Compressão de Nuvem de Pontos Incorporando Codificação de Região de Interesse

Gustavo Sandri, Victor F. Figueiredo, Philip A. Chou, e Ricardo de Queiroz

***Resumo*— Introduzimos codificação de região-de-interesse (ROI) para atributos de nuvem de pontos, utilizando uma medida de distorção de entrada ponderada, onde a ROI determina os pesos. Na codificação, é usada a transformada hierárquica adaptativa por região (RAHT), que depende de um conjunto de pesos. Usamos uma interpretação teórica da medida de distorção do RAHT para determinar quais os pesos da transformação devem ser definidos para os pesos da medida de distorção. A ROI é escolhida como a região 3D do rosto, detectada a partir de projeções 2D. Resultados experimentais mostram, subjetivamente, melhorias significativas na ROI com degradações subjetivamente insignificantes na não-ROI.**

***Palavras-Chave*— nuvem de pontos, região de interesse, RAHT.**

***Abstract*— We introduce region-of-interest (ROI) coding for point cloud attributes, using an input-weighted distortion measure where the ROI determine the weights. In terms of coding, we use the region adaptive hierarchical transform (RAHT), which relies on a set of weights. We use a measure-theoretic interpretation of RAHT to determine that the weights of the transform should be set to the weights of the distortion measure. The ROI is chosen as the 3D region of the face, which is detected from a set of 2D projections. Experimental results show subjectively meaningful improvements in a face ROI with subjectively insignificant degradations in the non-ROI.**

***Keywords*— Point cloud, region of interest, RAHT.**

I. Introdução

As nuvens de pontos (do inglês, *point clouds*), que representam o mundo 3D por amostragem, tem se tornado cada vez mais importantes nos últimos anos devido à proliferação de imagens computacionais voltadas para a detecção 3D.

Assim como imagens e vídeos brutos, as nuvens de pontos e as nuvens de pontos dinâmicas contêm grande quantidade de dados. Portanto, a compressão de nuvens de pontos é necessária para qualquer aplicação prática. O MPEG está atualmente padronizando um formato de compressão de nuvem de pontos (PCC) com essa finalidade [1].

Como as imagens e vídeos, as nuvens de pontos geralmente têm regiões de interesse (ROI) que têm significado ou relevância especiais - por exemplo, rostos - para os quais a preservação da alta fidelidade durante a compressão pode ser importante. Para imagens e vídeos, a compressão por ROI ou *codificação ROI* é bem estudada. (Veja, por exemplo, [2].) No entanto, para nuvens de pontos, não há literatura disponível sobre codificação de ROI. Neste artigo, propomos a codificação de ROI para nuvens de pontos.

Gustavo Sandri, Universidade de Brasília, e-mail: gustavo.sandri@ieee.org; Victor F. Figueiredo, Universidade de Brasília, e-mail: victorfabre@icloud.com; Philip A. Chou, Google, Seattle, WA USA, e-mail: pachou@ieee.org; Ricardo de Queiroz, Universidade de Brasília, e-mail: queiroz@ieee.org.

As nuvens de pontos consistem em geometria e todos os seus atributos. A parte geométrica de uma nuvem de pontos é, simplesmente, uma lista de posições 3D $\{\mathbf{x}_i\} = \{(x_i, y_i, z_i)\}$, $i = 1, \ldots, N$, onde N é o número de pontos da nuvem de pontos. Os atributos de uma nuvem de pontos são dados por uma lista de atributos $\{\mathbf{a}_i\} = \{(a_{i1}, \ldots, a_{iD})\}$, i=$1, \ldots, N$, onde D é o número de atributos por ponto. Comumente, os atributos incluem componentes de cor (Y_i, U_i, V_i), mas podem também incluir transparência, normais, vetores de movimento e mais. Uma vez que a geometria é dada, os atributos podem ser vistos como um sinal definido por um conjunto de pontos.

A maioria dos *codecs* de nuvens de pontos existentes na literatura comprime a geometria primeiro e depois comprime os atributos dada a geometria. Abordagens típicas para a codificação dos atributos incluem codificação de transformada usando graph fourier transform (GFT) [3], [4], [5], [6], [7], [8], o gaussian process transform (GPT) [9], [10], e a transformada hierárquica adaptativa por região (RAHT) [11], [12]. RAHT, diferentemente da GFT e da GPT, não requer uma auto-decomposição, e tem sido uma das transformadas inicialmente adotadas no MPEG PCC [1]. Codificação ROI para nuvens de pontos pode ser aplicada para geometria, atributos ou ambos. Por exemplo, para compressão da geometria, a ROI pode ser usada para ajustar o nível geométrico de detalhe ao ajustar a profundidade da *octree*. No entanto, neste artigo, focamos na codificação ROI dos atributos. Para compressão dos atributos, a ROI pode ser usada, por exemplo, para ajustar o passo de variação dos coeficientes da transformada, como é comum em codificação ROI de imagens e vídeos. Contudo, temos uma abordagem diferente.

Inspirados por uma recente interpretação de medida teórica da RAHT [13], na qual a RAHT é mostrada como uma transformada wavelet 3D separável que é ortonormal em respeito a uma medida de contagem uniforme de um conjunto de pontos, nós implementamos a codificação ROI ao modificar a medida e usando RAHT normalmente. Modificar a medida é equivalente a modificar os pesos em uma medida de distorção ponderada. Portanto, pode-se considerar nossa abordagem à codificação ROI como modificando a medida de distorção de acordo com a ROI e então realizando a codificação para minimizar a medida de distorção, formalmente conhecida como medida de distorção de entrada ponderada [14], [15], [16].

Nossa abordagem à codificação ROI tem a vantagem de ser independente de codec. Em vez de ajustar cada codec de uma maneira específica para ajustar sua fidelidade na ROI, defendemos o uso da ROI para modificar a medida de

distorção. A medida de distorção modificada é então disponibilizada para qualquer codec para sua otimização usual, por exemplo, otimização de distorção de taxa. Há um mapeamento simples da ROI para a medida de distorção, que pode ser quantificada (por exemplo, usando experimentos perceptivos) independentemente de qualquer codec específico. Como nosso codec, escolhemos transformar a codificação com RAHT, porque o RAHT é automaticamente otimizado para a medida de distorção em virtude de sua interpretação de medida teórica.

II. Medida de distorção ponderada por ROI e RAHT de medida teórica

Consideramos um único atributo escalar, digamos Y_i, nos pontos $\mathbf{x}_i$, $i = 1, \ldots, N$, da nuvem de pontos. O *erro quadrado ponderado* entre $Y = \{Y_i\}$ e sua reprodução $\hat{Y} = \{\hat{Y}_i\}$ é definido por

$$d(Y, \hat{Y}) = \sum_i w_i (Y_i - \hat{Y}_i)^2, \tag{1}$$

onde w_i, $i = 1, \ldots, N$, são os *pesos*. Se o peso w_i reflete a importância semântica ou perceptiva do ponto $\mathbf{x}_i$, então $d(Y, \hat{Y})$ pode ser chamado de medida de distorção *ponderada por ROI*. Um codec que minimize esta medida de distorção sujeita a uma restrição de taxa tenderá a reproduzir Y_i como $\hat{Y}_i$ com um erro quadrado inversamente proporcional a w_i. Por exemplo, suponha que $w_i = 16$ quando $\mathbf{x}_i \in R$ e $w_i = 1$ caso contrário, onde R é a região de interesse. Então o erro do valor quadrático médio (RMS) na ROI será em torno de $1/4$ do erro RMS em outro lugar. Esse é um jeito natural de especificar o objetivo da codificação ROI.

Os pesos no erro quadrado ponderado podem ser interpretados como uma medida. Uma *medida* em um espaço mensurável é uma função μ que atribui um número real a cada conjunto, de forma que a medida da união de qualquer sequência de subconjuntos disjuntos é a soma das medidas dos subconjuntos. Exemplos de medidas são a medida Lebesgue na reta real, a contagem nos inteiros e qualquer probabilidade em um espaço de probabilidades. Nós focamos em $\mathbb{R}^3$ como o espaço mensurável e definimos $\mu(S) = \sum_{i:\mathbf{x}_i \in S} w_i$ para qualquer conjunto mensurável $S \subseteq \mathbb{R}^3$.

A definição de medida induz a definição da integral, $\int f(\mathbf{x}) d\mu(\mathbf{x}) = \liminf_{\epsilon \to 0} \sum_n \mu(\{f(\mathbf{x}) \geq n\epsilon\}) = \sum_i w_i f_i$, onde $f_i = f(\mathbf{x}_i)$. Por sua vez, a definição da integral induz a definição do produto interno, $\langle f, g \rangle = \int f(\mathbf{x}) g(\mathbf{x}) d\mu(\mathbf{x}) = \sum_i w_i f_i g_i$. Por sua vez, a definição do produto interno induz a definição de ortogonalidade, $f \perp g \Leftrightarrow \langle f, g \rangle = 0$, e norma, $||f|| = (\langle f, f \rangle)^{1/2}$. No total, estes induzem um espaço de Hilbert. O erro quadrado ponderado $d(Y, \hat{Y})$ é precisamente a norma quadrada $||f - \hat{f}||^2$ desse espaço de Hilbert, onde $f_i = Y_i$ e $\hat{f}_i = \hat{Y}_i$.

A RAHT é adaptativa por região para manter-se ortonormal independentemente da localização dos pontos. Recentemente, demonstrou-se que a RAHT é interpretável como uma spline wavelet constante e separável por partes que é ortonormal em relação ao produto interno $\langle f, g \rangle$ definido pelos pesos w_i [13]. Assim, se os pesos forem definidos para os pesos na medida de distorção ponderada pela ROI, a transformada permanecerá ortonormal. Além disso, a quantização escalar uniforme dos coeficientes de transformada (passo de quantização constante) minimizará a medida de distorção ponderada pela ROI, ao menos em altas taxas.

Para ser específico, faça $\mathbb{R}^3$ ser particionado uniformemente em cubos de tamanho $2^{-m} \times 2^{-m} \times 2^{-m}$, blocos de tamanho $2^{-m} \times 2^{-m} \times 2^{-m+1}$, e $2^{-m} \times 2^{-m+1} \times 2^{-m+1}$, e sejam $\mathcal{F}_{3m}$, $\mathcal{F}_{3m+1}$, e $\mathcal{F}_{3m+2}$ os espaços de todas as funções $f_\ell : \mathbb{R}^3 \to \mathbb{R}$ que são constantes por partes nesses blocos, por $\ell = 3m$, $3m + 1$, e $3m + 2$, respectivamente. A sequência aninhada de espaços de função $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F}_\ell \subseteq \mathcal{F}_{\ell+1} \subseteq \cdots$ aproxima-se cada vez mais finamente (em relação à norma, isto é, ao erro quadrado ponderado) o espaço de funções contínuas por partes.

Tendo que $B_{\ell,n}$ denota um bloco no nível ℓ indexado por n, seja $1_{B_{\ell,n}}(\mathbf{x})$ sua função indicadora e $w_{\ell,n} = \mu(B_{\ell,n})$ sua medida. Então, $\mathcal{F}_\ell$ é abrangida pelas funções base

$$\phi_{\ell,n}(\mathbf{x}) = w_{\ell,n}^{-1/2} 1_{B_{\ell,n}(\mathbf{x})}, \tag{2}$$

que são ortogonais uma a outra e são normalizadas em relação ao produto interno e norma induzidos pela medida ponderada. Analogamente, $B_{\ell+1,n_0}$ e $B_{\ell+1,n_1}$ denotam sub-blocos de $B_{\ell,n}$, seja $\mathcal{G}_\ell$ o complemento ortogonal de $\mathcal{F}_\ell$ em $\mathcal{F}_{\ell+1}$. Então $\mathcal{G}_\ell$ é abrangida pelas funções base

$$\psi_{\ell,n}(\mathbf{x}) = \frac{-w_{\ell+1,n_0}^{-1} 1_{B_{\ell+1,n_0}}(\mathbf{x}) + w_{\ell+1,n_1}^{-1} 1_{B_{\ell+1,n_1}}(\mathbf{x})}{(w_{\ell+1,n_0}^{-1} + w_{\ell+1,n_1}^{-1})^{-1/2}} \tag{3}$$

que são ortogonais entre si e às funções em (2), e são normalizadas, como pode ser verificado pelo leitor interessado. Assim, qualquer função $f_{\ell+1} \in \mathcal{F}_{\ell+1}$ pode ser escrita como

$$f_{\ell+1}(\mathbf{x}) = \sum_n F_{\ell,n} \phi_{\ell,n}(\mathbf{x}) + \sum_n G_{\ell,n} \psi_{\ell,n}(\mathbf{x}), \tag{4}$$

onde $F_{\ell,n} = \langle f_{\ell+1}, \phi_{\ell,n} \rangle$ são conhecidos como coeficientes passa-baixa e $G_{\ell,n} = \langle f_{\ell+1}, \psi_{\ell,n} \rangle$ são conhecidos como coeficientes passa-alta. Após alguma manipulação algébrica, (2) e (3) podem ser expressadas recursivamente como as "equações de duas escalas"

$$\phi_{\ell,n}(\mathbf{x}) = a\, \phi_{\ell+1,n_0} + b\, \phi_{\ell+1,n_1} \tag{5}$$

$$\psi_{\ell,n}(\mathbf{x}) = -b\, \phi_{\ell+1,n_0} + a\, \phi_{\ell+1,n_1}, \tag{6}$$

onde $a = \frac{\sqrt{w_{\ell+1,n_0}}}{\sqrt{w_{\ell,n}}}$ e $b = \frac{\sqrt{w_{\ell+1,n_1}}}{\sqrt{w_{\ell,n}}}$. Substituindo isso na definição de $F_{\ell,n}$ e $G_{\ell,n}$, nós obtemos

$$\begin{bmatrix} F_{\ell,n} \\ G_{\ell,n} \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} F_{\ell+1,n_0} \\ F_{\ell+1,n_1} \end{bmatrix}, \tag{7}$$

que é uma rotação de Givens cujo ângulo de rotação depende dos pesos relativos dos sub-blocos.

A RAHT aplica (7) recursivamente para expandir $f_L \in \mathcal{F}_L$ como

$$f_L(\mathbf{x}) = \sum_n F_{0,n} \phi_{0,n}(\mathbf{x}) + \sum_{\ell=0}^{L-1} \sum_n G_{\ell,n} \psi_{\ell,n}(\mathbf{x}), \tag{8}$$

onde L é escolhido com largura o suficiente de modo que cada cubo $B_{L,n}$ contém ao menos um único ponto, digamos $\mathbf{x}_i$ com valor $f_i = f(\mathbf{x}_i)$. O número de coeficientes é N, isto é, a RAHT está criticamente amostrada [13]. Perceba que $\phi_{L,n}(\mathbf{x}) = w_i^{-1/2} 1_{B_{L,n}}(\mathbf{x})$, e, portanto, $F_{L,n} = \langle f, \phi_{L,n} \rangle =$

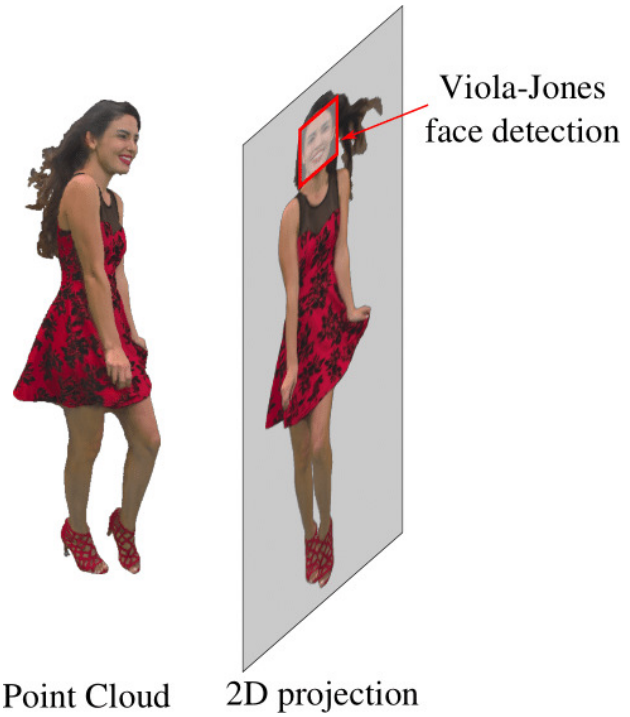

Fig. 1. Detecção de rosto utilizando projeção de nuvem de pontos em duas dimensões e o algoritmo Viola-Jones.

$w_i^{1/2} f_i$. Isso generaliza a RAHT em [11], para qual $w_i = 1$ para todos os pontos $i = 1, \ldots, N$.

Os coeficientes da RAHT são uniformemente quantizados com passos de tamanho $\Delta(F_{0,n})$ e $\Delta(G_{\ell,n})$, $\ell = 0, \ldots, L-1$, e codificados entropicamente. Como as rotações de Givens são ortonormais, a energia é preservada. Portanto, o erro quadrático de quantização é

$$\sum_n (F_{0,n} - \hat{F}_{0,n})^2 + \sum_{\ell=0}^{L-1} \sum_n (G_{\ell,n} - \hat{G}_{\ell,n})^2 = \sum_{i=1}^{N} w_i (f_i - \hat{f}_i)^2, \tag{9}$$

que é o mesmo que a medida de distorção ponderada pela ROI (1) quando $f_i = Y_i$. Uma vez que um tamanho de passo constante $\Delta = Q_{step}$ minimiza o erro quadrático de quantização sujeito a uma restrição de entropia, ao menos em altas taxas [14], definir o tamanho dos passos dos coeficientes RAHT para uma constante também minimiza a medida de distorção ponderada pela ROI desejada para a codificação ROI.

Resumindo, com RAHT, no codificador, voxels na ROI devem ter pesos iniciais definidos para $w_i = w$ e atributos escalados por $\sqrt{w}$. O decodificador deve escalar de volta os atributos.

III. Determinação da Região de Interesse e Sinalização

No nosso trabalho, a ROI é escolhida como sendo o rosto do sujeito na nuvem de pontos. Uma vez que nosso cérebro é mais sensível a artefatos introduzidos no rosto em imagens reconstruídas, acreditamos que priorizar a qualidade do rosto do sujeito durante a compressão levará a uma melhor qualidade subjetiva.

(a) *largura = 1* (b) *largura = 2* (c) *largura = 4* (d) *largura = 8*

Fig. 2. Artefatos na identificação da ROI devido a obliquidade das projeções dos lados da face.

O rosto é identificado como ilustrado na Fig. 1 e o processo é descrito no diagrama da Fig. 3 . A nuvem de pontos é girada para um determinado ângulo de visão definido por um par de ângulos de elevação e azimute e, em seguida, projetada para uma imagem 2D. Usando o algoritmo de Viola-Jones [17], o rosto é detectado e os voxels correspondentes são marcados como *face*. Esse processo é repetido para diferentes ângulos de visão. Foi escolhido variar o azimute de $0°$ até $360°$ em passos de $10°$, e variar a elevação de $-70°$ até $90°$ em passos de $10°$. Os voxels marcados como *face* em pelo menos 20% dos ângulos de visão são marcados como pertencentes à ROI.

Esse processo gera alguns buracos na ROI, uma vez que alguns voxels do rosto são ocluídos dependendo do ângulo de vista. Isso é mais visível nas bochechas, já que a maioria das projeções que o Viola-Jones é capaz de detectar um rosto são projeções frontais ou semi-frontais.

Fig. 2(a) mostra um exemplo destes buracos. Para superar esse problema, nós expandimos a ROI para seus voxels vizinhos. A nuvem de pontos é regularmente dividida em cubos de largura fixa, referidos como blocos. Se ao menos um voxel dentro de cada cubo for marcado como ROI, todos os outros voxels dentro do mesmo cubo também são marcados como ROI. Na Fig. 2, nós mostramos o resultado de expandir a ROI usando cubos de diferentes larguras. Larguras maiores resultam em menos buracos e foi decidido usar cubos de largura 8 até o fim deste projeto. Esse método de expansão da ROI foi escolhido por sua simplicidade, uma vez que pode ser facilmente implementado utilizando o código Morton [18] associado a cada voxel.

A localização da ROI precisa ser transmitida ao decodificador. Sejam M blocos ocupados. Precisamos codificar um vetor binário $b = [b_0, b_1, \ldots, b_{M-1}]$ indicando se cada bloco pertence ou não à ROI. Se ordenarmos os blocos por seus códigos de Morton, para preservar as vizinhanças, os b_i bits podem ser usados para gerar um vetor diferencial $\overline{b} = [\overline{b_0}, \overline{b_1}, \ldots, \overline{b_{M-1}}]$ onde

$$\overline{b_i} = \begin{cases} b_0 & i = 0 \\ 1 & b_{i-1} \neq b_i, i > 0 \\ 0 & b_{i-1} = b_i, i > 0 \end{cases}. \tag{10}$$

O vetor $\overline{b}$ tem uma longa sequência de zeros. Ele é codificado com um algoritmo baseado no codificador *run-length Golomb-Rice*, com a exceção que somente as *run-lengths* são codificadas com Golomb-Rice. Outros codificadores binários também podem ser usados.

IV. Resultados Experimentais

Para testar o codificador propostos utilizamos 6 nuvens de pontos: Boxer, Longdress, Loot, Redandblack, Soldier

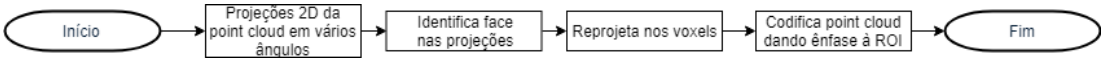

Fig. 3. Diagrama de detecção da ROI e codificação da nuvem de pontos.

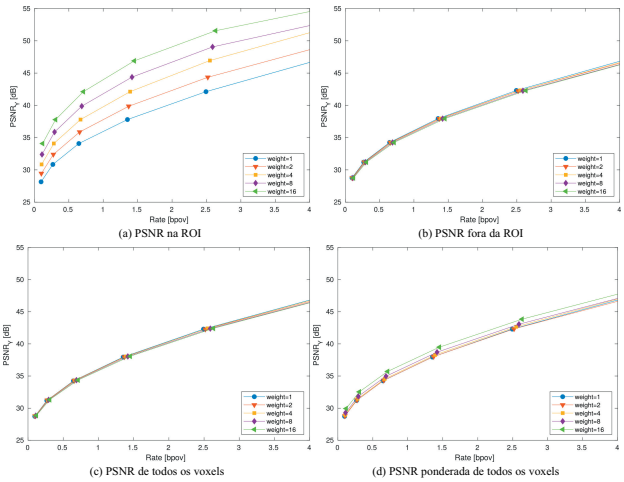

Fig. 4. Curvas de taxa-distorcão média.

e Thaidancer, todas voxelizadas com profundidade 10 (i.e. 1024x1024x1024 voxels), contendo 849452, 857966, 805285, 757691, 1089091 e 689953 voxels ocupados, respectivamente [19], [20].

Fig. 4 mostra as curvas médias de taxa-distorção calculadas para as 6 nuvens de pontos testadas. Bits para as informações secundárias estão inclusos. Pesos para ROI foram definidos para $w = 1, 2, 4, 8, 16$, enquanto voxels fora da ROI tem peso 1. Na Fig. 4 (a) a PSNR é calculada somente para os voxels na ROI. Quando $w = 1$ não existe diferença no peso para voxels dentro e fora da ROI e, neste caso particular, não há necessidade de codificar o vetor $\overline{b}$. A medida que w aumenta, a transformada favorece os voxels na ROI e é visto um aumento na PSNR para qualquer taxa, porque pesos maiores para voxels na ROI resultam em uma maior qualidade de reconstrução. O efeito é o oposto para voxels fora da ROI (Fig. 4 (b)), uma vez que existe uma transferência de bits do resto da nuvem para a ROI, para a mesma taxa de bits. A queda na PSNR geral é insignificante (Fig. 4 (c)), enquanto a PSNR ponderada (i.e. $10 \log 255^2 / WMSE$, onde $WMSE$ é o erro quadrado médio ponderado), melhora levemente com $w > 1$ (Fig. 4 (d)).

Na Fig. 5, a nuvem de pontos Redandblack foi codificada com diferentes pesos de ROI. Subjetivamente, Fig. 5 (b) parece ter uma qualidade melhor já que nosso cérebro é mais sensível a artefatos no rosto do que no resto da cena. Fig. 6 destaca a face para as nuvens de pontos reconstruídas mostradas na Fig. 5.

V. Conclusão

Introduzimos a codificação ROI para nuvens de pontos, tomando uma nova abordagem para codificação ROI ao modificar a medida de distorção para uma medida de distorção ponderada. Os pesos da medida de distorção ponderada são refletidos na medida sob a transformada RAHT. Para detectar a ROI 3D, nós combinamos algoritmos 2D como o bem conhecido algoritmo Viola-Jones. Resultados experimentais revelam, objetiva e subjetivamente, melhorias significativas (7-8 dB PSNR) na ROI com degradações subjetivamente insignificantes (abaixo de 1 dB PSNR) fora da ROI, sem nenhuma mudança na complexidade do codificador ou do decodificador.

Trabalhos futuros incluem otimizar os pesos da ROI utilizando estudos perceptivos, estendendo a abordagem para pesos multi-nível, otimizando as informações secundárias e aplicar codificação ROI para geometria de nuvens de pontos.

(a) peso = 1
Qstep = 128
9810 bytes

(b) peso = 16
Qstep = 154
9866 bytes

Fig. 5. nuvem de pontos Redandblack ($N_{vox} = 757691$) codificada com diferentes pesos para voxels na ROI. O Q_{step} foi ajustado para resultar em tamanhos de arquivo semelhantes.

(a) peso = 1

(b) peso = 16

Fig. 6. Destaque do rosto das nuvens de pontos mostradas na Fig. 5.

Referências

[1] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuca, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. Tourapis, e V. Zakharchenko, "Emerging MPEG standards for point cloud compression," *IEEE J. Emerging Topics in Circuits and Systems*, aceito para publicação.

[2] H. Hadizadeh e I. V. Bajic, "Saliency-aware video compression," *IEEE Trans. Image Process.*, vol 23, Jan. 2014.

[3] C. Zhang, D. Florêncio, e C. Loop, "Point cloud attribute compression with graph transform," em *2014 IEEE Int'l Conf. Image Processing (ICIP)*, Oct 2014.

[4] D. Thanou, P. A. Chou, e P. Frossard, "Graph-based motion estimation and compensation for dynamic 3d point cloud compression," em *IEEE Int'l Conf. Image Processing (ICIP)*, Sept 2015.

[5] ——, "Graph-based compression of dynamic 3d point cloud sequences," *IEEE Trans. Image Processing*, vol. 25, no. 4, April 2016.

[6] E.Pavezand e P.A.Chou, "Dynamic polygon cloud compression," em *IEEE Int'l Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Março 2017.

[7] E. Pavez, P. A. Chou, R. L. de Queiroz, e A. Ortega, "Dynamic polygon clouds: representation and compression for VR/AR," *APSIPA Transactions on Signal and Information Processing*, vol. 7, e 15, Nov. 2018.

[8] R.A.Cohen,D.Tian, e A.Vetro,"Attributecompression for sparse point clouds using graph transforms," em *IEEE Int'l Conf. Image Processing (ICIP)*, Sept 2016.

[9] P. A. Chou e R. L. de Queiroz, "Gaussian process transforms," em *IEEE Int'l Conf. Image Processing (ICIP)*, Sept 2016.

[10] R. L. de Queiroz e P. A. Chou, "Transform coding for point clouds using a Gaussian process model," *IEEE Trans. Image Processing*, vol. 26, no. 8, Aug. 2017.

[11] ——, "Compression of 3D point clouds using a region adaptive hierarchical transform," *IEEE Trans. Image Process.*, vol. 25, no. 8, Aug. 2016.

[12] G. Sandri, R. L. de Queiroz, e P. A. Chou, "Comments on "Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform"," *ArXiv e-prints*, Maio 2018. [Online]. Disponível: https://arxiv.org/abs/1805.09146v1

[13] M. Krivokuca, P. A. Chou, e M. Koroteev, "A volumetric approach to point cloud compression," *ArXiv e-prints*, Sep. 2018. [Online]. Disponível: https://arxiv.org/abs/1810.00484

[14] R. M. Gray e D. Neuhoff, "Quantization," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, Oct. 1998.

[15] T. Linder e R. Zamir, "High-resolution source coding for non-difference distortion measures: The ratedistortion function," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, Mar. 1999.

[16] J. Li, N. Chaddha, and R. M. Gray, "Asymptotic performance of vector quantizers with a perceptual distortion measure," *IEEE Trans. Inf. Theory*, vol. 45, no. 4, Maio 1999.

[17] P. Viola e M. Jones, "Rapid object detection using a boosted cascade of simple features," em *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, Dec 2001, pp. I–I.

[18] G. M. Morton, "A computer oriented geodetic data base and a new technique in file sequencing,"IBM Ltd., Ottawa 4, Ontario, Canada, Tech. Rep., Mar. 1966.

[19] E. d'Eon, B. Harrison, T. Myers, e P. A. Chou, "8i voxelized full bodies — a voxelized point cloud dataset," ISO/IEC JTC1/SC29/WG1 & WG11 JPEG & MPEG,inputdocumentsM74006&m40059,Jan.2017.

[20] M. Krivokuca, P. A. Chou, e P. Savill, "8i voxelized surface light field (8iVSLF) dataset," ISO/IEC JTC1/SC29/WG11 MPEG, input document m42914, Jul. 2018.