Rate-Distortion Optimized Graph-Based Representation for Multiview Images with Complex Camera Configurations

Xin Su, Thomas Maugey, Christine Guillemot

Rate-Distortion Optimized Graph-Based Representation for Multiview Images with Complex Camera Configurations

Xin Su, Thomas Maugey *Member, IEEE* and Christine Guillemot *Fellow, IEEE*

***Abstract*—Graph-Based Representation (GBR) has recently been proposed for describing color and geometry of multiview video content. The graph vertices represent the color information, while the edges represent the geometry information, *i.e.* the disparity, by connecting corresponding pixels in two camera views. In this paper, we generalize the GBR to multiview images with complex camera configurations. Compared with the existing GBR, the proposed representation can handle not only horizontal displacements of the cameras but also forward/backward translations, rotations, *etc*. However, contrary to the usual disparity that is a 2-dimensional vector (denoting horizontal and vertical displacements), each edge in GBR is represented by a one-dimensional disparity. This quantity can be seen as the disparity along an epipolar segment. In order to have a sparse (*i.e.*, easy to code) graph structure, we propose a rate-distortion model to select the most meaningful edges. Hence the graph is constructed with "*just enough*" information for rendering the given predicted view. The experiments show that the proposed GBR allows high reconstruction quality with lower or equivalent coding rate compared with traditional depth-based representations.**

***Index Terms*—Geometry information, graph-based representation (GBR), complex camera configurations, disparity, distortion model**

I. Introduction

THE recent and upcoming imaging and display hardware encourage the development of new multiview video coding approaches, such as 3DTV and free viewpoint TV [1]. The very large volume of data requires the use of a proper representation of the data, compact and efficiently compressed. Basically, a representation of multiview data should capture both color and geometry contained in the 3D signal. The color information is typically described by 2D images. The scene geometry information can be explicitly represented by depth or disparity [2], [3]. In order to construct a compact representation of multiview data, inter-view redundancy must be removed with the help of scene geometry given by the depth or disparity maps. The captured geometry information depends on the camera configurations. If the cameras are assumed to be aligned, as in conventional 3DTV applications, the displacement of the projection of one scene point from one view to the other is horizontal. For other applications such as free viewpoint television [1], virtual reality [4] or augmented reality [5], the camera configurations, hence the displacement of scene points from one view to the other, is more complex. In this paper, we focus on the problem of geometry compact representation with complex camera configurations.

In the popular multi-view plus depth (MVD) [6] format, the geometry is represented by a depth map, a gray scale image describing the distance between the camera and the scene. The depth map is used by image-based rendering techniques to render virtual views at any viewpoint [7]. Lossy compressions of MVD format is classically performed by the high efficiency video coding standard, namely 3D-HEVC [8], [9]. However, the lossy compression of depth data may cause edge displacement artifacts around foreground objects in the rendered views due to the smoothing of depth edges. As shown in [10], depending on the camera acquiring configuration, a different ratio between depth and color bitrate may be needed. To solve this problem, rate-distortion models (*e.g.*, in [11]) have been proposed to guarantee less depth error around edges. Another approach has been proposed in [12] in which depth edges are losslessly encoded in order to keep the piecewise contour aspect of the depth images.

Disparity, as an alternative to depth, describes the scene geometry by the distance between two pixels representing the same 3D point in two different views. Compared with the depth that gives an exhaustive representation of the geometry with respect to a single viewpoint, the disparity represents the geometry relation between two views. Based on the camera parameters of the two views, the disparity in each point can be easily derived from the depth information. In multiview video coding (MVC) [13], the disparity is used for inter-view prediction, but its "*block similarity*" assumption may fail when the foreground color is similar to the background color or when homographic transforward occurs between the two views. More recently, a graph-based representation (GBR) [14] has been proposed, in which, the graph connections are derived from the disparity and hold pixel-based "*just enough*" geometry information to synthesize the considered predicted views. However, only horizontal translations of the cameras were considered in the GBR proposed in [14].

In this paper, we extend the promising GBR approach to multiview systems with complex camera configurations. Beyond horizontal camera translations, the proposed GBR can handle more complex camera displacements, such as rotations and forward/backward translations. In the former GBR, the edges describe the disparity as follows. Each inter-view edge of the graph links one pixel and its (horizontal) neighboring pixel in the 3D scene (the gap between the two pixels is

Xin Su is with the Institut National de Recherche en Informatique et en Automatique, Rennes 35042, France (e-mail: xin.su@inria.fr).

Thomas Maugey is with the Institut National de Recherche en Informatique et en Automatique, Rennes 35042, France (e-mail: thomas.maugey@inria.fr).

Christine Guillemot is with the Institut National de Recherche en Informatique et en Automatique, Rennes 35042, France (e-mail: christine.guillemot.fr).

the disparity). The extension to complex camera configurations is not straightforward since the disparity becomes two-dimensional (horizontal and vertical displacement). As it was preliminary studied in [15], in order to circumvent this complexification, we use the concept of *epipolar segment* to keep the disparity one-dimensional. An *epipolar segment* (the purple segment shown in Fig.1) is a line segment consisting of all possible projections of a pixel with varying depth. A one-dimensional quantity is enough to denote the true projection position on the epipolar segment. Thus, in the proposed GBR the edge (*e.g.*, the blue link in Fig.1) representing the disparity can be presented by a one-dimensional value. This one-dimensional value, namely unidimensional disparity in this paper, can be seen as the disparity on the epipolar segment.

In order to have a sparse (*i.e.*, easy to code) graph structure, we horizontally group neighboring pixels to form a segment and only one connection is assigned to this segment, instead of one connection per pixel. Moreover, a rate-distortion model taking into account the reconstruction quality and the bitrate needed for coding the geometry is used to remove the less important edges of the graph and regroup a larger set of pixels described by the same edge. The proposed rate-distortion model minimizes a cost function consisting of distortion and bitrate. The constructed graph edges are finally represented by the unidimensional disparity maps. To code the unidimensional disparity maps, we first propose a lossless compression scheme using the mix of JBIG [16] and arithmetic edge coding [17] for the position of non-zero values in unidimensional disparity maps and DPCM for the unidimensional disparity values. A lossy compression with HEVC of the unidimensional disparity maps has also been considered to further reduce the bitrate. Our experimental results demonstrate that the proposed GBR leads to high reconstructing quality with less or comparable coding rate compared with traditional depth-based representations.

The proposed GBR can be considered as *a lossy representation followed by a lossless compression* of multiview geometry, compared with state of art depth-based approaches which are *a lossless representation followed by a lossy compression*. Since the edges of the constructed graph connect pixels across the views, providing more *neighboring* (inter views) information than typical 2D images, new compression approaches based on graph transform [18] may be developed based on the GBR structure for coding the pixels color values, which is out of the scope of this paper.

The rest of this paper is organized as follows. In Section II, we discuss related work. In Section III, the construction of graph is introduced in detail. Section IV presents the proposed rate-distortion model that selects the most meaningful edges of the graph. The view reconstruction from a graph is detailed in Section V. Finally, in Section VI, we show the experiments conducted to compare depth-based representations and our GBR.

II. Multiview Geometry and View Synthesis

A. Depth Image Based Rendering (DIBR)

Let us consider a scene captured by two cameras views $\mathcal{I}_1$ and $\mathcal{I}_2$ of size $X \times Y$ with camera configurations $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$,

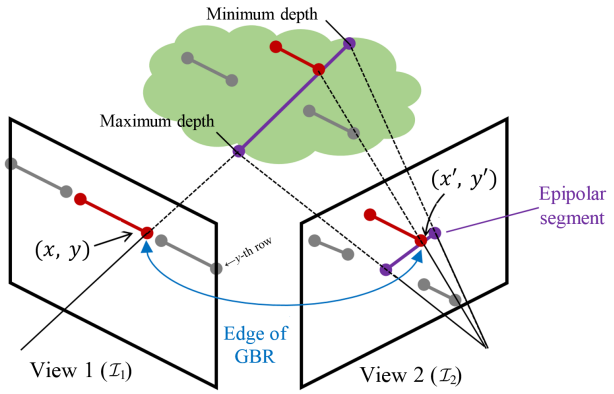


Fig. 1. The concept of GBR: The vertices correspond to pixels of multiview images; The edges link pixels in one view and their projections in another view.

where $\mathbf{\Phi}_i = \{\mathbf{M}_i, \mathbf{R}_i, \mathbf{T}_i\}$ are the parameters of the camera. $\mathbf{M}_i$ is the intrinsic matrix, $\mathbf{R}_i$ is the rotation matrix and $\mathbf{T}_i$ is the position of the camera ($[\mathbf{R}_i|-\mathbf{R}_i\mathbf{T}_i]$ is also known as the extrinsic matrix). As detailed in [7] and illustrated in Fig.1, pixel (x, y) in $\mathcal{I}_2$ (with depth $z_2(x, y)$) can be projected to $\mathcal{I}_1$ by

$$\begin{cases} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \mathbf{R}_2^{-1}\mathbf{M}_2^{-1} \begin{bmatrix} xz_2(x,y) \\ yz_2(x,y) \\ z_2(x,y) \end{bmatrix} + \mathbf{T}_2 \\ \begin{bmatrix} x'z_1(x',y') \\ y'z_1(x',y') \\ z_1(x',y') \end{bmatrix} = \mathbf{M}_1\mathbf{R}_1 \left(\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} - \mathbf{T}_1 \right) \end{cases}, \tag{1}$$

where, $[x_r, y_r, z_r]^{\mathrm{T}}$ are the coordinates of the corresponding point in the 3D scene. The relation between pixel (x, y) in $\mathcal{I}_2$ and pixel (x', y') in $\mathcal{I}_1$ can be denoted as $[x, y, z_2(x, y)]^{\mathrm{T}} \underset{\mathbf{\Phi}_1,\mathbf{\Phi}_2}{\rightleftharpoons} [x', y', z_1(x', y')]^{\mathrm{T}}$, which means they satisfy Eq.(1). Under the Lambertian assumption, the color at (x, y) in $\mathcal{I}_2$ is the same as the color at (x', y') in $\mathcal{I}_1$.

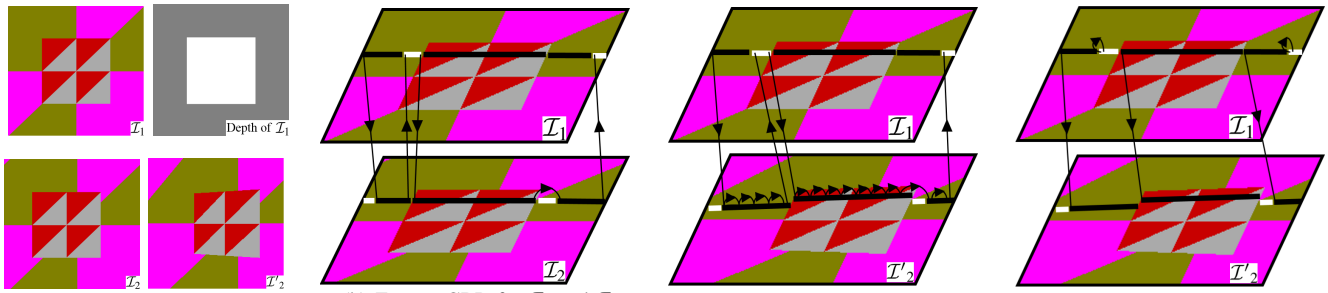
Considering $\mathcal{I}_1$ as the reference view (the color and depth are known), the predicted view ($\mathcal{I}_2$) can be generated by Eq.(1) from the reference view. This is referred to as *forward projection* (or forward warping), which is classically used in depth-image-based rendering (DIBR).

B. Depth vs. Disparity

A depth value is the distance from the observed object to one camera, while disparity is the displacement of pixels between two cameras. The position of the projected point (x', y') in $\mathcal{I}_1$ of a pixel (x, y) in $\mathcal{I}_2$ corresponding to a 3D point of depth $z_2(x, y)$ is given by Eq.(1). The disparity is thus defined as

$$\vec{d}(x, y) = (\Delta x, \Delta y) = (x' - x, y' - y) \ , \tag{2}$$

where $\Delta x, \Delta y \in \mathbb{R}$. When the displacement of the camera from one view to the other is a horizontal translation, the geometrical correlation between two views is only horizontal. In this case, the disparity vector $\vec{d}(x, y) = (\Delta x, \Delta y = 0)$ is simplified to a real number $d(x, y) = \Delta x$. Note that the coordinates (x, y) in Eq.(1) are integer numbers, while the coordinates (x', y') (the projection position) are not integer.



(a) Multiview images. (b) Former GBR for $\mathcal{I}_1$ and $\mathcal{I}_2$ (c) Former GBR for $\mathcal{I}_1$ and $\mathcal{I}'_2$ (d) New GBR for $\mathcal{I}_1$ and $\mathcal{I}'_2$

Fig. 2. GBR extensions to complex camera configuration. (a) Multiview images. From left to right and from top to bottom, the color images of $\mathcal{I}_1$, the depth of $\mathcal{I}_1$, the color images of $\mathcal{I}_2$ and $\mathcal{I}'_2$. The camera displacements from $\mathcal{I}_1$ to $\mathcal{I}_2$ are only horizontal translations, while the displacements from $\mathcal{I}_1$ to $\mathcal{I}'_2$ are horizontal translations and rotations. In this example, $\mathcal{I}_1$ is reference view, while $\mathcal{I}_2$ or $\mathcal{I}'_2$ is predicted view. (b) The former GBR [14] is constructed for $\mathcal{I}_1$ and $\mathcal{I}_2$. The white pixels in $\mathcal{I}_2$ are disocclusions while the white ones in $\mathcal{I}_1$ are occlusions. Each row of $\mathcal{I}_2$ can be reconstructed from $\mathcal{I}_1$ by following the graph edges. Since the camera displacement is simple and the disparity is horizontal, the graph edges are sparse. (c) A naive extension of the former GBR for $\mathcal{I}_1$ and $\mathcal{I}'_2$. $\mathcal{I}'_2$ can be recovered from $\mathcal{I}_1$ by following the graph edges, however since the camera displacements are complex (the disparity is two-dimensional), the graph edges are denser than the ones in (b). (d) The proposed extension of GBR for $\mathcal{I}_1$ and $\mathcal{I}'_2$, in which the edges connect a pair of straight segments (across two views). The graph edges are as sparse as the simple case handled by the former GBR in (b).

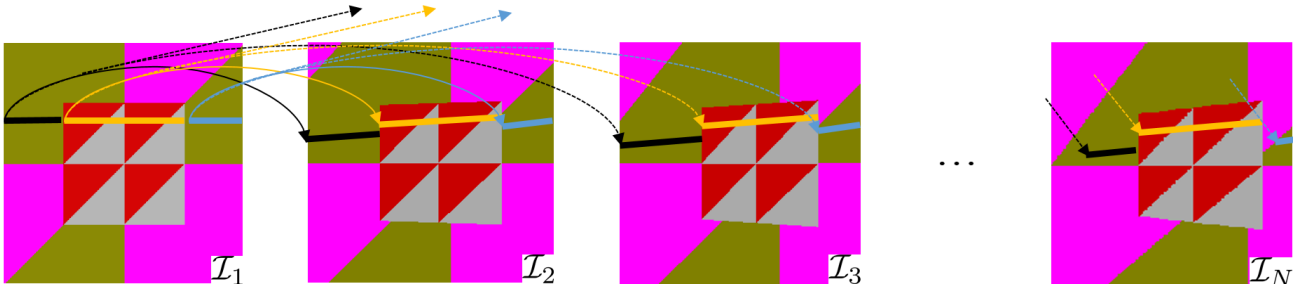


Fig. 3. Graph for multiple views. Only the edges connecting reference view $\mathcal{I}_1$ and $\mathcal{I}_2$ are necessary, marked by solid line. Other edges connecting (marked by dotted lines) linked to view $\mathcal{I}_3, \cdots, \mathcal{I}_N$ can be estimated according the solid edges.

However, for the sake of simplicity, the coordinates (x', y') are often rounded to the nearest integer in the literature. In this paper, the geometry is represented by a position on a 1D epipolar segment and not in the image plane. Therefore, the disparity presented by the position can provide sub-pixel precision, which is not integer value commonly used in disparity based representation. This 1D disparity measure is further explained in the sequel.

We see here that the fundamental difference between depth and disparity resides in the fact that disparity corresponds to two cameras, while depth to only one of them. Compared with depth-based representations, the GBR representation introduced in this paper, as well as the disparity and former GBR, simplifies the depth by considering the predicted views. Moreover, the proposed GBR uses the concept of epipolar segment to keep the disparity unidimensional even for complex camera configurations.

III. Extensions of GBR

Fig.2.a shows a set of multiview images. Among them $\mathcal{I}_1$ is used as reference and $\mathcal{I}_2$ and $\mathcal{I}'_2$ are predicted. The camera displacement from $\mathcal{I}_1$ to $\mathcal{I}_2$ is only a horizontal translation, while the displacement from $\mathcal{I}_1$ to $\mathcal{I}'_2$ is horizontal translation and rotation. An illustration of the former GBR [14] of one row in $\mathcal{I}_1$ and $\mathcal{I}_2$ is shown in Fig.2.b. The principle of the graph edges is connecting pixels and their neighboring pixels in the 3D scene. Since the camera displacement is simple (only horizontal), the pixel displacement has only horizontal component. Each row in $\mathcal{I}_2$ can be recovered from the same row in $\mathcal{I}_1$ by copying color values pixel by pixel from left to right and following the graph edges.

When the camera displacement becomes complex, for instance from $\mathcal{I}_1$ to $\mathcal{I}'_2$, a naive extension of the former GBR can be constructed but with dense edges, as shown in Fig.2.c. This is because for pixels in one row of $\mathcal{I}_1$, their corresponding pixels (corresponding to the same points in the 3D real world) in $\mathcal{I}'_2$ are no longer located at the same row due to the complex camera configurations. More edges are thus needed to reconstruct the rows of $\mathcal{I}'_2$. To reduce the graph density in the complex camera configuration cases, a different principle for graph edges is proposed in this paper. As shown in Fig.2, the idea consists in connecting pairs of straight segments across two views. The predicted view can be reconstructed segment by segment still following the graph edges. The interest of the proposed GBR is that the sparsity of the edges is of a similar order than the previous GBR, even when the camera displacement is complex. An earlier version of this new graph construction is presented in [15]. However, the graph in [15] is constructed only for the 2 input views with regards to the distortion of rendering results, instead of the rate-distortion model proposed in this paper. The proposed GBR in this paper can easily handle multiple views representation and virtual synthesis tasks.

IV. GBR Representation

This section details how to construct a graph for multiview images and how to reconstruct views from the graph.

A. Graph Construction

Let us denote the constructed graph by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where vertices $\mathcal{V} = \{v_i\}$ $(\forall i = 1, \ldots, NXY)$ correspond to the pixels in multiview images $\{\mathcal{I}_n\}_{n=1,\ldots,N}$, and edges $\mathcal{E} = \{e_{ij}\}$ $(i, j = 1, \ldots, NXY)$ connect pairs of pixels across two views.

1) Graph for two views: As mentioned before, pixels in $\mathcal{I}_1$ (reference view) are grouped into a set of straight horizontal segments. Pixels in each segment are supposed to have the same depth. For instance, as shown in Fig.2.d, pixels in one row of $\mathcal{I}_1$ are grouped into 3 segments. Note that we can select the straight segments horizontally (row by row), or vertically (column by column) or in zigzag order. However, since the cameras are usually placed horizontally and camera rotations are also in a horizontal plane, we thus form the straight segments horizontally. Each segment in $\mathcal{I}_1$ is linked to another segment in $\mathcal{I}_2$ by an edge of the graph, *i.e.*, the first pixels of the two segments are connected to each other, as shown in Fig.3. The two linked segments correspond to the same *segment* in the 3D real world. The link, *i.e.*, the graph edge, thus describes the disparity of the segment for two camera views. Since the depth of the segment can be estimated from the edge by Eq.(1), one segment in $\mathcal{I}_1$ needs only one edge to locate its corresponding segment in $\mathcal{I}_2$. The proposed GBR results in an edge set that only connects the pixels in two different views. Fig.2.d gives an illustration of the edges of the constructed graph between two views.

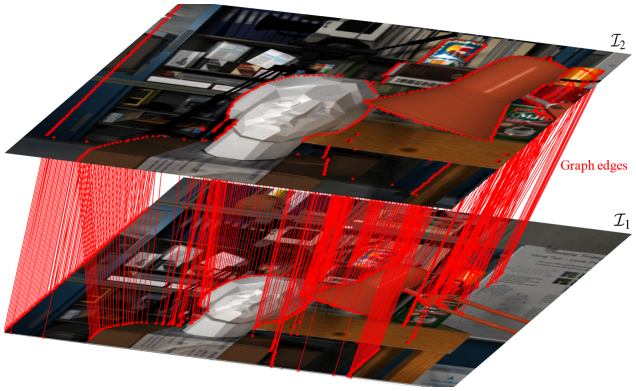
2) Graph for multiple views: Given multiple views (more than 2 views), the graph construction is repeat from one view to another. As shown in Fig.3, each straight segment in $\mathcal{I}_1$ is connected to its corresponding segments in all other views. Using Eq.(1) with camera parameters, the positions of these corresponding segments (in different views) can be estimated from each other. Thus, in the constructed graph, only the graph edges between $\mathcal{I}_1$ and $\mathcal{I}_2$ are kept (marked by solid lines in Fig.3), while the edges linked to other views $\mathcal{I}_3, \cdots, \mathcal{I}_N$ are redundant and can be removed (marked by dotted lines in Fig.3).

3) Graph for disoccluded pixels: The disoccluded pixels in $\mathcal{I}_n$ $(2 \leq n \leq N-1)$ are pixels not visible in $\mathcal{I}_1, \cdots, \mathcal{I}_{n-1}$, but first visible in $\mathcal{I}_n$. The proposed GBR thus also builds edges for these disoccluded pixels. Similar to pixels in reference view $\mathcal{I}_1$, disoccluded pixels are grouped into straight segments according to their depth and one segment has one edge. Only edges linked to $\mathcal{I}_{n+1}$ are kept. The color of disoccluded pixels of $\mathcal{I}_n$ are also stored in the vertices of the graph and used to render the following views $\mathcal{I}_{n+1}, \cdots, \mathcal{I}_N$. In other words, the disoccluded pixels in $\mathcal{I}_n$ is treated as "*reference pixels*" for $\mathcal{I}_{n+1}, \cdots, \mathcal{I}_N$.

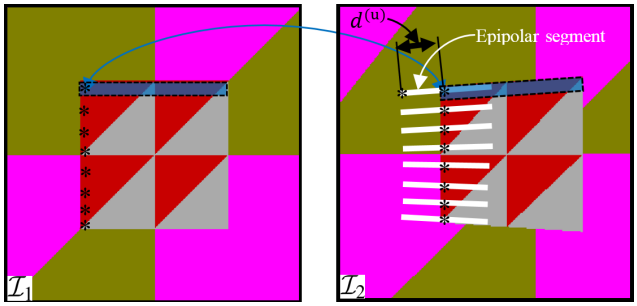
B. Graph Description (unidimensional disparity description for edges)

Basically, the graph constructed in section IV-A captures both vertices $\mathcal{V}$ (corresponding to the color information of multiview images) and edges $\mathcal{E}$ (corresponding to the disparity of multiview images) of the graph. Since the compression of color information is out of the scope of this paper, no compression has been applied on the graph vertices $\mathcal{V}$. The graph only represents the vertices corresponding to $\mathcal{I}_1$ and the disocclusion pixels in $\mathcal{I}_2, \cdots, \mathcal{I}_N$. The color information of the other pixels can be recovered from these vertices.

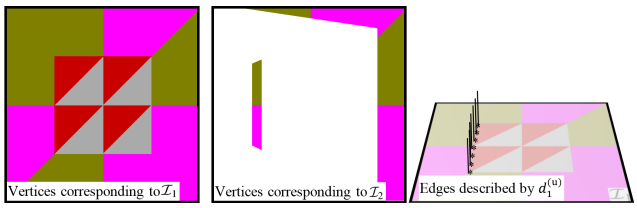
Fig.4.a gives an illustration of the constructed graph for a real dataset (Tsukuba dataset [19]), in which the red lines are the edges. As explained in the previous section, the edges for reference view (between $\mathcal{I}_1$ and $\mathcal{I}_2$) or for disoccluded pixels of $\mathcal{I}_n$ (between $\mathcal{I}_n$ and $\mathcal{I}_{n+1}$) can be described by a huge binary matrix of size $2XY \times 2XY$, which is a connectivity matrix between the $2XY$ pixels. However, since the edges link pixels across two views, the edges can be represented by



(a) An example of the graph for two views.



(b) The unidimensional disparity $d^{(u)}$



(c) The description of the graph.

Fig. 4. Graph construction and description for reference view $\mathcal{I}_1$ (which is similar to the graph for disoccluded pixels of $\mathcal{I}_n$). (a) An example of the graph for real dataset (Tsukuba dataset [19]). The red lines are the graph edges. (b) $d^{(u)}$ denotes the disparity along the epipolar segment (the white lines in $\mathcal{I}_1$). It is the distance between the true projection and the *boundary* projection. (c) The graph is thus described by 2D color values on the graph vertices and a $d^{(u)}$ map describing graph edges.

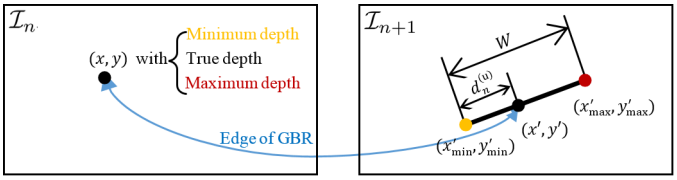


Fig. 5. The unidimensional disparity $d^{(u)}$ is the disparity on the epipolar segment, which can be seen as an index denoting the true projection position on the epipolar segment.

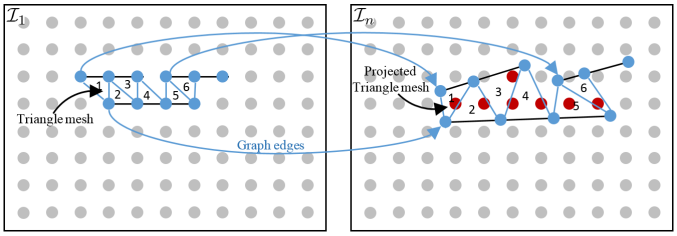


Fig. 6. Mesh based rendering. On the left, Every 3 neighboring pixels with the same or similar depth form a triangle mesh. On the right, the triangle mesh is projected to other views by Eq.(1) with estimated depth. The color values of the pixels inside the projected triangle mesh are the weighted mean of the color values of the triangle mesh's three vertices.

a binary matrix of size $XY \times XY$. For each pixel in $\mathcal{I}_n$, all its possible projections (with varying depth) in $\mathcal{I}_{n+1}$ are located on an epipolar segment, as shown in Fig.5. In addition, the true projection (x', y') in $\mathcal{I}_n$ is between the *boundary* projections $(x'_{\text{min}}, y'_{\text{min}})$ and $(x'_{\text{max}}, y'_{\text{max}})$, where projection $(x'_{\text{min}}, y'_{\text{min}})$ is located by Eq.(1) using the minimum depth and projection $(x'_{\text{max}}, y'_{\text{max}})$ is related to the maximum depth. Therefore, a real number is enough to denote the projection position, *e.g.* the distance between $(x'_{\text{min}}, y'_{\text{min}})$ and (x', y'). Thus, a new quantity $d_n^{(u)}(x, y)$ named *unidimensional disparity* denoting the graph edge linking pixel (x, y) in $\mathcal{I}_n$ and position (x', y') in $\mathcal{I}_{n+1}$ is given by

$$d_n^{(u)}(x,y) = \begin{cases} 0, \text{ if pixel } (x,y) \text{ has no edge} \\ \text{round}\left(\frac{|\vec{d}(x,y)|}{|\vec{d}_{\max}(x,y)|} W + 0.5\right), \text{otherwise} \end{cases}, \tag{3}$$

where,

$$\frac{|\vec{d}(x,y)|}{|\vec{d}_{\max}(x,y)|} = \frac{\sqrt{(x'_{\text{min}} - x')^2 + (y'_{\text{min}} - y')^2}}{\sqrt{(x'_{\text{min}} - x'_{\text{max}})^2 + (y'_{\text{min}} - y'_{\text{max}})^2}} . \tag{4}$$

By using the unidimensional disparity, the connectivity matrix can be replaced by a smaller binary matrix with size of $W \times XY$, where W is the maximum quantization bins within the epipolar segment. Since $d^{(u)}$ measures the distance between $(x'_{\text{min}}, y'_{\text{min}})$ and (x', y'), it can be considered as *a disparity along the epipolar segment*.

Fig.4.b gives an illustration of unidimensional disparity in the toy example. Each pixel (black stars) in $\mathcal{I}_1$ can be projected to $\mathcal{I}_2$ with varying depth. These projections are located on the white lines in $\mathcal{I}_2$, which are the epipolar segments. The unidimensional disparity $d^{(u)}$ denotes the location of the true projection along the epipolar segment with respect to the position $(x'_{\text{min}}, y'_{\text{min}})$ corresponding to the projection of the pixel (x, y) assuming the minimum depth value.

C. View Reconstruction from a Graph

The pixels in the reference view $\mathcal{I}_1$ and the disoccluded pixels in $\mathcal{I}_n$ are recovered directly by copying the color from the corresponding vertices of the graph. The reconstruction of the remaining pixels in $\mathcal{I}_n$ $(2 \leq n \leq N)$ relies on the following steps.

- **$d^{(u)} \rightarrow$ graph edge**: The edges between $\mathcal{I}_1$ and $\mathcal{I}_2$ are recovered directly from the unidimensional disparity $d^{(u)}$ values; For view $\mathcal{I}_n$ $(2 < n \leq N)$, the edges are estimated using Eq.(1) from the ones between $\mathcal{I}_1$ and $\mathcal{I}_2$. The edges for disoccluded pixels in $\mathcal{I}_2, \cdots, \mathcal{I}_{n-1}$ can be obtained by the same way.
- **Graph edge $\rightarrow$ depth**: The depth of each segment is estimated from the edge on the segment by Eq.(1).
- **Mesh based projection [20]**: Each segment and its neighbor segments in previous row can form some triangle meshes, as shown in the left image of Fig.6. The 3 vertices (pixels) of one triangle mesh have the same or similar depth, which means the 3 pixels belong to the same object. The triangle mesh is projected to other views by Eq.(1) with estimated depth, as shown in the right of Fig.6.
- **Color interpolation**: The color values of pixels located in the projected triangle mesh, *i.e.,* the red pixels in Fig.6, are the weighted mean of the color values of the triangle's three vertices. Let us assume that pixel (x', y') is located inside the triangle mesh with vertices at (x'_1, y'_1), (x'_2, y'_2) and (x'_3, y'_3). The color at position (x', y') can be calculated as

$$\begin{aligned} \mathcal{I}(x', y') = &\frac{\omega_1}{\omega}\mathcal{I}(x'_1, y'_1) + \frac{\omega_2}{\omega}\mathcal{I}(x'_2, y'_2) \\ &+ \frac{\omega_3}{\omega}\mathcal{I}(x'_3, y'_3) , \end{aligned} \tag{5}$$

where

$$\begin{aligned} &\omega_j = \exp\left(-(x' - x'_j)^2 - (y' - y'_j)^2\right) , \\ &\omega = \omega_1 + \omega_2 + \omega_3 . \end{aligned} \tag{6}$$

D. Virtual view Synthesis by the Graph

Although it is not the core purpose of the proposed GBR, the geometry described by the graph edges can be used to synthesize virtual views. The depth of reference view is first estimated from the graph edges. Virtual views thus can be synthesized with the estimated depth. However, the graph is constructed with the input of multiple views instead of the virtual views, especially the graph is sparsified and optimized to render the input of multiple views (detailed in section V). The rendering quality of the virtual viewpoints by the graph thus may be not as good as when using depth based representation.

V. Graph Sparsification

According to the GBR's principles presented in section IV, pixels on each segment have the same depth. In order to sparsify the graph, *i.e.*, reducing the number of edges, we can

group pixels even if their depth are not exactly equal. The segment depth can thus vary within a range $[z-\Delta z, z+\Delta z]$ ($\Delta z \geq 0$). It is obvious that large Δz leads to an inaccurate geometry, which further leads to a low rendering quality but leads to a graph less costly to code. Instead of having a predefined Δz, we adapt it to the encoding bitrate and rendering quality.

A. Graph Coding

1) AEC-based coding: In this paper, we only consider the geometry bitrate. The edges of the graph have been described by $d^{(\mathrm{u})}$ maps (a toy example shown in Fig.4.b and c). Each view $\mathcal{I}_n$ ($1 \leq n < N$) has a $d^{(\mathrm{u})}$ map (note that for $d^{(\mathrm{u})}$ maps of predicted views $\mathcal{I}_2, \cdots, \mathcal{I}_{N-1}$, the non-zeros values are only located in the disoccluded regions). Fig.7.b is a real $d^{(\mathrm{u})}$ map, in which pixels with edges are shown in non-white color (different grays show different $d^{(\mathrm{u})}$ values). Comparing Fig.7.a and b, we can see that the contours in the $d^{(\mathrm{u})}$ map correspond to the contours in color image. It is natural that a new segment appears with an edge in the graph when the depth is discontinuous. We thus encode the $d^{(\mathrm{u})}$ map contour by contour. Let us assume that the corresponding $d^{(\mathrm{u})}$ map of $\mathcal{I}_n$ consists of P contours $\{\mathcal{C}_n(p)\}, \forall p = 1, \ldots, P$ (*i.e.*, continuous set of pixels with 1 pixel per segment). The encoding of each contour $\mathcal{C}_n(p)$ proceeds as follows.

- **First pixel location**: The exact location of the first pixels of each contour is described by a bright point in a bi-level image, as shown in Fig.7.c. This bi-level image is then encoded by JBIG (Joint Bi-level Image Experts Group) [16];
- **Direction stream $\{\Delta x\}$**: The exact locations of other pixels (except first pixels) of $\mathcal{C}_n(p)$ are encoded by the arithmetic edge coding (AEC) method [17] (the direction stream in Fig.7.c);
- **$d^{(\mathrm{u})}$ stream $\{d^{(\mathrm{u})}(x,y)\}$**: Once the locations of $\mathcal{C}_n(p)$ have been encoded, the $d^{(\mathrm{u})}$ values on $\mathcal{C}_n(p)$ have also to be losslessly encoded from top to bottom (the $d^{(\mathrm{u})}$ stream in Fig.7.c).

Note that we finally use a Differential Pulse Code Modulation (DPCM) coder to encode the $d^{(\mathrm{u})}$ stream, *i.e.* $\{\Delta d^{(\mathrm{u})}(x,y)\}$ stream. The bitrate for encoding the $d^{(\mathrm{u})}$ map is thus computed contour by contour

$$\mathcal{R}(\mathcal{E}) = \sum_{n=1}^{N-1} \sum_{p=1}^{P} \mathcal{R}(\mathcal{C}_n(p)) \ , \tag{7}$$

where $\mathcal{C}_n(p)$ is the p-th contour in the $d^{(\mathrm{u})}$ map of view $\mathcal{I}_n$. The bitrate for encoding the contour $\mathcal{C}_p$ is

$$\mathcal{R}(\mathcal{C}_n(p)) = \underbrace{R_{\mathrm{J}}(\mathcal{C}_n(p))}_{\text{First pixel}} + \underbrace{R_{\mathrm{D}}(\mathcal{C}_n(p))}_{\text{Direction}} + \underbrace{R_{\mathrm{V}}(\mathcal{C}_n(p))}_{d^{(\mathrm{u})}\text{Value}} \tag{8}$$

where,

$$\begin{aligned} &R_{\mathrm{J}}(\mathcal{C}_n(p)) = c \text{ (bits)} \ , 0 < c < 32 \\ &R_{\mathrm{D}}(\mathcal{C}_n(p)) = \text{entropy}(\{\Delta x\}) \ , \\ &R_{\mathrm{V}}(\mathcal{C}_n(p)) = \text{entropy}(\{\Delta d^{(\mathrm{u})}(x,y)\}) \ . \end{aligned} \tag{9}$$

R_{J} is the bitrate for encoding the exact locations of first pixels of contour $\mathcal{C}_n(p)$. Since the JBIG encoder has been applied, R_{J} highly depends on the image content. However, for the sake of simplicity, we assume that R_{J} is constant for each contour. This constant value c is between 0 and 32, where 32 is the maximum number of bits for coding the coordinates of one pixel (16 bits for x-coordinate and 16 bits for y-coordinate). In this paper, we set $c = 24$. R_{D} is the rest of the contour $\mathcal{C}_n(p)$, which is measured by the entropy of the direction stream $\{\Delta x\}$. R_{V} is the bitrate needed for encoding the $d^{(\mathrm{u})}$ values on the contour $\mathcal{C}_n(p)$.

2) HEVC-based coding: AEC-based coding is a lossless compression of $d^{(\mathrm{u})}$ images (the edges of the graph). However, to further reduce the bitrate, we can compress the $d^{(\mathrm{u})}$ images with a lossy compression method, *i.e.,* HEVC. Before being compressed by HEVC, the zero values in $d^{(\mathrm{u})}$ images are replaced by their left first non-zeros values to smooth the $d^{(\mathrm{u})}$ images. The filled $d^{(\mathrm{u})}$ images are seen as a video (I-P-P-P-$\cdots$ sequence) then compressed by HEVC.

B. Rate-Distortion model

In this paper, we measure the rendering quality by the color distortion of the predicted views:

$$\mathcal{D}(\mathcal{E}) = \sum_{n=2}^{N} \sum_{y=1}^{Y} \sum_{x=1}^{X} \left|\left| \mathcal{I}_n(x,y) - \hat{\mathcal{I}}_n(x,y) \right|\right|_2^2 \ , \tag{10}$$

where $\hat{\mathcal{I}}_n(x,y)$ is the rendered color at position (x,y) in $\mathcal{I}_n$ which depends on the graph edges, and $\mathcal{I}_n(x,y)$ corresponds to the targeted color (*e.g.,* the original color).

A Lagrangian rate-distortion optimization is performed searching to minimize the cost function Eq.(11), where the bitrate $\mathcal{R}(\mathcal{E})$ given by Eq.(7) and the distortion $\mathcal{D}(\mathcal{E})$ given by Eq.(10). α is the Lagrangian multiplier which represents the relation between bitrate and rendering quality (distortion):

$$\mathcal{J} = \mathcal{D}(\mathcal{E}) + \alpha \mathcal{R}(\mathcal{E}) \ , \tag{11}$$

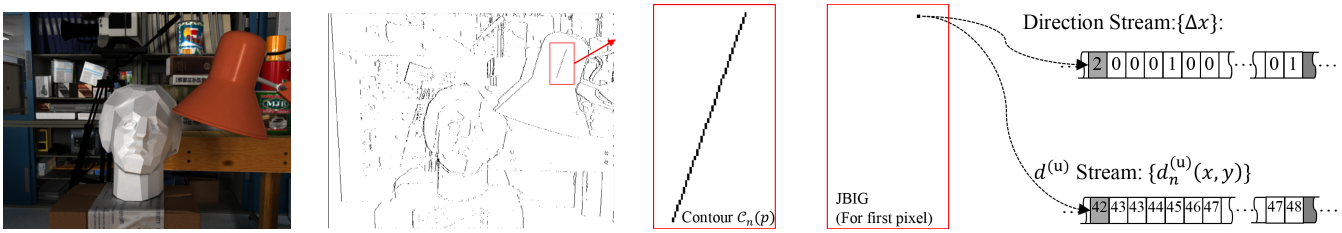
where $\mathcal{J}$ is the Lagrangian cost (smaller $\mathcal{J}$ values mean better optimal status). In this paper, we propose to simplify the graph by searching the set of edges which can be removed such that the Lagrangian cost function is minimized:

$$\begin{aligned} \mathcal{E} &= \underset{\mathcal{E}'}{\text{argmin}} \ \mathcal{J} \\ &= \underset{\mathcal{E}'}{\text{argmin}} \ \mathcal{D}(\mathcal{E}) + \alpha \mathcal{R}(\mathcal{E}) \ . \end{aligned} \tag{12}$$

However, since the solution of Eq.(12) is complex, we present two approximate minimizations in the following.

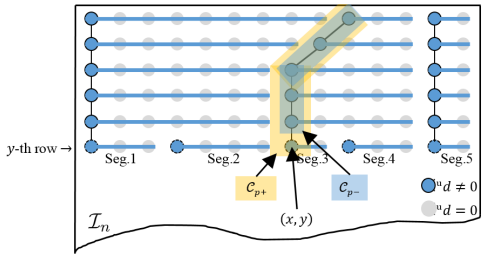
C. Row-wise minimization by shortest path algorithm

1) Row-wise rate-distortion model: Instead of carrying out a global minimization over the entire graph, we perform the minimization in Eq.(12) row-wise as illustrated in Fig.8. Taking the y-th row of view $\mathcal{I}_n$ as an example, all the pixels are initially grouped into 5 segments based on their depth (pixels on each segment have the same depth). Then, some edges are

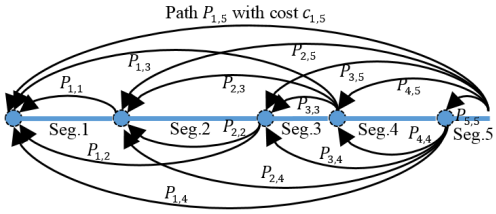


(a) Color image of $\mathcal{I}_n$. (b) $d^{(u)}$ map. (c) Encoding of contour $\mathcal{C}_n(p)$.

Fig. 7. Encoding of graph edges. (a) Color image of $\mathcal{I}_n$; (b) Edges of graph described by $d^{(u)}$ map, in which pixels with edges are shown in non-white color (different color denotes different $d^{(u)}$ value); (c) $d^{(u)}$ map is encoded contour by contour, for instance the locations of contour $\mathcal{C}_n(p)$ are encoded by a bi-level image (for first pixel) and direction stream (for other pixels), $d^{(u)}$ values on contour $\mathcal{C}_n(p)$ are encoded by $d^{(u)}$ stream.



(a) Initial segments on y-th row.



(b) Row-wise minimization (edge removing) by shortest path algorithm.

Fig. 8. Row-wise minimization. (a) Initialization. Pixels on y-th row have been grouped into five segments. Each segment has a constant depth. (b) Edge removing (segment merging). Some edges are removed (the segments are merged into their left neighboring segments) with respect to the row-base Lagrangian rate-distortion cost. The removing process is performed by shortest path minimization.

removed (i.e., some segments are merged) if they reduce the row-wise rate-distortion cost function Eq.(13):

$$\mathcal{J}_n(y) = \sum_{m=n+1}^{N} \sum_{x=1}^{X} \underbrace{\mathcal{D}_m(d_n^{(u)}(x,y))}_{\text{Distortion}} + \alpha \underbrace{\mathcal{R}_n(d_n^{(u)}(x,y))}_{\text{Bitrate}} , \quad (13)$$

where $\mathcal{D}(d_n^{(u)}(x,y))$ is the distortion when projecting pixel (x,y) from the reference view $\mathcal{I}_n$ to the other view $\mathcal{I}_m$ (using mesh-based rendering). The bitrate term $\mathcal{R}(d_n^{(u)}(x,y))$ is the bitrate of encoding $d_n^{(u)}(x,y)$ in $\mathcal{I}_n$, which can be calculated as

$$\mathcal{R}(d_n^{(u)}(x,y)) = \mathcal{R}(\mathcal{C}_{p+}) - \mathcal{R}(\mathcal{C}_{p-}), \quad (14)$$

where pixel (x,y) is on contour $\mathcal{C}_n(p)$. $\mathcal{C}_n(p+)$ denotes the contour with pixel (x,y) (and pixel (x,y) is the end point of contour $\mathcal{C}_n(p+)$, as shown in Fig.8.a), while $\mathcal{C}_n(p-)$ is the contour without pixel (x,y). The unidimensional disparity $d_n^{(u)}$ in each position (x,y) of the y-th row in view $\mathcal{I}_n$ is obtained by minimize the Lagrangian cost $\mathcal{J}_n(y)$, *i.e.*,

$$\{d_n^{(u)}(x,y)\} = \underset{\{d_n^{(u)}(x,y)\}}{\text{argmin}} \ \mathcal{J}_n(y) \ , \ \forall x = 1, \ldots, X \ . \quad (15)$$

can be considered as an approximate minimization of Eq.(12) (under the assumption that $\min\mathcal{J} \approx \sum_n \sum_y \min\mathcal{J}_n(y)$).

2) Minimization by shortest path algorithm: The removing of one edge, i.e., merging one segment into its left segment, can be modeled as a shortest path problem. For instance in the example shown in Fig.8.b, the path $P_{4,5}$ connects the beginning of segment 4 and the end of segment 5, and represents the fact of merging segments 4 and 5. Similarly, path $P_{1,5}$ denotes the merging of segments from 1 to 5. The cost of path $P_{s,t}$ (that segments from s to t are regrouped into one segment) is defined as

$$c_{s,t} = \sum_{m=n+1}^{N} \sum_{x=s_{start}}^{t_{end}} \mathcal{D}_m(d_n^{(u)}(x,y)) + \alpha \mathcal{R}_n(d_n^{(u)}(s_{start})) \quad (16)$$

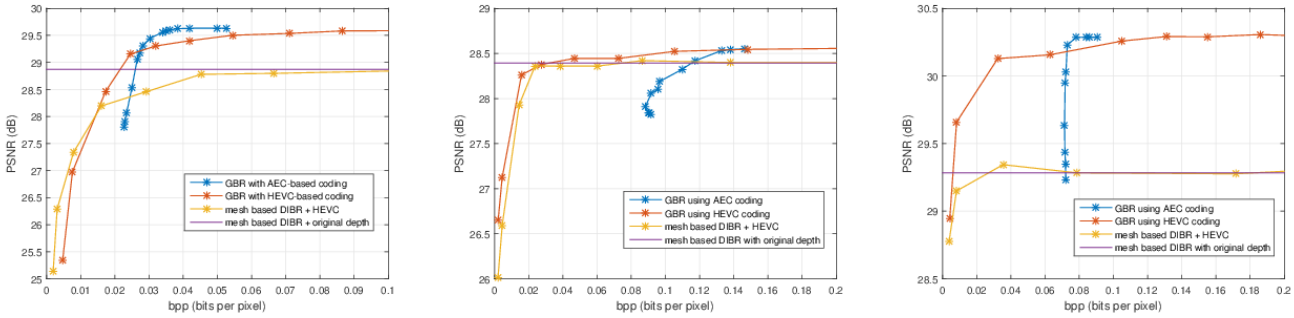
where s_{start} is the first pixel of segment s and t_{end} is the end of segment t. The optimal solution of the shortest path algorithm provides the *shortest* path from the end to the beginning of the y-th row of view $\mathcal{I}_n$.

VI. Experiments

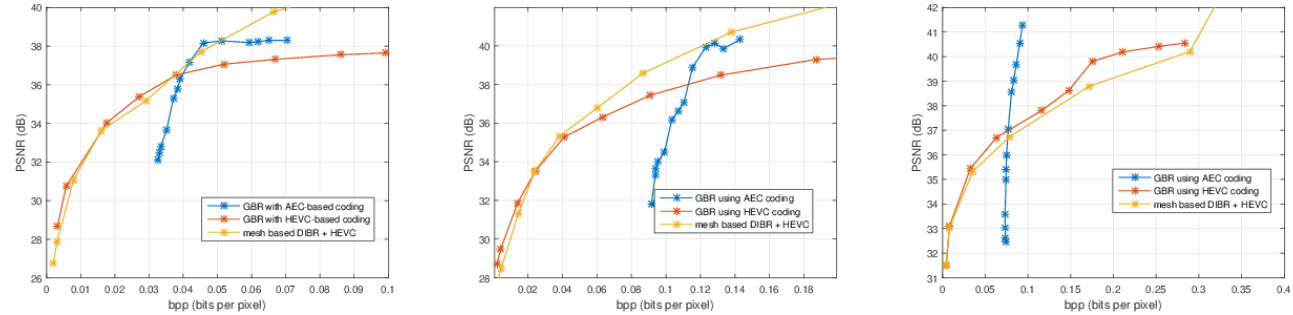
A. Experimental setup

Data sets. In this section, we present experiments to evaluate the performance of the proposed GBR. Three data sets have been tested: 1) Tsukuba dataset [21]: Fig.9.a is a 1 minute video sequence (1800 frames). Each frame contains a stereo pair with ground truth depth maps. Five frames of indices $\{1, 25, 50, 75, 100\}$ have been selected as a set of multiview images. The first frame is reference view and the others are predicted views. 2) MSR dataset [22]: the Ballet dataset is tested. The image of camera 4 is reference view (as shown in Fig.9.b) and the image from camera 3∼0 are predicted views (totally 5 views). 3) Akko&kayo dataset [23] Fig.9.c: the first frames of each video captured by camera 27∼30 are selected (totally 4 views).

This paper only focuses on geometry information, the 2D image representing the color information of $\mathcal{I}_1$ is assumed to have been transmitted separately. To assess the quality of the graph representation, the original (without compression) color information has been used. In addition, the color values



(a) PSNR-rate performance with the true original multiview images as reference. From left to right, Tsukuba dataset, Ballet dataset and Akko&kayo dataset.



(b) PSNR-rate performance with the rendered images as reference. The rendered images are obtained by DIBR with $\mathcal{I}_1$. From left to right, Tsukuba dataset, Ballet dataset and Akko&kayo dataset.

Fig. 10. PSNR-rate performance of multiview representation by different methods on Tsukuba dataset (left), Ballet dataset (middle) and Akko&kayo dataset (right). (a) The PSNR values are computed with the true original multiview images as reference. (b) The rendered images obtained by DIBR with $\mathcal{I}_1$ are the reference.

(a) Tsukuba (b) Ballet (c) Akko&Kayo

Fig. 9. Data sets. From top to bottom: color image of reference view $\mathcal{I}_1$, depth of $\mathcal{I}_1$ and color image of predicted view $\mathcal{I}_3$.

of disoccluded pixels in predicted views are also transmitted separately.

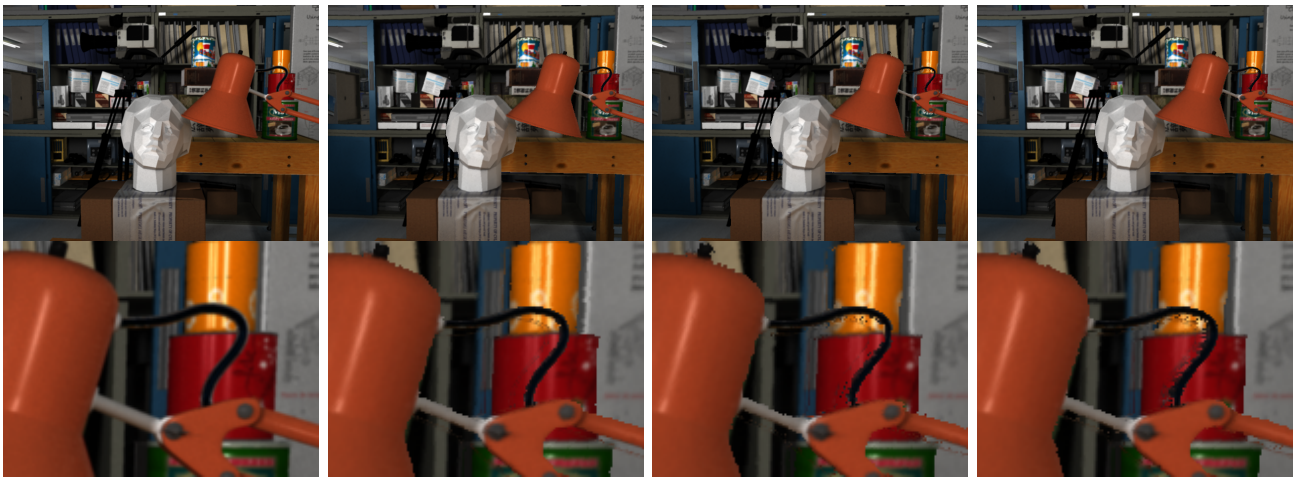
Baseline methods. The proposed GBR is compared with traditional depth-based approaches, in which the depth maps are compressed by HEVC [24]. In these baseline methods, the DIBR [25] method is applied to reconstruct the predicted views. Since the proposed GBR represents the geometry of both reference view $\mathcal{I}_1$ and disocclusions of predicted views $\mathcal{I}_n$ $(2 \leq n \leq N-1)$, the depth maps of reference view $\mathcal{I}_1$ and predicted views $\mathcal{I}_n$ $(2 \leq n \leq N-1)$ are compressed by HEVC as a video sequence.

Experiments. In section VI-B, we evaluate the capability of GBR to represent compactly a set of multiview image geometry. The PSNR values are computed over the whole multiview set using either the true original multiview images or the synthesized ones by DIBR with the reference view as reference. In section VI-C, even if this is not the main goal of GBR, we evaluate its ability to perform virtual view synthesis. In this case, the PSNR values are computed using the views rendered with original depth as reference signal.

B. Multiview representation

1) Parameter setting: The Lagrangian multiplier α varies from 10^0 to 10^4 in the row-wise minimizations. In the HEVC-based graph compression, the QP parameter ranges from 0 to 51.

2) Results: We start by assessing the PSNR-rate performance of the multiview coding scheme based on the proposed GBR. Fig.10.a shows the rate-distortion curves by GBR with AEC-based coding, GBR with HEVC-based coding and DIBR with depth compressed by HEVC. DIBR with non-compressed depth has also been tested as a baseline method, whose



(a) Tsukuba dataset. Bitrate around 0.03 bpp.

(b) Ballet dataset. Bitrate around 0.1 bpp.

(c) Akko&Kayo dataset. Bitrate around 0.08 bpp.

Fig. 11. Visual results. From left to right: Original $\mathcal{I}_3$, rendered view of $\mathcal{I}_3$ by GBR with AEC-based coding, rendered view of $\mathcal{I}_3$ by GBR with HEVC-based coding and rendered view of $\mathcal{I}_3$ by DIBR with HEVC.

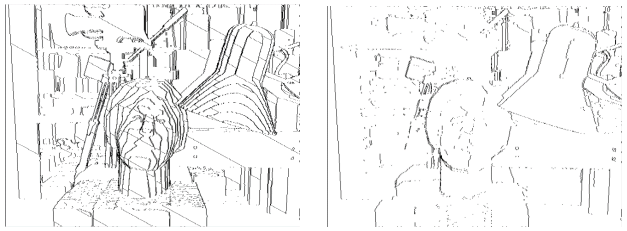
results can be seen as the *maximum* rendering quality since the non-compressed depth has been used. From the rate-distortion curves, we can see that the proposed GBR needs less bitrate to obtain the *maximum* PSNR (the one obtained by DIBR with non-compressed depth). Compared with GBR with AEC-based coding, GBR with HEVC-based coding costs less bitrate when achieving similar PSNR. In addition, GBR even yields the higher PSNR than the one obtained by DIBR+non-compressed depth. This is because sometimes removing an edge may have lower distortion than keeping it. And we use the *real* distortion in the Lagrangian rate-distortion cost instead of a model of distortion, which can help to remove these edges. This is further discussed in section VI-D. Fig.11 illustrates the visual results of the reconstructed $\mathcal{I}_3$ using different methods with similar bitrates. With similar cost in terms of bitrate, the visual results obtained with GBR usually are better than when using DIBR with HEVC (less artifacts on the edges of objects). However, we can notice that the row-wise minimization of GBR may introduce some artifacts in the regions with homogeneous depth, such as the displacement of the poster on the wall in Ballet dataset.

We then evaluate the PSNR-rate performance of the multi-view representation with different original input views. We modified the multiview datasets as follows. The reference view $\mathcal{I}_1$ is the same as in the experiment in Fig.10.a. The predicted views $\mathcal{I}_2, \cdots, \mathcal{I}_N$ are replaced by rendered results $\mathcal{I'}_2, \cdots, \mathcal{I'}_N$ obtained by DIBR using $\mathcal{I}_1$ and the original depth of $\mathcal{I}_1$. The graph is constructed using the same parameters as in the experiment in Fig.10.a. The PSNR values of rendered views are computed with $\mathcal{I'}_2, \cdots, \mathcal{I'}_N$ as reference, which is commonly used in view synthesis literatures [26], [27]. Fig.10.b shows the rate-distortion performance. From these curves, we can find that at low bitrate our GBR is better than or comparable with HEVC, while at high bitrate HEVC generally outperforms our GBR. Compared with results in Fig.10.a, results presented in Fig.10.b only differs on the quality evaluation metric.

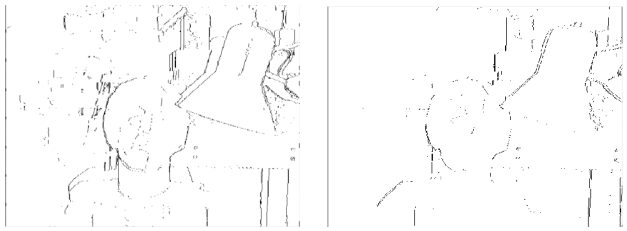
As we can see from Fig.10, the PSNR-rate performance obtained by the GBR with AEC-based coding is always shown by "*vertical*" curves (the blue curves). This is because the lossless compression of AEC-based coding. It does not have too much compression techniques to reduce the bitrate, compared with the advanced and efficient (lossy) compression by HEVC-based coding (the red curves).

C. View synthesis

In some scenarios as FTV, the geometry information can be used to synthesize virtual viewpoints. It is well-known that the depth enables such efficient view synthesis tasks. While GBR's core purpose is to represent a set of input multiview images, the geometry described by the graph edges might be used to generate virtual views. This is what we study here. At the encoder, the GBR representation is constructed for some input multiple views. However, the viewpoints to be synthesized do not belong to the input multiview set. At the decoder, these virtual viewpoints can still be rendered from the constructed graph, as detailed in section IV-D.



(a) Depth discontinuity of reference view. (b) $d^{(u)}$ image with frame 1 as predicted view.



(c) $d^{(u)}$ image with frame 50 as predicted view. (d) $d^{(u)}$ image with frame 90 as predicted view.

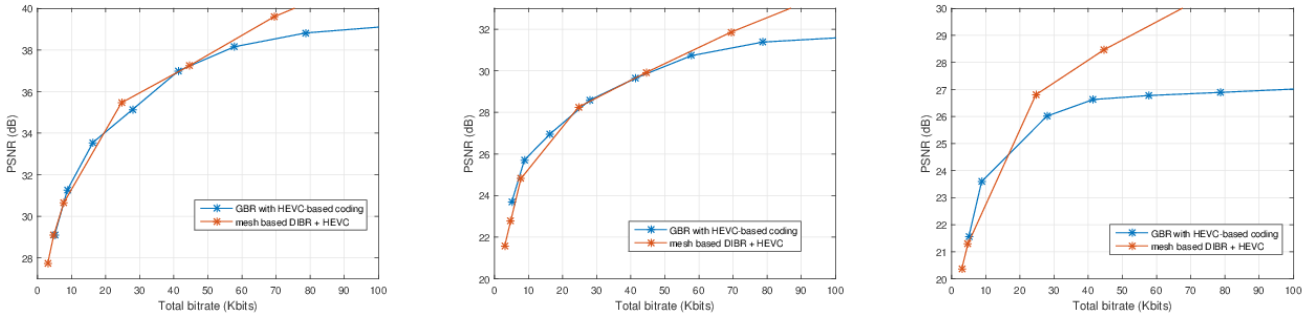
Fig. 13. The proposed GBR simplifies the depth information with respect to the given rendering task. (a) The depth discontinuity of the reference view (frame 100). (b) $d^{(u)}$ image with frame 1 as predicted view. (c) $d^{(u)}$ image with frame 50 as predicted view. (d) $d^{(u)}$ image with frame 90 as predicted view.

1) Experiment setting: This experiment is tested on T-sukuba dataset. Since the multiview images are five frames $\{1, 25, 50, 75, 100\}$ of 1 minute video, another three frames 60, 105 and 130 are tested as virtual views. The constructed graph in section VI-B is used here. These three virtual views (intermediate view, close virtual view and distant virtual view) are rendered with the constructed graph. Since the graph is constructed with (or is optimized to) the input multiple views, the rendering quality of these virtual viewpoints by the graph may depend on how close the virtual viewpoints and the input views are. More precisely, the intermediate view (frame 60) and the close virtual view (frame 105) are *near* the input multiple views, thus their rendered results with the graph are supposed to be comparable with or better than the ones obtained by depth based representation. While, for the distant virtual view (frame 130) which is *far away* from the input views, the rendered results by the graph should be worse than the ones obtained by depth based representation.

2) Results: The rendering results of the intermediate view (frame 60) and close virtual view (frame 105) are shown in Fig.12.a and b, from which we can find that at low bitrate the proposed GBR is comparable with HEVC. However, since the graph is a simplified representation of depth with respect to the input views, it is not as good as depth (the full representation) when rendering virtual view, especially the complex ones, *e.g.,* the virtual view (frame 130) shown in Fig.12.c.

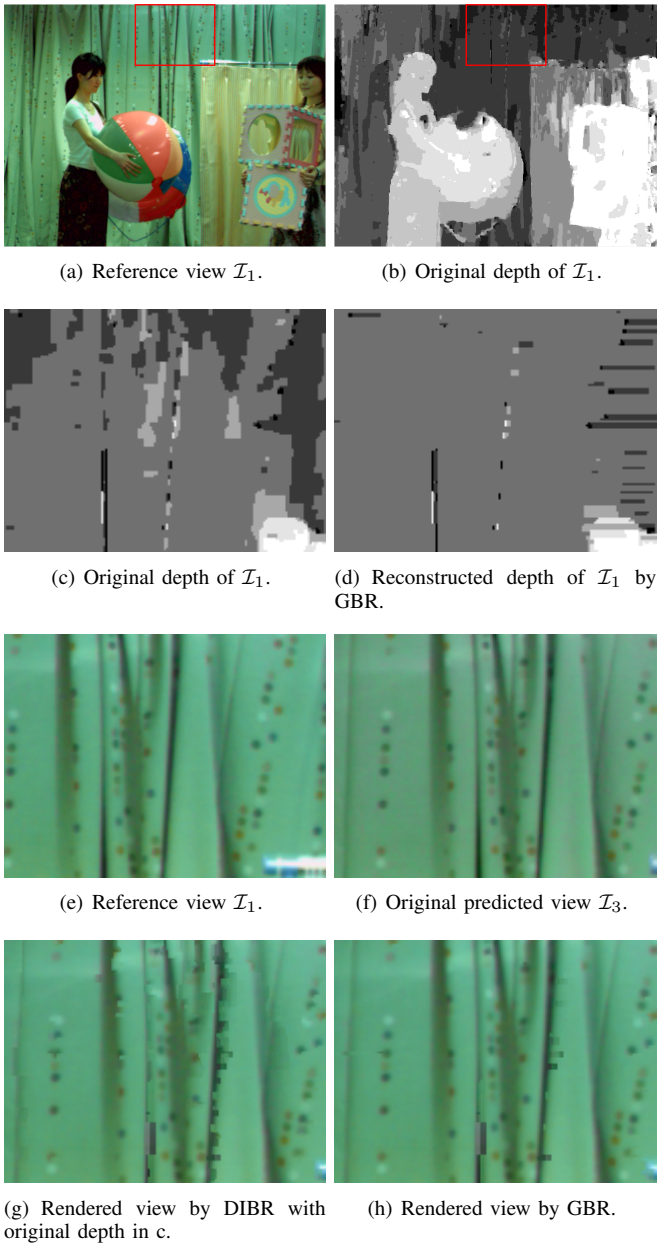
D. Analysis of GBR

1) Simplification of depth: As introduced in section I, GBR simplifies the depth information so that it is sufficient to describe a given multiview set. Fig.13 gives an illustration of how the proposed GBR adapts to the rendering difficulty.



(a) Intermediate view (frame 60). (b) Intermediate view (frame 105). (c) Intermediate view (frame 130).

Fig. 12. Rate-distortion performance of virtual view rendering. The PSNR values are computed with rendered results by DIBR (with original depth) as reference.



(a) Reference view $\mathcal{I}_1$. (b) Original depth of $\mathcal{I}_1$.

(c) Original depth of $\mathcal{I}_1$. (d) Reconstructed depth of $\mathcal{I}_1$ by GBR.

(e) Reference view $\mathcal{I}_1$. (f) Original predicted view $\mathcal{I}_3$.

(g) Rendered view by DIBR with original depth in c. (h) Rendered view by GBR.

Fig. 14. Depth correction by GBR.

In this experiment, frame 100 in Tsukuba dataset is considered as the reference view. Three frames 1, 50 and 90 are considered as predicted views, for which the baseline with the reference view decreases. The geometry relation between the reference view and the predicted view becomes simpler. Theoretically, GBR needs to describe less geometry information to render frame 90 than for example rendering frame 1. Fig.13.b-d show the $d^{(u)}$ images of the constructed graphs with different predicted views, frame 1, 50 and 90 respectively. These three $d^{(u)}$ images are obtained by row-wise minimization with the same parameter α, *i.e.*, the same target quality. We can see that the $d^{(u)}$ image in Fig.13.b becomes *simpler* (less graph edges) as long as the predicted view gets close to the reference view due to the proposed rate-distortion graph sparsification. It illustrates how the GBR adapts its amount of geometry depending on the rendering task.

2) Depth correction: As illustrated in the experiments in section VI-B, the proposed GBR can yield higher PSNR than the DIBR with original depth. This is because the depth images are not always accurate (*e.g.*, estimated by block similarity of color images). Thanks to the graph sparsification, the proposed GBR can *improve* the depth by removing some *noisy* edges. For instance, Fig.14 gives an illustration of *depth correction* done by the proposed GBR. Fig.14.a and b are the color and depth of reference view $\mathcal{I}_1$. The reconstructed depth of $\mathcal{I}_1$ by GBR is shown in Fig.14.d, in which we can find that the depth has been smoothed. Compared with the true original $\mathcal{I}_3$ in Fig.14.f, the rendered view by GBR in Fig.14.h is of better visual quality than the rendered view by DIBR (with original depth) in Fig.14.g.

VII. Conclusion

In this paper, we have proposed an alternative to depth for multiview geometry representation. Contrary to the original GBR in [14], the proposed GBR can deal with multiview images with complex camera configurations. A rate-distortion model has been proposed to simplify the graph. The proposed GBR representation simplifies the depth of multiview images for reconstructing the given predicted views, *i.e.*, the GBR costs less bitrate when obtaining the same high rendering quality. Future work will focus on the full representation of both color and geometry, in which the connections of the graph

should be used as a support for a better texture representation. The extensions of our GBR to other datasets, such as light fields (2D camera arrays), point cloud and so on, are also interesting challenges in the future.

ACKNOWLEDGEMENT

This work has been funded by the regional council of the Brittany Region.

REFERENCES

[1] M. Tanimoto, "Free viewpoint television (FTV)," in *Digital Holography and Three-Dimensional Imaging*. Optical Society of America, 2007, p. DWD2.

[2] H. Y. Shum, S. B. Kang, and S. C. Chan, "Survey of image-based representations and compression techniques," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 11, pp. 1020–1037, 2003.

[3] K. Müller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643–656, 2011.

[4] A. Wexelblat, *Virtual reality: applications and explorations*. Academic Press, 2014.

[5] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators and virtual environments*, vol. 6, no. 4, pp. 355–385, 1997.

[6] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Image Processing, IEEE International Conference on*, vol. 1. IEEE, 2007, pp. 201–204.

[7] E. Martinian, A. Behrens, J. Xin, and A. Vetro, "View synthesis for multiview video compression," in *Picture Coding Symposium*, vol. 37, 2006, pp. 38–39.

[8] K. Müller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. H. Rhee *et al.*, "3D high-efficiency video coding for multi-view video and depth data," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3366–3378, 2013.

[9] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of high efficiency video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 35–49, 2016.

[10] E. Bosc, V. Jantet, M. Pressigout, L. Morin, and C. Guillemot, "Bit-rate allocation for multi-view video plus depth," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*. IEEE, 2011, pp. 1–4.

[11] H. Yuan, S. Kwong, J. Liu, and J. Sun, "A novel distortion model and Lagrangian multiplier for depth maps coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 24, no. 3, pp. 443–451, 2014.

[12] J. Gautier, O. Le Meur, and C. Guillemot, "Efficient depth map compression based on lossless edge coding and diffusion," in *Picture Coding Symposium (PCS), 2012*. IEEE, 2012, pp. 81–84.

[13] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H. 264/MPEG-4 AVC standard," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, 2011.

[14] T. Maugey, A. Ortega, and P. Frossard, "Graph-based representation for multiview image geometry," *Image Processing, IEEE Transactions on*, vol. 24, no. 5, pp. 1573–1586, 2015.

[15] X. Su, T. Maugey, and C. Guillemot, "Graph-based representation for multiview images with complex camera configurations," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1554–1558.

[16] B. Martins and S. Forchhammer, "Lossy/lossless coding of bi-level images," in *Data Compression Conference, 1997. Dcc'97. Proceedings*. IEEE, 1997, p. 454.

[17] I. Daribo, D. Florencio, and G. Cheung, "Arbitrarily shaped motion prediction for depth video compression using arithmetic edge coding," *Image Processing, IEEE Transactions on*, vol. 23, no. 11, pp. 4696–4708, 2014.

[18] T. Maugey, Y. H. Chao, A. Gadde, A. Ortega, and P. Frossard, "Luminance coding in graph-based representation of multiview images," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 130–134.

[19] M. Peris, A. Maki, S. Martull, Y. Ohkawa, and K. Fukui, "New Tsukuba Stereo Dataset." [Online]. Available: http://cvlab-home.blogspot.fr/2012/05/h2fecha-2581457116665894170-displaynone.html

[20] S. M. Muddala, "Free View Rendering for 3D Video," Ph.D. dissertation, Mid Sweden University, 2015.

[21] M. Peris, A. Maki, S. Martull, Y. Ohkawa, and K. Fukui, "Towards a simulation driven stereo vision system," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 1038–1042.

[22] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM Transactions on Graphics (TOG)*, vol. 23. ACM, 2004, pp. 600–608.

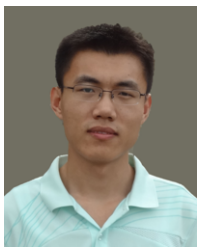
[23] T. Saito, "Nagoya University Multi-view Sequences." [Online]. Available: http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/

[24] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012.

[25] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004, pp. 93–104.

[26] B. T. Oh, J. Lee, and D.-s. Park, "Depth map coding based on synthesized view distortion function," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1344–1352, 2011.

[27] P. Carballeira, J. Cabrera, F. Jaureguizar, and N. García, "Analysis of the depth-shift distortion as an estimator for view synthesis distortion," *Signal Processing: Image Communication*, vol. 41, pp. 128–143, 2016.



Xin Su received the B.S. degree in Electronic Engineering from Wuhan University, Wuhan, China, in 2008 and the Ph.D. degree in image and signal processing from Télécom ParisTech, Paris, France, in 2015. He is currently a Post-Doctoral Researcher with team SIROCCO at Institut National de Recherche en Informatique et en Automatique, Rennes, France. His research interests include multiview coding, 3-D video communication, image rendering, image processing and computer vision.

Thomas Maugey (S'09 - M'11) received the Engineering Degree from Ecole Supérieure d'Electricité, Supélec, Gif-sur-Yvette, France, in 2007; the M.Sc. degree in fundamental and applied mathematics from Supélec, and Université Paul VerlaineMetz, Metz, France, in 2007; and the Ph.D. degree in image and signal processing from Télécom ParisTech, Paris, France, in 2010.

He was a Post-Doctoral Researcher with the Signal Processing Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland, from 2010 to 2014. He is currently a Research Scientist with the team SIROCCO at Institut National de Recherche en Informatique et en Automatique, Rennes, France. His research interests include monoview and multiview distributed video coding, 3-D video communication, data representation, video compression, network coding, and view synthesis.

Christine Guillemot (F'13) is currently Director of Research at INRIA (Institut National de Recherche en Informatique et Automatique) in France. She holds a PhD degree from ENST (Ecole Nationale Supérieure des Telecommunications) Paris (1992). From 1985 to 1997, she has been with France Télécom in the areas of image and video compression for multimedia and digital television. From 1990 to mid 1991, she has worked as visiting scientist at Bellcore Bell Communication research) in the USA. Her research interests are signal and image processing, and in particular 2D and 3D image and video coding, joint source and channel coding for video transmission over the Internet and over wireless networks, and distributed source coding.

She is currently senior Area Editor for IEEE Trans. on Image Processing (since Feb. 2016), associate editor of the International journal on mathematical imaging and vision (JMIV, since 2014), and member of the IEEE Trans. on Multimedia steering committee, as SPS (signal processing society) representative (nomination in Jan. 2016). She has served as associate editor for IEEE Trans. on Image Processing (2000-2003, 2014-2015), associate editor for IEEE Trans. on Circuits and Systems for Video Technology (from 2004 to 2006), associate editor for IEEE Trans. on Signal Processing (2007-2009), associate editor for the EURASIP journal on image communication (2010-2016), and senior board member of the IEEE journal on selected topics in signal processing (2013-2015). She has served on the IEEE IMDSP - International Multidimensional Digital Signal Processing - technical committee (2002-2007) as well as of the IEEE MMSP - International Multimedia Signal Processing - technical committee (2005-2008). She is currently a member of the IEEE IVMSP - Image Video Multimedia Signal Processing - technical committee (since 2013). She has been (and is) the member of technical programme committees of a number of international conferences. She has been general co-chair of the Picture Coding Symposium (PCS) in 2003, technical co-chair of the packet video workshop in 2003, special session co-chair at IEEE ICME conference in Hannover, in May 2008, general co-chair of the IEEE Multimedia Signal Processing (MMSP) workshop (Saint Malo, Oct. 2010), technical program co-chair of the packet video workshop (Munchen, May 2012), keynote chair of the organizing committee of the IEEE International Conference on Image Processing (ICIP) (Paris, 2014), general co-chair of the IEEE Multimedia Signal Processing (MMSP) workshop, Montreal, Sept. 2016, technical program co-chair of the IEEE Image, Video and Multidimensional Signal Processing (IVMSP) workshop, on the theme Manifold-based approaches for image and video processing, Bordeaux, July 2016, tutorial chair of the IEEE International Conf. on Image Processing (ICIP), to be held in Athens, in 2018. She is vice-chair of Inria's evaluation committee (since Sept. 2015). She has been promoted to the grade of "Chevalier de la legion d'Honneur" by decret of the president of French Republic (April 2, 2010), and IEEE fellow in January 2013. She has received a Google faculty award in 2015, and has received an ERC advanced grant for a project on computational light field imaging (CLIM) which has started on Sept. 1st 2016.