo baseado no codificador run-

length Golomb-Rice [23], embora com uma modificação onde apenas os *run-lengths* são codificados usando Golomb-Rice.

Para alcançar um totalmente *embedded* que priorize a região de interesse, algumas adaptações precisaram ser feitas nos trabalhos anteriormente desenvolvidos. O método proposto é usar o que chamamos de um entrelaçado. Ou seja, para o mesmo bit-stream, há uma parte do que transmite informações dos voxels da ROI e uma parte que transmite informações sobre os voxels não-ROI. Os bits são entrelaçados de tal forma que, como o decodificador já conhece a ROI, ele pode decodificar o completo como um *embedded bit-stream*.

A ROI é priorizada à medida que o codificador envia uma proporção diferente de bits ROI e não-ROI, no nível de entrelaçamento da codificação. Digamos que tenhamos um nível de entrelaçamento ℓ . Um nível de entrelaçamento de 0 é o mesmo que uma codificação regular sem nenhuma priorização da ROI. À medida que aumentamos ℓ , a ROI é mais priorizada. ℓ significa que, para cada bit não-ROI enviado ao decodificador, 2^ℓ bits ROI são enviados.

As figuras 1 e 2 ilustram diferentes reconstruções da nuvem de pontos "Longdress" com aproximadamente o mesmo número de bits usados pelo decodificador em diferentes níveis de entrelaçamento.

IV. RESULTADOS EXPERIMENTAIS

Para testar o codificador proposto, usamos 6 nuvens de pontos: Boxer, Longdress, Loot, Redandblack, Soldier e Thaidancer, todas voxelizadas com profundidade 10 (ou seja, $1024 \times 1024 \times 1024$ voxels) [24], [25]. A PSNR é calculada como uma PSNR ponderada definida como

$$PSNR(\mathcal{P}, \hat{\mathcal{P}}, \{w_i\}) = 10 \log_{10} \left(\frac{255^2}{E_{noise}} \right), \quad (2)$$

onde w_i é o peso do voxel, potencialmente diferente para voxels ROI e não-ROI, e a energia do ruído pode ser redefinida como

$$E_{noise} = \frac{\sum_i w_i (Y_i - \hat{Y}_i)^2}{\sum_i w_i}. \quad (3)$$

\mathcal{P} é a nuvem de pontos original, $\hat{\mathcal{P}}$ é a nuvem de pontos reconstruída, Y_i é o valor original e \hat{Y}_i é o valor reconstruído, para o canal Y , onde o peso de cada voxel é $w_{ROI} = 2^\ell$, se estiver dentro da ROI, e 1, caso contrário.

A figura 3 mostra as curvas médias de taxa-distorção calculadas para as 6 nuvens de pontos testadas para diferentes valores de ℓ . Usamos $\ell = 0, 2, 4, 6, 8, 10$, resultando em $w_{ROI} = 1, 4, 16, 64, 256, 1024$, enquanto os voxels fora da ROI têm peso $w = 1$. Na Figura 3a, o PSNR é calculado apenas para voxels na ROI. Bits para as informações laterais são considerados. No entanto, quando $w = 1$, não há diferença no peso entre voxels situados dentro e fora da ROI. Neste cenário específico, codificar o vetor \bar{b} é desnecessário. Conforme ℓ aumenta, o codificador favorece os bits da ROI e vemos um aumento no PSNR para voxels dentro da ROI. O oposto acontece para voxels fora da ROI (figura 3b). A PSNR ponderada aumenta drasticamente para $\ell > 4$ e melhora



Fig. 1: Projeções da nuvem de pontos "Longdress" codificadas com diferentes níveis de entrelaçamento. O número de bits usado na decodificação foi ajustado para que cada reconstrução tivesse aproximadamente 0.91 bpov.



(a) $\ell = 0$



(b) $\ell = 2$

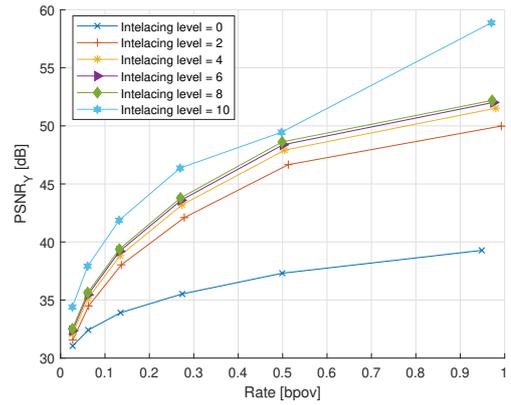


(c) $\ell = 10$

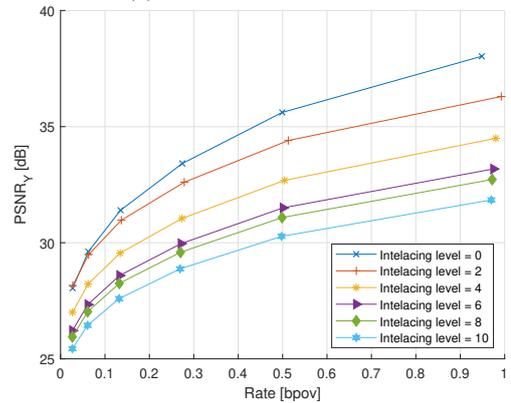
Fig. 2: Close-up na ROI das nuvens de pontos reconstruídas mostradas na Figura 1. Note que para $\ell = 2$ a qualidade da ROI pode ser suficientemente alta.

ligeiramente para taxas de bits mais baixas com $0 < \ell \leq 4$ (Figura 3c). A PSNR sobre todos os voxels diminui para $\ell > 0$ (figura 3d).

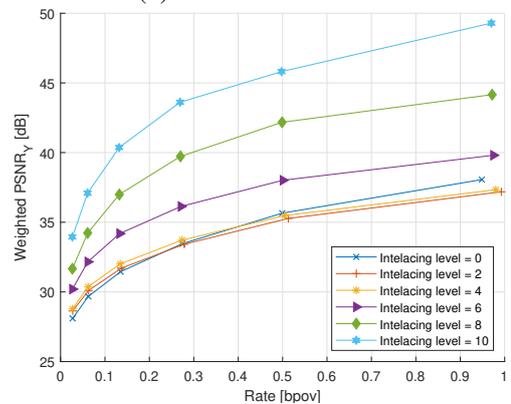
Subjetivamente, da figura 2, as reconstruções com maior nível de entrelaçamento parecem ter uma qualidade maior, uma vez que nosso cérebro é mais sensível a artefatos no rosto do que no restante da cena. A reconstrução com $\ell = 2$ parece ter uma qualidade da ROI suficientemente alta. Essencialmente, a escolha de ℓ representa um compromisso e um valor ideal pode ter que ser cuidadosamente escolhido para um valor em que a qualidade da ROI seja satisfatória e um aumento adicional de qualidade possa não valer a pena a diminuição adicional de qualidade das regiões não-ROI.



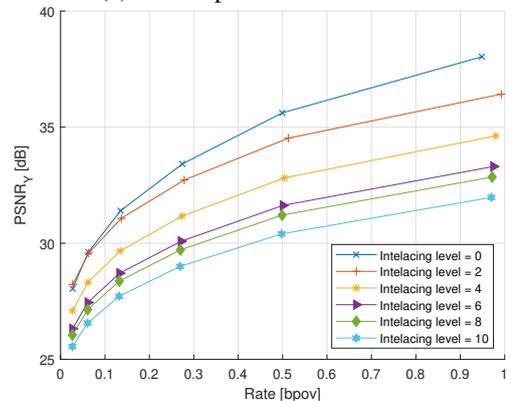
(a) PSNR dentro da ROI.



(b) PSNR fora da ROI.



(c) PSNR ponderada total PC.



(d) PSNR total da PC.

Fig. 3: Curvas médias de taxa-distorção.

V. CONCLUSÕES

Introduzimos a codificação *embedded* de atributos de região de interesse para nuvens de pontos, por meio do uso de *embedded bit-streams* entrelaçados para os voxels da ROI e não-ROI da nuvem de pontos. O método de entrelaçamento proposto é agnóstico ao codec progressivo de nuvem de pontos ou ao algoritmo de detecção de região de interesse. Os resultados mostraram um aumento na qualidade dos voxels reconstruídos dentro da região de interesse às custas de alguma degradação fora da ROI, com pouca alteração na complexidade do codificador ou decodificador. A escolha do nível de entrelaçamento deve ser feita cuidadosamente, uma vez que representa um compromisso entre qualidade da ROI e não-ROI e, nesse ponto, o aumento adicional na qualidade da reconstrução da ROI pode não valer a pena a diminuição na qualidade da não-ROI. Acreditamos que nosso codificador pode ser muito útil em várias aplicações.

REFERÊNCIAS

- [1] M. Levoy and T. Whitted, “The use of points as a display primitive,” Technical report, University of North Carolina at Chapel Hill, North Carolina, United States, January 1985.
- [2] S. Liu, M. Zhang, P. Kadam, and C. C. J. Kuo, *3D Point Cloud Analysis: Traditional, Deep Learning, and Explainable Machine Learning Methods*, Springer, 1st edition, 2021.
- [3] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, “An overview of ongoing point cloud compression standardization activities: video-based (V-PCC) and geometry-based (G-PCC),” *APSIPA Transactions on Signal and Information Processing*, vol. 9, pp. e13, 2020.
- [4] 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.
- [5] 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)*, 2019, pp. 4370–4374.
- [6] V. F. Figueiredo, G. L. Sandri, R. L. de Queiroz, and P. A. Chou, “Saliency maps for point clouds,” in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020, pp. 1–5.
- [7] C. Christopoulos, A. Skodras, and T. Ebrahimi, “The jpeg2000 still image coding system: an overview,” *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, 2000.
- [8] L. Itti, “Automatic foveation for video compression using a neurobiological model of visual attention,” *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1304–1318, 2004.
- [9] Y. Liu, Z. Guo Li, and Y. C. Soh, “Region-of-interest based resource allocation for conversational video communication of h.264/avc,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 134–139, 2008.
- [10] D. Yee, S. Soltaninejad, D. Hazarika, G. Mbuyi, R. Barnwal, and A. Basu, “Medical image compression based on region of interest using better portable graphics (bpg),” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 216–221.
- [11] H. Hadizadeh and I. V. Bajić, “Saliency-aware video compression,” *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 19–33, 2014.
- [12] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, 1996.
- [13] V. Ruiz, J. Fernández, and I. García Fernandez, “Image compression for progressive transmission,” *The Nineteenth IASTED International Conference on Applied Informatics: Advances in Computer Applications*, 01 2001.
- [14] S. McCanne, V. Jacobson, and M. Vetterli, “Receiver-driven layered multicast,” in *Conference Proceedings on Applications, Technologies, Architectures, and Protocols for Computer Communications*, New York, NY, USA, 1996, SIGCOMM ’96, p. 117–130, Association for Computing Machinery.
- [15] A.E. Mohr, E.A. Riskin, and R.E. Ladner, “Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction,” *Selected Areas in Communications, IEEE Journal on*, vol. 18, pp. 819 – 828, 07 2000.
- [16] A. L. Souto, V. F. Figueiredo, P. A. Chou, and R. L. de Queiroz, “Set partitioning in hierarchical trees for point cloud attribute compression,” *IEEE Signal Processing Letters*, vol. 28, pp. 1903–1907, 2021.
- [17] R. L. de Queiroz and P. A. Chou, “Compression of 3D point clouds using a region-adaptive hierarchical transform,” *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3947–3956, aug 2016.
- [18] P. A. Chou, M. Koroteev, and M. Krivokuća, “A volumetric approach to point cloud compression—part i: Attribute compression,” *IEEE Transactions on Image Processing*, vol. 29, pp. 2203–2216, 2020.
- [19] E. Kang, H. Choi, and S. Ko, “Progressive region of interest coding using an improved embedded zerotree wavelet coding,” in *Proceedings of IEEE. IEEE Region 10 Conference. TENCN 99. 'Multimedia Technology for Asia-Pacific Information Infrastructure' (Cat. No.99CH37030)*, 1999, vol. 1, pp. 609–612 vol.1.
- [20] M.M. Subedar, L.J. Karam, and G.P. Abousleman, “An embedded scaling-based arbitrary shape region-of-interest coding method for jpeg2000,” in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004, vol. 3, pp. iii–681.
- [21] E. Alexiou, I. Viola, T. M. Borges, T. A. Fonseca, R. L. de Queiroz, and T. Ebrahimi, “A comprehensive study of the rate-distortion performance in mpeg point cloud compression,” *APSIPA Transactions on Signal and Information Processing*, vol. 8, pp. e27, 2019.
- [22] G. M. Morton, “A computer oriented geodetic data base; and a new technique in file sequencing,” Technical report, IBM, Ottawa, Canada, 1966.
- [23] H.S. Malvar, “Adaptive run-length / golomb-rice encoding of quantized generalized gaussian sources with unknown statistics,” in *Data Compression Conference*. March 2006, Institute of Electrical and Electronics Engineers, Inc.
- [24] E. d’Eon, B. Harrison, T. Myers, and P. A. Chou, “8i voxelized full bodies — a voxelized point cloud dataset,” input documents M74006 & m40059, ISO/IEC JTC1/SC29/WG1 & WG11 JPEG & MPEG, Jan. 2017.
- [25] M. Krivokuća, P. A. Chou, and P. Savill, “8i voxelized surface light field (8iVSLF) dataset,” input document m42914, ISO/IEC JTC1/SC29/WG11 MPEG, July 2018.