

Rate Distortion Optimized Graph Partitioning for Omnidirectional Image Coding

Mira Rizkallah, Francesca de Simone, Thomas Maugey, Christine Guillemot, Pascal Frossard

▶ **To cite this version:**

Mira Rizkallah, Francesca de Simone, Thomas Maugey, Christine Guillemot, Pascal Frossard. Rate Distortion Optimized Graph Partitioning for Omnidirectional Image Coding. EUSIPCO 2018 - 26th European Signal Processing Conference, Sep 2018, Rome, Italy. pp.1-5. hal-01807613

HAL Id: hal-01807613
https://inria.hal.science/hal-01807613

Submitted on 4 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Rate Distortion Optimized Graph Partitioning for Omnidirectional Image Coding

Mira Rizkallah*, Francesca De Simone‡, Thomas Maugey†, Christine Guillemot† and Pascal Frossard‡
*IRISA, Université de Rennes 1, Rennes, France
†INRIA, Rennes Bretagne Atlantique, France
‡Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Suisse

***Abstract*—Omnidirectional images are spherical signals captured by cameras with 360-degree field of view. In order to be compressed using existing encoders, these signals are mapped to planar domain. A commonly used planar representation is the equirectangular one, which corresponds to a non uniform sampling pattern on the spherical surface. This particularity is not explored in traditional image compression schemes, which treat the input signal as a classical perspective image. In this work, we build a graph-based coder adapted to the spherical surface. We build a graph directly on the sphere. Then, to have computationally feasible graph transforms, we propose a rate-distortion optimized graph partitioning algorithm to achieve an effective trade-off between the distortion of the reconstructed signals, the smoothness of the signal on each subgraph, and the cost of coding the graph partitioning description. Experimental results demonstrate that our method outperforms JPEG coding of planar equirectangular images.**



(a) Pool (b) Farm (c) Metro (d) Hotel

Fig. 1: Omnidirectional images (in equirectangular format) used in our experiments: two outdoor images (a and b) and two indoor images (c and d) [6]

I. Introduction

Nowadays, omnidirectional images are widely used for popular applications such as virtual reality and immersive communications. Omnidirectional images are spherical signals captured by cameras with 360-degree field of view. In order to use existing image and video processing algorithms, these signals are usually mapped to planar domain and stored as rectangular lattices. A commonly used planar representation for omnidirectional content is the so-called *equirectangular* representation [1] (Figure 1). This representation presents strong warping distortions around the polar areas and corresponds to an *equi-angular* sample distribution on the spherical surface, which is non-uniform (Figure 2). An equirectangular image can be fed as input to existing state of the art encoders, but the equirectangular signal statistics differ from those of classical perspective images. Thus, using existing compression algorithms is sub-optimal [2].

In this paper, we propose a graph-based representation for omnidirectional images, which takes into account the spherical geometry and provides a flexible way to efficiently store and compress the visual data. Specifically, we propose to represent an omnidirectional image by a graph, where the graph vertices correspond to the image pixels defined on the spherical surface. The edge weights capture the sampling grid on which the signal is defined. Such a flexible representation permits to go beyond traditional transform coding by moving from classical fixed transforms such as the Discrete Cosine Transform (DCT) to graph-based transforms that are adapted to the actual signal support, such as the Graph Fourier Transform (GFT) [3] [4].

Due to the high spatial resolution typical of omnidirectional images [1] , the graph that we propose to build would have a huge number of vertices (>500K). Consequently, the GFT computation in an actual coding pipeline would be unfeasible. Indeed, one would think about using sampling techniques on the sphere. However such coding schemes involve interpolation which makes the distortion control very complex. Instead, we propose an efficient graph partitioning strategy, which takes into account the geometrical information in order to optimize the smoothness of the signals on the subgraphs while keeping a small overhead to code the description of the partition. Finally, we propose a complete GFT-based lossy compression scheme using this partitioning and compare its performance to the classical DCT-based JPEG coding [5]. Experimental results show that the partitioning provides an effective tradeoff between the smoothness of signals on the subgraphs and the cost of coding the partition. Moreover, the proposed coding scheme outperforms JPEG coding of planar equirectangular images, in terms of Rate-Distortion (RD) analysis using multiple quality metrics.

II. 360-degree image as signal on a Graph

A 360-degree image I can be represented by a signal $x \in \mathbb{R}^N$ defined on an undirected, 4-connected, weighted *global graph* $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ (See Figure 2), which consists of a finite set of vertices $\mathcal{V}$ defined on the image surface, with $|\mathcal{V}| = N$, a set of edges $\mathcal{E}$, and a weighted adjacency matrix $\mathbf{W}$. For $i = 1, ..., N$, the signal value x_i corresponds to the pixel color value at vertex $i \in \mathcal{V}$. If there is an edge $e = (i, j)$

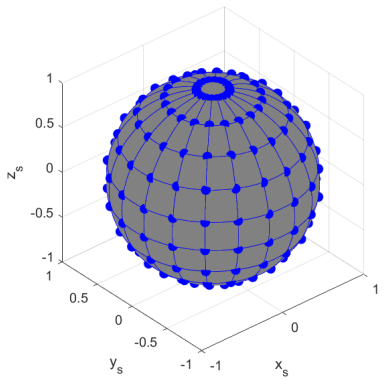


Fig. 2: Equi-angular (i.e., non uniform) sampling on the sphere corresponding to the planar equirectangular representation. A graph is drawn on the spherical surface with vertices corresponding to pixels and edges connecting each pixel with its four closest neighbors (three closest neighbors at the poles).

connecting vertices i and j, the entry $W_{i,j}$ represents the weight of the edge, otherwise, $W_{i,j} = 0$. We define the weight of an edge connecting adjacent vertices i and j via a Gaussian kernel weighting function:

$$W_{i,j} = exp\left(-\frac{d_{geo}(i,j)^2}{2\theta_{geo}^2}\right) \tag{1}$$

for some parameters θ_{geo} , where $d_{geo}(i,j)$ represents the geodesic distance between vertices i and j capturing the sampling grid on which the vertices are defined (Figure 2). One known operator defined on the graph is the unnormalized graph Laplacian defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ where $\mathbf{D}$ is a diagonal matrix whose i-th diagonal element $D_{i,i}$ is equal to the sum of the weights of all the edges incident to vertex i. $\mathbf{L}$ is a real symmetric matrix, thus, it has a complete set of orthonormal eigenvectors $\mathbf{u}_l = \{u_l\}$ associated to real nonnegative eigenvalues $\lambda_l = \{\lambda_l\}$, with $l = 0, 1, ..., N-1$, satisfying:

$$\mathbf{L}\mathbf{u}_l = \lambda_l \mathbf{u}_l. \tag{2}$$

We assume that the graph Laplacian eigenvalues are ordered, i.e., $0 = \lambda_0 \leq \lambda_1 ... \leq \lambda_{N-1}$. The graph spectrum is defined as $\sigma(\mathbf{L}) = \{\lambda_0, \lambda_1, ...\lambda_{N-1}\}$. Once we have a graph and a signal defined on its vertices, the eigenvectors of the Laplacian are used to define the graph Fourier transform (GFT) [4] of the signal x. Particularly, the GFT $\hat{x}$ of any function $x : \mathcal{V} \rightarrow \mathbb{R}$ defined on the vertices of $\mathcal{G}$ is the expansion of x in terms of the eigenvectors of the graph Laplacian:

$$\hat{x}(\lambda_l) = \langle x, \mathbf{u}_l \rangle = \sum_{i=1}^{N} x(i) u_l^*(i). \tag{3}$$

The inverse graph Fourier transform is then given by:

$$x(i) = \sum_{l=0}^{N-1} \hat{x}(\lambda_l) u_l(i). \tag{4}$$

The graph Laplacian eigenvalues and eigenvectors carry a notion of frequency. The eigenvector u_0 associated with the eigenvalue λ_0 is constant and equal to $1/\sqrt{N}$ at each vertex. The eigenvectors associated with small eigenvalues vary slowly across the graph, i.e., if two vertices are connected by an edge with a large weight, the values of the eigenvector at those locations are likely to be similar. The eigenvectors associated with larger eigenvalues oscillate more rapidly and are more likely to have dissimilar values on vertices connected by an edge with high weight.

A signal x is considered to be smooth on $\mathcal{G}$ if strongly connected vertices have similar signal values [7]. This is usually quantified in terms of the laplacian quadratic form:

$$S(x) = x^T \mathbf{L} x. \tag{5}$$

In general, graph-based image compression methods use a graph representation as defined above, and perform a GFT to capture the main characteristics of the signal. The coefficients are then encoded instead of original values. The smoother the signal on a graph (smaller $S(x)$), the more its energy is concentrated in the low frequency GFT coefficients and the more it is easily compressible.

III. Problem formulation

In our case, the common computational limitation of the *global graph* representation is the maximum acceptable number of vertices in the graph for GFT computation, which limits the resolution of the visual signal that can be supported.

In order to cope with the feasibility of the graph-based transform of the signal in high resolution omnidirectional images, the *global graph* $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ should be separated into several connected components, e.g. M subgraphs $\{\mathcal{G}_1, ..., \mathcal{G}_i, ..., \mathcal{G}_M\}$ by pruning some unreliable edges. The i-th subgraph is $\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i, \mathbf{W}_i\}$ where $\mathcal{V}_i$ are the vertices in the subgraph, with $|\mathcal{V}_i| = N_i < N$, $\mathcal{E}_i$ are their edges, and $\mathbf{W}_i$ is the weights matrix. x_i is the signal defined on the i-th subgraph. The signals on each of the subgraphs are then independently processed, and transformed separately using their respective local Laplacian $\mathbf{L}_i$.

If the topology and weights of the *global graph* are fixed, in order to obtain a good compression performance, the graph partition should be chosen such that it leads to smooth representations of the signals inside different subgraphs. On the other hand, it should also be easy to encode, since it has to be transmitted to the decoder for signal reconstruction. Our problem is therefore how to split the fixed *global graph* into connected components, so that we achieve optimal RD performance and such that all the connected components contain less than N_{max} nodes..

We first pose the problem as a rate-distortion optimization problem defined as:

$$\begin{aligned} \min_{\tilde{\mathcal{G}}=\{\mathcal{G}_i\}} \quad & \mathcal{D}(\tilde{\mathcal{G}}) + \gamma \mathcal{R}_C(\tilde{\mathcal{G}}) + \beta \mathcal{R}_B(\tilde{\mathcal{G}}) \\ \text{subject to} \quad & N_i < N_{max}, \ \forall i \end{aligned} \tag{6}$$

$\tilde{\mathcal{G}} = \{\mathcal{G}_i\}$ is the *global graph* based on the geometry defined a priory using Equation (1) , where some edges are removed. $\mathcal{D}(\tilde{\mathcal{G}})$ is the distortion between the original image and the reconstructed one, $\mathcal{R}_C(\tilde{\mathcal{G}})$ is the rate cost of the transform coefficients, and $\mathcal{R}_B(\tilde{\mathcal{G}})$ is the rate cost of the

boundaries for the graph partitioning description. Each of these terms possibly depend on the chosen partition of the graph and of the coding scheme envisioned. We detail each one of them in the next section.

IV. R-D optimized Graph Partitioning

A. Distortion Estimation

Since the GFT is orthonormal and independant in each subgraph, the distortion term in the above problem $\mathcal{D}(\tilde{\mathcal{G}})$ is equal to the sum of distortions on all subgraphs:

$$\mathcal{D}(\tilde{\mathcal{G}}) = \sum_{i=1}^{M} \mathcal{D}(\mathcal{G}_i) = \sum_{i=1}^{M} \|x_i - \tilde{x}_i\|^2 = \sum_{i=1}^{M} \|\hat{x}_i - \hat{x}_{iq}\|^2,$$

where x_i and $\tilde{x}_i$ are the original signal and decoded signal in the i^{th} subgraph respectively. $\hat{x}_i$ and $\hat{x}_{iq}$ are the original and quantized signal GFT coefficients in the i^{th} subgraph.
If we consider a uniform scalar quantizer with small quantization step q for all N coefficients, $\mathcal{D}(\tilde{\mathcal{G}})$ can be approximated by:

$$\mathcal{D}(\tilde{\mathcal{G}}) = q^2 \frac{N}{12} \tag{7}$$

and is thus independent from $\tilde{\mathcal{G}}$. Therefore, the optimization problem (6) is reduced to minimizing the rate terms.

B. Rate approximation of transform coefficients

We can evaluate the rate of the GFT coefficients $\mathcal{R}_C(\mathcal{G}_i)$ in a subgraph i using the approximation in [8]:

$$\mathcal{R}_C(\mathcal{G}_i) = S(x_i) = x_i^T \mathbf{L}_i x_i = \sum_l \lambda_l \hat{x}_{i,l}^2. \tag{8}$$

The parameters λ_l and $x_{i,l}$ are the eigenvalues of the local Laplacian, and the corresponding GFT coefficients of the signal x_i respectively. Hence, It is an eigenvalue-weighted sum of squared transform coefficients which depends on the underlying local graph $\mathbf{L}_i$.
Such proxy assumes that the bitrate of the transform coefficients increases when the smoothness of a signal on the graph decreases. While the bitrate needed to code the DC component is not captured by this approximation, we assume that it is only dependent on the number of subgraphs, which can be tuned using the N_{max} constraint in our optimization problem. The higher the N_{max}, the lower the bitrate needed to code the DC coefficients.

C. Rate approximation of the subgraphs boundaries

In fact, in our problem we impose that the pixels of the same subgraph form a connected component. Thus, a common way to code the subgraph membership is to code the boundaries. In order to approximate the coding rate of a boundary B_{ij} between two adjacent subgraphs $\mathcal{G}_i$ and $\mathcal{G}_j$, we use the 4-directional differential freeman chaincodes (*DCC*) [9] and estimate the coding rate of the boundary as its entropy computed as follows:

$$\mathcal{C}_B(ij) = -\#_l \sum_{k=1}^{4} p_k \log_2 p_k, \tag{9}$$

where $\#_l$ is the number of chaincodes of the boundary and $p_k, k = 1:4$ are the probabilities of each of the 4 directions.

D. Minimization of the total coding rate

Using (7) (8) and (9), the optimization problem in (6) becomes:

$$\begin{aligned} &\min_{\tilde{\mathcal{G}}=\{\mathcal{G}_i\}} \sum_{i=1}^{M} x_i^T \mathbf{L}_i x_i + \alpha \frac{1}{2} \sum_{i=1}^{M} \sum_{j \in \mathcal{N}_i} \mathcal{C}_B(ij), \\ &\text{s.t. } N(\mathcal{G}_i) < N_{max},\ \forall i \end{aligned} \tag{10}$$

where $\mathcal{N}_i$ is the neighborhood of the subgraph $\mathcal{G}_i$. The second term is divided by 2 since we only have to code the boundary between any two neighboring regions once.
Finding this optimal partition is in general a combinatorial task, so we solve it using traditional agglomerative approximation. To initialize our optimization process, we use the Normalized Cut [10] which is well known for favoring the highest smoothness inside partitions. For that, we build a new graph $\mathcal{G}_{\mathrm{NC}}$ with the same connectivity as $\mathcal{G}$ but with weights taking into account both the geodesic distance on the sphere and the euclidean distance in the Y space as:

$$w(i,j) = \exp\left(-\frac{d_{geo}(i,j)^2}{2\theta_{geo}^2}\right) \exp\left(-\frac{d_Y(i,j)^2}{2\theta_Y^2}\right). \tag{11}$$

Note that this graph $\mathcal{G}_{\mathrm{NC}}$ is only used in the Normalized Cut algorithm, and do not serve as a support for GFT thus will not be transmitted. To limit the computation time, the segmentation is performed with recursive 2-way cut algorithm: at each iteration, only the first 2 eigenvectors are computed exploiting the sparse nature of the laplacians. At the output of the initialization, we have an over-segmentation with non-overlapping subgraphs $R = \{\mathcal{G}_1, ..., \mathcal{G}_i, ..., \mathcal{G}_K\}$. To model their spatial locality, we construct a subgraph neighborhood matrix $\mathbf{E}$ where $\mathbf{E}(i,j) = 1$ indicates that the subgraphs $\mathcal{G}_i$ and $\mathcal{G}_j$ are adjacent in the image. In fact, merging any two adjacent subgraphs $\mathcal{G}_i$ and $\mathcal{G}_j$ implies re-considering the connections between adjacent pixels on the boundary between them (from the *global graph*), hence removing the boundary itself. At each iteration of the merging process, we find the two adjacent subgraphs $\mathcal{G}_i^*$ and $\mathcal{G}_j^*$, which if merged, bring the most significant decrease of the criterion in (10) while not exceeding N_{max} nodes in the merged region. In other words,

$$\begin{aligned} \{\mathcal{G}_i^*, \mathcal{G}_j^*\} = &\max_{\mathcal{G}_i, \mathcal{G}_j \in R} \Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j) + \lambda \Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j), \\ &\text{s.t. } \mathbf{E}(i,j) = 1,\ N(\mathcal{G}_i \cup \mathcal{G}_j) < N_{max} \end{aligned} \tag{12}$$

where

$$\begin{aligned} \Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j) &= x_i^T \mathbf{L}_i x_i + x_j^T \mathbf{L}_j x^j \\ &\quad - \begin{bmatrix} x_i^T \\ x_j^T \end{bmatrix} \begin{bmatrix} \mathbf{L}_i + \mathbf{D}_{ij} & \mathbf{W}_{ij} \\ \mathbf{W}_{ji} & \mathbf{L}_j + \mathbf{D}_{ji} \end{bmatrix} \begin{bmatrix} x_i & x_j \end{bmatrix} \\ \Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j) &= \mathcal{C}_B(ij). \end{aligned}$$

$\Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j)$ and $\Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j)$ essentially capture the difference in the rate needed to code the coefficients and the rate to code the boundaries between the two regions before and after merging, respectively. If $\Delta\mathcal{R}_C(\mathcal{G}_i^*, \mathcal{G}_j^*) + \lambda\Delta\mathcal{R}_G(\mathcal{G}_j^*, \mathcal{G}_j^*) > 0$, we merge $\mathcal{G}_i^*$ and $\mathcal{G}_j^*$ into one subgraph, and repeat the process until the total rate cannot be further reduced.
In the previous formulation and subgraph merging process, we

assume that all subgraphs are having the same contribution to the global rate of the whole scheme. However, in our omnidirectional image application, we are interested in giving ideally more rate to the most useful part of the signal, allowing more rate to the subgraphs occupying the biggest surface in the sphere, favoring the merging on smaller surfaces in the spherical domain. Hence, we modify the initial RD gain of Equation (12) adding a normalizing factor equal to the area occupied by the merged region on the sphere:

$$\{\mathcal{G}_i^*, \mathcal{G}_j^*\} = \max_{\mathcal{G}_i, \mathcal{G}_j \in R} \frac{\Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j) + \lambda\Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j)}{A_{ij}}, \quad (13)$$

where A_{ij} is the area on the sphere that the merged region occupies. Such normalization gives more priority to merging in the poles than in the equator. The final algorithm of the partitioning is detailed in Algorithm 1.

Algorithm 1: Rate-distortion optimized Graph partitioning for omnidirectional image coding

Data: NcutLabels, Maximum number of Nodes in a Subgraph: $\mathcal{G}_{Ncut}$, N_{max}
Result: Labels after merging $\mathcal{G}_{final}$
Initialization: $\mathcal{G} = \{\mathcal{G}_i\} = \mathcal{G}_{Ncut}$;
Construct the region neighborhood matrix $\mathbf{E}$;
Compute $\Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j) + \lambda\Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j)$ for all i, j where $\mathbf{E}(i,j) = 1$ and $N(\mathcal{G}_i \cup \mathcal{G}_j) < N_{max}$;
Repeat
  Find $\{\mathcal{G}_i^*, \mathcal{G}_j^*\}$, Eq. (13)
  $\mathcal{G} = \mathcal{G} \setminus \{\mathcal{G}_i^*, \mathcal{G}_j^*\} \cup \{\mathcal{G}_i^* \cup \mathcal{G}_j^*\}$;
  Update $\mathbf{E}$ based on the newly merged region
Until $\max\left(\Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j) + \lambda\Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j)\right) < 0$;
$\mathcal{G}_{final} = \mathcal{G}$;

V. Experimental validation

We now move to describe how we use the above graph partitioning algorithm in an omnidirectional image compression scheme. As pointed out in the previous sections, once we solve our optimization problem, we have two kind of information to transmit to the decoder side: the GFT coefficients of the signals in all subgraphs and the description of the partitioning. The transform coefficients are quantized using a uniform quantizer with a fixed step size q for all the bands, then coded with a simple entropy coder.
In order to code the partition map, we use the arithmetic edge coder (*AEC*) proposed in [11]. The contours are first represented by differential chaincode (*DCC*) [9] and divided into segments. Then, to efficiently encode a sequence of symbols in a segment, *AEC* uses a linear regression model to estimate probabilities, which will be subsequently used in the arithmetic coder.

A. Validation of our rate proxy

Although we do not explore the true rate needed to code boundaries using *AEC* in our graph partitioning, we can show the accuracy of our proxy. During the optimization process, we compare the rate needed to code a boundary using *AEC* to our rate proxy using the entropy. Results are shown in Figure 3 in which the x-axis corresponds to the rate needed using *AEC*, and the y-axis corresponds to the our proposed rate proxy of $\mathcal{R}_B$. Although our rate proxy of the subgraphs' boundaries has a very small computation time with respect to AEC, the positive linear trend observed in the plot shows that it is a good approximation.

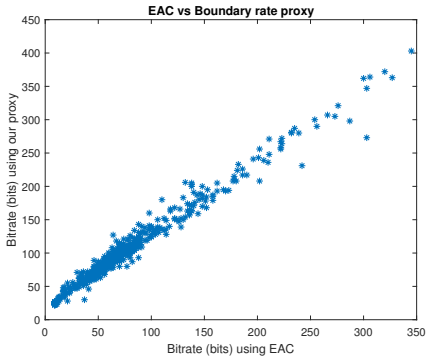


Fig. 3: Accuracy of the our rate proxy. x-axis: rate needed to code a boundary using *EAC*. y-axis: rate computed using our proxy



(a)



(b)

Fig. 4: Graph partitions represented in the equi-angular domain (a) and in the spherical domain (b)

B. Coding results

We test our method on four grayscale omnidirectional images, namely *Metro*, *Pool*, *Farm* and *Hotel* shown in Figure 1. Each omnidirectional image is of size (512×1024).
We test two versions of our scheme that we call *WithoutGeometry* and *WithGeometry*. In the first version, the geodesic distance is not taken into account which comes down to set $\theta_{geo} = \infty$ in the construction of graphs $\mathcal{G}$ and $\mathcal{G}_{\mathrm{NC}}$. Moreover, the merging is done as explained in Equation (12). On the other hand, the second version corresponds to our detailed scheme of the previous section taking into account the geometrical information in all stages: normalized cut, merging with equation (13) and transform coding. To evaluate the compression performance, we compute the PSNR in two different domains: the equirectangular domain between

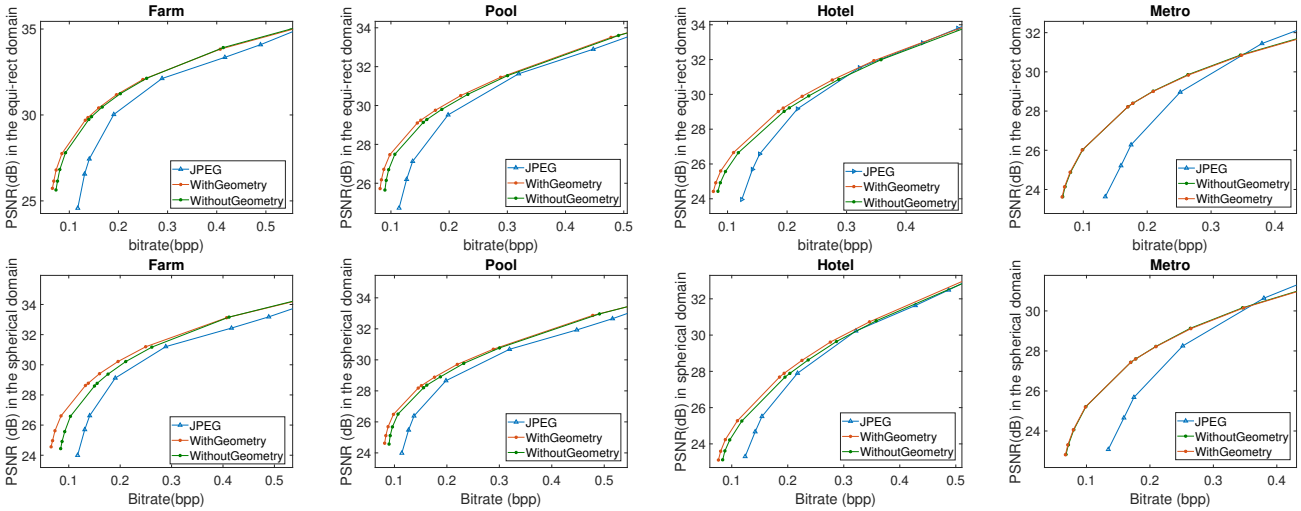


Fig. 5: Rate-distortion comparison. 1^s row: performance in the equirectangular domain. 2^{nd} row : performance in the Spherical domain

original and decoded omnidirectional images, and in the spherical domain after performing a uniform sampling of the spherical surface [12].

Figure 4 shows an example of subgraphs obtained using *WithGeometry* scheme after fixing λ to 600 for the *Pool* image, in the equirectangular and the spherical domains. It is clear that subgraphs are adhering to the objects borders in both domains and larger subgraphs are formed around the poles.

Results in Figure 5 show that *WithGeometry* leads to a better rate-distortion performance in both domains, compared to *WithoutGeometry*. There is two major explanations of this behavior. First, the global graph in the first case is more adapted to the omnidirectional signal: more specifically, the signal values in horizontally adjacent pixels around the poles are assumed to be more correlated than those which are horizontally adjacent in the equator. This is the case of most of omnidirectional images where poles usually consist of the floor or the sky. In practice, some images like *Metro* do not totally follow this assumption which explains the comparable performance observed for the two schemes. A second explanation is that in the *WithGeometry* case, the total rate is allocated more carefully taking into account the area occupied on the sphere. Furthermore, the obtained results show that our proposed schemes outperform classical DCT transform coding scheme in JPEG especially in the low bitrate range, although they can be further improved by optimizing the coding step namely in the quantization and arithmetic coding parts.

VI. Conclusion

In this paper, we have proposed a new graph-based framework for omnidirectional image compression. We introduced a new R-D optimized graph partitioning to cope with the feasibility of graph Fourier Transform on global graphs defined on high resolution images. The partition obtained provides an effective tradeoff between the smoothness of signals inside subgraphs and the cost of coding the partition description. Also, we showed that our methods outperform traditional DCT coding schemes at low bitrates. As future work, we investigate the use of different forms of laplacians and focus on the adpatation of the quantization and other coding tools, which may lead to further improvements to the coding performance. Comparison with traditional coding of planar representations other than the equirectangular one [1], as well as analysis of RD performance on higher resolution test material, will also be performed.

References

[1] Z. Chen, Y. Li, and Y. Zhang, "Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation," *Signal Processing*, vol. 146, pp. 66 – 78, 2018.

[2] F. D. Simone, P. Frossard, P. Wilkins, N. Birkbeck, and A. Kokaram, "Geometry-driven quantization for omnidirectional image coding," in *2016 Picture Coding Symposium (PCS)*, Dec 2016, pp. 1–5.

[3] G. Fracastoro, D. Thanou, and P. Frossard, "Graph transform learning for image compression," in *2016 Picture Coding Symposium, PCS 2016, Nuremberg, Germany, December 4-7, 2016*, 2016, pp. 1–5.

[4] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[5] *Digital compression and coding of continuoustone still images*, ISO/IEC IS 10918-1 ITU-T Std. Recommendation T.81.

[6] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba, "Recognizing scene viewpoint using panoramic place representation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2695–2702.

[7] D. Zhou and B. Scholkopf, "A regularization framework for learning from graph data," in *ICML workshop on statistical relational learning and Its connections to other fields*, 2004.

[8] W. Hu, G. Cheung, A. Ortega, and O. C. Au, "Multiresolution graph fourier transform for compression of piecewise smooth images," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 419–433, Jan 2015.

[9] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, vol. EC-10, no. 2, pp. 260–268, June 1961.

[10] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.

[11] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression," in *2012 19th IEEE International Conference on Image Processing*, Sept 2012, pp. 1541–1544.

[12] M. Yu, H. Lakshman, and B. Girod, "Content adaptive representations of omnidirectional videos for cinematic virtual reality," in *Proceedings of the 3rd International Workshop on Immersive Media Experiences*, ser. ImmersiveME '15. New York, NY, USA: ACM, 2015, pp. 1–6.