GRAPH-BASED DETECTION OF SEAMS IN 360-DEGREE IMAGES

Francesca De Simone*
Distributed & Interactive Systems
Centrum Wiskunde & Informatica
Amsterdam, the Netherlands

Roberto G. de A. Azevedo†, Sohyeong Kim,
Pascal Frossard
LTS4
Ecole Polytechnique Federale de Lausanne
Lausanne, Switzerland

ABSTRACT

In this paper, we propose an algorithm to detect a specific kind of distortions, referred to as seams, which commonly occur when a 360-degree image is represented in planar domain by projecting the sphere to a polyhedron, e.g., via the Cube Map (CM) projection, and undergoes lossy compression. The proposed algorithm exploits a graph-based representation to account for the actual sampling density of the 360-degree signal in the native spherical domain. The CM image is considered as a signal lying on a graph defined on the spherical surface. The spectra of the processed and the original signals, computed by applying the Graph Fourier Transform, are compared to detect the seams. To test our method a dataset of compressed CM 360-degree images, annotated by experts, has been created. The performance of the proposed algorithm is compared to those achieved by baseline metrics, as well as to the same approach based on spectral comparison but ignoring the spherical nature of the signal. The experimental results show that the proposed method has the best performance and can successfully detect up to approximately 90% of visible seams on our dataset.

Index Terms— Omnidirectional image, cube map projection, compression, visual distortion, quality metric.

1. INTRODUCTION

Omnidirectional, i.e., 360-degree, images and videos are spherical visual signals, captured by cameras with a 360-degree field of view. They are commonly mapped to a 2D plane (the planar domain) and stored as rectangular lattices in order to use existing image and video processing technologies, such as lossy compression algorithms. During runtime, the signal in the planar domain is mapped back to its native spherical domain and, at each instant, the portion of the sphere in the field of view of the user, i.e., the viewport, is rendered and presented to the end user. When consumed via a Head Mounted Display (HMD), the viewports (one per eye) are seamlessly updated following the user’s head movements to provide an increased sense of presence.

A commonly used planar representation for omnidirectional content compression is what we refer to in this paper as baseball arrangement [1]. The baseball arrangement is the result of the Cube Map (CM) projection [1]—i.e., the projection of the visual samples from the spherical surface to the faces of a cube enclosing the sphere—followed by an arrangement of the cube faces in a rectangular shape, as depicted in Fig. 1. Compared to other planar representations, such as the equirectangular (ER) one [1], CM is more compact—it needs 25% fewer pixels to represent the same uncompressed equatorial sampling of ER—and corresponds to a more uniform density of samples in the native spherical domain (Fig. 2).

The arrangement of cube faces in a rectangular lattice is inevitably characterised by visual discontinuities, due to the mosaicing of non-neighbouring faces. When applying classical block-based lossy encoding on this rectangular image, the discontinuities might be treated as high frequencies and modified by the quantization process of the transform coefficients.

*Both authors contributed equally to this work.
†This research has been partially supported by YouTube.
or by de-blocking filters used in modern video compression methods\(^2\) [2]. As a result, when the content is mapped back to the spherical domain and rendered to be visualized by the end-user, visual distortions might appear in the viewports, along the borders of adjacent cube faces (Fig. 3). We refer to these distortions as *seams*\(^3\). The seams are highly localized on the sphere in correspondence with the borders of cube faces that belong to a discontinuity in the baseball arrangement.

In this paper, we propose a Full-reference (FR) [3] detection algorithm to infer the visibility of seams. Our algorithm analyses both the original and compressed 360-degree visual signals by exploiting a graph-based representation that accounts for both the native geometry and distribution of samples on the sphere [4]. To test our method, a dataset of compressed CM 360-degree images, annotated by experts, has been created. The performance of the proposed algorithm is compared to that achieved by (1) classical FR metrics applied to portions of the planar signal consisting of adjacent cube faces and (2) an approach similar to the proposed one but without making use of the graph representation, i.e., ignoring the spherical nature of the 360-degree signal. The experimental results show that our method has the best detection performance and that the graph-based representation offers advantages over the planar one.

Although approaches to quantify the overall visual quality of 360-degree images and videos have been proposed in literature [5–11], they consider uniformly distributed distortions due to the processing of content in ER domain, rather than localized distortions that might occur when processing the signal in other domains, such as the CM one [2].

To the best of our knowledge, our algorithm is the first method focusing on the detection of localized coding distortions affecting 360-degree content processed in CM domain. It can be compared to existing metrics for classical 2D images aiming at the detection of specific compression artifacts, e.g., blockiness, blur, and ringing [12], to be used within the lossy encoding loop. Also, it can be used as a first step towards the design of a complete quality metric, accounting for multiple distortions, for compressed 360-degree content.

The rest of the paper is organized as follows. Section 2 details the proposed algorithm. Section 3 describes the dataset and ground truth creation, and the performance evaluation. Section 4 discusses the conclusions and future work.

### 2. SEAM DETECTION ALGORITHM

As input, our algorithm receives the reference (\(I_r\)) and the processed (\(I_p\)) 360-degree images, both in CM baseball arrangement. \(I_r\) is the image before compression, i.e., distortion-free. \(I_p\) is the same image after lossy compression. The method consists of three main steps. First, a weighted graph (\(G_b\)) is built on the sphere, along each border (b) between two adjacent faces of the cube that is enclosing the sphere (Fig. 4). The position of the vertices of the graph is obtained by projecting the discrete grid of pixels from the cube faces to the sphere, via the inverse gnomonic projection [13]. The reference (processed) signal lying on the graph, \(I_{r,b}\) (\(I_{p,b}\)), consists of the corresponding pixel values in \(I_r\) (\(I_p\)). The graph spectral representation of both the reference and the processed signal is computed by applying the Graph Fourier Transform (GFT) [14]. If all color channels, namely the luminance and chrominance channels, are considered, each is treated as a separate signal, and the GFT is computed for each of them. Second, the spectra of the reference and processed signals are compared by means of a divergence measure, to assess whether the spectral characteristics of the signal on the graph have been altered due to the presence of a seam. Finally, a classification model, trained over a set of ground truth annotated data, is used to define a function that maps the value of the divergence measure (three values if all color channels are considered) to a detection decision for a potential seam. Each step of the algorithm is detailed hereafter. Algorithm 1 summarizes the entire process.

#### Algorithm 1 Graph-based detection of seams

**Input:** \(I_r\): reference (uncompressed) CM image. 
\(I_p\): processed (compressed) CM image. 
\(ColorChannels\): \(Y\) or \{\(Y, Ch, Cr\)\}, i.e., the color channel(s).

**Model:** a pre-trained classification model for visible seams based on a divergence metric, \(M\), between signal spectra.

**Result:** isSeam: array of binary values with 0 = non-visible seam, 1 = visible seam.

```matlab
for b = 1, ..., 12 do
    \(G_b\) and \(I_{r,b}\) ← build_border_graph(\(I_r, b\));
    \(G_b\) and \(I_{p,b}\) ← build_border_graph(\(I_p, b\));
    for each \(ch \in ColorChannels\) do
        \(I_{\hat{r},b}(ch)\) ← compute_GFT(\(G_b, I_{r,b}(ch)\))
        \(I_{\hat{p},b}(ch)\) ← compute_GFT(\(G_b, I_{p,b}(ch)\))
        \(M_b(ch)\) ← compute_M(\(I_{\hat{r},b}(ch), I_{\hat{p},b}(ch)\))
    end for
    isSeam(b) ← classify_border(Model, \{\(M_b(ch)\)\});
end for
```

**Graph-based representation.** We define an undirected, 4-connected, weighted graph, referred to as *border-graph,*
$\mathcal{G}_b = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$, which consists of a finite set of vertices $\mathcal{V}$ lying on a sphere, $|\mathcal{V}| = N$, a set of edges $\mathcal{E}$, and a weighted adjacency matrix $\mathbf{W}$ [15]. Our vertices are a subset of the samples resulting from the inverse gnomonic projection [13] of pixels from the cube faces enclosing the sphere to the spherical surface: they are located along the border $b$ between two adjacent cube faces. The border-graph includes $L$ vertices on each cube face (i.e., $|\mathcal{V}| = N = 2L$). Each vertex is connected to its 4 nearest neighbors, apart from the vertices at the corner and the borders of the graph, which are connected to 2 and 3 neighbors, respectively. This connectivity matches the pattern of neighbouring pixels on the cube faces. Thus, a border-graph creates the structure depicted in Fig. 4. If the vertices $i$ and $j$ are connected by an edge, the entry $W_{i,j}$ in $\mathbf{W}$ represents the weight of the edge, otherwise, $W_{i,j} = 0$. We define the weight of an edge connecting vertices $i$ and $j$ via a Gaussian kernel weighting function [15]:

$$W_{i,j} = \exp\left(-\frac{d(i,j)^2}{2\theta^2}\right)$$

for a parameter $\theta$, where $d(i,j)$ is the geodesic distance between vertices $i$ and $j$ on the sphere.

The portion of the 360-degree image along the border between two adjacent cube faces defines the signal $x: \mathcal{V} \to \mathbb{R}$ lying on the border-graph. Particularly, for $i = 1, \ldots, N$, the signal value on the vertex $i$, $x(i)$, is the value of the corresponding pixel on a cube face in the reference (processed) CM image, $I_r$ ($I_p$).

The spectrum of both the reference and processed signals on each border-graph is computed by applying the GFT [14], as follows. Known operators defined on a graph are the graph Laplacian, defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, and the symmetric normalized Laplacian, defined as $\mathcal{L} = \mathbf{D}^{-1/2}(\mathbf{L})\mathbf{D}^{-1/2}$, 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 $i$. $\mathcal{L}$ is a real symmetric matrix, thus, it has a complete set of orthonormal eigenvectors, $u_i = \{u_i\}$, associated to real non-negative eigenvalues, $\lambda_i = \{\hat{\lambda}_i\}$, with $i = 0, 1, \ldots, N - 1$ and $0 = \lambda_0 \leq \lambda_1 \leq \cdots \lambda_{N-1} \leq 2$ [15]. The eigenvectors of $\mathcal{L}$ are used to define the GFT $\hat{x}$ of the signal $x$:

$$\hat{x}(\lambda_i) = \langle x, u_i \rangle = \sum_{i=1}^{N} x(i)u_i^*(i).$$

We refer to $I_{r,b}$ ($I_{p,b}$) as the reference (processed) signal defined on $\mathcal{G}_b$: thus, $\hat{I}_{r,b}$ ($\hat{I}_{p,b}$) indicates its GFT. By computing $\hat{I}_{r,b}$ and $\hat{I}_{p,b}$, we are able to capture the main characteristics of the signal before and after compression in the spectral domain. Particularly, the eigenvectors of the two graphs will be the same and the only difference will be in the graph Fourier coefficients of the two signals. These can be compared to detect a seam on each border-graph.

**Spectral comparison.** To assess whether the characteristics of the signal on a border-graph have been altered due to the presence of a seam, the probability distributions of the graph Fourier coefficients of the reference and the processed signals lying on the border-graph are compared by means of the Kullback-Leibler divergence (KLD) [16]. The KLD can be computed on the entire set of graph Fourier coefficients, or over a subset of them, selected by applying a band-pass filter before computing the divergence. The use of a filter permits to localize the analysis to the portion of the signal spectrum most affected by the seam.

If multiple color channels are considered, each channel is considered separately, i.e., a value of KLD is computed for each channel.

**Classification.** To solve the binary classification problem (i.e., visible or non-visible seam), based on the luminance-only or the multi-channel KLD values, a classification model, trained on an annotated dataset, is used. A cross-validation approach is used to demonstrate that the performance achieved in terms of accuracy of detection are stable across multiple partitions of the same dataset into training and test sets.

### 3. EXPERIMENTS

**Test material and ground truth.** The test material used in the experiments has been created starting from 50 360-degree images in ER format, at 9104x4552 pixels resolution, selected from the SUN360 dataset [17]. Half of them depict indoor scenes, the other half outdoor scenes. We applied the CM projection and generated our set of reference images in baseball arrangement at 3414x2276 pixels resolution. To generate our set of processed images, we applied lossy JPEG compression to the reference CM images. We used quality parameters $\{30, 50, 70\}$ in the Matlab function `imwrite` to generate half of the processed images, and $\{40, 60, 80\}$ to generate the other half. The resulting set of 150 compressed images, mixing 50 scenes and 6 JPEG compression levels, was divided into three subsets, each visualized and annotated by one expert viewer using a Lenovo Mirage HMD. Particularly, each viewer visually inspected the portion of the image corresponding to each of the 12 borders between two adjacent cube faces and indicated *if a seam was visible or not*. An interface, allowing the overlayed visualization of the location of the border between cube faces on the 360-degree image, was developed to facilitate the annotation task. As a result, our dataset contains 1800
annotated CM borders, out of which 223 are classified as visible seams.

**Planar metrics.** None of the metrics proposed in literature [5–11] addresses the detection of localized distortions in 360-degree content due to lossy compression in CM domain. Therefore, the performance of the proposed method is compared to that of a set of classical FR quality metrics for 2D images, i.e., *planar metrics*. These are applied to what we refer to as the *planar signal*: a portion of the image obtained by merging two adjacent cube faces and considering only L pixels on each face around the border between them. A planar signal is built for each of the 12 borders between cube faces, where a seam could potentially be identified. The metrics included in our experiments are: the PSNR, the FSIM [18], the SSIM [19], the GSIM [20], and the GMSD [21]. A direct comparison between our method and the same approach based on spectral analysis but without making use of the graph representation, i.e., ignoring the spherical nature of the 360-degree signal, is also considered. In this case, the spectrum of the planar signal, both for the reference and processed CM image, is computed via the Fast Fourier Transform (FFT) or the Discrete Cosine Transform (DCT). The corresponding planar metrics are referred to as FFT KL and DCT KL.

**Classification models & cross-validation.** Three classification models are tested: logistic regression (Logit) [22], linear discriminant (LD) [23], and support vector machine (SVM) [24]. A 200 × 5-fold cross-validation is performed on our dataset. First, some samples are selected from the original dataset to have a balanced set: since there are 223 visible seams in the dataset, this selection results in 446 samples. Then, the selected samples are randomly divided into 5 folds. For each fold: (1) the 4 others are used to learn the model parameters; (2) the model is validated with the selected fold. This process is repeated 200 times, ensuring that for each repetition the same set of pixels are used for all metrics and classification techniques.

**Results.** All metrics were computed on a vertically subsampled version of the cube faces, corresponding to a face resolution of 1138x569 pixels. Table 1 reports the detection accuracy achieved by each metric, using the different classification techniques, with L = 4 and L = 8. The proposed method is referred to as GFT KL or GFT BP KL, when the entire or a band-pass filtered version of the spectrum is considered, respectively. The average, maximum, and standard deviation of the accuracy index are reported, computed over the 200 iterations of the cross-validation. Our results show that: (1) the best overall average and maximum accuracy (91.80% and 94.22%) are achieved by our method using L = 8 pixels and all color components in conjunction with a band-pass filter, which permits to localize the analysis to the portion of the signal spectrum most affected by the seam; (2) the graph-based approach, which takes into consideration the spherical nature of the underlying signal, is always consistently better than the spectrum-based ones in planar domain, i.e., the best GFT classification model is always better than the best FFT and DCT one, for different values of L.

<table>
<thead>
<tr>
<th>metric (L=4)</th>
<th>logit(y)</th>
<th>lda(y)</th>
<th>svm(y)</th>
<th>logit(y)</th>
<th>lda(y)</th>
<th>svm(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (L=4)</td>
<td>86.00 (87.95 ; 1.12)</td>
<td>85.39 (87.95 ; 1.12)</td>
<td>85.27 (87.95 ; 1.12)</td>
<td>85.77 (89.95 ; 1.35)</td>
<td>85.69 (89.95 ; 1.39)</td>
<td>85.74 (89.95 ; 1.41)</td>
</tr>
<tr>
<td>SSIM (L=4)</td>
<td>83.33 (86.17 ; 1.38)</td>
<td>82.77 (86.17 ; 1.38)</td>
<td>82.65 (86.17 ; 1.38)</td>
<td>83.05 (86.42 ; 1.50)</td>
<td>82.97 (86.42 ; 1.54)</td>
<td>83.02 (86.42 ; 1.57)</td>
</tr>
<tr>
<td>FSIM (L=4)</td>
<td>79.56 (83.17 ; 1.45)</td>
<td>79.01 (83.17 ; 1.45)</td>
<td>78.98 (83.17 ; 1.45)</td>
<td>80.01 (83.80 ; 1.53)</td>
<td>79.93 (83.80 ; 1.57)</td>
<td>80.00 (83.80 ; 1.60)</td>
</tr>
<tr>
<td>GSIM (L=4)</td>
<td>76.55 (81.17 ; 1.48)</td>
<td>76.01 (81.17 ; 1.48)</td>
<td>75.98 (81.17 ; 1.48)</td>
<td>77.60 (83.24 ; 1.66)</td>
<td>77.06 (83.24 ; 1.70)</td>
<td>77.60 (83.24 ; 1.73)</td>
</tr>
<tr>
<td>GMSD (L=4)</td>
<td>73.54 (78.35 ; 1.50)</td>
<td>73.00 (78.35 ; 1.50)</td>
<td>72.97 (78.35 ; 1.50)</td>
<td>75.65 (81.74 ; 1.69)</td>
<td>75.12 (81.74 ; 1.73)</td>
<td>75.65 (81.74 ; 1.76)</td>
</tr>
<tr>
<td>L=4</td>
<td>84.85 (87.69 ; 1.12)</td>
<td>84.39 (87.69 ; 1.12)</td>
<td>84.36 (87.69 ; 1.12)</td>
<td>85.97 (90.45 ; 1.38)</td>
<td>85.54 (90.45 ; 1.42)</td>
<td>85.97 (90.45 ; 1.45)</td>
</tr>
<tr>
<td>DCT KL (L=4)</td>
<td>81.33 (86.17 ; 1.52)</td>
<td>80.92 (86.17 ; 1.52)</td>
<td>80.89 (86.17 ; 1.52)</td>
<td>82.99 (88.47 ; 1.70)</td>
<td>82.56 (88.47 ; 1.74)</td>
<td>82.99 (88.47 ; 1.77)</td>
</tr>
<tr>
<td>FFT KL (L=4)</td>
<td>84.85 (87.69 ; 1.12)</td>
<td>84.39 (87.69 ; 1.12)</td>
<td>84.36 (87.69 ; 1.12)</td>
<td>86.06 (90.54 ; 1.30)</td>
<td>85.63 (90.54 ; 1.34)</td>
<td>86.06 (90.54 ; 1.37)</td>
</tr>
</tbody>
</table>

**Table 1:** Experimental results showing the average and maximum accuracy and the standard deviation (in the format: "avg. (max. : std.)") of the 200x5-fold cross-validation using different seams widths and regression methods: Logit (Logistic Regression); LDA (Linear Discriminant Analysis); and SVM (Support Vector Machine). The best result, per value of L and number of channels, is highlighted in bold.

---

4. CONCLUSION

This paper proposes a FR algorithm for the detection of seams in compressed CM 360-degree images, which exploits a graph-based representation of the 360-degree images to account for the native spherical nature of the signals. Our method outperforms traditional 2D image quality metrics as well as the proposed approach applied in the planar domain, thus, ignoring the spherical geometry. It can be considered as a first step towards the design of a complete quality metric for compressed 360-degree content, accounting for multiple distortions. Future work will focus on its adaptation to the no-reference scenario, to detect other kinds of discontinuities [2].
5. REFERENCES


