Visual data compression: beyond conventional approaches

Thomas Maugey

ANNÉE 2021-2022

UNIVERSITÉ DE RENNES 1

HDR / UNIVERSITÉ DE RENNES 1
Domaine : Signal, Image, Vision
Ecole doctorale MathSTIC

présentée par

Thomas Maugey

préparée au centre de recherche de
Inria Rennes-Bretagne Atlantique

Visual data compression: beyond conventional approaches

soutenue à Rennes
le 27 Juin 2022

devant le jury composé de :

Frédéric DUFAUX
DR, CNRS, CentraleSupelec, Univ. Paris-Saclay / rapporteur
Enrico MAGLI
Prof., Politecnico di Torino / rapporteur
Marta MRAK
Prof., Queen Mary University of London & BBC / rapportrice
Marc ANTONINI
DR, CNRS, Université Côte d'Azur / examinateur
Luce MORIN
Prof., INSA / examinatrice
Dong TIAN
Senior Scientist, InterDigital / examinateur

"Je voudrais être un arbre, boire à l'eau des orages
Pour nourrir la terre, être ami des oiseaux
Et puis avoir la tête si haut dans les nuages
Pour qu'aucun homme ne puisse y planter un drapeau"

Renaud

First of all, I would like to sincerely thank all the jury members for assessing my work. For their time and for their very interesting feedbacks.

Thanks for those who attended in presence or remotely my defense.

I would like to thank all the permanent researchers of the SIROCCO team (Christine, Aline, Olivier, Laurent) for their warm welcoming, their advises and the nice collaborations we have.

I would like to give my deep consideration to Mira and Navid, the two first PhD students I supervised. I feel really lucky that I was able to start my supervision experience with you. Thank you for your great great work, for your trust and for all what you taught to me.

I would like to also thank the students/engineer/postdocs I have supervised and those I am currently supervising. Thanks also for your trust and for all the energy and knowledge you bring to me and to the team. Thanks to the whole SIROCCO team for the very good research and work atmosphere that you are creating everyday. More generally, thanks to all my collaborators, for their trust and support.

I would like to have a deep thank for all the people who support our research: Huguette and Caroline, our assistant. Also to all the service info, the service gén'eral, service achat, the HR, the service com etc. that are always here to assist us.

Thanks to the running team of Irisa who gives me beautiful breaths several times a week. Thanks for your attention and support.

I would like to end by warmly thanking my friends and family. Some were coming from far away, it is priceless for me. Thanks for your support and understanding.

Last but not least, a particular thanks to Charlotte, and my three boys for your love and the joy you give to me everyday.

Contents

1 Introduction **9**
- 1.1 Basics of visual data compression . . . 9
  - 1.1.1 Block scheduler . . . 9
  - 1.1.2 Prediction . . . 10
  - 1.1.3 Transform . . . 11
  - 1.1.4 Quantization . . . 12
  - 1.1.5 Entropy coder . . . 12
- 1.2 Raising of new modalities . . . 12
  - 1.2.1 3D image definition . . . 13
  - 1.2.2 Point clouds and meshes . . . 13
  - 1.2.3 Omnidirectional images . . . 15
  - 1.2.4 Light field images . . . 15
  - 1.2.5 Stereo / multi-view images . . . 16
- 1.3 Limitations . . . 16
- 1.4 Contributions . . . 17

2 Visual data compression with random access at the user's side **19**
- 2.1 What is "Compression with random access" ? . . . 19
- 2.2 Segment the input data for navigation . . . 20
  - 2.2.1 Navigation domain partitioning . . . 21
  - 2.2.2 Segment representation . . . 22
  - 2.2.3 Optimal partitioning and results . . . 24
- 2.3 What achievable performances ? How to reach them ? . . . 26
  - 2.3.1 Modeling Random Access by means of a navigation graph . . . 27
  - 2.3.2 Optimal achievable performance . . . 28
  - 2.3.3 Practical scheme and experiments . . . 30
- 2.4 Practical interactive video coder for omnidirectional images . . . 34
  - 2.4.1 Proposed scheme's overview . . . 35
  - 2.4.2 Experimental comparison . . . 36
  - 2.4.3 Extension to 3D mesh texture coding . . . 36
- 2.5 Conclusion . . . 38

3 Graph construction: exploiting the geometry of 3D images **39**
- 3.1 Graph construction problem . . . 39
- 3.2 "Closer is more correlated" . . . 40
  - 3.2.1 Nearest neighbor for 3D data . . . 41
  - 3.2.2 Far/Near model for Light field images . . . 42
  - 3.2.3 Geodesic distance for 360° images . . . 43
- 3.3 Transmitting the graph instead of the geometry . . . 45

3.3.1 Graph-based representation (GBR) . . . 45
3.3.2 Retrieving the geometry from the graph . . . 47
3.3.3 Color compression using GBR . . . 48
3.3.4 Extensions . . . 49
3.4 Conclusion . . . 51

4 Graph-based transform for high-dimensional data **53**
4.1 Graph-based transform and complexity issue . . . 53
4.2 Graph segmentation . . . 54
4.2.1 Motivations and problem . . . 54
4.2.2 Super-ray segmentation . . . 54
4.2.3 Rate-distortion optimized segmentation . . . 56
4.3 Separable transform . . . 57
4.3.1 Definition . . . 57
4.3.2 Separable Laplacian . . . 58
4.3.3 Dimension factorization . . . 58
4.3.4 Application to Light field compression . . . 61
4.4 Graph reduction . . . 62
4.4.1 Motivations . . . 62
4.4.2 Graph coarsening principles . . . 63
4.4.3 Application to Light Field compression . . . 64
4.5 Conclusion . . . 65

5 Conclusion and Perspectives **67**
5.1 Disseminate the work on interactive coding . . . 67
5.2 Multi-view 360 view synthesis . . . 68
5.3 Learning on the sphere . . . 69
5.4 Coding for Machines . . . 70
5.5 Data Repurposing . . . 71

Chapter 1

Introduction

1.1 Basics of visual data compression

Despite the great disparity between the visual data formats, the coding/decoding pipeline adopted for their compression generally follows the same architecture (depicted in Figure 1.1). In a nutshell, the T consecutive input images $\mathbf{I}_t$ are generally split into blocks. Each block $\mathbf{x}_k$ is first predicted, with a prediction function f taking as inputs some previously encoded/decoded blocks $\{\tilde{\mathbf{x}}_l\}_{l<k}$. The residue $\mathbf{z}_k = \mathbf{x}_k - f(\{\tilde{\mathbf{x}}_l\}_{l<k})$ is then transformed and quantized. An entropy coder is finally used to build a compact bitstream $\mathbf{b}_k$. Let us now explain role of each of these steps and the motivation behind them.

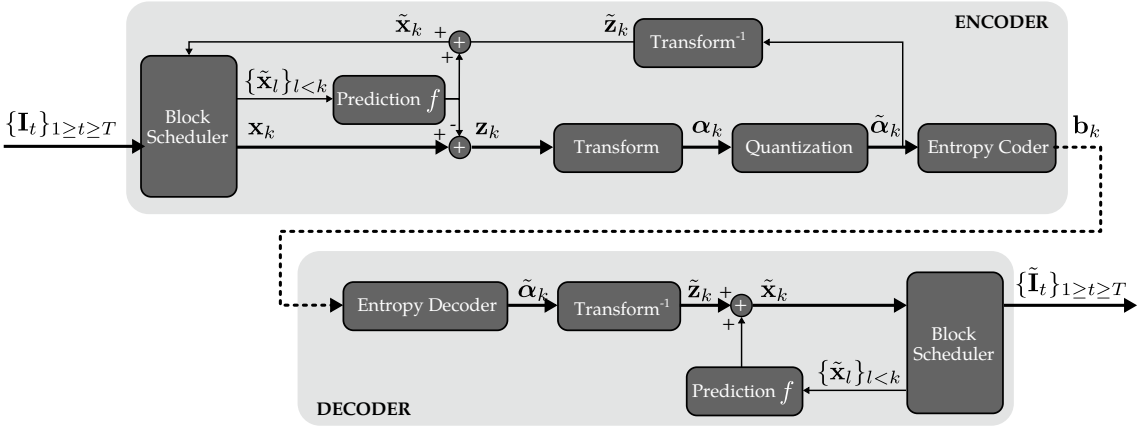


Figure 1.1: Structure of a conventional video encoder/decoder. The performance are measured by both the rate $R = \sum_k |\mathbf{b}_k|$ and the distortion $D = \sum_k ||\mathbf{x}_k - \tilde{\mathbf{x}}_k||_2^2$.

1.1.1 Block scheduler

The block scheduler consists in spatially dividing the input image/video into smaller entities called *blocks*. As depicted in Figure 1.2, the blocks can be 2D squares/rectangles for regular images or videos, 2D curved patches for omnidirectional images or videos or even 3D cubes for meshes. Performing such a spatial domain partitioning is justified by two main reasons: complexity and non-stationarity of the color signal.

The most obvious reason for patching the visual data is that the processing can be subdivided into smaller tasks. Indeed, compression tools such as prediction and transform

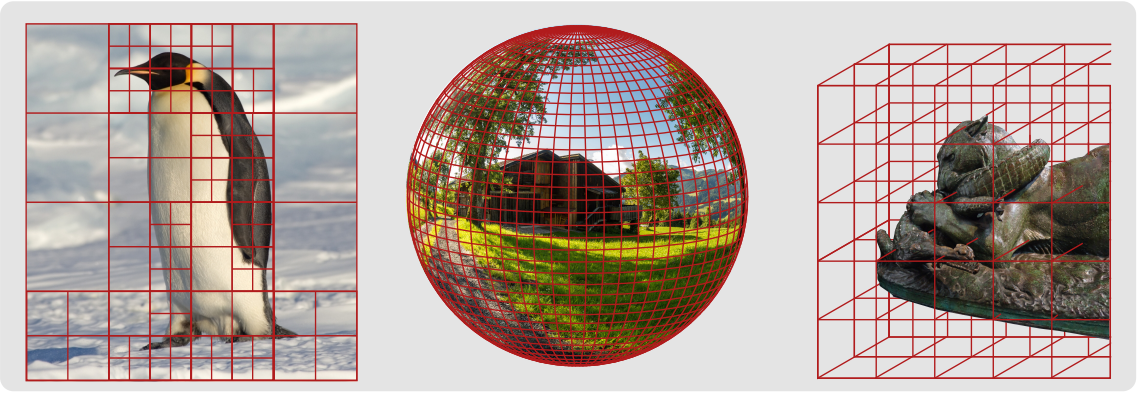


Figure 1.2: Depending on the visual data format, the blocks can have different shapes. They can delimit 2D or 3D areas. The can have a constant or optimized size.

can be very complex, and they usually scale with the signal's dimension. On top of limiting the complexity expense, defining fixed or predefined patch size (typically 8×8 to 64×64 pixels for 2D images/videos) enables to implement the compression tools (such as transform or quantization) once for all and in an optimal manner in the hardware. Thus when dealing with multiple visual data resolutions, the same optimized tool can be applied by just adapting the number of patches.

Even though complexity is an important justification for a blockwise processing of the visual data, the most important reason is the non-stationarity of the color signal. In a nutshell, compression tools aim at tracking the signal redundancies and remove them. Since, pixels are usually correlated with their close neighborhood, tracking the redundancy should naturally be a local operation, *i.e.*, within a patch.

In the recent 2D video standards, this block splitting is a key element for achieving huge coding gains [1]. Powerful optimizations are run such that the size of each block suits exactly the local statistics of the image.

1.1.2 Prediction

Even though pixels are usually more correlated with their direct neighborhood, longer-term correlations exist and cannot be removed if only intra-block operations are performed. Prediction step tackles this issues by simply constructing an estimation of a block $\mathbf{x}_k$ based on other decoded blocks $\{\tilde{\mathbf{x}}_l\}_{l<k}$.

A first category of prediction function f models the motion and spatially displaces the image's content accordingly. In that case, the $\{\tilde{\mathbf{x}}_l\}_{l<k}$ belong to a previous frame $\mathbf{I}_{t'}$ with $t' < t$. The function f consists in copying one patch belonging to the previous frame (*i.e.*, $f(\mathbf{x}_k) \in \bigcup_{l<k} \tilde{\mathbf{x}}_l$). The position of this patch is signaled to the decoder.

A second category of prediction function f performs texture extrapolation by propagating the information belonging to the already decoded neighbor information into $\mathbf{x}_k$ through an optimal direction. This direction is transmitted to the decoder.

Alternative prediction tools exist, for example those dealing with the recent advances in deep learning [2, 3]. They still take as input the pastly decoded blocks $\{\tilde{\mathbf{x}}_l\}_{l<k}$ and produces a prediction of $\mathbf{x}_k$.

1.1.3 Transform

The transform operation is nothing else than an orthonormal basis change that enables to i) decorrelate the signal and ii) to compact the energy in a small number of coefficients. Thanks to i) and ii), coding the signal in the transformed domain is more efficient than in the spatial domain. The signal to transform is the residue $\mathbf{z}_k$. If no prediction is done, then $\mathbf{z}_k = \mathbf{x}_k$.

i) *Decorrelate the signal*: the signal $\mathbf{z}_k$ can be seen as a vector of N random variables Z_i, that are correlated with each other. For the sake of clarity, let us assume for the moment that one has to code this vector losslessly. The minimum achievable transmission rate is the joint entropy $H(Z_1, \ldots, Z_N)$ (in the sense of Shannon entropy [4]). However, this rate is difficult to achieve in practice because the joint probability distribution of $(Z_1, \ldots, Z_N)$ is not straightforwardly obtainable. Let us now assume that the signal has been transformed to another random vector set $(Z'_1, \ldots, Z'_N)$ thanks to an orthogonal basis change. The achievable rate remains the same, *i.e.*, $H(Z'_1, \ldots, Z'_N) = H(Z_1, \ldots, Z_N)$. However, if the variable decorrelation is achieved, we have that $H(Z'_1, \ldots, Z'_N) = H(Z'_1) + \ldots + H(Z'_N)$, which means that each variable can be coded separately without any rate increase. In that case the joint probability distribution is not needed, and the optimal rate becomes easily achievable in practice, just requiring each of the Z'_i distributions.

ii) *Compact the energy*: another interest of the transform is to compact the energy in a small number of coefficients. Since the transform is orthonormal, then the amount of signal energy is the same in both signal and transform domains according to Parseval's theorem. While, in the original domain, the energy in the signal tends to be distributed relatively evenly over the nodes, with a compact representation (*i.e.*, compact transform), the frequency coefficients do not contain the same amount of energy as illustrated in Figure 1.3. In other words, some coefficients are more representative of the signal and have to be coded with a greater precision. On the contrary, some coefficients are less representative and can be coded coarsely or even can be withdrawn without a great impact on the decoded signal quality, but with a significant bitrate decrease. This is controlled by the quantization as we will explain in the next paragraph. It is important that the energy is contained in the low frequencies (*e.g.*, power-law decay as in the general case) or otherwise known law, which may lead to a good model (*e.g.*, zerotree [5]). The exact indices may not be exactly known in advance but still coded efficiently using SPIHT [6] or similar.

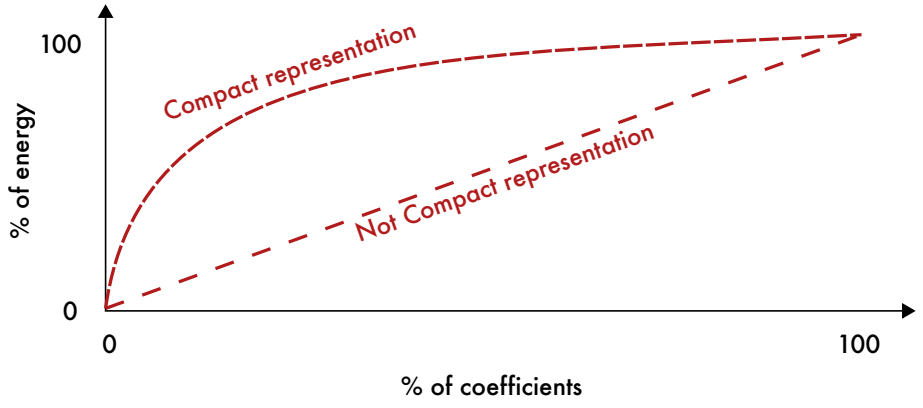


Figure 1.3: Behavior of a compact and a not compact representation.

Finding the optimal transform is possible when the data's covariance matrix $\mathbf{\Sigma}$ is known. In that case, the transform basis, called the Karhunen-Loeve Transform (KLT), is chosen

such that it diagonalizes $\mathbf{\Sigma}$ [7]. This approach is however almost never used in compression because i) $\mathbf{\Sigma}$ has to be known and ii) the transform basis has to be transmitted to. In practical coder, hand-crafted transform are used such as the popular Discrete Cosine Transform (DCT) that can be designed for signals of every dimension (*e.g.*, [8], 2D [9], 3D [10], 4D [11]).

1.1.4 Quantization

As explained before, the output of the signal transform stage is a set of transform coefficients. They are most of the time decorrelated. Also, a large proportion of the total signal energy is contained in a handful of coefficients. From a compression perspective, the transform coefficients cannot be sent as they are. The number of bits needed to represent float or double values (with high precision) can easily explode, without necessarily improving the signal reconstruction quality. Therefore, restrictions on the number of bits used to represent those coefficients is necessary. This is usually done by a scaling and rounding procedure, what we call a scalar uniform quantization:

$$\tilde{\boldsymbol{\alpha}} = Q \times \text{round}(\frac{\boldsymbol{\alpha}}{Q}). \tag{1.1}$$

Depending on the quantization step size Q, the transform coefficients are rounded to the nearest multiple of Q. With such procedure, the precision of the sent transform coefficients $\tilde{\boldsymbol{\alpha}}$ can be varied and the reconstruction quality (*i.e.*, the distortion of the signal reconstructed) is impacted.

Different ways exist to define Q. The easiest way is to choose one fixed quantization step for all coefficients. However, since the energy is concentrated in the first coefficients, standards have decided to group coefficients with respect to their energy (equivalent to grouping random variables with a similar distribution), and thus define different Q values per group of coefficients. For example, for the coding of 2D images, matrices of quantization steps are usually precoded and scaled to vary the bitrate [12, 13].

1.1.5 Entropy coder

In all compression schemes, the quantized coefficients are further compressed in a lossless manner, thanks to entropy coding. This enables to exploit the probability distribution to decrease the number of bits needed to code the quantized values. This entropy coding can be done using, simple coders (*e.g.*, arithmetic [14], Huffman [15]) or more evolved ones (*e.g.*, CABAC [16, 17]).

1.2 Raising of new modalities

In the recent years, in parallel of the development of more and more powerful compression tools, new types of image modalities have emerged. Most of them were developed in the context of immersive experience such as virtual/augmented reality, 6 degree-of-freedom (6-DoF) visualization. These images tend to represent the 3D world and are therefore referred to as 3D images [18]. It implies that the dimension of the images captured with such devices is huge, and requires efficient compression algorithms.

1.2.1 3D image definition

3D images are classically defined in opposition to 2D images. Indeed, by "2D images", we usually denote the images acquired with a traditional pinhole camera under a perspective projection model, *i.e.,* everyday-life camera. By "3D" we usually consider all what is not "2D", hence including many types of capturing devices. The word "3D" is general and does not always mean that the image is defined in an $\mathbb{R}^3$ topology. However, it always reflects the fact that an extra information (implicit or explicit) exists and can be used to assist the image processing task. This information corresponds to the scene *geometry*, called γ in the following. We now review the most common types of 3D images from the most explicit to the most implicit geometry.

1.2.2 Point clouds and meshes

A point cloud refers to a set of data points in space (Figure 1.4). Each point is defined by a 3D coordinate and a color value. Point clouds are usually captured with one or several laser device(s) (such as lidar, time-of-flight cameras) coupled with classical image camera(s). They are used when remote visualization is desired (medical imaging, virtual tour, virtual reality, etc.). For realistic rendering, a high number of points is used (several millions), increasing the need for efficient compression algorithms. Point clouds are sometimes converted to polygon or triangular meshes, in which a mesh topology is added which describes the connectivity between 3D points. The color information can be represented in a huge color vector [19, 20] or texture maps [21] that are 2D projection of the 3D object. They enable the use of traditional coding tools. Both point clouds and 3D meshes can be static or dynamic. In case of dynamic content (sequence of frames), the number of points and the topology may vary between frames. The geometry of a point cloud is said *explicit* since it directly corresponds to the set of 3D coordinates. The compression of this *geometry* data is based on decorrelation's principle to eliminate the statistical redundancy.

The compression of 3D mesh *geometry* has been widely studied. A survey of such methods can be found in [22, 23]. In general, static mesh compression approaches are divided into three categories: single-rate algorithms try to build a compact representation of an input mesh. In progressive algorithms, as in [24, 25, 26], the input mesh is iteratively decimated until a base mesh is generated. This provides successive levels of details for the input mesh in which a coarse version of the mesh can be quickly displayed to the user and this coarse mesh is progressively refined as more data are decoded. Random-accessible algorithms like [27] allow decompressing only the required parts of the mesh to avoid the need to load and decompress the full model.

In contrast with 3D meshes, the lack of connectivity information in point clouds is the main difficulty to overcome [28]. In this case, various types of tree-based representations are considered to process geometry information, including octrees [29], binary trees [30] and kd-trees [31]. Different approaches are used to compress the geometry, including methods to decompose the mesh into Levels of Details [32, 33], clustering methods [34] and transform-based techniques [35]. A survey regarding the compression of point clouds is provided in [36]. The approaches used to decorrelate the point cloud data are classified into three main families: 1) 1D traversal compression techniques that provide 1D prediction using tree-based connectivity induced by the native geometric distances between the points in the cloud [33]. 2) 2D projection-based methods map the 3D point cloud into 2D images/videos and then use existing image/video coding techniques to compress the data [32]. 3) 3D decorrelation techniques directly exploit the 3D correlation [35].

Figure 1.4: Example of a 3D point cloud: *Stanford Bunny*.

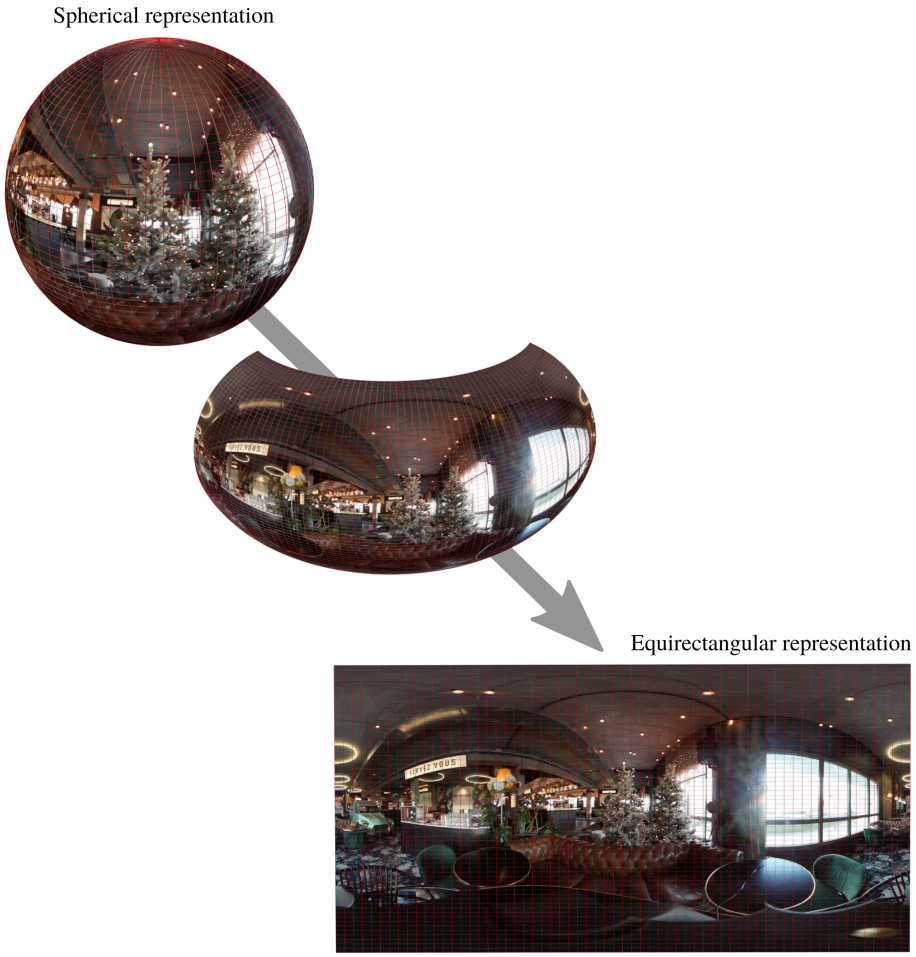


Figure 1.5: Example of omnidirectional image: *Sopha* [37]

1.2.3 Omnidirectional images

Omnidirectional images, also called 360° images, describe the visual information coming from *every direction* and converging to a point (the camera center). They can also be seen as spherical images, in which pixels are lying on a unitary sphere, and their position on the sphere corresponds to a light ray direction. Here, the geometry is still explicit since it corresponds to the position on the sphere. It is however of only 2 dimensions (longitude and latitude), being only able to position the color point on a line (rather than at an exact 3D position for a point cloud).

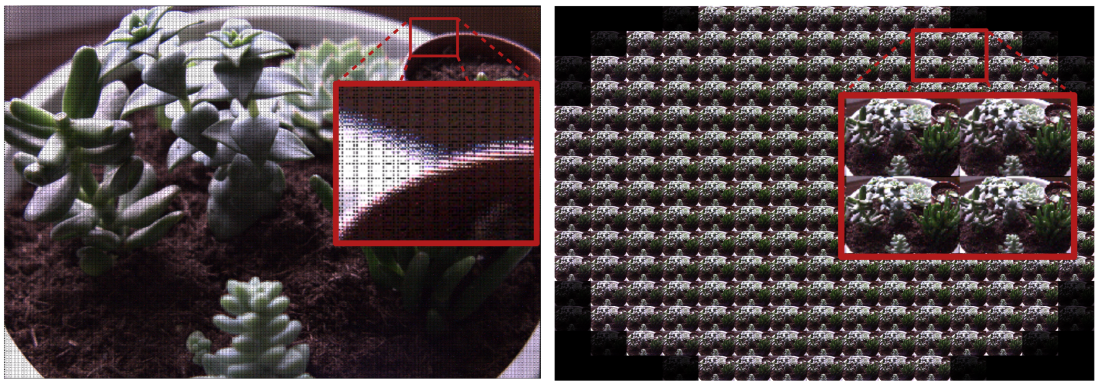
Omnidirectional images are generally mapped to 2D images to enable traditional image compression (Figure 1.5). Different mappings exist: equirectangular [38], cube map [39], rhombic dodecahedron [40], Dyadic [41], etc. Each of them presents certain drawbacks such as not non-uniform pixel's distribution on the sphere, connectivity loss between the difference mapping surfaces.

1.2.4 Light field images

Light Fields represent light rays emitted by every point in a scene and along different orientations [42]. It is described by the plenoptic function $L(x, y, z, \theta, \phi, \lambda, t)$, where x, y and z are the 3D coordinates, θ and ϕ are the direction angles, λ is the light frequency and t is the time instant. A light field camera, also known as plenoptic camera, is able to capture for a given position: i) the light color and ii) the light direction, contrary to regular cameras that only capture the light color. In other words, light field cameras provide a finer sampling of the plenoptic function. This has been made possible, for example, by placing a 2D array of micro-lenses in front of the photo-receptor. The image captured by the 2D sensor (called lenslet-based plenoptic image), is in fact seen as a 4D table of pixels where two dimensions correspond to the pixel position, and two correspond to the ray angle (Figure 1.6). The scene geometry information is, this time, *implicit* and can be retrieved and estimated from the plenoptic image content.

Estimating this geometry always relies on the *motion parallax effect*: when switching from one view to another, an object "moves" (*i.e.*, it is not at the same position in the images), and the way it moves is directly linked to its depth (*i.e.*, its distance to the cameras). More precisely, the closer the object, the faster it moves between the different images. The whole point of geometry estimation is to estimate this motion, called *disparity*, in order to retrieve the scene *depth*.

Having very narrow baselines (distance between the views), lenslet-based plenoptic images could not be efficiently used in stereo matching techniques as they usually involve interpolation with blurriness due to sub-pixel shifts in the spatial domain. Therefore, research has been devoted to find different constraints and cues for estimating the depth. One way is to compute cross-correlation between microlens images to estimate the disparity map [43]. Other approaches rely on structure tensors to estimate vertical and horizontal slopes in epipolar images [44]. Alleviating some ambiguities and difficulties encountered due to occlusions and large displacements, researchers have combined different cues like defocus and correspondence in [45], with occlusion handling [46]. More recently, learning-based methods are proposed [47, 48, 49] and show significant improvement when a sufficient amount of data is available for learning. Once a disparity map or a depth map is estimated, this geometry information needs to be compressed and transmitted. If a dense disparity map is needed, then it can be compressed using traditional image coding methods with adapted criterion [50, 51]. If only a sparse set of disparity or depth values are needed, they can be coded using arithmetic coding techniques.



Raw lenslet image Sub-aperture images

Figure 1.6: Example of Light field image: *Succulents* [52]

1.2.5 Stereo / multi-view images

Multi-view imaging refers to a synchronous capture of a scene taken from different angles. A particular example is the stereo capture where two points of view are captured mimicking the human visual system, usually for rendering a 3D impression at the user's side. Other systems such as super multi-view exist [53]. Since the captures are usually done with perspective cameras, no geometry information is available. However, exploiting the information from multiple points of view enables to retrieve it, again thanks to the motion parallax effect. Some other nice properties of the perspective projection, such as the so-called epipolar geometry, can be used to relate disparity and depth. Therefore, from multiple synchronous captures of a scene, the depth, and thus the 3D points position can be retrieved. This is why multi-view imaging is considered as "3D", the geometry being *implicit* and deduced from the pixel redundancies across views.

Geometry estimation and compression is close in spirit to the ones for light-field. However, with larger baselines (distance between the views), stereo matching techniques [54, 55, 56] are mostly used to estimate the geometry *i.e.* disparity. Those approaches can be broadly classified into two categories: the intensity-based matching and the feature-based matching techniques. Learning based approaches for deep stereo matching and depth estimation have also been proposed, (*e.g.*, StereoNet in [57, 58]).

1.3 Limitations

With the advent of the aforementioned images modalities, some major problems arise. First their size explodes, and this revolutionizes the way they are consumed. Second, they mostly rely on topologies that are irregular. We describe here, how such evolutions make the use of conventional compression suboptimal or even impossible.

Random Access - Given the new nature of the signals (omnidirectional, 3D, etc.) or their huge size, users are not anymore willing to watch (hence to receive) the entire image at once. While classical 2D images describe a single viewpoint, 3D images represent many of them, and a user cannot access all of them at the same time. This simple statement brings several major changes: first, a user should be given the possibility of *interactively* choose his/her viewpoint. Second, the coding scheme should be able to send only what is needed to the users, in order to save bandwidth. However, since no live re-encoding is conceivable in practice, the encoder should encode the image blindly to the user's request.

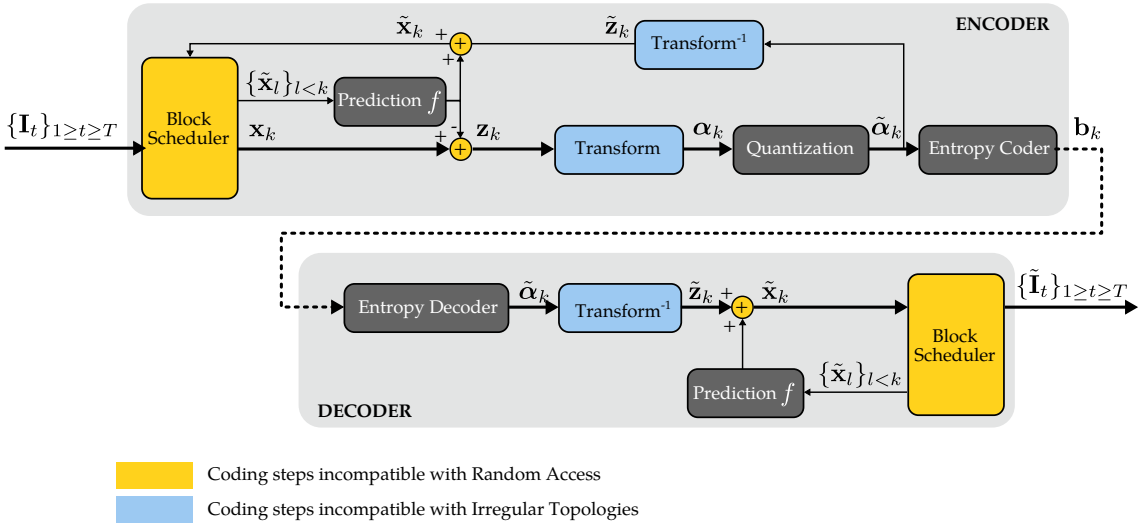


Figure 1.7: Limitations of conventional coding architecture.

This is why *interactivity* is also called *random access*. This randomness is precisely what makes the conventional compression approach suboptimal. As depicted in Figure 1.7, the predictive approach relies on *prediction and residue* and thus imposes that the coding order is the same at the encoder and the decoder. Unfortunately, the set of blocks requested by a user does not necessarily contain all the blocks in the predefined order. As a results, it is more likely that a significant number of non-displayed blocks have to be transmitted, which makes the conventional approach far from optimal.

Irregular topologies - Instead of the 2D cartesian grid used for classical image, the 3D data relies on sphere, arbitrary shaped surfaces, arbitrary shaped 4D volumes etc. None of these domains of definition is euclidean, which makes the usage of traditional signal processing tools *impossible*. In the conventional coding architecture, the transform is not defined anymore (see Figure 1.7). Indeed, the usual transforms adopted by conventional coders (such as DCT, Fourier Transform, Wavelets, etc.) are only defined on euclidean space.

1.4 Contributions

The work presented in this manuscript aims at tackling the two aforementioned issues.
★ In Chapter 2, we present solutions to perform image compression under random access. We first investigate how to segment the input data such that less useless information is transmitted while maintaining good compression performance. We then derive the optimal coding performance that can be expected from the theoretical point of view. We show, in particular, that the widely used segmentation-based approach is not optimal. Finally, inspired by such intuitions, we develop a practical solution aiming at achieving these optimal performance. We then build a practical coder for omnidirectional and 3D mesh data. We demonstrate that it is possible to send only what is requested without any loss of coding performance.
★ In Chapter 3, we present how we have tackled the non-regular topologies. We describe how the 3D data topology can be modeled with a graph. We then derive practical coding solutions for different modalities such as light field, omnidirectional images and 3D meshes. In a second time, we use this graph to describe the geometry information instead

of, for example, depth maps. We show that such representation embed the necessary "amount of geometry" for efficient compression.
★ In Chapter 4, we focus on the construction of graph-based transforms. Playing an important role in the coding schemes investigated in Chapter 3, they are at the same time, very complex, especially when the data resolution is increasing. We propose solutions to decrease this computational complexity. First, we propose to optimally segment the graph, enabling to reduce the transform calculation and at the same time to preserve the smoothness of the signal on the graph (and thus good compression performances). In a second time, we investigate how the graph-based transform can be designed such that it is computed in a separable manner along "graph dimensions". We arise problems of basis misalignment that may occur and propose solution to tackle them. Finally, we use graph coarsening techniques to represent the graph in a reduced dimension when the signal is sufficiently smooth.

Chapter 2

Visual data compression with random access at the user's side

2.1 What is "Compression with random access" ?

Compression with random access refers to the compression of data for which only a subpart is nedeed/desired at the decoder side (for visualization or processing). Typical examples of such data are visual data that cannot be watched entirely at a given time. Omnidirectional videos or 3D model (mesh or point clouds) fall into this category since, by nature, they represent the light field from different possible viewpoints (Section 1.2). Other examples are databases made of numerous items (*e.g.*, signals captured by sensors, images or videos) that a given user may partly consume.

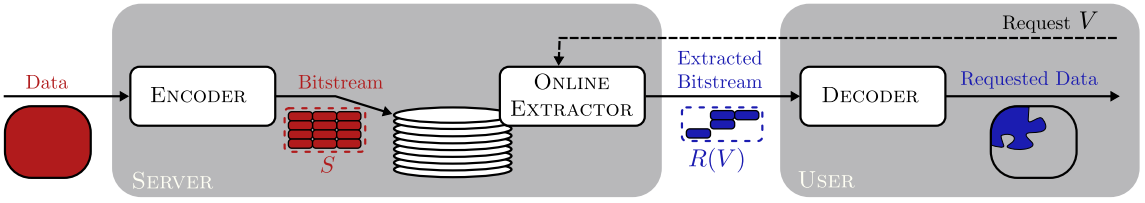


Figure 2.1: Compression with Random Access

The fact that i) the entire data must be represented and at the same time ii) only a part of it may be consumed by a user makes the compression highly challenging and novel. A first novelty is that it splits the traditional compression rate into two quantities:

- the **storage rate**, S, measuring the size of the file that is needed to represent the whole data. This file is usually stored on a server.
- the **transmission rate**, R, measuring the amount of information that is needed for the partial decoding at user's side.

A second novelty is that the transmission rate depends on the user's behavior and thus requires a modeling of it to be defined. For that purpose, we define:

- the user's **request**, V, as the subset of the data that is needed at user's side. This can be considered as a random variable with a set of realization $\mathcal{V}$ and distribution p_V.

If $R(V)$ is the transmission rate measured for a given user's request V, we can define the transmission rate as:

$$R = \mathbb{E}_{V \sim p_V}[R(V)]. \tag{2.1}$$

A third novelty raises together with this new rate evaluation. Indeed, this two-headed rate term implies a need for novel coding architecture. Naturally, a simple encoder-decoder architecture is no longer meaningful. Moreover it is not conceivable to perform re-encoding (after user's request) because of obvious complexity issues. Therefore, we define the following coding architecture pipeline:

- the **encoder** that compresses the entire input data and stores it as a bitstream
- the **online extractor** that, upon a user's request, extracts from the stored bitstream what is necessary.
- the **decoder** that reconstructs the desired data from the extracted bitstream.

The global coding scheme for compression with Random Access is depicted in Figure 2.1. The main challenges are the following:

★ Model the user's request V in order to estimate the transmission rate and to anticipate the user's behavior at the encoding

★ Design a Encoding-Extraction-Decoding strategy that optimizes the S and R rates.

2.2 Segment the input data for navigation

The traditional compression scheme introduced in Figure 1.1 imposes the prediction f to be the same at the encoder and the decoder (because of the residue computation). A huge consequence of that requirements is that the *block scheduler* should encode and decode the blocks of a visual data *with the same order*. This is naturally not compatible with the random access introduced in the previous section. As an illustrative example, if a user requests one block, that is predicted from another one, the extractor must send these two blocks to the decoder even though only one is required. In other words, the block scheduler introduces a dependency chain that cannot be broken at the extraction.

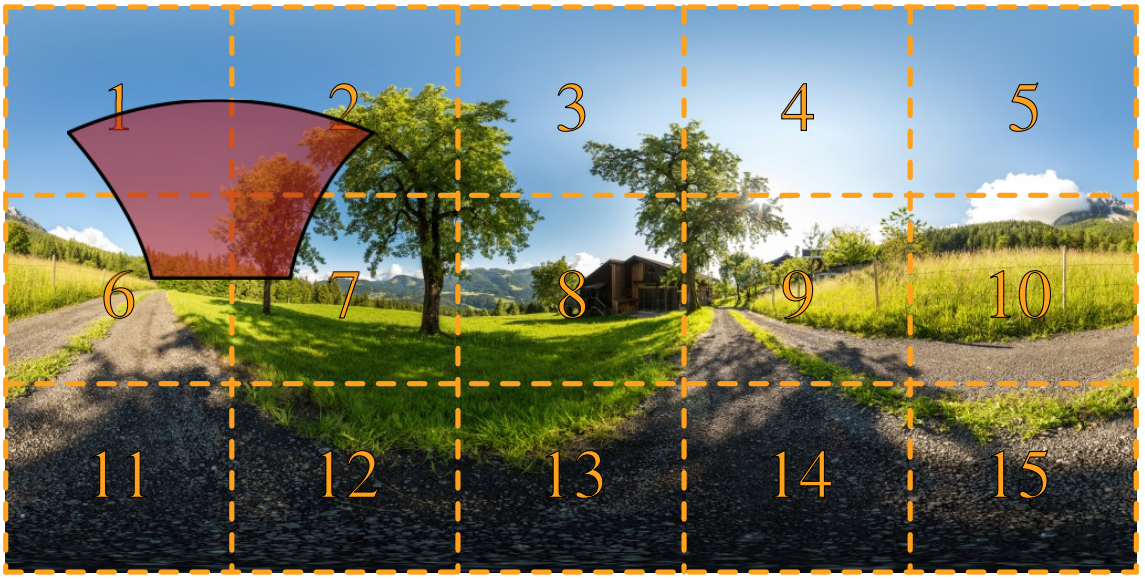


Figure 2.2: An omnidirectional image represented in equirectangular format and divided into 15 tiles. As an example, when the user's viewport corresponds to the red zone, the tiles 1, 2, 6 and 7 are transmitted instead of the whole image.

The historical way of tackling this drawback consists of segmenting the input data into small indivisible entities. The most straightforward example is the concept of *Group-of-Pictures* (*GoP*). It consists of a set of frames in a video that are coded together: one intra frame and several frames predicted from it (see (J23)). More recent instances are the *tiles* in omnidirectional videos [59, 60, 61]. Basically, the input omnidirectional image (or video) is split into spatial blocks (see Figure 2.2). Each tile is encoded separately and transmitted as soon as a subpart of it is requested by a user (*i.e.*, his viewport lies on its domain of definition). This enables to reduce the amount of useless (*i.e.*, not watched) pixels that are transmitted.

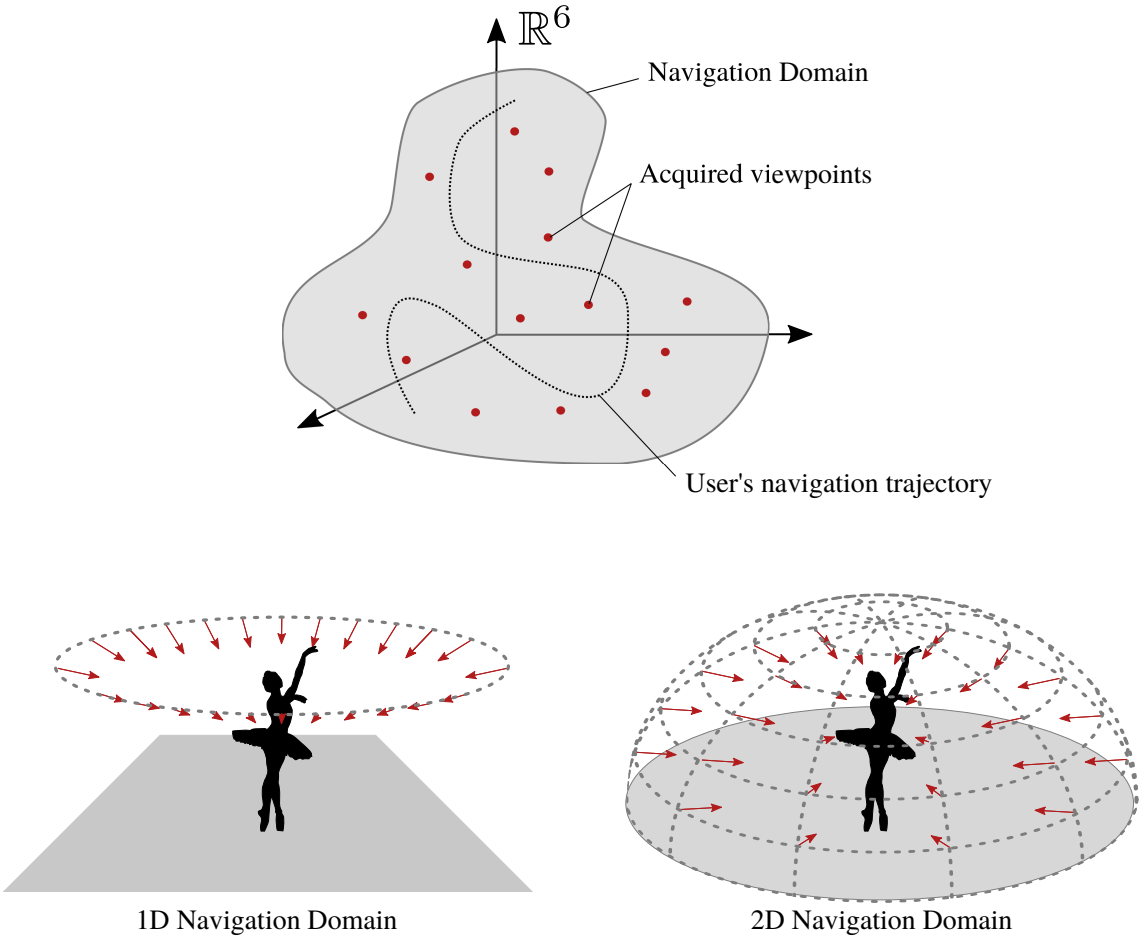


Figure 2.3: The navigation domain is a manifold included in $\mathbb{R}^6$, corresponding to the 6 degrees of freedom a user can play with during his navigation. The set of acquired views (in red) corresponds to a discrete set included in this navigation domain.

2.2.1 Navigation domain partitioning

In (J3) and (J18), we have extended this concept to the *navigation domain*, *i.e.*, the set of viewpoints that could be synthesized to a user. Let us assume a *free viewpoint viewing* (*FVV*) system where a user observes a scene from the viewpoint he desires. We define the *user's viewpoint* by the 6 parameters positioning the pinhole camera in the world domain: three positions for the camera center (t_x, t_y and t_z) and three angles (θ_x, θ_y and θ_z). The navigation domain is thus the manifold included in $\mathbb{R}^6$ that contains all viewpoints that

can be chosen by a user. The concept of navigation domain is illustrated in Figure 2.3 and depends on the application.

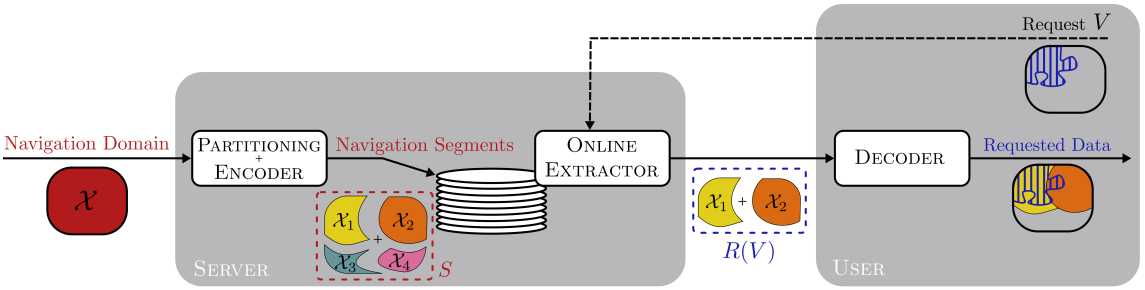


Figure 2.4: The navigation domain is partitioned into navigation segments, and each navigation segment is encoded and stored on a server. Users interact with the server to request the navigation segments needed for the navigation.

We denote by $\mathcal{X}$ the navigation domain and by X a given viewpoint in $\mathcal{X}$. We assume that a probability density fonction P exists on $\mathcal{X}$, where $P(X)$ gives the probability of X to be chosen by a user. We thus call $P(X)$ the *popularity* of a viewpoint X. As mentioned previously, we propose to partition $\mathcal{X}$ into *navigation segments* $\mathcal{X}_i$, in order to make it suitable for an interactive compression (see Figure 2.4). The partition is chosen such that it optimally minimizes the following criterion:

$$\min_{N,\mathcal{X}_1,\ldots,\mathcal{X}_N} \quad \sum_{i=1}^{N} R(\mathcal{X}_i) + \lambda S(\mathcal{X}_i), \tag{2.2}$$

where $S(\mathcal{X}_i)$ is the storage cost of a navigation segment, $R(\mathcal{X}_i)$ is the expected rate for the navigation segment and λ is a weighting factor balancing the importance of storage cost compared to transmission one. More precisely, the expected rate can be expressed as:

$$R(\mathcal{X}_i) = \left(\int_{X \in \mathcal{X}_i} P(\mathcal{X}_i) \right) S(\mathcal{X}_i). \tag{2.3}$$

We can see that the rate cost is equal to the storage cost (*i.e.*, the total amount of bits needed to describe a navigation segment) weighted by the popularity of the navigation segment. When minimized, this rate term enforces the partition to advantage low storage costs (*e.g.*, smaller segments) for more popular parts of the navigation domain.

2.2.2 Segment representation

In (C17, C18, C29, J3, J13), we propose to represent a navigation segment $\mathcal{X}_i$ with the two following quantities:

- the color and depth information of a *reference view* (denoted by Y_i), that is a well-chosen point of view belonging to the navigation segment. All the viewpoints X belonging to this navigation segment will be generated from Y_i.
- an *auxiliary information* φ_i that is a light helper for the synthesis of the views in the navigation segment.

The proposed representation is illustrated in Figure 2.5. Using view synthesis tools such as *Depth image based rendering* (*DIBR*) techniques [62, 63, 64], the reference view is able to synthesize a portion of each view of the navigation segment. This portion corresponds to the regions of the 3D scene that is visible from the reference view. The remaining region

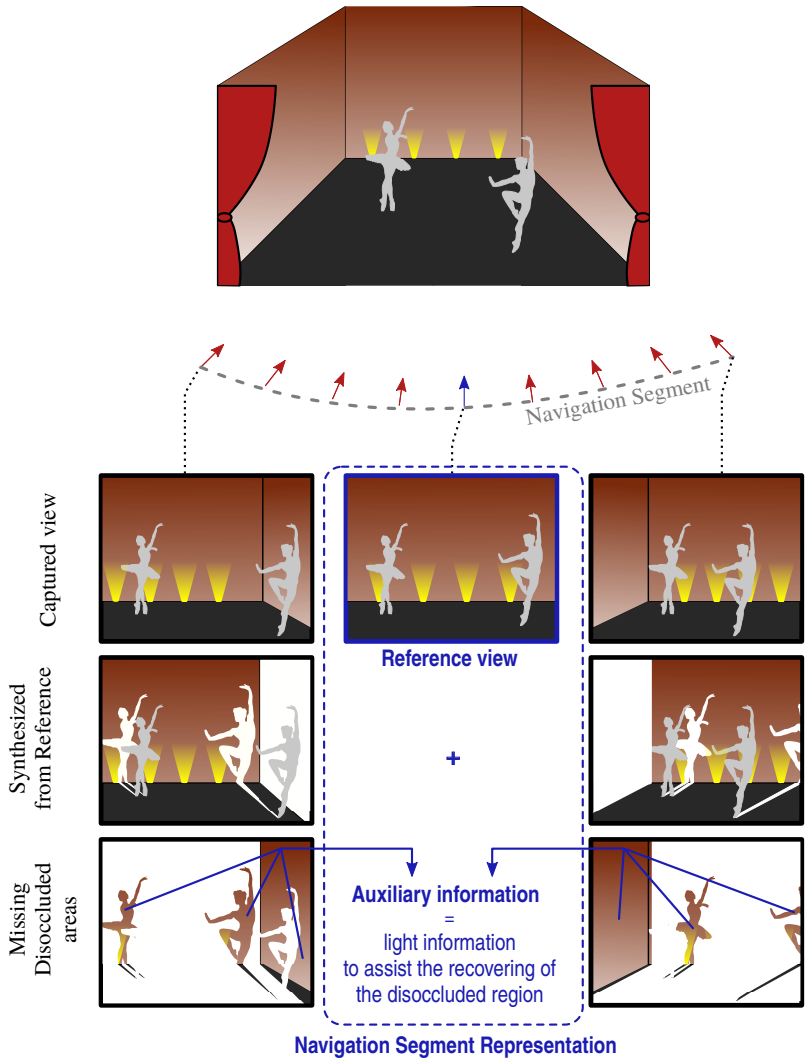


Figure 2.5: Proposed representation of a navigation segment: a reference view and an auxiliary information.

is called the *disoccluded* area and corresponds to the scene content that is hidden in the reference view.

These holes in the synthesized views can be filled in using inpainting algorithms [65]. However, inpainting targets *plausibility* of the content rather than *fidelity* to an input signal (since these algorithms are usually used when no ground-truth signal exists). The consequence is that with classical inpainting solutions, the synthesized views look realistic, but are most of the time significantly different from the original view. This can have huge consequences on the visual quality (*e.g.*, unconsistency over time or across the views). This is the reason why we have proposed to send an auxiliary information to assist the inpainting algorithm. More precisely it helps the inpainting to converge to a reconstruction that is plausible and at the same time close to the input view. This auxiliary information is not sent for every block to be filled in. In Figure 2.6, we show that the level of uncertainty when inpainting a hole strongly depends on the content and the occlusion shape. The auxiliary information should thus be designed in such a way that it guides the inpainting only when necessary.

In order to build an auxiliary information, we have explored different solutions. In

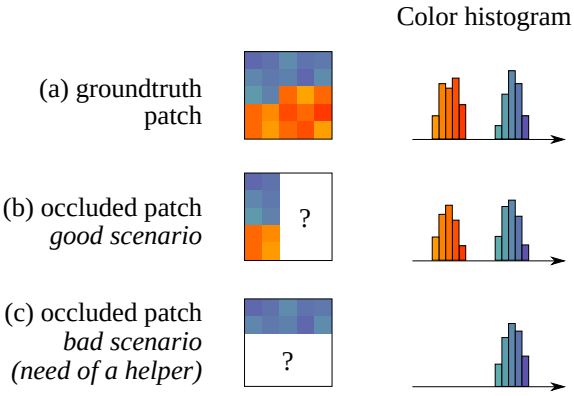


Figure 2.6: Illustration of different incertainty levels when inpainting occluded blocks. Completing the occluded patch in (b) is intuitive, while guessing that an orange region exists only relying on the blue region in (c) is nearly impossible without an helper.

(C17,J3), the side information consists of DCT coefficients of some well-chosen blocks. In (C18, J13), the side information is made of intra/inter mode values (*i.e.*, the type of mode and the prediction parameter). The choice of where to position this auxiliary information in the image is made by solving a rate-distortion criterion. Finally, in (C29), we build a texture dictionary made of all the patches of the known region (the non-occluded one). We cluster this dictionary in sub-dictionaries. The auxiliary information is simply the index of the cluster when this one cannot be deduced from the context when doing inpainting. Said differently, the auxiliary information helps the inpainting to choose the "mode" of the inpainting (*i.e.*, the type of content) and let it fill in the disocclusion.

In (J8), we have alternatively proposed to describe the navigation segment $\mathcal{X}_i$ using a *Layered Depth Image*. As explored in [66] for a different context, it consists in projecting all views $X \in \mathcal{X}_i$ onto the reference image Y_i (using DIBR). Naturally, some pixels of X are already present in Y_i and are removed to avoid redundancy. Other pixels can be hidden in Y_i (because of occlusion) and are represented in another layer. Some pixels are out of the initial Y_i boundary, and should be represented in an extended version of Y_i. When all pixels are projected, the resulting image is an extended layered depth image.

Whatever the solution adopted (auxiliary information or layered depth image), the purpose is to represent compactly (without any redundancy) the information necessary to synthesize any $X \in \mathcal{X}_i$. The remaining question is how to decide the boundaries of the different navigation segments ? This is tackled in the next section.

2.2.3 Optimal partitioning and results

In this Section, we explain how we have proposed to solve the problem formalized in (2.2), when the navigation segments $\mathcal{X}_i$ are coded with one reference view Y_i and an auxiliary information φ_i. The rate of the reference view Y_i is considered as constant, meaning that coding a view costs approximately always the same rate and does not depend on the chosen view. The rate of the auxiliary information is modeled as proportional to what we call the *innovation* between one view and another. This innovation between X and Y_i is the size of the disoccluded region when predicting X from Y_i (as in Figure 2.5), *i.e.*, the number of pixels of X that are occluded in Y_i. This innovation can also be computed between one reference view Y_i and a set of views (for example the remaining views in the navigation segment $\mathcal{X}_i \setminus Y_i$). In that case, the innovation is computed in the 3D world (number of

disoccluded voxels instead of number of disoccluded pixels) and the disoccluded area is the union of the disoccluded voxels corresponding to each view. This model is content dependent in the sense that it does not only rely on the distance between cameras. We see in Figure 2.7 that the rate of the auxiliary information is indeed proportional to the innovation.

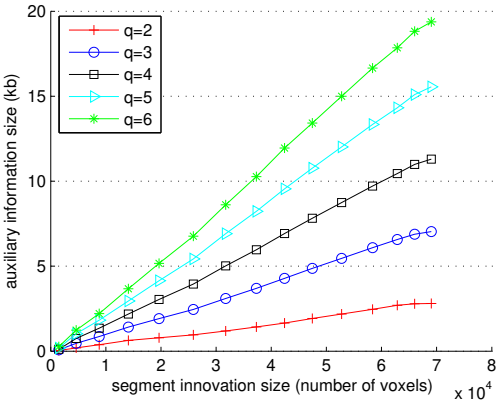


Figure 2.7: Illustration of the evolution of the size of auxiliary information $R(\varphi_i)$ as a function of the number of voxels in the segment innovation. The auxiliary information is coded with a DCT-based scheme with uniform quantization of the coefficients, where q corresponds to the number of bits used to describe each DCT coefficient.

Based on the aforementioned model, the problem in (2.2) has been solved using an extension of Lyold algorithm in (J3) or a Dijkstra algorithm in (J18). In Figure 2.8, we show some typical results we have obtained. We can see that our approach that takes into account the scene geometry obtains a better rate performance (for a same decoding quality) than a solution only relying on the camera poses.

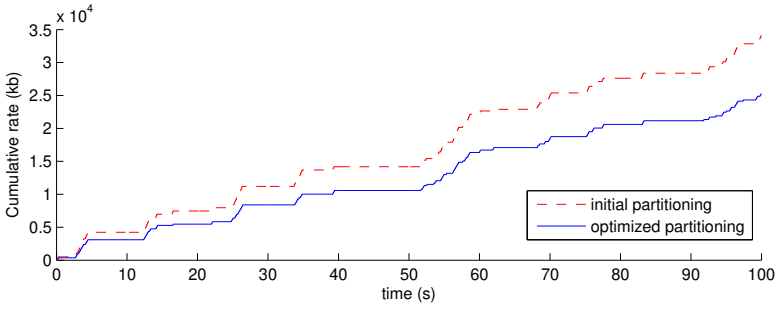


Figure 2.8: Cumulative rate computation during a user's navigation. Both navigation domain partitioning are compared: the *initial* one based on the camera distance only, and the optimized one, for which our method is adopted to adapt the size of the navigation segments to the scene content.

(**C17**) T. Maugey, P. Frossard, G. Cheung *Consistent view synthesis in interactive multiview imaging* In international Packet Video Workshop, Orlando, USA, Sep 2012

(**C18**) I. Daribo, T, Maugey, G. Cheung, P. Frossard *RD optimized auxiliary information for inpainting-based view synthesis*, 3DTV Conference Zurich, Switzerland, Oct., 2012

(**J3**) T. Maugey, I. Daribo, G. Cheung, P. Frossard *Navigation domain partitioning for interactive multiview imaging*, in IEEE Transactions on Image Processing, Vol. 22, p. 3459-3472, Sep. 2013.

(**J8**) U. Takyar, T. Maugey, P. Frossard, *Extended Layered Depth Image Representation in Multiview Navigation*, in IEEE Signal Processing Letters Vol. 21, p. 22-25, 2014.

(**C29**) T. Maugey, and P. Frossard, C. Guillemot, *Guided inpainting with cluster-based auxiliary information,* IEEE ICIP, Quebec, Canada, Sep. 2015.

(**J13**) Y. Gao, G. Cheung, T. Maugey, P. Frossard, J. Liang, *Encoder-driven Inpainting Strategy in Multiview Video Compression,* in IEEE Transactions on Image Processing, Vol. 25(1), p. 134-149, Jan. 2016.

(**J18**) R. Ma, T. Maugey, P. Frossard, *Optimized Data Representation for Interactive Multiview Navigation*, in IEEE Transactions on Multimedia, Vol. 20(7), p. 1595-1609, Jul 2018.

(**J23**) M. Q. Pham, A. Roumy, T. Maugey, E. Dupraz, M. Kieffer *Optimal Reference Selection for Random Access in Predictive Coding Schemes* in IEEE Transactions on Communications, vol. 68(9), pp. 5819-5833, Sep. 2020.

2.3 What achievable performances ? How to reach them ?

While the segmentation introduced above takes into account the view popularity, and thus the user's request statistics, it cannot prevent that useless information has to be sent. Indeed, a navigation segment is transmitted entirely even though some viewpoints in it are never requested by users. In the following, we tackle the question: is it possible to send only what is needed and thus to achieve the minimal rate, *i.e.*, the one that we would achieve without interactivity ? From what we can see in the literature, the answer is "yes", but generally, the price to pay is a gigantic storage rate expense (*i.e.*, anticipate all possible requests at the encoder side). We thus propose to study this question from the theoretical point of view: what are the minimum R and S that can be achieved ?

In order to answer this question, we introduce the coding scheme in Figure 2.9. It is similar to the one in Figure 2.1, with some additional precisions. The data is explicitly composed of a set of items that can be individually accessed: the signals $\mathbf{x}_l$ generated by L correlated sources denoted $\{X_l\}_{1\leq l\leq L}$. After offline compression, the stored data is composed of several separable bitstreams denoted by $\{\mathbf{b}_i\}_{1\leq i\leq B}$. We express the *storage*

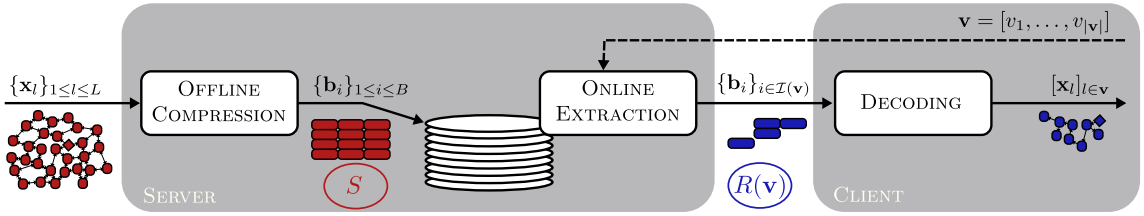


Figure 2.9: Massive Random Access with a Navigation graph

cost S as the size of the bitstreams stored on the server:

$$S = \frac{1}{L} \sum_{i=1}^{B} |\mathbf{b}_i|, \tag{2.4}$$

where $|.|$ denotes the sub-stream size expressed in number of bits, and where the normalization factor L is in order to have a cost per source.

The user's request $\mathbf{v}$ is a vector of source indices and is ruled by some restrictions imposed by the application. In Sec. 2.3.1, we propose to model these restrictions with a *navigation graph*. Based on this request, only some of the separable bitstreams are transmitted to the decoder. Their index set is denoted by $\mathcal{I}(\mathbf{v})$. We thus define the *per request transmission cost* as the cumulated size of the bitstreams sent to a client for a given request:

$$R(\mathbf{v}) = \frac{1}{|\mathbf{v}|} \sum_{i \in \mathcal{I}(\mathbf{v})} |\mathbf{b}_i|, \tag{2.5}$$

where the normalization leads to a per source criterion. Finally, to obtain a criterion that does not depend on the client's request, we assume that a probability distribution p over the clients' requests is available. This leads to the *expected transmission cost*:

$$R = \mathbb{E}_{\mathbf{v}}(R(\mathbf{v})) = \sum_{\mathbf{v} \in \mathcal{V}} p(\mathbf{v}) R(\mathbf{v}). \tag{2.6}$$

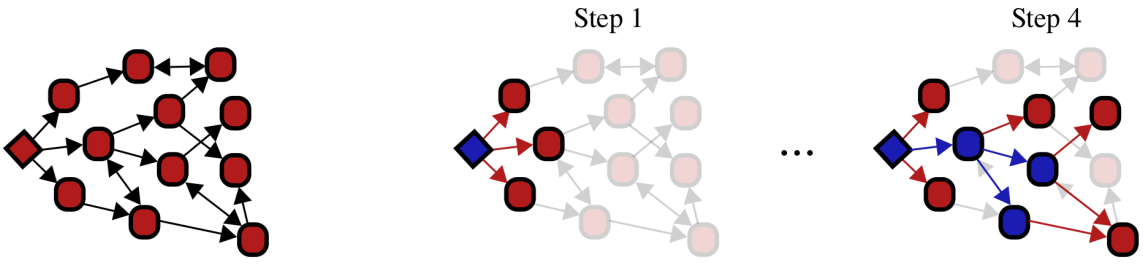
In Section 2.3.2, we derive the optimal values for S and R from the theoretical point of view. We compare this optimal value with three baseline schemes of the literature. Finally, in Section 2.3.3, we propose solutions able to reach these optimal storage and transmission costs. For ease of presentation, we consider in the next two Sections a lossless compression scheme. A lossy extension may be obtained, *e.g.*, with a quantization of the input sources [67].

2.3.1 Modeling Random Access by means of a navigation graph

Access to a database is usually proposed with some restrictions. For instance, in Free Viewpoint Navigation or 6DoF systems, the client observes a scene by navigating from one viewpoint to another. But, to offer a smooth client experience, the navigation might be limited to neighboring viewpoints only.

Before showing how to integrate these restrictions, we first model a request as a vector of source indices. The navigation of the client is therefore equivalent to a request of ordered source indices.

To describe the set of allowed requests to the database that may be performed by a client, we introduce the oriented navigation graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. $\mathcal{N}$ is a set of $L + 1$ nodes



(a) Navigation graph (b) Example of clients' navigation

Figure 2.10: A navigation graph and an example of client's navigation. (a) The nodes of the graph depicted by a circle represent the sources. The square node represents a dummy source used for the initialization of the navigation. A directed edge exists between node i and j if the source X_j can be requested once the source X_i has been previously requested and stored in the clients' memory. (b) In the example of navigation, the blue nodes correspond to the requested sources, the red ones to the sources that are allowed to be requested, and the faded red ones represent sources that can not be requested.

and $\mathcal{E}$ is a set of directed edges between these nodes. The nodes represent the L sources plus a (dummy) source X_0 used to initiate the navigation. A directed edge $e_{j,i}$ from node j to node i indicates that the source X_i can be accessed by a client, once X_j has been previously requested and stored in the clients' memory. The only source that may be directly accessed, without having previously requested another source, is the (dummy) source X_0 which corresponds to node 0. To summarize, the graph $\mathcal{G}$ introduces the constraints on the way sources may be accessed. The set of all possible requests consistent with $\mathcal{G}$ is denoted by $\mathcal{V}$.

2.3.2 Optimal achievable performance

In such conditions, several schemes have been investigated in the literature. In order to express their achieved performance, we introduce the following notations $\forall i \in [1, L], \forall j \in [0, L]$:

$$h_{i|j} = H(X_i^{n_i}|X_j^{n_j}) \text{ and } h_i = H(X_i^{n_i}), \tag{2.7}$$

where n_i and n_j stand for the signal length of the sources X_i and X_j respectively, $X_i^{n_i}$ denotes the random vector $(X_{i,1}, X_{i,2}, \ldots, X_{i,n_i})$, and $H(X)$ and $H(X|Y)$ denote the entropy and the conditional entropy respectively. We also recall that $\forall\, i,\ H(X_i|X_0) = H(X_i)$, since X_0 is a dummy source used for initialization only. The three reference coding architectures are:

- The *All Intra* (*AI*) scheme codes each source *independently* [68, 69]. While totally flexible, this solution does not exploit the correlation between the sources. The storage cost achieved by the *AI* scheme is thus[1]

$$S_{\text{AI}} = \frac{1}{L}\sum_{i=1}^{L} h_i \tag{2.8}$$

$$R_{\text{AI}} = \sum_{\mathbf{v}\in\mathcal{V}} p(\mathbf{v}) \frac{1}{|\mathbf{v}|} \sum_{i\in\mathbf{v}} h_i. \tag{2.9}$$

[1]We recall that, for sake of clarity, these costs are expressed for lossless transmission.

- The *Multiple Prediction* (*MP*) scheme stores one residue per possible prediction and transmit the actual one once the request is known [70, 71, 72, 73]. These predictions are built from each possible adjacent source available.

$$S_{\mathrm{MP}} = \frac{1}{L}\sum_{i=1}^{L}\sum_{j:e_{j,i}\in\mathcal{E}} h_{i|j}. \tag{2.10}$$

$$R_{\mathrm{MP}} = \sum_{\mathbf{v}\in\mathcal{V}} p(\mathbf{v})\frac{1}{|\mathbf{v}|}\sum_{i\in\mathbf{v}} h_{i|\pi_{\mathbf{v}}(i)}. \tag{2.11}$$

  Since $h_{i|j} \leq h_i$, the transmission cost is reduced with respect to the *AI* scheme, and the cost reduction increases with the correlation of the requested sources. On the other hand, the storage cost increases significantly with the averaged degree of the graph, *i.e.*, with the flexibility offered to the client to navigate within the database.
- The *Compound* (*C*) scheme uses channel-based coding to build one single bitstream able to correct any prediction [74, 75]. This is possible if this bitstream corresponds to the worst prediction, *i.e.*, the worst channel model. In that case,

$$S_{\mathrm{C}} = \frac{1}{L}\sum_{i=1}^{L}\max_{j:e_{j,i}\in\mathcal{E}} h_{i|j}. \tag{2.12}$$

$$R_{\mathrm{C}} = \sum_{\mathbf{v}\in\mathcal{V}} p(\mathbf{v})\frac{1}{|\mathbf{v}|}\sum_{i\in\mathbf{v}}\max_{j:e_{j,i}\in\mathcal{E}} h_{i|j}. \tag{2.13}$$

  The *C* scheme is thus a good way of achieving reasonable transmission cost (between R_{AI} and R_{MP}) while having a smaller storage cost than those of *MP* and *AI* schemes. However, while it is reasonable to consider that one needs to anticipate the worst scenario at the server side, this is unfortunate to transmit this bitstream in all the cases, even if the prediction is of a better quality.

The storage and transmission costs are summarized in Figure 2.11.

In our work in (C28) and (J19), we have shown that when compressing one single source with several potential side informations available at the decoder, it is possible to use incremental coding in order to only send the necessary amount of bits. Based on the graph-based client's navigation formalism, we have generalized this result to the multi-source scenario in (J19). We call this new coding scheme *Incremental coding Based Extractable Compression* (*IBEC*).

For each signal $\mathbf{x}_i$ to be compressed, we first identify the parents of the source of index i in the navigation graph $\mathcal{G}$. These neighbors are used to build potential predictions $\hat{\mathbf{x}}_{i|j}$. They are then sorted from the best to the worst (*i.e.*, from the smallest $h_{i|j}$ to the largest). Then, we build a first bitstream $\mathbf{b}_i^{j_1}$ able to decode the best prediction assuming that $\mathbf{x}_{j_1}$ is already decoded at the receiver. For the second bitstream, the coding scheme is able to use $\mathbf{b}_i^{j_1}$ plus an additional bitstream $\mathbf{b}_i^{j_2}$ of size $h_{i|j_2} - h_{i|j_1}$. This incremental construction is applied in the same way to all predictions. As a result, the stored bitstream, for each source, has the same size than the *C* scheme, *i.e.*, the one corresponding to the highest $h_{i|j}$, but is split into several sub-streams so that only the necessary information can be extracted. All schemes are illustrated in Figure 2.11.
The global performance of the *IBEC* scheme is:

$$S_{\mathrm{IBEC}} = \frac{1}{L}\sum_{i=1}^{L}\max_{j:e_{j,i}\in\mathcal{E}} h_{i|j}. \tag{2.14}$$

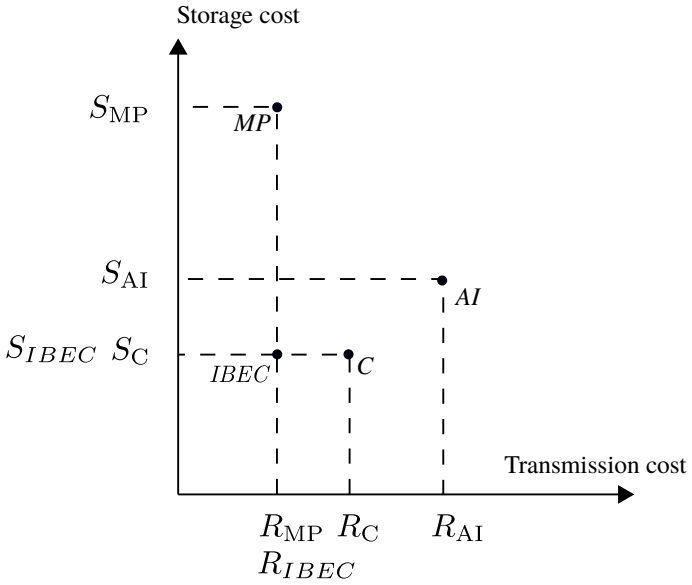


Figure 2.11: Storage and rate transmission costs summary. Our proposed *IBEC* scheme, obtains the best theoretical storage and transmission costs.

and

$$R_{\mathrm{IBEC}} = \sum_{\mathbf{v}\in\mathcal{V}} p(\mathbf{v}) \frac{1}{|\mathbf{v}|} \sum_{i\in\mathbf{v}} h_{i|\pi_{\mathbf{v}}(i)} \tag{2.15}$$

As it can be seen from Figure 2.11, the proposed *IBEC* scheme obtains the smallest storage and transmission costs of all the conventional architectures, *i.e.*,

$$S_{\mathrm{IBEC}} = S_{\mathrm{C}} \leq S_{\mathrm{AI}} \leq S_{\mathrm{MP}} \tag{2.16}$$

and

$$R_{\mathrm{IBEC}} = R_{\mathrm{MP}} \leq R_{\mathrm{C}} \leq R_{\mathrm{AI}}. \tag{2.17}$$

2.3.3 Practical scheme and experiments

In (J22) we have proposed a practical coding solution, able to reach such promising performance.

Incremental coding principle: The source coding problem with one source and one side information at the decoder can be solved in practice by channel codes [76, 77, 78, 79, 80]. Based on similar intuitions, we propose to construct a coding scheme based on channel codes. However, the channel code needs to tolerate variable rate to adapt to all the potential side informations. In practice, rate adaptation is achieved by choosing a rate among a finite set of predefined source coding rates: $R \in \{\frac{1}{M}, \ldots, \frac{m}{M}, \ldots, \frac{M}{M}\}$.

Assume that the decoder requests the source X_i, and has previously requested the source X_j, with $j \in \{j_1, ..., j_J\}$ (see Figure 2.12(a)). Note that the size J of the neighborhood depends on the node i, but for ease of presentation, we remove the dependence with respect to i in the notation J. Let us further assume that the sources X_j, with $j \in \{j_1, ..., j_J\}$ are sorted in increasing order of conditional entropy, *i.e.*, from the most to the least correlated source X_j,

$$h_{i|j_1} \leq h_{i|j_2} \leq \ldots \leq h_{i|j_J}. \tag{2.18}$$

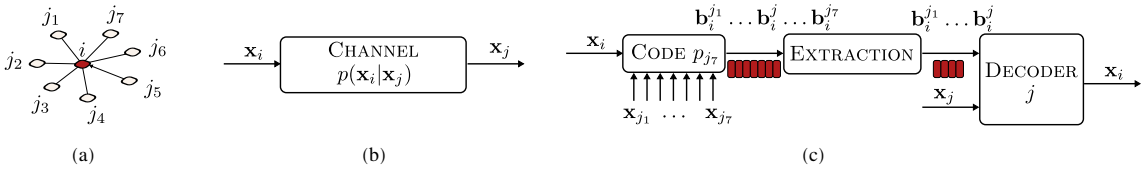


Figure 2.12: (a) Navigation graph: source X_i can be requested after one of the source X_j, with $j \in \{j_1, ..., j_7\}$. (b) Correlation channel: the correlation between the sources X_i and X_j is modeled by a channel with transition probability $p(\mathbf{x}_i|\mathbf{x}_j)$. (c) MRA compression scheme based on channel codes: encoding is performed by first computing the bitstreams $(\mathbf{b}_i^{j_1} \dots \mathbf{b}_i^{j_7})$ needed for the less correlated possible side information X_{j_7}. These bits are stored at the server. Then, upon request of the source with index i, and knowing that the source with index j has been previously requested, a subset of the bitstreams is extracted and used with the side information $\mathbf{x}_j$ to reconstruct the source vector $\mathbf{x}_i$.

We now explain how to encode the source vector $\mathbf{x}_i$ into an extractable bitstream $\mathbf{b}_i = (\mathbf{b}_i^{j_1}, \mathbf{b}_i^{j_2}, \dots, \mathbf{b}_i^{j_J})$.

Data encoding: let us consider that all correlated sources are marginally i.i.d., binary with uniform distribution, and that each source X_i generates a vector of length n. We model the pairwise correlation between the correlated sources by a channel with transition probability $p(\mathbf{x}_i|\mathbf{x}_j)$, see Fig. 2.12(b). Futher assume that the correlation channel is a binary symmetric channel. We use the rate-adaptive code called Low Density Parity Check Accumulate (LDPCA) Code introduced in [81]. Given a set of predefined target rates $\{\frac{1}{M}, \dots, \frac{m}{M}, \dots, \frac{M}{M}\}$ and a source vector length n, the LDPCA construction provides M parity check matrices denoted $(\mathbf{K}_1, \dots, \mathbf{K}_m, \dots, \mathbf{K}_M)$, where $\mathbf{K}_m$ is of size $n\frac{m}{M} \times n$ and where

$$\forall \mathbf{x}, \mathbf{K}_1\mathbf{x} \subseteq \mathbf{K}_2\mathbf{x} \subseteq \dots \subseteq \mathbf{K}_M\mathbf{x} \tag{2.19}$$

meaning that $\mathbf{K}_1\mathbf{x}$ is a subvector of the vector $\mathbf{K}_2\mathbf{x}$. In our simulations, we considered the *6336_irregDeg2to21* LDPCA code, whose parameters are available at [82].
We now explain how to encode the source vector $\mathbf{x}_i$. First, the so-called accumulated syndromes are computed as

$$\forall m,\ \mathbf{a}_{i,m} = \mathbf{K}_m\mathbf{x}_i, \tag{2.20}$$

where $\mathbf{a}_{i,m}$ is of length $n\frac{m}{M}$. Then, for each possible side information $\mathbf{x}_j$, and for each accumulated syndrome $\mathbf{a}_{i,m}$, a reconstruction is performed according to the maximum *a posteriori* criterion, *i.e.*, $\forall j \in \{j_1, ..., j_J\}, \forall m \in \{1, \dots, M\}$

$$\begin{aligned} &\forall j \in \{j_1, ..., j_J\}, \forall b \in \{1, \dots, B\}, \\ &\hat{\mathbf{x}}_{i,j,m} = \arg \max_{\mathbf{x}_i : \mathbf{a}_{i,m} = \mathbf{K}_m\mathbf{x}_i} p(\mathbf{x}_i|\mathbf{x}_j). \end{aligned} \tag{2.21}$$

Note that this defines a modified channel decoder, since the search space is the coset of syndrom $\mathbf{a}_{i,m}$ and not the coset of syndrom $\mathbf{0}$, as in classical channel coding. When a LDPC code is used, decoding is performed with the modified belief propagation (BP) algorithm proposed in [83].
Then, for each possible side information $\mathbf{x}_j$, we select the shortest accumulated syndrom $\mathbf{a}_i^j$ such that the BP decoder recovers $\mathbf{x}_i$ perfectly, *i.e.*, $\forall j \in \{j_1, ..., j_J\}, \forall m \in \{1, \dots, M\}$

$$m^*(j) = \arg\min_m \{|\mathbf{a}_{i,m}| = n\frac{m}{M}\ s.t.\ \hat{\mathbf{x}}_{i,j,m} = \mathbf{x}_i\} \tag{2.22a}$$

$$\mathbf{a}_i^j = \mathbf{a}_{i,m^*(j)}. \tag{2.22b}$$

From the inclusion property of the accumulated syndromes (2.19), and from the ordering of the side information vectors (2.18), the optimal accumulated syndromes satisfy $\mathbf{a}_i^{j_1} \subseteq \mathbf{a}_i^{j_2} \subseteq \cdots \subseteq \mathbf{a}_i^{J}$.
Finally, for the source X_i, the stored sequence of bitstreams $\mathbf{b}_i = (\mathbf{b}_i^{j_1}, \mathbf{b}_i^{j_2}, \ldots, \mathbf{b}_i^{j_J})$ is constructed from the $\mathbf{a}_i^j$ as follows. First, $\mathbf{b}_i^{j_1} = \mathbf{a}_i^{j_1}$. Then, the second bitstream $\mathbf{b}_i^{j_2}$ is obtained by retaining the bits in $\mathbf{a}_i^{j_2}$ that are not in $\mathbf{a}_i^{j_1}$, *i.e.*, $\mathbf{b}_i^{j_2} = \mathbf{a}_i^{j_2} \setminus \mathbf{a}_i^{j_1}$. More generally, we have $\mathbf{b}_i^{j_k} = \mathbf{a}_i^{j_k} \setminus \mathbf{a}_i^{j_{k-1}}$.
The resulting storage cost for the source X_i is $S_i = |\mathbf{a}_i^J|/n$, where the overall storage cost is $S = \frac{1}{L}\sum_{i=1}^{L} S_i$.

Data extraction: Upon request of the source X_i, and knowing that the source X_j is available at the decoder, the server extracts from $\mathbf{b}_i$ the subsequence $(\mathbf{b}_i^{j_1}, \mathbf{b}_i^{j_2}, \ldots, \mathbf{b}_i^{j}) = \mathbf{a}_i^j$, and sends it to the decoder. This leads to a transmission cost $R_i^j = |\mathbf{a}_i^j|/n$. Then, the transmission cost of a request, $\mathbf{v} = (l_1, \ldots, l_{|\mathbf{v}|})$ is

$$R(\mathbf{v}) = \frac{1}{|\mathbf{v}|}\sum_{i\in\mathbf{v}} R_i^{\pi_{\mathbf{v}}(i)} \tag{2.23}$$

Data decoding: Upon request of the source X_i, the decoder receives $(\mathbf{b}_i^{j_1}, \mathbf{b}_i^{j_2}, \ldots, \mathbf{b}_i^{j}) = \mathbf{a}_i^j$. The decoder then performs BP decoding taking into the previously received side information $\mathbf{x}_j$. From the rate adaptation performed at the encoder (2.22), the reconstruction is performed without any error.

IBEC vs C scheme: The C scheme shares similarities with the proposed IBEC scheme since, in both cases, a channel code is used to perform data encoding. In the IBEC scheme, the index of the previous request j is used to adapt the transmission and send the complement information only, as shown in (2.22). In the C scheme, this knowledge is not used. The sent accumulated syndrome is the one that allows perfect reconstruction for any possible side information (2.24a), in particular for the worst one (2.24b).

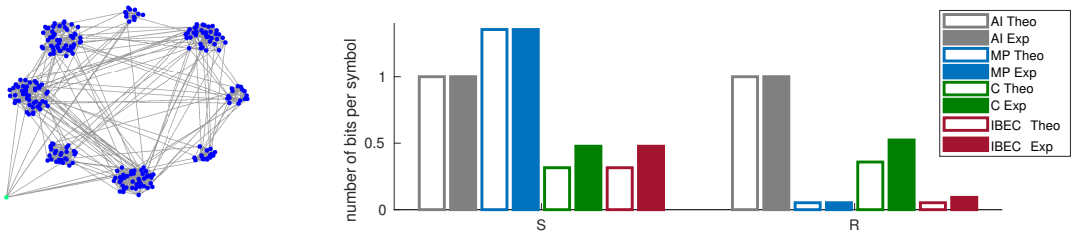
$$m^* = \arg\min_m\{|\mathbf{a}_{i,m}| = n\frac{m}{M} \; s.t. \; \forall j \; \hat{\mathbf{x}}_{i,j,m} = \mathbf{x}_i\} \tag{2.24a}$$

$$= \arg\min_m\{|\mathbf{a}_{i,m}| = n\frac{m}{M} \; s.t. \; \hat{\mathbf{x}}_{i,j_J,m} = \mathbf{x}_i\} \tag{2.24b}$$

$$\forall j, \; \mathbf{a}_i^j = \mathbf{a}_{i,m^*}. \tag{2.24c}$$

IBEC and C schemes vs MP: The IBEC and C schemes use channel coding to perform data encoding. By contrast, in the MP scheme all possible residues $\hat{\mathbf{x}}_i - \mathbf{x}_{i|j}, \forall(i,j)$ are encoded with a variable length source code and then stored. Upon request of the source of index i, after having requested the source of index j, only the compressed bitstream of $\hat{\mathbf{x}}_i - \mathbf{x}_{i|j}$ is sent.

Experimental results: we now show that our IBEC solution enables to outperform the baseline schemes and reach the performance promised by our theoretical derivation. We have implemented the *AI*, *MP*, *C* schemes and our proposed *IBEC*. For *AI*, *MP* schemes the coding of the source or the residue after prediction is done using an arithmetic coder. For *C* scheme, the encoding is done with the *6336_irregDeg2to21* LDPCA code available at [82] (as in (2.22), among the set of codes, we choose the LDPC code with minimum rate that allows perfect reconstruction of the vector). For each graph, we have encoded the corresponding data $\mathbf{X}$, and simulated 100 client's navigations, recording, each time, the transmission cost. We have also calculated, for each case, the theoretical expected performance, based on the entropy calculation, see Equations (2.8,2.9) for *AI*, Equations (2.10,2.11) for



(a) Navigation graph (b) Rate/Storage comparison

Figure 2.13: Experimental results obtained for th *Community network* [84] and $L = 256$ sources.

MP, Equations (2.12,2.13) for *C* and Equations (2.14,2.15) for *IBEC*. A sample of the obtained results in (J22) is shown in Figure. 2.13.

The *IBEC* scheme theoretically achieves the smallest storage and transmission costs, respectively in Eq. (2.16) and Eq. (2.17). Said differently, our *IBEC* scheme achieves the smallest transmission rate as *MP* scheme while reaching also the smallest storage cost as the *C* scheme, which validates its potential advantage.

The difference between theoretical and practical costs is small for the schemes *AI* and *MP* since they use an arithmetic coder whose performance is not far from the Shannon bounds. On the contrary, the channel codes used in the *C* scheme and by the incremental coders in the *IBEC* scheme have a more significant gap between theory and practice. Nevertheless, despite this disadvantage, the practical performances comparison still demonstrate the benefits of our scheme. This is indeed visible from the following observations:

- Compared to the *AI* scheme, both storage and transmission costs are always smaller with the *IBEC* scheme. The results demonstrate the ability of the *IBEC* scheme to take into account this correlation at the storage and transmission stages.
- Given the fact that the *MP* scheme reaches the best transmission rate possible (thanks to en extensive storage cost), we observe that our *IBEC* scheme is really efficient. Indeed, the transmission rate achieved by the *IBEC* scheme is almost the same (or slightly higher) than the *MP* scheme, for a storage cost that is much lower. To be able to reach the best transmission cost, the *MP* scheme has to store many residues, exploding the storage cost, while this storage cost remains small with our scheme.
- Instead of storing any possible navigation transition, the *C* scheme stores for each source, the *worst one*, as the *IBEC* scheme does. This is the reason why the storage costs of the *C* and *IBEC* schemes are the same (and the minimum ones) in the theoretical and practical performances. However, instead of transmitting the whole codeword for every request as the *C* scheme, our *IBEC* scheme only transmits the necessary subpart.

As a conclusion, in both theoretical and practical aspects, one observes that the *IBEC* scheme reaches or is very close to the best expected transmission costs, with a minimal storage cost.

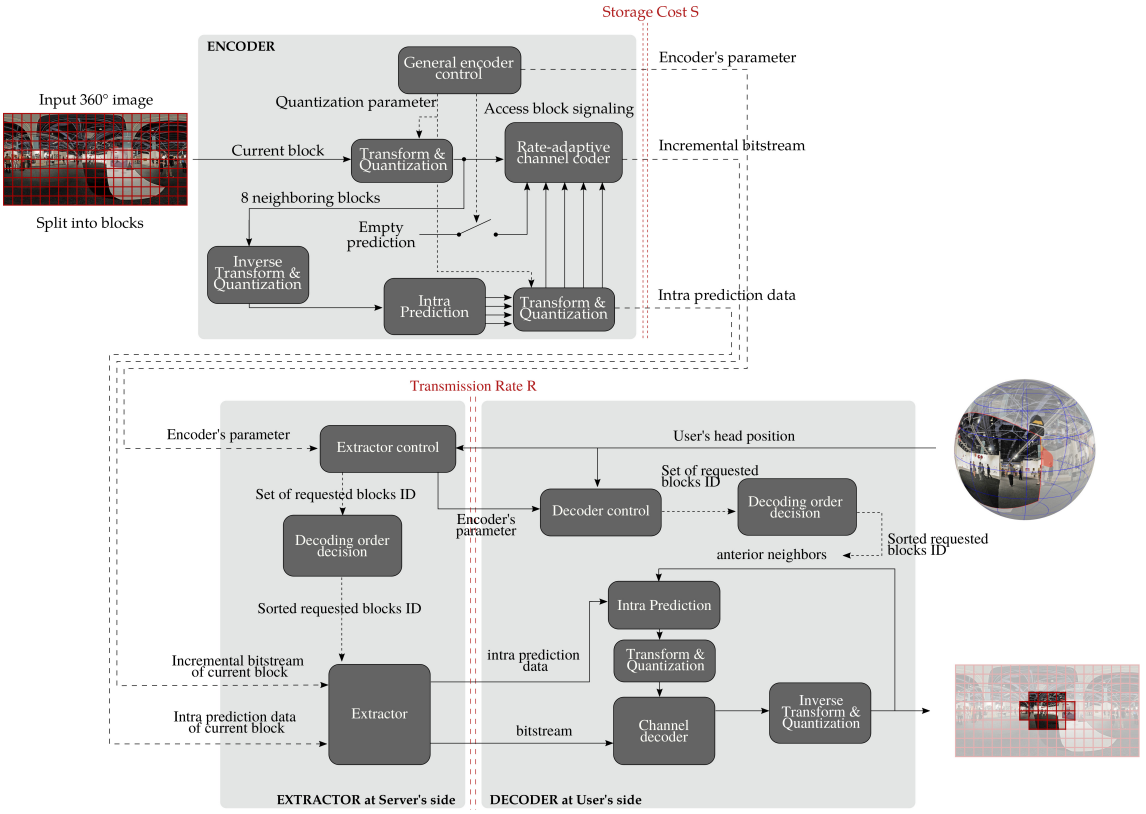


Figure 2.14: Proposed interactive coding scheme for omnidirectional data.

(**C28**) A. Roumy and T. Maugey, *Universal lossless coding with random user access: the cost of interactivity*, IEEE ICIP, Quebec, Canada, Sep., 2015 (Top 10% papers)

(**J19**) E. Dupraz, T. Maugey, A. Roumy, M. Kieffer, *Rate-Storage Regions for Extractable Source Coding with Side Information* in Physical Communication, Elsevier, Special Issue on Coding and Information Theory for Emerging Communication Systems, Vol. 37, 2019.

(**J22**) T. Maugey, A. Roumy, E. Dupraz, M. Kieffer, *Incremental coding for extractable compression in the context of Massive Random Access* in IEEE Transactions on Signal and Information Processing over Networks, vol. 6(1), pp. 251-260, Dec. 2020.

2.4 Practical interactive video coder for omnidirectional images

Based on the intuition above, we have built in (J24), and entire coding scheme for omnidirectional videos, greatly inspired from the principles exposed in (J22) and the previous Section. In (C43), we have extended this scheme to handle 3D mesh texture.

2.4.1 Proposed scheme's overview

The proposed scheme is shown in Figure 2.14 and descibed in detail in (J24). It relies on the following principles:

- the omnidirectional image is *split into blocks*[2], and these blocks constitute the sources $\mathbf{x}_l$ (of previous Section and Figure 2.9) that are requested partly.
- given that the sources $\mathbf{x}_l$ are the image blocks, the navigation graph simply links a block with its 4 direct neighbors. This comes from the fact that the requested region is made of a connected set of blocks.
- some blocks, the so-called access blocks, are intra coded. They correspond to the neighbors of the dummy source X_0 in the graphs.
- the decoding is done as depicted in Figure 2.15. First the access block is decoded. And then, the blocks are decoded in an optimized order *that can be decided at the server's side* (and not at the encoder as in conventional compression schemes). The already decoded blocks serve to predict the next block to be decoded.

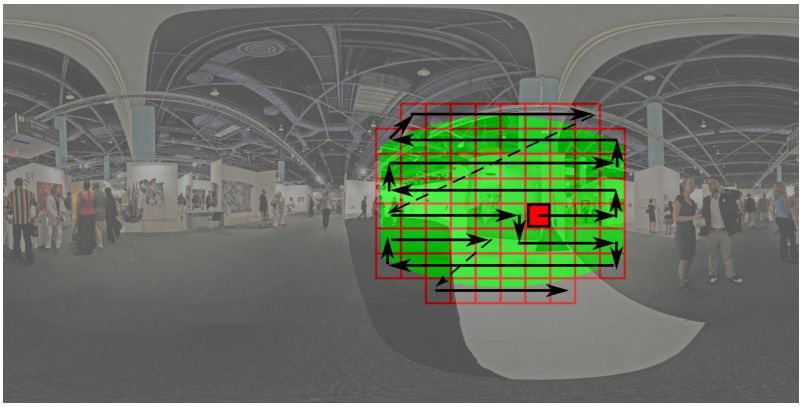


Figure 2.15: Decoding process. The red block is the access block. The arrows indicate the decoding order.

- the prediction of a block $\mathbf{x}_l$ is done using intra prediction [85] using the available neighboring blocks. In order to anticipate the different scenarios at the decoder, several predictions are considered depending on the possible available blocks (see Figure 2.16).

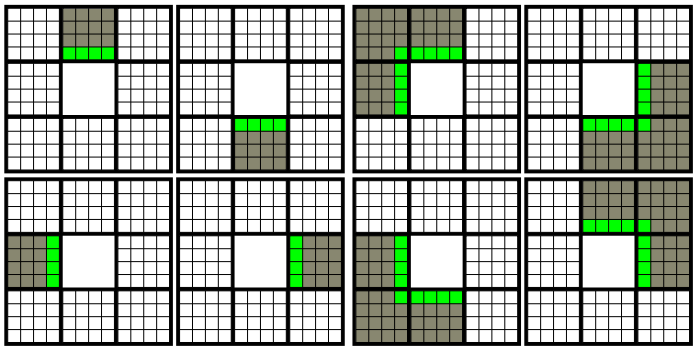


Figure 2.16: Possible available blocks when predicting a block (the central one in white).

[2] Any omnidirectional image format that supports block decomposition is compatible with our scheme.

- the encoder uses the IBEC's principle and store for each block an extractable codeword able to handle any prediction.

With these principles, we are able to apply the IBEC's strategy to an omnidirectional image.

2.4.2 Experimental comparison

Assessing the performance of a coding scheme enabling random access raises several questions. First, the transmission cost R naturally depends on user behavior. Second, the rate cost is split into two terms: the storage and the transmission costs. In (C41), we have proposed an evaluation strategy that basically consists in:

- record or simulate several user's navigation, and evaluate, for each of them the transmission cost. The final transmission cost R is simply their average.
- evaluate the weighted rate as
$$R + \lambda S, \tag{2.25}$$
where λ sets the relative importance between the storage and the transmission costs.

We show the performance of the proposed scheme (called 360-IBEC) in Table 2.1 for different values of λ. The results are presented as the Bjontegaard gain over the scheme that consists in transmitting the whole image. We first compare our method with the exhaustive storage (ES) approach, that consists of a predictive coding scheme for which every prediction is stored for every block. We can see that, when the storage size is not important ($\lambda = 1e^{-3}$), ES reaches the best performance. Theoretically, our 360-IBEC should have reached the same performance, but it suffers from the small sub-optimality of the proposed rate-adaptive LDPC. We also compare with the tile-based approaches (called $T.2 \times 2$ and $T.7 \times 7$). These methods are the best when the transmission cost R is not important ($\lambda = 0.1$). Indeed, sending useless pixels is not penalized, and only the storage overhead matters. We can, however, see the benefits of the proposed approach when both transmission and storage costs are important. The LDPC sub-optimality becomes negligible and being able to transmitted only the requested blocks without exploding the storage cost is a great advantage.

Table 2.1: Weighted BD for requests of length 1 sec averaged over all users relative to the no tiling approach (T. 1x1).

		Market	Street	Mountain	Church	Seashore	Park	Jacuzzi	Cafe	Average
$\lambda = 0.1$	T. 2x2	-27.73	-24.53	-19.46	-30.02	-15.20	-27.82	-29.46	-24.82	-24.88
	T. 7x7	**-46.16**	**-45.84**	**-44.55**	**-45.84**	**-42.52**	**-45.65**	**-42.65**	**-45.16**	**-44.80**
	ES	362.05	360.38	358.54	359.84	359.48	361.02	359.02	360.72	360.13
	360-IBEC	-29.05	-28.76	-27.39	-28.29	-25.82	-28.08	-26.42	-27.55	-27.67
$\lambda = 0.01$	T. 2x2	-41.87	-37.03	-29.37	-45.32	-22.95	-42.00	-44.48	-37.48	-37.56
	T. 7x7	-69.71	-69.23	-67.26	-69.23	-64.21	-68.93	-64.41	-68.22	-67.65
	ES	-19.47	-18.93	-17.53	-18.62	-15.85	-18.59	-16.68	-18.04	-17.96
	360-IBEC	**-73.35**	**-72.11**	**-69.44**	**-71.47**	**-66.90**	**-71.69**	**-68.18**	**-70.70**	**-70.48**
$\lambda = 1e^{-3}$	T. 2x2	-44.11	-39.02	-30.95	-47.75	-24.18	-44.26	-46.87	-39.50	-39.58
	T. 7x7	-73.46	-72.95	-70.87	-72.95	-67.67	-72.64	-67.87	-71.89	-71.29
	ES	-80.15	**-79.26**	**-77.34**	**-78.82**	**-75.56**	**-78.96**	**-76.43**	**-78.28**	**-78.10**
	360-IBEC	**-80.40**	-79.00	-76.13	-78.34	-73.45	-78.63	-74.83	-77.57	-77.29

2.4.3 Extension to 3D mesh texture coding

In (C43), we have proposed an extension of the IBEC's principle to 3D mesh compression. As it can be seen in Figure 2.17, the color information of a 3D mesh can be represented

as a color atlas or texture map, that is basically an image gathering the pieces of the 3D mesh.

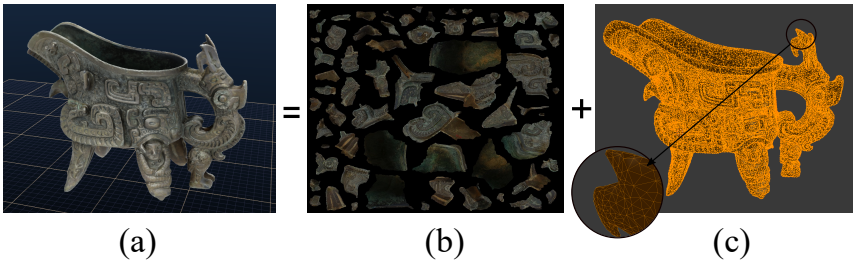


Figure 2.17: Decomposition of a 3D model (a) into its texture atlas (b) and its triangular mesh (c).

As an omnidirectional image, the atlas in Figure 2.17(b) can be split into blocks. And they can be coded with the same principles than the 360-IBEC scheme described above: several predictions can be generated, and an incremental codeword is generated such that it is able to correct any of them at the decoder side. Contrary to the omnidirectional image case, we can see in Figure 2.18, that a request in the atlas can be a disconnected set of blocks. One simply has to navigate in the geometry information to retrieve the neighborhood information of the blocks on the border of a patch in the atlas.

Figure 2.18: Example of user's navigation around the 3D model (left), what he observes at a given instant (center), and the corresponding visible blocks $\mathcal{V}$ (in red) in the atlas (right). For better visibility of the blocks, the black background area is turned into white in the atlas.

As for the 360-IBEC, we have shown that our scheme is able to transmit only what is requested, while exploiting the correlation between the blocks and keeping the extra storage negligible.

(J24) N. Mahmoudian-Bidgoli, T. Maugey, A. Roumy, *Fine granularity access in interactive compression of 360-degree images based on rate adaptive channel codes* accepted in IEEE Transactions on Multimedia, 2020

(C43) N. Mahmoudian Bidgoli, T. Maugey, A. Roumy, F. Nasiri and F. Payan, *A geometry-aware compression of 3D mesh texture with random access* Picture Coding Symposium (PCS), Ningbo, China, Nov. 2019

(C41) N. Mahmoudian Bidgoli, T. Maugey , A. Roumy, *Evaluation framework for 360-degrees visual content compression with user-dependent transmission* IEEE ICIP, Tapei, Taiwan, Sep. 2019

2.5 Conclusion

In this Chapter, we have studied the problem of compression under user's random access. We have first shown that the conventional approach was incompatible with such assumptions. Then, we have extended the concept of image partitioning to the 6D navigation domain. We have proposed compact representations for describing the information of many contiguous viewpoints, and we have proposed an optimal partitioning algorithm. In a second group of works, we have developed a complete study *from the theory to the practice* enabling to reach optimal coding performance. Based on innovative information theoretical results, we have built a proof of concept scheme and two practical coding schemes for omnidirectional images and 3D mesh. The groundbreaking result is that we have shown that it was theoretically and practically feasible to send *only what is requested by a user* with a small storage overhead.

Chapter 3

Graph construction: exploiting the geometry of 3D images

We recall that a typical 3D image is composed of two entities: i) the color (also called texture) and ii) the geometry (denoted by γ). The compression of such data has been studied in the context of standardization (*e.g.*, [86] for point clouds, [87] for 360° videos, [88] for multi-view and [89] for Light fields) yielding to already efficient gain. However, the non-euclidean topology inherent to these image modalities has not been taken into account properly or only indirectly (thanks to mapping for example) leading to coding sub-optimality. In this chapter, we describe how graph-based coding techniques have been able to compress color signal exploiting the full benefit of the geometry data [90]. In particular, we discuss how transform operations can be efficiently defined on these irregular topologies. We first pose the problem of graph construction in Section 3.1. We then present examples of graph construction techniques for different image modalities in Section 3.2. In these two first Sections, we assume that the geometry information is coded *separately*, using specific coding tools (which have been reviewed in Section 1.2), and in Section 3.3, we show that it can be beneficial to code the graph itself instead of the geometry data.

3.1 Graph construction problem

Modeling an irregular topology with graph is equivalent to setting pairwise relationships between the N pixels (considered as the nodes). These connections are then used to define processing operations on the signal defined on this topology. Before discussing what is the link between the pairwise relationships and the signal property, we first recall the mostly adopted transform defined on the graph: the *graph-fourier transform* (*GFT*) (also called graph-based transform (GBT).

The GFT is based on the eigendecomposition of the Laplacian operator as explained in [91, 92]. For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, the Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ and $\mathbf{A}$ are the degree and adjacency matrices. The eigenvectors and eigenvalues of $\mathbf{L}$ are respectively denoted by $\mathbf{u}_l$ and λ_l (with $1 \leq l \leq N$). By definition, the matrix $\mathbf{U}$, whose columns are the $\mathbf{u}_l$, diagonalizes $\mathbf{L}$, *i.e.*, $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ with $\mathbf{\Lambda} = \mathrm{diag}([\lambda_l]_{1 \leq l \leq N})$. The graph Fourier transform of the signal $\mathbf{z}$ is defined as its projection on the Laplacian eigenvectors:

$$\boldsymbol{\alpha} = \mathbf{U}^\top \mathbf{z}. \tag{3.1}$$

The vector $\boldsymbol{\alpha}$ contains the transformed coefficients or the spectrum of signal $\mathbf{z}$. As recalled previously, a transform is efficient for compression if it i) decorrelates and ii) compacts

the signal energy. Let us analyze why and when the Laplacian can be considered as a good transform for compression.

A first remark is that, based on the optimality of the Karhunen-Loeve Transform (KLT) [93], a sufficient condition for the graph-Fourier transform to be able to decorrelate the signal $\mathbf{z}$ is that its covariance matrix looks like the Laplacian $\mathbf{L}$. In other words, the weights w_{ij} that constitute the offline elements of $\mathbf{L}$ should reflect the amount of correlation between the source Z_i and the source Z_j (still considering that the signal $\mathbf{z}$ is a realization of a vector of random variable $(Z_1, \ldots, Z_N)$). A second remark is that the Laplacian operator is strongly related to the notion of variation on the graph. Said differently, the eigenvectors in $\mathbf{U}$ are ranked by their level of variation measured by the eigenvalues λ_i, *i.e.*, diagonal elements of $\mathbf{\Lambda}$. The graph transform is thus able to compact the signal energy in some coefficients, *i.e.*, some eigenvectors, if the signal $\mathbf{z}$ exhibits a similar behaviour to that of a few eigenvectors over the graph $\mathcal{G}$. Based on these two remarks, we are able to state that if the signal $\mathbf{z}$ is smooth on the graph, then the graph-Fourier transform is efficient for compression. By smooth, we mean that the signal $\mathbf{z}$ does not vary too much along the graph edges, or more exactly should vary according to the weights (*i.e.*, large weights imply low variation). Keeping this idea in mind, a usual way to measure the smoothness of a signal is computing what we call a total variation of the signal on the graph (also called Laplacian quadratic form) as follows:

$$\mathrm{TV}_{\mathbf{L}}(\mathbf{z}) = \mathbf{z}^{\top}\mathbf{L}\mathbf{z} = \sum_{i,j} w_{ij}(z_i - z_j)^2 = \sum_{l} \lambda_l \alpha_l^2, \tag{3.2}$$

where the set of λ_l are the eigenvalues of the Laplacian matrix $\mathbf{L}$. The smaller the total variation on the graph, the more the energy of the transformed signal is concentrated in the smallest eigenvectors. The role of the graph construction is to build a graph such that the signal $\mathbf{z}$ is smooth on it. As such, the energy will be concentrated in a few coefficients corresponding to low frequencies, and only those that are the most representative need to be transmitted to the decoder side. In light with these principles we highlight two challenges for graph construction: the topology and the weights.

Topology design: contrary to 2D images that benefit from a natural underlying 2D grid, the topology of 3D data is not straightforwardly defined and has to be carefully constructed depending on the data type. The topology design consists in finding the edges $\mathcal{E}$ from a given set of nodes $\mathcal{V}$. Based on the previous discussions, an edge should connect two nodes if their attached signal values are correlated.

Weights adjustment: it simply consists in estimating the matrix $\mathbf{W}$ whose elements w_{ij} are the weights assigned to each edge e_{ij}. As mentioned before, a good weight w_{ij} should depict the correlation between pixel colors on node i and j.

3.2 "Closer is more correlated"

The graph should be constructed identically at the encoder and the decoder. Therefore the graph construction cannot be driven by the input signal $\mathbf{z}$ since it is not available at the decoder. However, the geometry $\boldsymbol{\gamma}$ is available at the encoder and decoder and can therefore be exploited to construct the graph. In order to define a graph on which the signal $\mathbf{x}$ is smooth, the following hypothesis is formulated:

Hypothesis: *The correlation between pixels is decreasing with the distance between their corresponding points in the 3D space:*

$$\forall (i,j),\ \ w_{ij} = \phi(||\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j||_2^2) \quad \textit{with } \phi \textit{ monotonically decreasing.} \tag{3.3}$$

This hypothesis is justified by the fact that two points close in space are more likely to belong to the same object and therefore to have similar color. On the contrary, pixels that are far away in space do not belong to the same part of the scene and their color can thus be seen as independent. Despite its simplicity, this hypothesis already leads to an efficient compression performance.

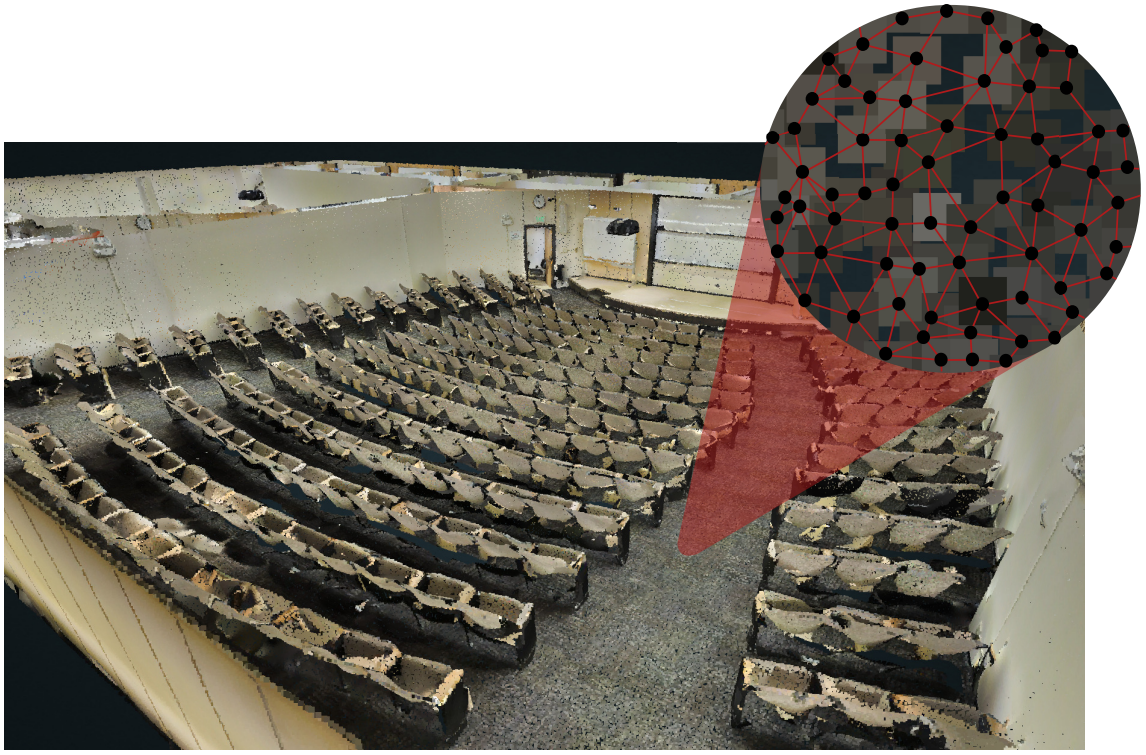


Figure 3.1: Graph topology for point clouds.

3.2.1 Nearest neighbor for 3D data

A first way of applying the Hypothesis in Equation (3.3) in practice is to build the topology *from scratch*, relying mostly on the geometry. This is for example the case of Point Clouds, for which two approaches can be considered.

In a first one, a node is linked to any other node in a given neighborhood (as illustrated in Figure 3.1). More formally:

$$e_{ij} \in \mathcal{E} \quad \text{if } v_i \in \mathcal{N}(v_j),$$

where $\mathcal{N}(v_j)$ stands for the neighborhood of vertex v_j. The neighborhood can simply be a ball of a given radius around vertex v_j, *i.e.*,

$$\mathcal{N}(v_j) = \{v_i \mid ||\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j||_2^2 < \varepsilon\}.$$

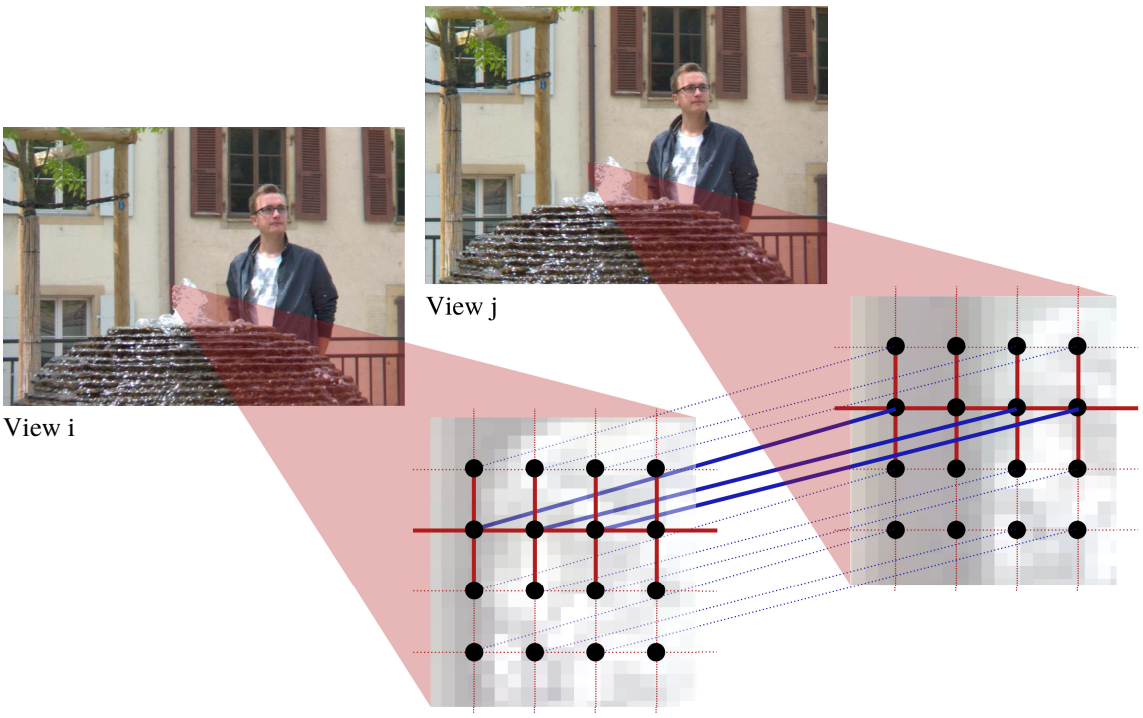


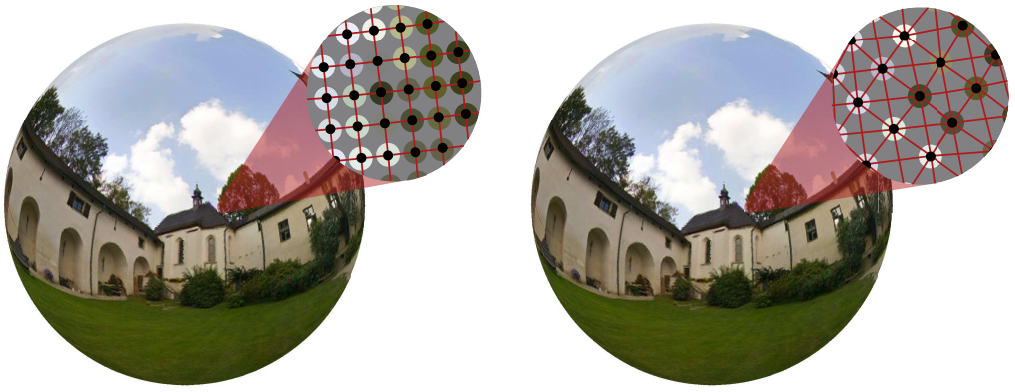
Figure 3.2: Graph topology for light fields.

The neighborhood can also be defined based on octrees [94]. The vertices v_j are usually placed at the center of cubes that pave the 3D space. A given cube is adjacent to 26 other cubes in space. These 26 cubes are taken as the neighborhood in [95, 96]. In a second approach, a node can be linked to the k-nearest neighbors [97, 98, 99]. Said differently, a node is linked to the k nodes that have the smallest distance $||\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j||_2^2$.
With both topologies, the edges do not always link nodes with the same distance. In order to fit with the hypothesis in Equation (3.3), several continuous functions can be considered: $w_{ij} = \frac{1}{||\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j||_2^2}$ (inverse-distance model) as in [95, 96] or $w_{ij} = \exp\left(-\frac{||\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j||_2^2}{2\sigma^2}\right)$ (exponential model) as in [97].

We have extended the nearest neighbor principle to the coding of 3D mesh when they are presented under the form of an atlas (see Section 2.4.3). In (C39), we have built the following graph construction strategy. Since the atlas relies on a 2D grid, we have kept the 2D grid connections when meaningful, *i.e.*, when the 4 neighbors are available. When one or several neighbors is not available, as pixels at the border of a patch, we find the pixel whose geometrical information $\boldsymbol{\gamma}$ is the closest. Said differently, we find the pixel that is a neighbor on the 3D shape. After having built such a graph, we define a graph-based transform that is used for compression. We experimentally demonstrate that compressing the data with the proposed graph topology is more efficient than coding the atlas directly with a classical image coder such as JPEG (see Figure 3.4(b)).

3.2.2 Far/Near model for Light field images

Light field images are very redundant (see Section 1.2). Indeed, pixels in different views usually correspond to the same 3D point in the scene. Their color should be similar if not equal. The idea of our proposal in (C32) is to detect these redundant pixels and link them within a graph $\mathcal{G}$. These edges are represented in blue in Figure 3.2. At the same



Equirectangular sampling Healpix sampling

Figure 3.3: Graph topology for omnidirectional images based on two different sampling approaches.

time, the 2D image grids in each image (red in Figure 3.2) are kept. It means that a node is connected to its 4 neighbors in the same view, and to its corresponding pixels in other views. While the inter-view (blue) edges connect pixels that are very likely to be correlated, the intra-view (red) edges in each 2D image grid are not always meaningful since two neighboring pixels could correspond to two different objects. This is the reason why, the weights corresponding to red edges can be refined based on the geometry information, *i.e.*, disparity map. A far/near model can be adopted:

$$w_{ij} = \begin{cases} 1 & \text{if } ||\gamma_i - \gamma_j||_2^2 < \varepsilon \\ a & \text{otherwise} \end{cases},$$

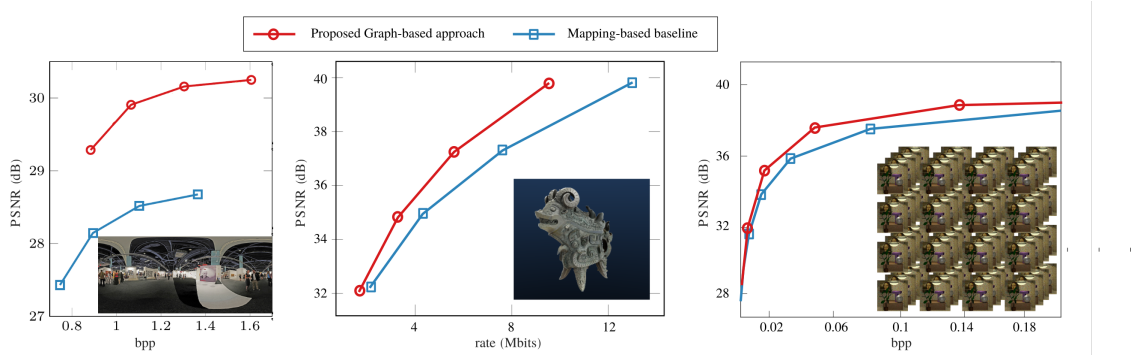
with a being an arbitrary small value.

Based on this graph, a coding scheme based on GFT is built. The disparity map is transmitted so that the graph can be rebuilt at the decoder side. Experiments have shown interesting gains compared to baseline approaches (see Figure 3.4(c)). This basic method has however been improved (see Chapter 4) to take into account the local statistics of the signal $\mathbf{x}$.

3.2.3 Geodesic distance for 360° images

Before specifying the graph topology, the most important problem of spherical data representation is to define the position of the pixels. Several sampling methods of the sphere exist [100], each of them presenting advantages and drawbacks. In this section, we will only focus on two of them: equirectangular and uniform sampling. All what is said hereafter is compatible with *any* other sphere mapping.

Equirectangular sampling consists in uniformly sampling the longitude and the latitude of a sphere (as commonly done for representing the earth). The resulting pixels can then be mapped easily into a 2D image, which makes it compatible with 2D processing tools. For this reason, it has been widely adopted. However, the pixels are not uniformly sampled on the sphere. Indeed pixel distribution is denser at the poles than at the equator. In this case, the topology is simply derived from the 2D grid (Figure 3.3 left) but the weights can be adjusted such that this heterogeneous distribution is taken into account.



(a) 360° (b) 3D mesh (c) Light Fields

Figure 3.4: Rate-distortion comparison between our graph-based compression approach and mapping-based solution. The mapping-based baseline respectively corresponds to (a) JPEG in equirectangular format, (b) JPEG in the atlas format, (c) HEVC with the Light Field views considered as a video.

Uniform sampling consists in spreading N pixels uniformly over the sphere. Even though this problem is mathematically unsolved, pseudo-optimal solutions exist such as the sampling called HealPix introduced in [101]. In that case, the edges are built such that each pixel is linked to its k-neighbors ($k = 8$ in Figure 3.3 right). Here also, the weights can then be defined to take into account the distance between nodes, as for example in our proposed method in (C44):

$$w_{ij} = \exp\left(-\frac{d^2_{\text{geo}}(\boldsymbol{\gamma}_i, \boldsymbol{\gamma}_j)}{2\sigma^2}\right),$$

where d_{geo} stands for the geodesic distance, *i.e.*, the shortest distance *on the sphere* between two points. The geodesic distance, or *great-circle* distance, differs from the cartesian distance between two points and is given by:

$$d_{\text{geo}} = 2 \arcsin \frac{||\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j||_2}{2},$$

where $\boldsymbol{\gamma}_i$ refers to the 3D position of the pixels lying on a unitary sphere.

In (C44), we use the HealPix sampling to define an entire image coder operating directly on the sphere. Using the nice properties of HealPix sampling, we are able to define spherical blocks on the sphere on which we redefine the conventional coding operations. The transform is done using the aforementioned approach. We experimentally demonstrate that this method is more efficient than: a conventional compression of the equirectangular image (see Figure 3.4(a)). It validates the double intuition of i) working directly on the uniformly sampled sphere and ii) building a graph following the spherical geometry.

(**C32**) Xin Su, M. Rizkallah, T. Maugey, C. Guillemot *Graph-based light fields representation and coding using geometry information*, ICIP, Beijing, China, Sep., 2017.

(**C39**) F. Nasiri, N. Mahmoudian-Bigdoli, F. Payan, T. Maugey ., *A geometry-aware framework for compressing 3D mesh textures*, IEEE ICASSP, Brighton, UK, May. 2019. *cited in IEEE MMTC Review Letter of April 2019* Picture Coding Symposium (PCS), Ningbo, China, Nov. 2019

(**C44**) N. Mahmoudian Bidgoli, T. Maugey , A. Roumy, *Intra-coding of 360-degree images on the sphere* Picture Coding Symposium (PCS), Ningbo, China, Nov. 2019

3.3 Transmitting the graph instead of the geometry

In the solutions above, the geometry is sent separately to the decoder and the geometry serves to reconstruct the graph at the receiver. We have also wondered if it could be more efficient to code the graph directly instead of the depth. In (C19,C20, C21,C24,J10,J16), we have proposed a new representation for multi-view data, called *graph-based representation* (*GBR*), that precisely uses the graph to encode the geometry information γ. This solution enables to send the exact amount of geometry that is needed at the receiver and thus to have a more compact data representation.

3.3.1 Graph-based representation (GBR)

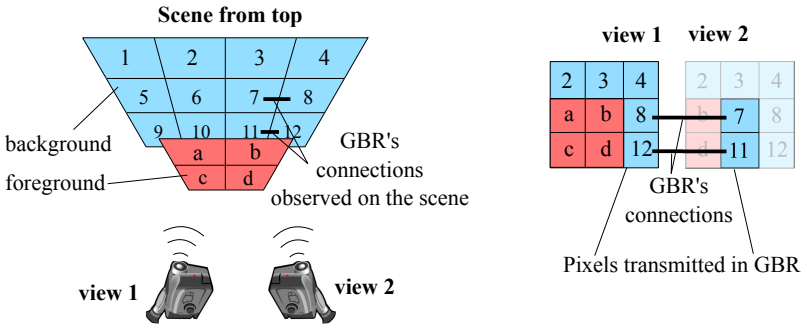


Figure 3.5: Illustration of GBR concepts for a simple scene.

Let us consider a scene captured by N cameras with the same resolution and focal length f. The image captured by the n^{th} camera is denoted by I_n, with $1 \leq n \leq N$, where $I_n(r, c)$ is the pixel at row r and column c in I_n. In a first step, we only consider translation between cameras, and we assume that the views are rectified (see Section 3.3.4 for more general camera configurations). In other words, the geometrical correlation between the views $\{I_n\}$ is only horizontal. We further assume that an accurate depth image, Z_n, is available at the encoder for every viewpoint I_n. We then compute $N-1$ dense disparity maps from these depth images. In what follows we assume that the set of images contains one *reference* view (typically the first, left-most, image) and $N-1$ *predicted* views.

We categorize the different types of pixels depending on how they change from one view to another (see Figure 3.6). Due to camera translation, a new part of the scene ap-

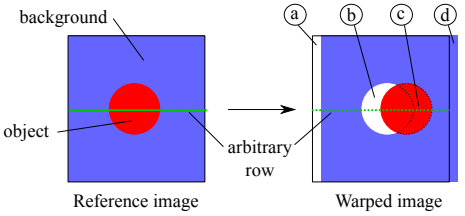


Figure 3.6: Illustration of camera translation for a simple scene with a uniform background, and one foreground object. Types of pixels in depth-based inter-view image warping: pixels can be a) appearing, b) disoccluded, c) occluded and d) disappearing. The green plain line is an arbitrary row in the reference view and the dashed line is the corresponding row in the target view.

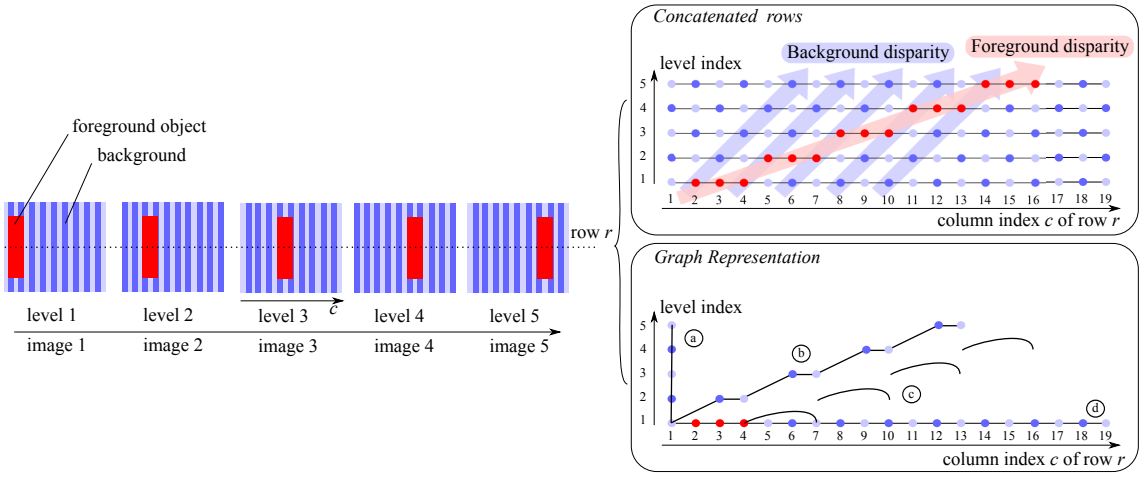


Figure 3.7: Graph construction example: the blue texture background has a disparity of 1 at each view and the red rectangle foreground has a disparity of 3 for each view. This example graph contains all different types of pixels: a) appearing, b) disoccluded, c) occluded and d) disappearing.

pears on the right or left of the image (*appearing* pixels) and another part disappears (*disappearing* pixels). As we move from one camera to a next, foreground objects move faster than the background. As a result, some background pixels may appear behind objects (*disoccluded* pixels). Conversely, some background pixels may become hidden by a foreground object (*occluded* pixels). If we consider a pair of views (reference and target), a row of the target view can be reconstructed by copying pixels from the corresponding row of the reference view, except when the above mentioned types of pixels occur (in which case "new" pixels have to be inserted). Our graph approach directly conveys this information by transmitting either i) a link to *the location in the reference row* where pixels should be copied from, or ii) *the values of new pixels* to be inserted.

A graph with N layers describes 1 reference view and $N-1$ predicted views. Its construction uses the information provided in the depth maps Z_n, $1 \leq n \leq N-1$. The constructed graph is made of two components, which are described by two matrices of size $N \times W$, where N is the number of layers (*i.e.*, the number of views encoded by the graph) and W is the image width. These two matrices are the color values $\mathbf{\Lambda}_r$ and the connections $\mathbf{\Gamma}_r$ and represent color and geometry information for all pixels of all views, where r is the row index (a pair of matrices per row). The matrices $\mathbf{\Lambda}_r$ and $\mathbf{\Gamma}_r$ are generated based on

the following principles. Pixel intensity values are stored in the layer (view) where they appear first. This means that a given layer only contains pixels that were not present in a lower layer. Then, the connections simply link these "new" pixels to the position of their neighbor in the previous layer. We show in Figure 3.7 a simple graph construction example for a given row r, with 5 levels (1 reference and 4 predicted views). At the decoder side, the reconstruction involves traversing the graph (left to right) and copying pixel values. An example for the synthesis of view 2 is depicted with the green arrows in Figure 3.8.

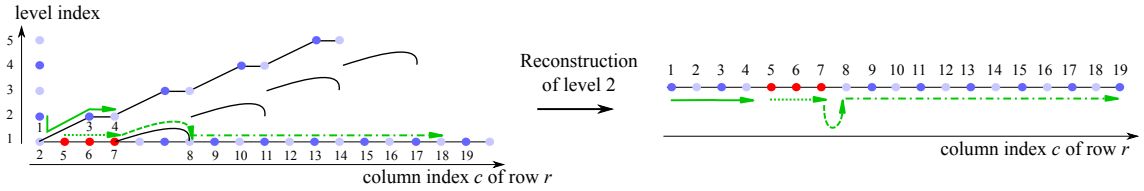


Figure 3.8: Reconstruction of level 2 with the toy example of Figure 3.7. The green arrows indicate the graph exploration order for view reconstruction.

Graph-based representation has several advantages: i) it removes inter-view redundancy (*i.e.*, a pixel appears only once), ii) it connects neighbors in the 3D scene, which is useful for color compression (see Section 3.3.3) and iii) it represents the geometry information in a compact form (see Sec 3.3.2).

For the sake of conciseness and clarity, we consider that the disparity is integer, but non-integer disparity have been tackled in the developed algorithms.

3.3.2 Retrieving the geometry from the graph

As claimed previously, the graph-based representation captures the geometry information. In other words, from the connections stored in the geometry matrices $\{\mathbf{\Gamma}_r\}_r$, we are able to retrieve the disparity information, hence the geometry of the scene.

Concretely, a disparity value corresponds to the shift that a pixel does between two views. Looking at the way the views are reconstructed from the proposed GBR (illustrated in Figure 3.8), the shift for each pixel can be estimated during the reconstruction process. Let us take the example of the reconstruction of view 2 in Figure 3.8. At the beginning, one appearing pixel is present (*i.e.*, pixel of index 1). The disparity of the pixel of index 2 in the view 1 is thus equal to 1. Then, two disoccluded pixels occur (*i.e.*, of index 3 and 4), implying that the disparity of pixels of index 5, 6 and 7 have a disparity of $1 + 2 = 3$. The rest of the disparity values are retrieved on the same principle. Since, the retrieved disparity corresponds to the shift used to synthesize the views, it naturally corresponds to the exact geometry precision needed at the decoder. We illustrate this nice property in Figure 3.9. A graph-based representation is built from the original depth maps in (a) and (b). The geometry retrieved at the decoder is depicted in (e) and (f). More precisions on how the graph is coded can be retrieved in (J10). As a comparison, we show in (c) and (d) the depth compressed with the same bitrate as the graph using a state-of-the-art depth coder. We can clearly see that the GBR depth is sharper and thus more adapted to the rendering task. This is confirmed by the rate-distortion comparison shown in Table 3.1.

Another illustration is proposed in Figure 3.10, where the disparity is retrieved for two configurations: rendering of view 2 and view 3 (that is more distant). We can see that, depending on the rendering task, the retrieved depth has different precision levels, demonstrating that GBR well adapts to what is needed for the rendering at the decoder.

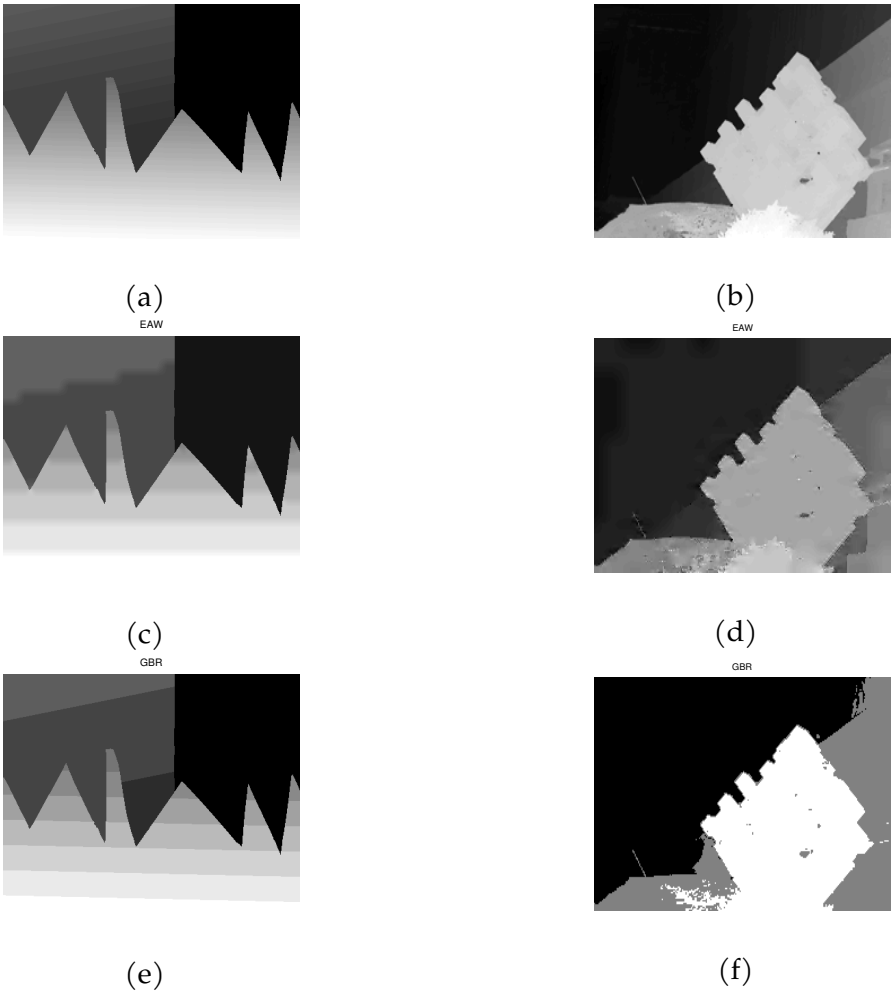


Figure 3.9: Geometry images for "Sawtooth" (left) and "Statue" (right) sequences. Subfigures (a) and (b) are the original depth maps. Subfigures (c) and (d) are the depth maps coded with edge-adaptive wavelet (EAW) based coder [102], while (e) and (f) are geometry images extracted from our GBR. In these visual examples, the geometry coding rate of EAW is equal to the rate of our GBR (30 kb for "Sawtooth" and 10 kb for "Statue").

3.3.3 Color compression using GBR

As seen in the previous section, graph-based representations describe the geometry in a compact form. Another advantage of such approach is that the topology links pixels that are neighbors in the 3D scene (exactly as a 3D mesh, see Figure 3.5). Therefore, this graph can be used to compress the color information. In (C24), we have designed a graph wavelet transform along with a adapted SPIHT algorithm to compress the color information. The geometry information can even be used to set graph weighs. An example of obtained rate-distortion results is shown in Figure 3.11. The proposed approach is denoted by *wGBT*. We consider a second version of the proposed approach denoted by *nGBT*, in which no weight is assigned on the edges. We compare these two approaches with a simple differential coding on the graph (*DC*) and a shape adaptive transform (*SA*) as proposed in [103]. We can conclude that the proposed graph wavelet is able to compact the color more efficiently than other baseline methods. Moreover, we can see that the weights are also useful to accurately describe the inter-pixel correlation.

Table 3.1: Rate comparison between GBR and baselines compression methods with synthesized views at 0.05 dB from the optimal quality.

	GBR vs JP2K	GBR vs HEVC	GBR vs EAW
"Couch"	-20.0%	-2.0%	-46.6%
"Statue"	-43.5%	$+161\%$	-6.87%
"Bikes"	-39.9%	-26.0%	-25.5%
"Church"	-45.0%	$+16.7\%$	-30.3%

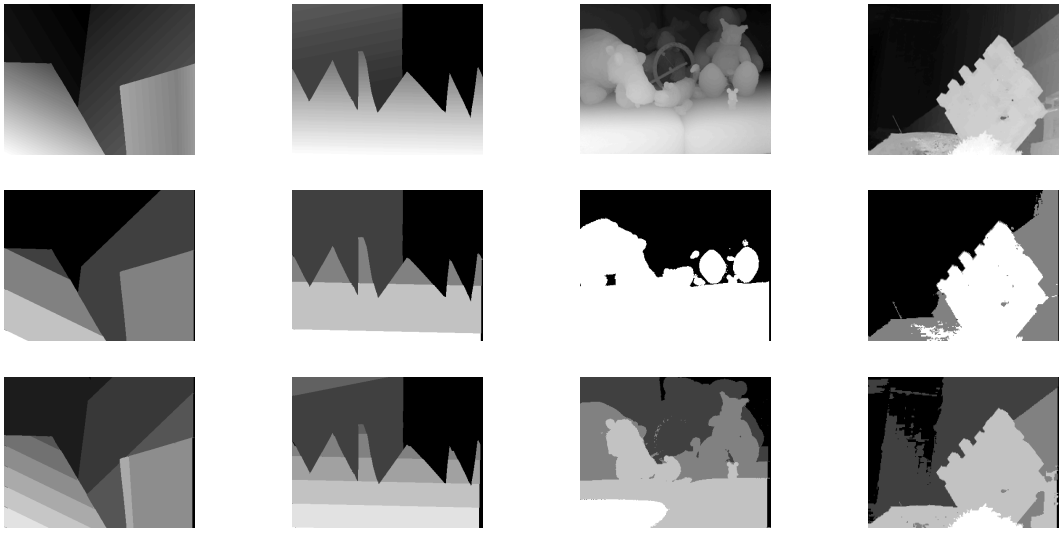


Figure 3.10: Illustration of depth images (first row), GBR geometry for view 2 prediction (second row), and GBR geometry for view 3 prediction (third row).

3.3.4 Extensions

The graph-based representation described above relies on the assumption that the views are vertically aligned. In practice, such camera configuration is not always conceivable. The cameras can for example have rotations between them. They can also lie on a 2D grid like for light field capture. In all these scenarios, the graph-based representation construction has to be adapted. In (J16), we have developed a graph-based representation for general camera configuration (including rotation, forward/backward displacements). Despite the complexity of the object transformation between the views, the constructed graph keeps its essential properties (see Figure 3.12): i) it is sparse, ii) it links neighboring pixels in the 3D scene and iii) it describes the geometry information. We have also extended our proposed representation to the Light Field format in (C30). In order to guarantee property i), we have proposed a graph sparsification method enabling to reduce the graph coding cost.

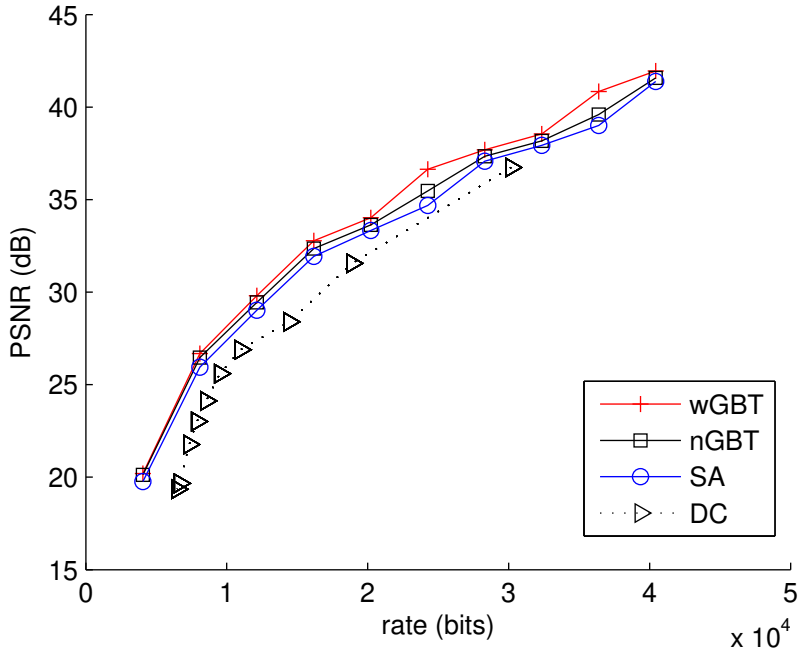


Figure 3.11: Rate-distortion evaluation for multi-view coding (geometry+luminance).

(**C19**) T. Maugey, A. Ortega, P. Frossard *Graph-based representation and coding of multiview geometry*, IEEE ICASSP, Vancouver, May, 2013

(**C20**) T. Maugey, A. Ortega, P. Frossard *Multiview image coding using graph-based approach*, IEEE IVMSP, Seoul, Korea, June, 2013

(**C21**) T. Maugey, A. Ortega, P. Frossard *Graph-Based vs Depth-Based Data Representation for Multiview Images*, IEEE Asilomar CSSC, Pacific Grove, CA, USA, Nov, 2013

(**C24**) T. Maugey, Y.H. Chao, A. Gadde, A. Ortega and P. Frossard *Luminance coding in graph-based representation of multiview images*, IEEE ICIP, Paris, France, Oct., 2014

(**C30**) X. Su, T. Maugey, C. Guillemot, *Graph-based representation for multiview images with complex camera configurations*, IEEE ICIP, Phoenix, Arizona, Sep. 2016.

(**J10**) T. Maugey, A. Ortega, P. Frossard *Graph-based representation for multiview image geometry*, in IEEE Transactions on Image Processing, Vol. 24(5) , pp. 1573 - 1586, 2015.

(**J16**) X. Su, T. Maugey, C. Guillemot *Rate-distortion optimized graph-based representation for multiview images with complex camera configurations*, in IEEE Transactions on Image Processing, Vol 26(6), p. 2644–2655, Jun. 2017 2017.

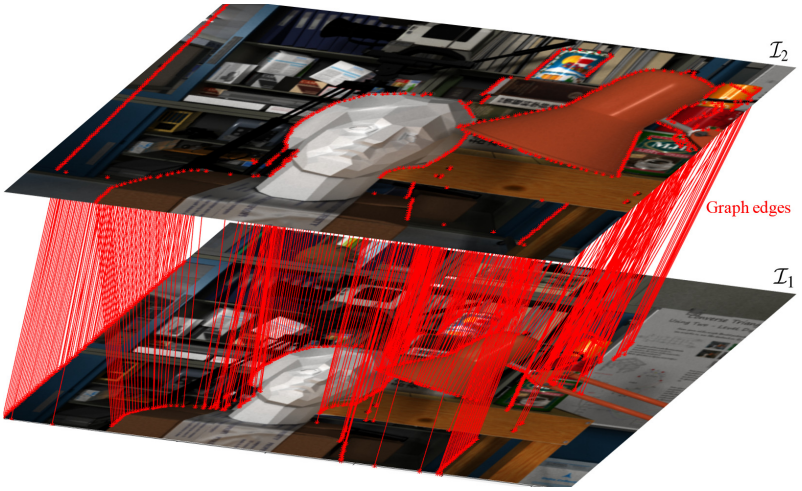


Figure 3.12: Graph-based representation for complex camera configuration.

3.4 Conclusion

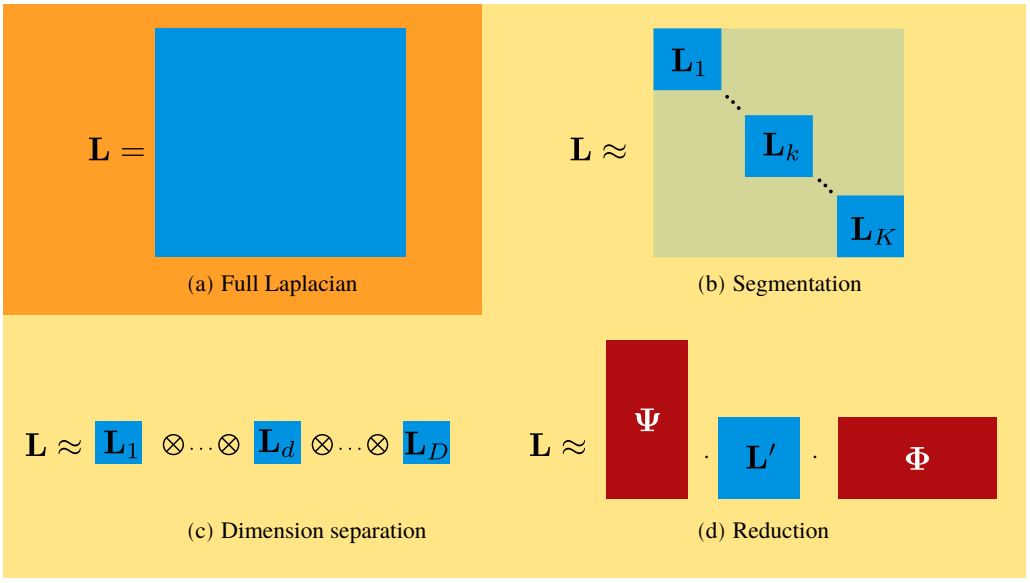
In this Chapter, we have presented some proposed solutions to tackle the fact that some 3D visual data rely on irregular domain. Based on the geometry information, we have built a graph connecting pixels that are close in the 3D space, assuming that they should be correlated. In particular, we have proposed graph construction strategy to handle various types of 3D data such as multi-view, 360°, Light-Field and 3D mesh. Each time, we have proven that the proposed graph enables to extend the benefits of a transform-based energy compaction on the 3D data topology directly even though it is irregular. We have even shown that this graph, if transmitted, could be used to retrieve the geometry information and thus to save bitrate by not sending it to the decoder. Despite its high potential, the graph-based transform is computed at the price of a huge computation cost, which is not always conceivable in practice. In the next Chapter, we explain the methods that we have proposed to circumvent this issue.

Chapter 4

Graph-based transform for high-dimensional data

4.1 Graph-based transform and complexity issue

As discussed in the previous chapter, graphs have been shown to be useful tools to model data correlation within many types of images, and to define transform supports for decorrelation and signal compaction. However, the high dimensionality and resolution of the data in hand have obvious implications on the storage footprint of the Laplacian matrix $\mathbf{L}$ (whose dimensions is $N \times N$, where N is the number of pixels in the image) and on the transform computational complexity, which can make graph-based transforms impractical. Indeed, estimating the transform basis implies to diagonalize the Laplacian matrix (see Equation (3.1)). Such operation has a complexity of $\mathcal{O}(N^3)$ and becomes rapidly intractable when the dimension of the Laplacian increases. One solution is thus to limit the Laplacian size, while preserving a good data representation efficiency. We have investigated three solutions to decrease the dimension of the Laplacian itself (see Figure 4.1 below): segmentation, dimension separation and reduction. We detail each of these solutions in the next three sections.



(a) Full Laplacian (b) Segmentation

(c) Dimension separation (d) Reduction

Figure 4.1: Envisaged solutions to decrease the Laplacian dimension.

4.2 Graph segmentation

4.2.1 Motivations and problem

To reduce the dimension of the graphs, one can construct local graphs by first segmenting the data. This amounts to approximating the full Laplacian $\mathbf{L}$ by a block diagonal matrix, $\tilde{\mathbf{L}}_{\text{seg}}$ where the blocks $\mathbf{L}_k$ can be diagonalized independently (see Figure 4.1(b)), decreasing the overall complexity. The question is how to efficiently split the graph into different sub-graphs ? Or said differently, what edges have to be cut ?

As discussed in the previous chapter, the graph-based transform, defined as the eigenvectors of $\mathbf{L}$, is efficient to compact a signal $\mathbf{z}$ if the signal is smooth on the graph. This smoothness is measured by the total variation

$$\text{TV}_{\mathbf{L}}(\mathbf{z}) = \mathbf{z}^\top \mathbf{L}\mathbf{z}. \tag{4.1}$$

The transform enables to compact $\mathbf{z}$ if $\text{TV}_{\mathbf{L}}(\mathbf{z})$ is small. When the graph is constructed based only on the 3D data geometry $\boldsymbol{\gamma}$ (as in the previous Chapter), it may happen that large variations occur along an edge. Hence, cutting such an edge tends to decrease the total variation. Said differently, a good graph segmentation leads to

$$\text{TV}_{\tilde{\mathbf{L}}_{\text{seg}}}(\mathbf{z}) \leq \text{TV}_{\mathbf{L}}(\mathbf{z}), \tag{4.2}$$

since it enables to reduce the transmission cost of the transformed coefficients. If the signal $\mathbf{z}$ is used to perform the graph segmentation, this segmentation has to be transmitted to the decoder as well, leading to an extra cost, hopefully balanced by the coefficient transmission cost reduction. In the following, we introduce two solutions proposed to perform such segmentation and their associated coding scheme. The first one estimates the segmentation based on the signal $\mathbf{z}$ of one view of a light field and extrapolates it on all other views. In the second one, we propose a rate-distortion optimized segmentation for 360° data.

4.2.2 Super-ray segmentation

A light field is typically made of more than 12 millions of pixels. Obviously, it is inconceivable to diagonalize a Laplacian matrix of size 12 millions $\times$ 12 millions. Segmenting the graph is thus unavoidable. As stated before, a good segmentation keeps, in each subgraph, pixels that are correlated (in order to keep the total variation low). Two types of correlation occur for light field.

Spatial correlation: within each sub-aperture image, neighboring pixels representing the light activity of two points close in the 3D world (*i.e.*, belonging to the same object) might be correlated. They can be gathered in a same group called *super-pixel* [104]. Different segmentation methods can be considered either using normalized cuts on graphs or graph-cut techniques [105], or using clustering as performed by the Simple Linear Iterative Clustering (SLIC) [106] or the SEED [107] algorithms, leading to so-called superpixels. In our proposed solutions in (C35, C35, J21), we have used the SLIC algorithm as depicted in Figure 4.2.

Angular correlation: across the light fields sub-aperture images, some pixels describe the light activity emitted from the same 3D point. They are naturally correlated between each other. This is the reason why, the concept of super-rays introduced in [108] extends super-pixels to 4D light fields by grouping light rays coming from the same 3D object. The method performs a k-means clustering of all light rays based on color and distance in the

Figure 4.2: An example of super-pixel segmentation for spatial correlation.

3D space. This method does not really suits for compression because the 4D segmentation is costly to describe. We have proposed an alternative solution that consists in projecting the SLIC-based super-pixels estimated on one view onto every other views using the disparity map. Therefore, only the super-pixel segmentation and the disparity map have to be transmitted to the decoder.

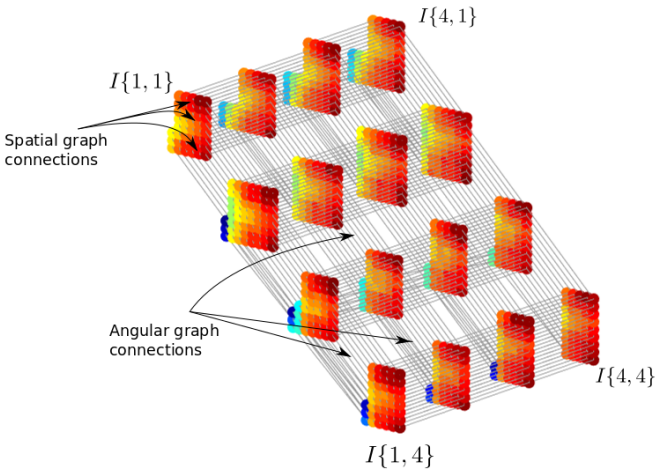
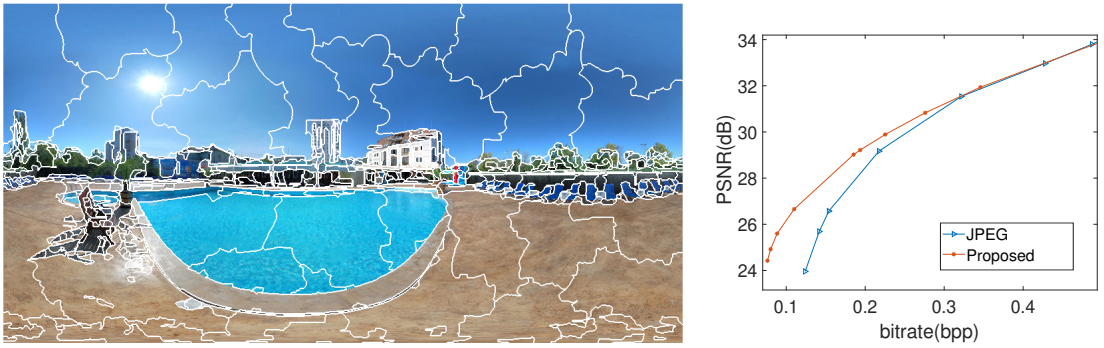


Figure 4.3: An example of a graph built over a super-ray. The color corresponds to the pixel intensity.

Coding strategy: Once the super rays are constructed, a graph is then built to connect neighboring pixels as depicted in Figure 4.3. Each super-ray k is assigned to a Laplacian $\mathbf{L}_k$ that is diagonalized. This approach remains quite complex since the different $\mathbf{L}_k$ are quite highly dimensional because of the large number of views in the light field. This



(a) (b)

Figure 4.4: Rate-distortion optimized graph segmentation for 360°.

approach will be referred as the *non-separable transform* in the following. Rate-Distortion comparison involving this method will be provided later in this Chapter.

4.2.3 Rate-distortion optimized segmentation

Even if the above methods, essentially relying on color similarity and distances in space, are designed so that super-pixels or super-rays adhere well to the boundaries of objects, they do not explicitly rely on smoothness constraints of the local graphs that can be constructed on these regions. Yet smoothness or total signal variation can be a useful criterion to define the local graph supports when the goal is signal energy compaction that has a direct impact on the bit rate in a compression context. Moreover, in the methods described above, the cost of transmitting the segmentation boundaries is not taken into account in the cutting decision. In (C36), we have proposed a rate-distortion optimized segmentation of the graph-based domain describing a 360° image. The problem is posed as follows:

$$\begin{aligned} \min_{\tilde{\mathcal{G}}=\{\mathcal{G}_k\}} \quad & \mathcal{D}(\tilde{\mathcal{G}}) + \gamma \mathcal{R}_C(\tilde{\mathcal{G}}) + \beta \mathcal{R}_B(\tilde{\mathcal{G}}), \\ \text{subject to} \quad & N_k < N_{max}, \ \forall i \end{aligned} \tag{4.3}$$

where $\tilde{\mathcal{G}} = \{\mathcal{G}_k\}$ is the *global graph partitioning*, *i.e.*, the global graph in which 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 graph partitioning description. The size of each graph N_k is constrained to be smaller than a maximum tolerable number of pixels N_{max}.

The impact of the partitioning on the distortion term $\mathcal{D}(\tilde{\mathcal{G}})$ is assumed to be negligible. The term $\mathcal{R}_C(\tilde{\mathcal{G}})$ is modeled using the total variation as it is done in [109]:

$$\mathcal{R}_C(\mathcal{G}_k) \propto \mathrm{TV}_{\mathbf{L}_k}(\mathbf{z}_k), \tag{4.4}$$

where $\mathbf{L}_k$ is the Laplacian of graph $\mathcal{G}_k$ and $\mathbf{z}_k$ is the signal lying on $\mathcal{G}_k$. The rate of the partitioning $\mathcal{R}_B(\tilde{\mathcal{G}})$ is modeled as a function of the boundary length and the regularity of its shape. The optimization problem is solve with an original iterative normalized cut algorithm. Visual result of the partitioning is depicted in Figure 4.4(a) and rate distortion evaluation is shown in Figure 4.4(b). We can see that the proposed method enables to outperform JPEG, showing that the optimization of criterion in Equation 4.3 leads to coding gain compared to a fixed size transform.

(**C35**) X.Su, M. Rizkallah, T. Maugey, C. Guillemot, *Rate-Distortion Optimized Super-Ray Merging for Light Field Compression* EUSIPCO, Athens, Greece, Sept. 2018.

(**C36**) M. Rizkallah, F. De Simone, T. Maugey, C. Guillemot, P. Frossard, *Rate Distortion Optimized Graph Partitioning for Omnidirectional Image Coding* EUSIPCO, Athens, Greece, Sept. 2018. *Best Student Paper*

(**J21**) M. Rizkallah, T. Maugey, C. Guillemot, *Prediction and Sampling with Local Graph Transforms for Quasi-Lossless Light Field Compression*, vol. 29, pp. 3282 – 3295, Dec. 2019.

4.3 Separable transform

As stated in Section 4.2.2, even after a graph segmentation, the size of each subgraph can remain too large. In that case, one needs to exploit other methods to decrease the complexity. In this Section, we explore how separable transform can be defined (see Figure 4.1(c)). First, we define what is a separable transform, and we discuss its impact on the Laplacian diagonalization. Then, we explain how this factorization can be considered on an irregular graph, and what problem it raises. Finally we show how we applied this separable transform in the context of light field compression.

4.3.1 Definition

Let us assume that the signal $\mathbf{z}$ is of dimension N, and that there exist $\{M_d\}_{d\in[1,D]}$ such that $N = M_1 \times M_2 \times \ldots \times M_D$. Let us further assume that there exists a function called the *index factorization*:

$$\begin{aligned} \sigma\colon [1:N] &\to [1:M_1] \times \ldots \times [1:M_D] \\ i &\mapsto (j_1, \ldots, j_D) \end{aligned} \tag{4.5}$$

that maps the index of i into a vector of size D. Each j_d may be interpreted as the position of index i with respect to the d^{th} dimension axis (such as time, space, *etc.*). Thanks to this function σ, we are able to rewrite $\mathbf{z}$ as a tensor $\mathbf{z}_\sigma$ of order D. Let $\mathbf{z}_\sigma^d(j_1, \ldots, j_{d-1}, j_{d+1}, \ldots, j_D)$ be the extracted vector along the d^{th} dimension at position $(j_1, \ldots, j_{d-1}, j_{d+1}, \ldots, j_D)$.

A separable linear transform [110, 111] is composed of a set of d matrices $\mathbf{U}_d$ each of them, operating on the d^{th} dimension axis. The separable transform is applied successively on each dimension. In more details, the transformed tensor $\hat{\mathbf{z}}_\sigma$ is first initialized with $\mathbf{z}_\sigma$. Then, the D^{th} transform is applied as follows $\forall\ (j_1, \ldots, j_{D-1})$:

$$\hat{\mathbf{z}}_\sigma^D(j_1, \ldots, j_{D-1}) \leftarrow \mathbf{U}_D \hat{\mathbf{z}}_\sigma^D(j_1, \ldots, j_{D-1}). \tag{4.6}$$

Then, successively, the d^{th} transforms are applied $\forall\ (j_1, \ldots, j_{d-1}, j_{d+1}, \ldots, j_D)$:

$$\hat{\mathbf{z}}_\sigma^d(j_1, \ldots, j_{d-1}, j_{d+1}, \ldots, j_D) \leftarrow \mathbf{U}_d \hat{\mathbf{z}}_\sigma^d(j_1, \ldots, j_{d-1}, j_{d+1}, \ldots, j_D). \tag{4.7}$$

When computing the d^{th} transform in (4.7), the elements of $\hat{\mathbf{z}}_\sigma$ are already transformed along the dimensions $d' > d$.

The order with which the transformed coefficients are computed is not important. However, given an order (let us say from D to 1), the above separable transform process is equivalent to perform the global transform:

$$\hat{\mathbf{z}} = \mathbf{U}\mathbf{z} \tag{4.8}$$

with

$$\mathbf{U} = \mathbf{U}_1 \otimes \ldots \otimes \mathbf{U}_d \otimes \ldots \otimes \mathbf{U}_D. \tag{4.9}$$

In that case, σ is simply the inverse of the usual "vec" that vectorizes a tensor across the dimensions. If the initial indices i are not arranged according to this natural index factorization, a permutation can be applied *a priori*. In the next Section, we discuss what could be the benefits of having such separable transform for the Laplacian diagonalization.

4.3.2 Separable Laplacian

From (3.1), the graph transform is obtained by a diagonalization of its Laplacian matrix $\mathbf{L}$. In the scenario where the graph's nodes can be factorized with a function σ as defined in the previous Section, and that the transform matrix $\mathbf{U}$ can be expressed as a separable transform (see (4.9)), we can write:

$$\begin{aligned}\mathbf{L} &= \mathbf{U}^\top \mathbf{\Lambda} \mathbf{U} \\ \mathbf{L} &= (\mathbf{U}_1 \otimes \ldots \otimes \mathbf{U}_D)^\top (\mathbf{\Lambda}_1 \otimes \ldots \otimes \mathbf{\Lambda}_D)(\mathbf{U}_1 \otimes \ldots \otimes \mathbf{U}_D).\end{aligned} \tag{4.10}$$

This expression can be factorized as

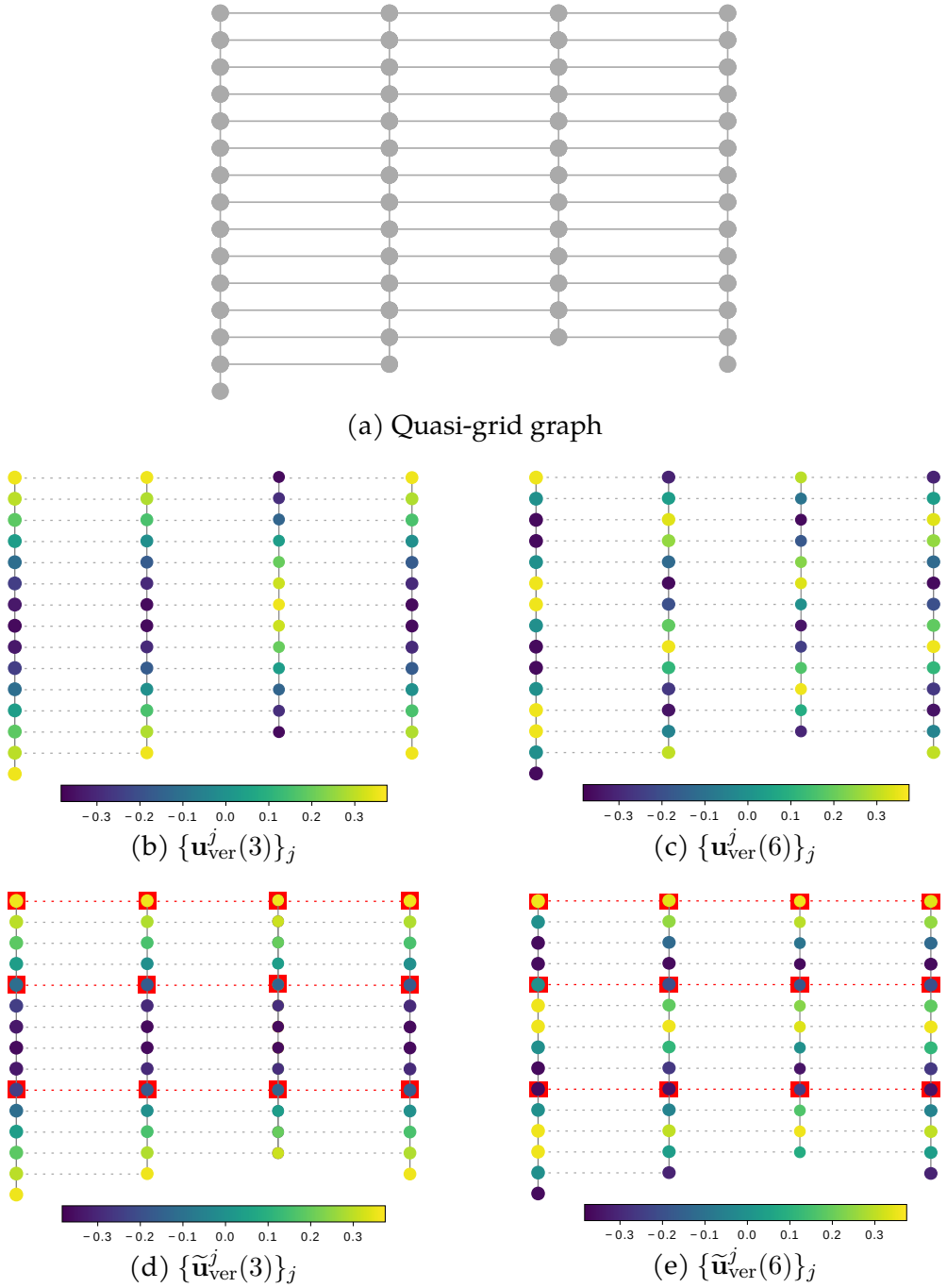
$$\begin{aligned}\mathbf{L} &= (\mathbf{U}_1^\top \mathbf{\Lambda}_1 \mathbf{U}_1) \otimes \ldots \otimes (\mathbf{U}_d^\top \mathbf{\Lambda}_d \mathbf{U}_d) \otimes \ldots \otimes (\mathbf{U}_D^\top \mathbf{\Lambda}_D \mathbf{U}_D) \\ \mathbf{L} &= \mathbf{L}_1 \otimes \ldots \otimes \mathbf{L}_d \otimes \ldots \otimes \mathbf{L}_D.\end{aligned} \tag{4.11}$$

It means that there is a direct link between a separable transform and a separable Laplacian. It implies that if one is able to write the Laplacian in a separable way, each sub-Laplacian can be diagonalized, and the global graph transform can be retrieved from each sub-transform matrix $\mathbf{U}_d$. This way of doing could considerably decrease the transform calculation complexity.

Defining a separable Laplacian is therefore interesting for complexity saving and parallelization. However, this is not always possible because of two reasons. First an index factorization σ does not always exist. This is even the general case, when the structure of the graph is arbitrary and is not ruled by natural underlying axes such as space or time. In Section 4.3.3, we explain how this separability can be approximated and what side problem it can raise. We finally propose a solution to tackle such problem and properly define a separable transform in practice.

4.3.3 Dimension factorization

First, it is important to recall that, in general, graphs *cannot* be expressed in a separable fashion. Indeed, the separability implies a grid shape of the graphs which remains quite specific. However, in some cases, some underlying dimensions (row, columns, angle, time, etc.) exist and the initial graph is not far from being approximated by a grid. We show in Figure 4.5(a), a toy example of such a graph. This graph is organized as rows and columns, with some missing nodes. Considering this graph as separable consists in defining a vertical and an horizontal transform. This however brings two problems.



(a) Quasi-grid graph

(b) $\{\mathbf{u}^j_{\text{ver}}(3)\}_j$ (c) $\{\mathbf{u}^j_{\text{ver}}(6)\}_j$

(d) $\{\widetilde{\mathbf{u}}^j_{\text{ver}}(3)\}_j$ (e) $\{\widetilde{\mathbf{u}}^j_{\text{ver}}(6)\}_j$

Figure 4.5: (a): a quasi-grid graph that is nearly separable, (b) and (c): examples of separable transform basis computed independently on the different "columns" of the graph, (d) and (e): separable transform basis aligned by minimizing Equation (4.12) with the matching nodes depicted in red.

First, the different vertical transforms may not be consistent between each other. More generally, in ideal separable scenario, the separable transform as computed in (4.7), uses the same transform matrix $\mathbf{U}_d$ along the dimension d. Said differently, the transform basis should be identical when applied at the different positions on the other factorized dimensions. It means, for the example of Figure 4.5, that the basis of the vertical transform should be the same for every column of the graph. This is by nature impossible because the number of nodes is not even the same. Moreover, when diagonalizing the sub-Laplacians, the eigenvectors are not consistent between each other, as depicted in Figure 4.5(b) and (c). Inspired by prior works [112], we have proposed a solution to align the basis. The goal is to build a consistent transform in each sub-graph. Let us denote by $\mathbf{L}_d^j = \mathbf{U}_d^{j\top} \mathbf{\Lambda}_d^j \mathbf{U}_d^j$ the d^{th} separable Laplacian for every position j over the remaining dimensions (*i.e.*, other than d). For the example of Figure 4.5, d corresponds to the vertical direction and the j corresponds to the different columns. When aligning the basis, one has to solve a general form as:

$$\min_{\{\widetilde{\mathbf{U}}_d^j\}_j} E_{\text{cons}}(\{\widetilde{\mathbf{U}}_d^j\}_j) + \lambda \sum_j E_{\text{comp}}(\widetilde{\mathbf{U}}_d^j). \tag{4.12}$$

On the one hand, the term $E_{\text{cons}}(\{\widetilde{\mathbf{U}}_d^j\}_j)$ controls the *consistency*, *i.e.*, the fact that the different basis functions of the transforms are consistent over the j. Some nodes of each column are identified as *matching nodes*, based, for example, on geometry information (depicted in red in Figure 4.5(d) and (e)). The term E_{cons} is set up such that the basis functions match on these matching nodes. More practically, a reference column j_{ref} is selected, and E_{cons} is defined as:

$$E_{\text{cons}}(\{\widetilde{\mathbf{U}}_d^j\}_j) = \sum_{j \neq j_{\text{ref}}} ||\mathbf{F}_{j_{\text{ref}}} \widetilde{\mathbf{U}}_d^{j_{\text{ref}}} - \mathbf{F}_j \widetilde{\mathbf{U}}_d^j||_2^2, \tag{4.13}$$

where the $\mathbf{F}_j$ are matrices with 1 on the matching nodes indices and 0 elsewhere. On the other hand, the term $E_{\text{comp}}(\widetilde{\mathbf{U}}_d^j)$ controls the *compaction* efficiency of each $\widetilde{\mathbf{U}}_d^j$. For example, in [112], this term is defined as the capacity of $\widetilde{\mathbf{U}}_d^j$ to diagonalize the corresponding Laplacian $\mathbf{L}_d^j$:

$$E_{\text{comp}}(\widetilde{\mathbf{U}}_d^j) = off(\widetilde{\mathbf{U}}_d^{j\top} \mathbf{L}_d^j \widetilde{\mathbf{U}}_d^j). \tag{4.14}$$

This optimization can be solved using a gradient descent. More details on the gradient expression can be found in (J20). An example of results obtained by such optimization is given in Figure 4.5(d) and (e). We can see how the basis now match on the red nodes, which implies more consistent behavior over the whole graph.

Once this first transform along the dimension d is applied, the second issue deals with the design of the next transform (*i.e.*, the horizontal one for the toy example of Figure 4.5). In an ideal grid, the *spectral bands* (*i.e.*, the coefficients corresponding to a given eigenvalue) are transformed together. However, in the case of a quasi-grid, the number of *bands* is different and they correspond to different eigenvalues over the different columns. Inspired by shape-adaptive DCT [113], this can be solved by regrouping the bands by eigenvalue *indices*. In other words, the transformed coefficients corresponding to eigenvalue i are coded together thanks to the eigenbases $\mathbf{U}_{d-1}^i$ (or $\mathbf{U}_{\text{hor}}^i$ in the example of Figure 4.5). Since the topology is not perfectly regular, it may happen that the number of nodes for each i is not the same. For example, in Figure 4.5, the 14^{th} and 15^{th} row are not complete. To deal with this situation, we propose to consider the uncomplete rows as new vertical subgraphs where additional edges are added so that the graph is connex. New $\mathbf{U}_{d-1}^i$ are computed (with the good number of nodes) and possibly aligned with the oth-

ers $\mathbf{U}_{d-1}^{i}$ as explained before. In the following, we apply such dimension factorization strategy to the problem of light field compression.

4.3.4 Application to Light field compression

When dealing with super-rays defined on light fields (see Figure 4.3), one typically has a quasi-grid graph as in Figure 4.5(a). Indeed, the super-pixel's shape may vary across the views due to, for example, occlusion of a background. This may cause basis misalignment problems that should be solved with our proposed method introduced in the previous Section. In Figure 4.6, we show an example of alignment of the spatial transform basis, *i.e.*, the basis of the transform performed in the super-pixel of each view.

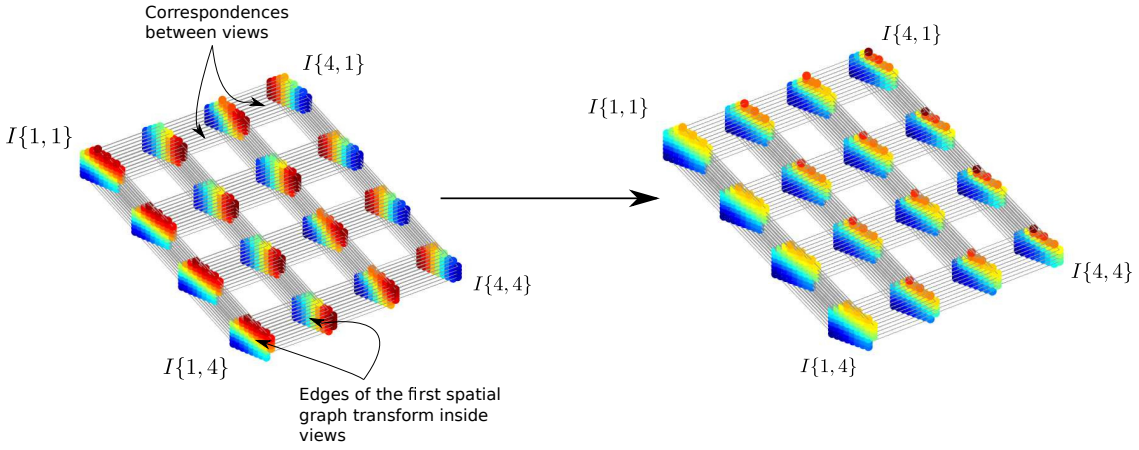


Figure 4.6: Second eigenvector of different super-pixels belonging to the same super-ray before and after optimization.

We implement a light-field coding scheme using these separable graph-based transform and we compare its performance with the non-separable case (introduced in Section 4.2.2) and the non-optimized transform, *i.e.*, no basis alignment. A typical rate-distortion result is shown in Figure 4.7.

We can first see the effect of the proposed alignment algorithm that clearly outperforms the scheme using the misaligned basis. Unfortunately, the optimized separable transform does not exactly reach the non-separable transform performance, especially at high bitrate. This is explained by the fact that the basis alignment is more difficult for high "frequencies" (*i.e.*, highly varying eigenvectors). To conclude, the complexity is drastically decreased with a separable transform with a small price to pay on the coding performance. In the next Section, we present a third approach to reduce the Laplacian diagonalization complexity, namely the graph reduction.

(J20) M. Rizkallah, T. Maugey, C. Guillemot, *Geometry-Aware Graph Transforms for Light Field Compact Representation* in IEEE Transactions on Image Processing, vol. 29, pp. 602–616, Jul. 2019.

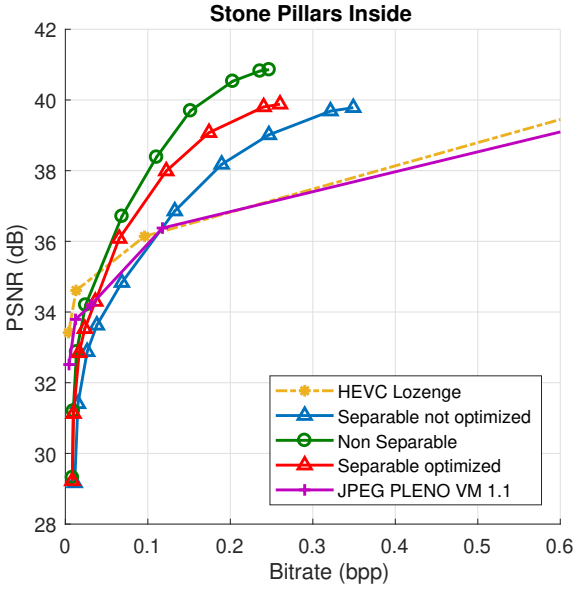
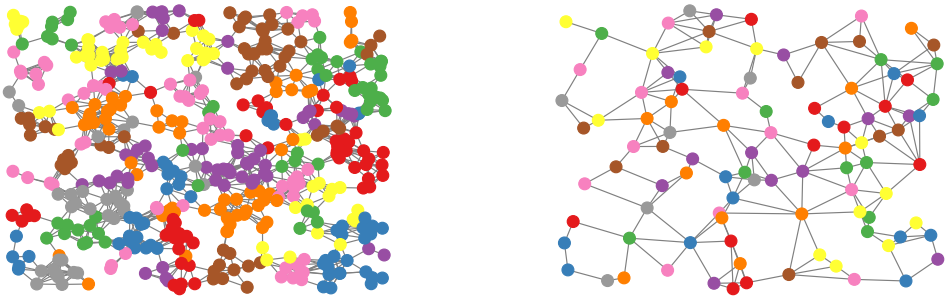


Figure 4.7: Example of Rate Distortion performance obtained with our graph based coding scheme (Non separable, not optimized and optimized separable graph transforms) compared to Light Field coding baselines.

4.4 Graph reduction

4.4.1 Motivations

As it was explained in Section 4.2.1, estimating the full Laplacian $\mathbf{L}$ with a block diagonal matrix aims at keeping the dimension of each sub-block $\mathbf{L}_k$ below a certain size while, at the same time, decreasing the total variation (see Equation (4.2)). Unfortunately, this goal may not be always achievable. As an example, let us take a signal $\mathbf{z}$ that is extremely smooth over the whole graph. Because of the Laplacian size constraint, one needs to cut the global graph into, for example, two subgraphs. Given that $\mathrm{TV}_{\mathbf{L}}(\mathbf{z})$ is already very low, it is more likely that one cannot find any edge sets for which $\mathrm{TV}_{\tilde{\mathbf{L}}_{\mathrm{seg}}}(\mathbf{z}) < \mathrm{TV}_{\mathbf{L}}(\mathbf{z})$. In that case, a segmentation would decrease the coding performance and alternative solutions have to be explored. Note that if the signal $\mathbf{z}$ is smooth on the graph, it means that in the transformed domain, a few coefficients could be sufficient to describe it. It means that its dimension could be reduced. The only barrier is that the computation of such transformed coefficient is intractable. To circumvent this issue, one may think of reducing the dimension *a priori*. We could approximate the full $N \times N$ Laplacian $\mathbf{L}$ by a Laplacian $\mathbf{L}'$ of smaller dimension $N' \times N'$, using projection and back-projection matrices (see Figure 4.1(d)). Same matrices should be used to reduce the dimension of the signal $\mathbf{z}$ as well. If this signal is sufficiently smooth, its low-dimension approximation $\mathbf{z}'$ may not cause too much error at the reconstruction stage. This is the intuition behind graph reduction techniques. In the next Section, we focus on one particular type of graph reduction, namely the graph coarsening. We then explain how we used this technique for the context of light field compression.



(a) Big graph (b) Coarsened graph

Figure 4.8: Illustration of coarsening result with [114]. Colors depicts the pixels that are merged during the coarsening process.

4.4.2 Graph coarsening principles

Graph reduction consists in finding a proper reduction matrix $\mathbf{P} \in \mathbb{R}^{N' \times N}$, used to define the reduced signal and Laplacian as:

$$\begin{aligned} \mathbf{z}' &= \mathbf{P}\mathbf{z} \\ \mathbf{L}' &= \mathbf{P}^{\mp}\mathbf{L}\mathbf{P}^{+}. \end{aligned} \tag{4.15}$$

Symbols $+$ and $\mp$ denote the pseudo-inverse and the transpose pseudo-inverse respectively. The reduced signal $\mathbf{z}'$ can be lifted back to the original dimension by doing the following operation:

$$\tilde{\mathbf{z}} = \mathbf{P}^{+}\mathbf{z}'. \tag{4.16}$$

Coarsening is a special case of graph reduction abiding to a set of constraints that render the graph transformation explainable. Contraction sets are formed from the vertices v_i. As such, every reduced variable corresponds to a small set of adjacent vertices in the original graph and coarsening amounts to a scaling operation. The set of adjacent vertices is denoted by $\mathcal{V}^{(r)}$ and is called a contraction set to produce one vertex in the reduced graph. Neighboring vertices belonging to the same contraction set are depicted with similar colors in Figure 4.10(a). The coarsening matrix $\mathbf{P}$ satisfies two important conditions. First each node belongs to one and only one contraction set. Second, the contraction set should be connected in the graph. This method of constructing the matrix $\mathbf{P}$ enables a simple inversion of $\mathbf{P}$. Moreover, if we constrain all non zero entries of $\mathbf{P}^{+}$ to be equally valued (each node is aggregated with the same weight), then the resulting coarsened matrix $\mathbf{L}'$ is also a graph Laplacian matrix. In that case the matrices $\mathbf{P}$ and $\mathbf{P}^{+}$ are defined $\forall r < N'$ and $\forall i < N$ as:

$$[\mathbf{P}](r,i) = \begin{cases} \frac{1}{\|\mathcal{V}^{(r)}\|} & \text{if } v_i \in \mathcal{V}^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{4.17}$$

$$[\mathbf{P}^{+}](i,r) = \begin{cases} 1 & \text{if } v_i \in \mathcal{V}^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{4.18}$$

The resulting Laplacian $\mathbf{L}'$ is of a smaller dimension and is thus simpler to diagonalize. The work in [114] shows that a graph can be reduced such that its fundamental structural properties are preserved namely its first eigenvectors and eigenvalues. Authors in

[114] are thus able to guarantee a low reconstruction error between $\mathbf{z}$ and $\tilde{\mathbf{x}}$ for sufficiently smooth signal $\mathbf{z}$. In the following, we explain how we have used this coarsening technique in a light field compression algorithm.

4.4.3 Application to Light Field compression

Using the technique described above, we have proposed a rate-distortion optimized super-ray partitioning for light field compression. Contrary to the fixed-size approach described in Section 4.2.2, we intend to adjust the size of the super-ray to the local signal statistics. The proposed strategy is summarized in Figure 4.9. Basically, if the signal is smooth on the graph, we define large super-rays and we use coarsening techniques to reduce its dimension. On the contrary, if the signal is not smooth, we cut the graph into two sub-graphs to lower the total variation. On top of the signal smoothness, we consider the cost of boundary description.

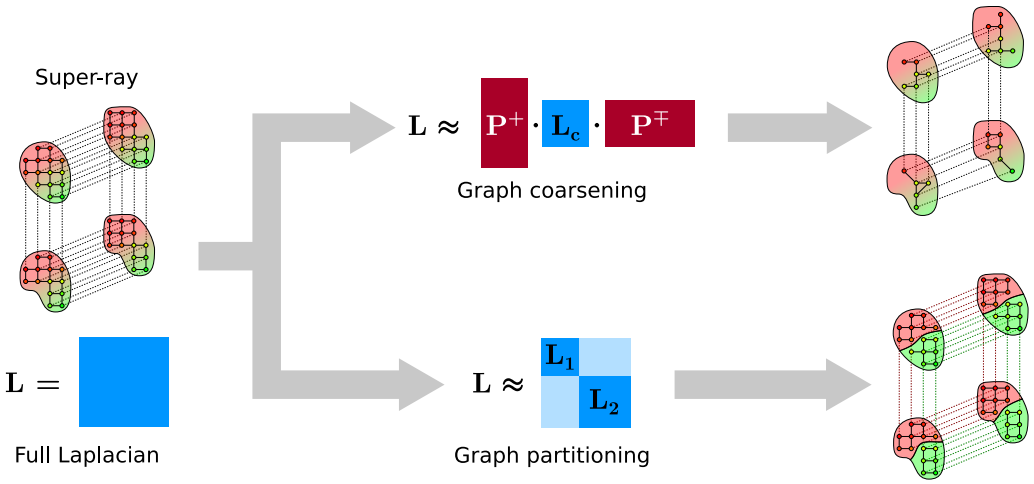


Figure 4.9: When the full Laplacian is to heavy to compute, two strategies are possible depending on how smooth is the signal on the graph.

The optimal partitioning problem is formalized as follows:

$$\begin{aligned} \min_{\tilde{\mathcal{G}}=\{\mathcal{G}_k\}} \quad & \mathcal{R}_C(\tilde{\mathcal{G}}) + \mathcal{R}_B(\tilde{\mathcal{G}}) \\ \text{subject to} \quad & \mathcal{D}(\mathcal{G}_k) < D_{max} \ \ \forall k \end{aligned} \tag{4.19}$$

$\tilde{\mathcal{G}} = \{\mathcal{G}_k\}$ represents the set of *local graphs* capturing local color information and the color variation inside the 4D light field. $\mathcal{D}(\mathcal{G}_k)$ is the distortion between the original signal and the reconstructed one on the k^{th} graph, $\mathcal{R}_C(\tilde{\mathcal{G}})$ is the rate cost of the quantized transform coefficients sent to the decoder side, and $\mathcal{R}_B(\tilde{\mathcal{G}})$ is the rate cost of the boundaries for the graph partitioning description. We assume that the maximum tolerated Laplacian size is $N_{\max}$. Which means that if $\mathcal{G}_k$ has a size $N_k > N_{\max}$, a coarsening is performed, possibly impacting $\mathcal{D}(\mathcal{G}_k)$.

We propose to solve this problem iteratively. We start from large size super-rays, and for each of them, we decide if it is beneficial (in terms of criterion in Equation 4.19) to split it into two sub-graphs. Then the same decision is estimated for each of the sub-graphs and so forth. An example of the obtained partitioning is shown in Figure 4.10. We can see that smoothest regions have large super-ray and more texture regions as the background is made of smaller super-rays. We show in Figure 4.11 a typical results obtained

Figure 4.10: Partitioning results obtained with our solution.

by our proposed partitioning. We can see that our proposed method clearly outperforms the fixed-size super-rays as proposed in Section 4.2.2. As shown by the blue and yellow curves, this gain is both due to the coarsening and a rate-distortion optimized partitioning.

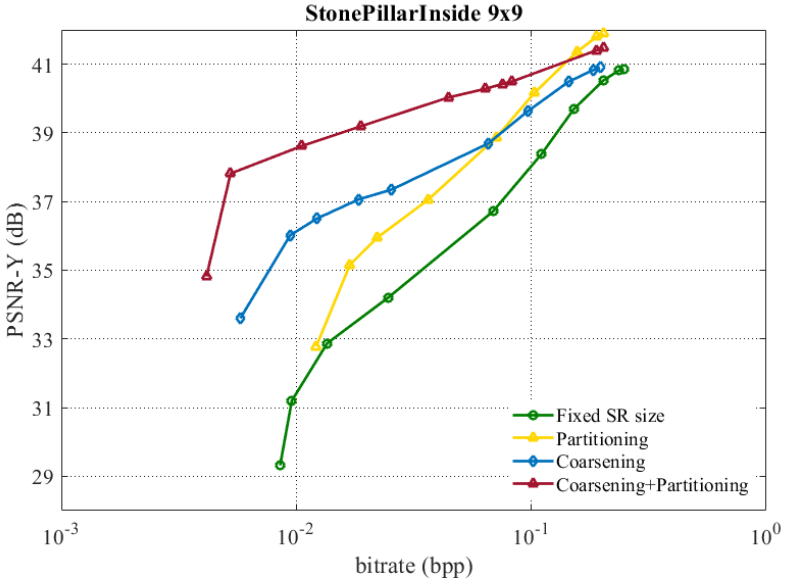


Figure 4.11: Rate-distortion comparison.

(J28) M. Rizkallah, T. Maugey, C. Guillemot *Rate-Distortion Optimized Graph Coarsening and Partitioning for Light Field Coding*, submitted to IEEE Transactions on Image Processing in IEEE Transactions on Image processing, vol. 29, pp. 3282 – 3295, Dec. 2019.

4.5 Conclusion

In this Chapter, we have proposed methods to compute the graph-based transform for high dimension data. In particular, we have proposed three approaches: graph segmentation, graph factorization and graph reduction. In all three cases, we have posed the

problem such that an optimal solution could be estimated. We have finally applied this to the coding of light field or 360° images. In particular, results show that partitioning and coarsening are two complementary strategies. When put in a global rate-distortion optimization solution, they are able to both bring coding gain. Coarsening reduces the image for large smooth areas, and partitioning handle the highly textured regions by following the object contours.

Chapter 5

Conclusion and Perspectives

The work presented in this manuscript has been realized when I was postdoctoral researcher at EPFL (between 2010 and 2014) and, later, research scientist at Inria (after 2014). An important part of these contributions has been done in the context of two PhD thesis co-supervisions:

- ★ Mira Rizkallah, *Graph based transforms for compression of new imaging modalities*, (PhD 2016-2019, co-supervised with Christine Guillemot)
- ★ Navid Mahmoudian Bidgoli, *Compression for interactive communications of visual contents*, (PhD 2016-2019, co-supervised with Aline Roumy)

I would like to take the opportunity of these few lines to deeply thank these two excellent students for their involvement.
In all contributions described above, we have started with the identification of some scenarios in which the conventional coding architecture was limited, *i.e.*, random access or irregular topologies. Then, we have identified and formalized the associated scientific problems. We have then proposed methodological answers to solve them. In the case of random access, we have even provided a theoretical study setting the performance that could be expected. Finally, we have, each time, proposed a practical coding solution to demonstrate the efficiency of the proposed methodology. Going "from the theory to practice" enabled us to have a good overview of the problem, and to propose solutions that were at the same time efficient and practical.

In the following, I introduce the perspective works, that I have started or that I plan to conduct with the same research methodology. These 5 research axes are ranked from the most short-term to the most prospective.

5.1 Disseminate the work on interactive coding

In the work presented in Chapter 2, we have built the proof-of-concept of an interactive video coder, that is able to reach the theoretical performance promised in our theoretical study, namely, no extra transmission cost and a low storage overhead. Despite the great promise of this results, we are aware about the revolution it brings to the whole coding architecture, and about the difficulty to include such solution rapidly in a real standard. In order to facilitate the extension and the reuse of our proof-of-concept, we have launched a project, called ICOV (Interactive Coder for Omnidirectional video)[1] that aims at developing a clean open-source version of our codec. The goal is to have a robust, clear and

[1] https://project.inria.fr/icov/

transferable implementation such that researchers or industrial can reuse it. This project is done in the context of the supervision of:

★ Sébastien Bellenous, *ICOV transfer project*, (Research Engineer 2020-2022).

Besides, the work conducted in Chapter 2, has led us to highlight the importance of saving storage size in a streaming applications, which is often overlook in the literature. With that goal in mind, we are working with the company *Mediakind* to explore new light video representations to be stored on the server, in the context of video streaming (sort of a random access, where the requests correspond to the different bandwidth conditions). This project is done in the context of the co-supervision of:

★ Reda Kaafarani, *Optimization of Multi-profiles encoding systems*, (PhD 2021-2024, co-supervised with Aline Roumy, Mederic Blestel and Michael Ropert).

5.2 Multi-view 360 view synthesis

In the context of interactive coding or graph-based representations, we have been led to study 360° images in depth. This type of data enables a user to change his angle of view, interacting with 3 degrees of freedom: yaw, roll and pitch. Clearly providing the sensation of being inside the scene, spherical imaging has thus been seen as the corner stone of immersive multimedia. However, ultimate free navigation in a scene is achieved when a translation $\mathbf{t}$ over x, y and z is additionally possible, which is not the case with a simple omnidirectional capture. This is the reason why, we have investigated the possibility of performing multi-view 360° capture. As it is illustrated in Figure 5.1, such system enables the user to do translations at sampled positions, *i.e.*,

$$\mathbf{t} \in \{\delta_i\}_{0 \leq i \leq N}, \tag{5.1}$$

where the δ_i are the position of the N cameras. In that case, 6 degrees of freedom are given to the users, in which 3 of them are discrete. If the number of camera N is sufficiently large or the distance between the δ_i is small, a quite good level of immersion sensation can be given to the users. In (C40) and on the FTV360 website[2], we have shared a dataset that we built based on multi-view 360° acquisition. The shared sequences consist of indoor and outdoor scenes captured with 40 omnidirectional cameras. The relative positions of the cameras are also provided.

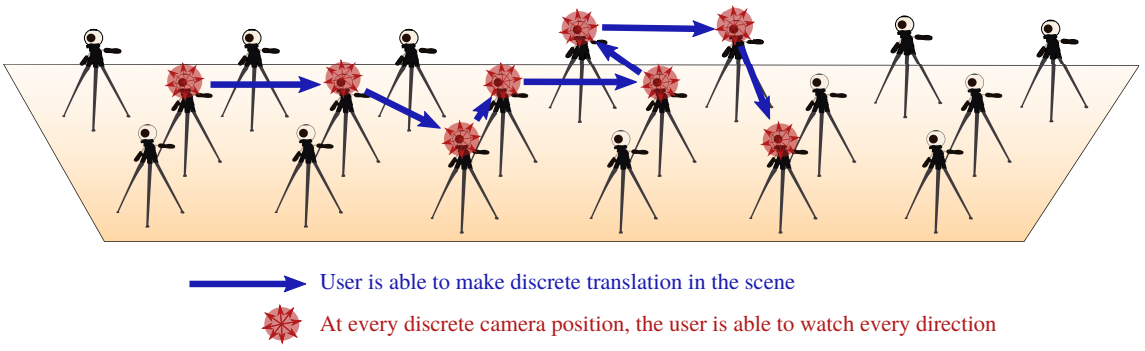


Figure 5.1: User navigation possibility in a multi-view 360° capture system.

Such new acquisition system opens exciting research challenges. The most obvious and impacting one is the virtual view synthesis. Indeed, having the possibility of generating virtual spherical images between the positions δ_i would enable to reach a smooth

[2]https://project.inria.fr/ftv360/

free navigation around the 6 degrees of freedom, and thus a total immersion in the scene. We are currently working on developing such view synthesis algorithm using omnidirectional captured views. This work is done in the context of the co-supervision of:

★ Kai Gu, *Spherical light field representation and reconstruction from omnidirectional imagery*, (PhD 2021-2024, co-supervised with Christine Guillemot and Sebastian Knorr).

and has already led to the work published in (C48).

(**C48**) K. Gu, T. Maugey, S. Knorr, C. Guillemot, *Omni-NeRF: Neural Radiance Field from 360° image captures*, IEEE ICME, Jul 2022, Taipei, Taiwan

(**C40**) T. Maugey , L. Guillo, C. Le Cam, *FTV360: a Multiview 360-degree Video Dataset with Calibration Parameters*, ACM Multimedia Systems Conference, Amherst, MA, US, June 2019.
Cited by the OmniCV workshop of CVPR 2020

5.3 Learning on the sphere

In Chapter 3 and 4 of this manuscript, we have proposed to use graph-based signal processing tools to handle the non-euclidean topology of spherical data. In particular, we have investigated how to define and use graph-based transforms. In order to go further, we have studied how more evolved tools such as learning tools could be efficiently defined directly on the sphere. In that case, we have noticed that the graph-based approach may be limited because of the isotropic property of the kernel filter that one is able to define in such context. We have thus investigated an alternative to graph-based approach, for doing learning on the sphere.

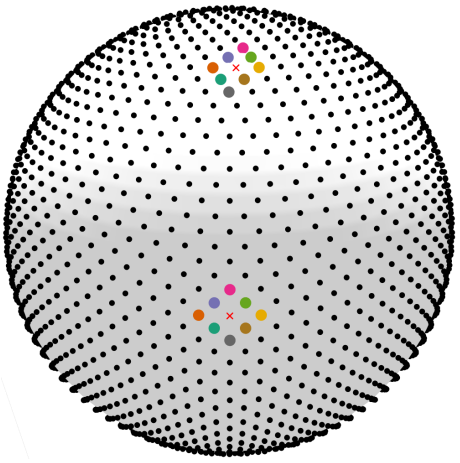


Figure 5.2: The convolution developed in our OSLO toolbox enables to easily translate a kernel on pseudo-uniform sampling on the sphere. At each position, the number of neighbors to the central point is constant and a different weight to each neighbor could be assigned for a high filter expressiveness.

In (J31s), we have proposed a toolbox called OSLO, that uses the properties of HEALPix uniform sampling of the sphere and redefines the mathematical tools used in deep learning models for omnidirectional images. In particular, we: i) propose the definition of a

new convolution operation on the sphere that keeps the high expressiveness and the low complexity of a classical 2D convolution (see Figure 5.2); ii) adapt standard CNN techniques such as stride, iterative aggregation, and pixel shuffling to the spherical domain; and then iii) apply our new framework to the task of omnidirectional image compression. Our experiments show that our proposed on-the-sphere solution leads to a better compression gain that can save 13.7% of the bit rate compared to similar learned models applied to equirectangular images. Also, compared to learning models based on graph convolutional networks, our solution supports more expressive filters that can preserve high frequencies and provide a better perceptual quality of the compressed images.

Such results demonstrate the efficiency of the proposed framework, which opens new research venues for other omnidirectional vision tasks to be effectively implemented on the sphere manifold, such as classification, segmentation, etc. On top of these different research directions, we are also working on the transfer of such technology to the industry, in the context of a start-up creation. The project *Anax* is lead by Navid Mahmoudian-Bidgoli and Simon Evain, and aims at using on-the-sphere AI technology for 360° video editing, and more particularly for building remote virtual tour. The *Anax* start-up project is currently funded by Inria Start up Studio incubator. I am involved as scientific advisor with Aline Roumy.

(J31s) N. Mahmoudian Bidgoli, R. Azevedo, T. Maugey, A. Roumy, P. Frossard, *OSLO: On-the-Sphere Learning for Omnidirectional images and its application to 360-degree image compression*, submitted to IEEE Transactions on Image Processing

5.4 Coding for Machines

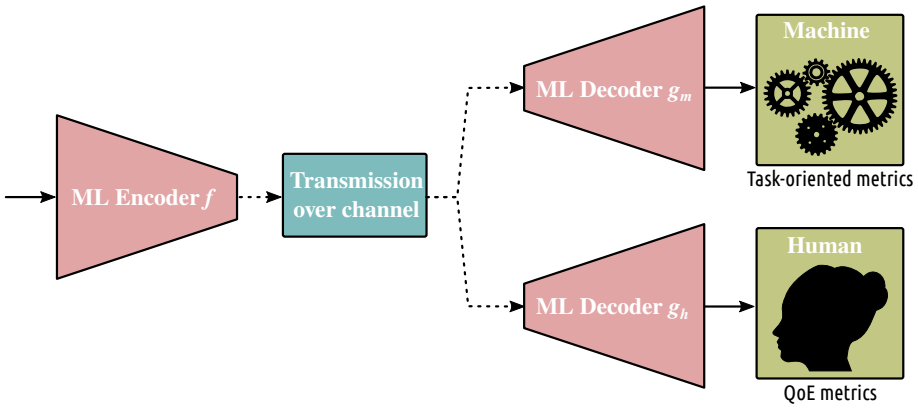


Figure 5.3: In the coding for machine scenario, the compressed and uncompressed data is not only watched, but also processed by algorithms.

While the perspectives drawn previously were in the continuation of the work presented in this manuscript, I present now two prospective research topics that tackle another limitations of the conventional coding architecture: its dependence to the *fidelity* metric. Indeed, as stated in Figure 1.1, the conventional architecture evaluates the quality of its compression as its ability to retrieve a decoded image/video that is as close as

possible to the input. However, with the explosion of data creation nowadays, the visual content that is created is not necessarily meant to be watched by a user. As illustrated in Figure 5.3, the decoded data may be processed by an algorithm for some given tasks (classification, segmentation, etc.).

This change of paradigm opens several interesting research questions such as, is there a trade-off between accuracy of the processing algorithm and the visual quality ? Is a conventional architecture optimal for task-oriented metrics ? Can a learning algorithm operate in the compressed domain (or must we consider a decoder) ? We will investigate those questions in the context of the following PhD thesis that I co-supervise:

★ Rémi Piau, *Video coding for learning: video content analysis in the compressed domain*, (PhD 2021-2024, co-supervised with Aline Roumy).

5.5 Data Repurposing

The limitation of the *fidelity* metric involved in conventional architectures can be seen under an other angle. This is what I develop in this other perspective research axis.

The era of data explosion we live in has led to cutting edge findings in big data analysis and deep learning algorithms but at an expensive cost in terms of data storage. Storage growth is exceeding even the highest estimates with no sign of it slowing down anytime soon: 2.5 quintillion bytes of data are created each day at our current pace [115], and it will only accelerate with the advent of IoTs, volumetric videos, and new sensors. The storage burden has been partially alleviated by state-of-the-art compression algorithms, which can substantially reduce the amount of bits needed to store one or multiple sources, *e.g.*, end-to-end learning-based image compression algorithms to minimize the compression rate [116], MPEG standards to ensure exploitation of spatial and temporal correlation [117], joint source compression [118, 119, 120, 121]. All these coding strategies have led to impressive compression ratio, which however will be scaling always with the number of sources. However, to contain the upcoming avalanche of data, there is the need for a much drastic compression rate, which cannot be reached till the ultimate goal of the compression algorithm is to represent *each* original source with high fidelity.

In this future work entitled *Data Repurposing*, we aim at addressing this challenge by proposing a new paradigm-shift for compression algorithm aimed instead at preserving a global information perceived by the final user. We define this information as *perceived information* (*PI*). Sources should be compressed in such a way that the *information of interest for the final user* – rather than per source information – is preserved.

Data collection sampling: In (J26), we proposed a first solution in the case of the encoder being a sampling algorithm. To achieve this goal, we first introduce the PI metric as the volume spanned by the sources features in a personalized latent space, *i.e.*, feature domain distorted by the user preferences. Then, we formalize our PI-based compression problem as a selection of the subset of sources that maximizes PI under sample size constraints and we propose an adaptive sampling algorithm to solve it. The latter selects for each user a subset of sources, which is the most representative of the original database, in terms of features most preferable by the user. Finally, we evaluate the performance of the proposed algorithm via simulation results, proving its gain against baseline algorithms taking into account user's preference or source redundancy disjointly. In particular, we show that our algorithm balances *features-perceived quality* (how relevant each feature is to the user) and *features-diversity* (how well features are represented within the selected

subsect).

We are extending such theoretical study to real image collection. This includes to build a proper latent representation for images, and model the user's preferences. This work is done in the context of the supervision of:

★ Anju J. Tom, *Data collection sampling*, (Postdoc 2020-2022).

Generative compression: In a second axis of the Data Repurposing project, we investigate how to semantically describe the database information in a concise representation, thus leading to drastic compression ratios exactly *as a music score is able to describe, for example, a concert in a compact and reusable form*. This enables the compression to withdraw tremendous amount of useless, or at least not essential, information while condensing the important information into a compact recycled signal. From this data collection digest form, the decoder generates (*i.e.*, invents) a content, coherent with the described semantic. For this task, guided GAN architectures can be used [122, 123]. The decoded signals target subjective exhaustiveness of the information description, rather than fidelity to the input data, as in the traditional compression algorithms. Naturally, not all the visual content is meant to be regenerated. Users might be willing to retrieve faithfully the content after decompression. Such approaches will therefore be designed according to user's profile taking into account their choice and interaction (as depicted in Figure 5.4). This is a complete change of paradigm for image and video compression, which must enable gigantic compression gains. This work will be conducted in the context of two PhD supervisions:

★ Tom Bachard, *Generative compression of image collection*, (PhD 2021-2024).
★ Tom Bordin, *Generative video compression*, (PhD 2022-2025).

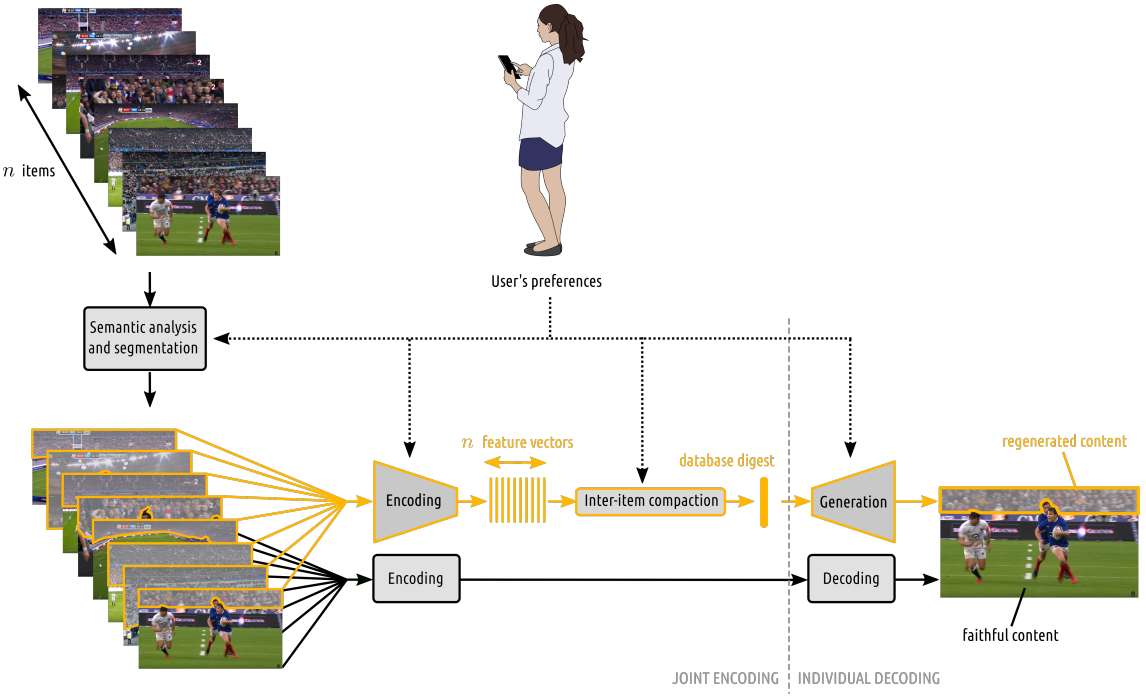


Figure 5.4: Generative compression framework, in which part of the image/video content is regenerated at the decoder.

Data streaming energy cost estimation

The well-known *rebound effect* or *Jevon's paradox* states that as technological improvements increase the efficiency with which a resource is employed, the total consumption of that resource may increase rather than decrease. In the particular case of video compression, this statement is more than confirmed. As the video compression has improved its efficiency, the streaming cost of a single video has been drastically reduced. However, this has lead to an explosion of the video streaming usage in our every day life. As *Data Repurposing* aims at reducing the energy cost spent to process and store the exploding amount of data, such rebound effect should be avoided. This can be done by *increasing the user's awareness*. We would like to provide an online simulator of video streaming cost. In a nutshell, a user will be able to design a streaming strategy (or load a predefined template corresponding to existing schemes) and to calculate the corresponding energy cost. This work will be done in the context of the supervision of:

★ Sébastien Bellenous, *Video streaming energy cost estimation*, (Research Engineer, 2022-2024).

(J26) T. Maugey, L. Toni, *Large Database Compression Based on Perceived Information*, in IEEE Signal Processing Letters, vol. 7, pp 1735–1739, Sep. 2020

Publications

Books

(B2) T. Maugey, *Acquisition, Representation and Rendering of Omnidirectional videos*, Immersive Video Technologies, G. Valenzise, M. Alain, E. Zerman, C. Ozcinar (Eds.), Elsevier, to appear

(B1) T. Maugey, M. Rizkallah, N. Mahmoudian Bidoli, A. Roumy, C. Guillemot, *Graph Spectral 3D Image Compression*, Graph spectral image processing, G. Cheung and E. Magli (Eds.), ISTE, 2021, pp. 105-128.

Journal papers, submitted

(J31s) N. Mahmoudian Bidgoli, R. Azevedo, T. Maugey, A. Roumy, P. Frossard, *OSLO: On-the-Sphere Learning for Omnidirectional images and its application to 360-degree image compression*, submitted to IEEE Transactions on Image Processing

Journal papers, accepted/published

(J30) N. Thomos, T. Maugey, L. Toni, *Machine learning for multimedia communications*, accepted in MDPI Sensors, 2022

(J29) P. Garus, F. Henry, J. Jung, T. Maugey, C. Guillemot, *Immersive Video Coding: Should Geometry Information be Transmitted as Depth Maps?*, accepted in IEEE Transactions on Circuits and Systems for Video Technology 2021

(J28) M. Rizkallah, T. Maugey, C. Guillemot *Rate-Distortion Optimized Graph Coarsening and Partitioning for Light Field Coding*, accepted in IEEE Transactions on Image Processing, 2021

(J27) F. Ye, N. Mahmoudian Bidgoli, E. Dupraz, A. Roumy, K. Amis, T. Maugey *Bit-Plane Coding in Extractable Source Coding: optimality, modeling, and application to 360° data*, accepted in IEEE Communication letters 2021

(J26) T. Maugey, L. Toni, *Large Database Compression Based on Perceived Information*, in IEEE Signal Processing Letters, vol. 7, pp 1735–1739, Sep. 2020

(J25) N. Mahmoudian-Bidgoli, T. Maugey, A. Roumy, *Excess rate for model selection in interactive compression using Belief-propagation decoding*, accepted in Annals of Telecommunications

(J24) N. Mahmoudian-Bidgoli, T. Maugey, A. Roumy, *Fine granularity access in interactive compression of 360-degree images based on rate adaptive channel codes* in IEEE Transactions on Multimedia, vol 23, pp. 2868-2882, Aug. 2021

(J23) M. Q. Pham, A. Roumy, T. Maugey, E. Dupraz, M. Kieffer *Optimal Reference Selection for Random Access in Predictive Coding Schemes* in IEEE Transactions on Communica-

tions, vol. 68(9), pp. 5819-5833, Sep. 2020.

(J22) T. Maugey, A. Roumy, E. Dupraz, M. Kieffer, *Incremental coding for extractable compression in the context of Massive Random Access* in IEEE Transactions on Signal and Information Processing over Networks, vol. 6(1), pp. 251-260, Dec. 2020.

(J21) M. Rizkallah, T. Maugey, C. Guillemot, *Prediction and Sampling with Local Graph Transforms for Quasi-Lossless Light Field Compression* in IEEE Transactions on Image processing, vol. 29, pp. 3282 – 3295, Dec. 2019.

(J20) M. Rizkallah, T. Maugey, C. Guillemot, *Geometry-Aware Graph Transforms for Light Field Compact Representation* in IEEE Transactions on Image Processing, vol. 29, pp. 602–616, Jul. 2019.

(J19) E. Dupraz, T. Maugey, A. Roumy, M. Kieffer, *Rate-Storage Regions for Extractable Source Coding with Side Information* in Physical Communication, Elsevier, Special Issue on Coding and Information Theory for Emerging Communication Systems, Vol. 37, 2019.

(J18) R. Ma, T. Maugey, P. Frossard, *Optimized Data Representation for Interactive Multiview Navigation*, in IEEE Transactions on Multimedia, Vol. 20(7), p. 1595-1609, Jul 2018.

(J17) C. Verleysen, T. Maugey, C. De Vleeschouwer, P. Frossard, *Wide baseline image-based rendering based on shape prior regularisation*, in IEEE Transactions on Image Processing, Vol 26(11), p. 5477 – 5490, Jul. 2017.

(J16) X. Su, T. Maugey, C. Guillemot *Rate-distortion optimized graph-based representation for multiview images with complex camera configurations*, in IEEE Transactions on Image Processing, Vol 26(6), p. 2644–2655, Jun. 2017 2017.

(J15) S. Khattak, T. Maugey, R. Hamzaoui, S. Ahmad, P. Frossard, *Temporal and Inter-view consistent error concealment technique for multiview plus depth video broadcasting*, in IEEE Transactions on Circuits and Systems for Video Technology, Vol. 26(5), p. 829-840, May 2016.

(J14) T. Maugey, G. Petrazzuoli, M. Cagnazzo and B. Pesquet-Popescu, P. Frossard, *Reference view selection in DIBR-based multiview coding*, in IEEE Transactions on Image Processing, Vol 25(4), p. 1808-1819, April 2016.

(J13) Y. Gao, G. Cheung, T. Maugey, P. Frossard, J. Liang, *Encoder-driven Inpainting Strategy in Multiview Video Compression,* in IEEE Transactions on Image Processing, Vol. 25(1), p. 134-149, Jan. 2016.

(J12) A. De Abreu, L. Toni, N. Thomos, T. Maugey, F. Pereira, P. Frossard, *Optimal Layered Representation for Adaptive Interactive Multiview Video Streaming*, in Journal of Visual Communication and Image Representation (Elsevier), Vol. 33, pp. 255-264, Nov. 2015.

(J11) L. Toni, T. Maugey, P. Frossard, *Optimized Packet Scheduling in Multiview Video Navigation Systems*, in IEEE Transactions on Multimedia, Vol. 17(9), pp. 1604 - 1616, Sep. 2015.

(J10) T. Maugey, A. Ortega, P. Frossard *Graph-based representation for multiview image geometry*, in IEEE Transactions on Image Processing, Vol. 24(5) , pp. 1573 - 1586, 2015.

(J9) G. Petrazzuoli, T. Maugey, M. Cagnazzo and B. Pesquet-Popescu *Depth-Based Multiview Distributed Video Coding*, in IEEE Transactions on Multimedia, Vol. 16(7), pp. 1834 - 1848, 2014.

(J8) U. Takyar, T. Maugey, P. Frossard *Extended Layered Depth Image Representation in Multiview Navigation*, in IEEE Signal Processing Letters, Vol. 21, p. 22 - 25 Jan. 2014.

(J7) L. Toni, T. Maugey, P. Frossard *Correlation-Aware Packet Scheduling in Multi-Camera Networks*, in IEEE Transactions on Multimedia, Vol. 16(2), pp. 496 - 509, 2014.

(J6) S. Khattak, T. Maugey, R. Hamzaoui, S. Ahmad, P. Frossard *Bayesian Early Mode De-*

cision Technique for View Synthesis Prediction-enhanced Multiview Video Coding, in IEEE Signal Processing Letters, Vol. 20, p. 1126 - 1129, Nov. 2013.

(J5) T. Maugey, J. Gauthier, M. Cagnazzo, B. Pesquet-Popescu *Evaluation of side information effectiveness in distributed video coding, Signal Processing*, in IEEE Transactions on Circuits and Systems for Video Technology, Vol. 23, p. 2116 - 2126, Dec. 2013.

(J4) B. Rajei, T. Maugey, P. Frossard *Rate-distortion analysis of multiview coding in a DIBR framework*, in Annals of Telecommunications (Springer), Vol. 68, p. 627-640, Dec. 2013.

(J3) T. Maugey, I. Daribo, G. Cheung, P. Frossard *Navigation domain partitioning for interactive multiview imaging*, in IEEE Transactions on Image Processing, Vol. 22, p. 3459-3472, Sep. 2013.

(J2) T. Maugey, P. Frossard *Interactive multiview video system with a non-complex navigation at the decoder*, in IEEE Transactions on Multimedia, Vol.15, p 1-13, Aug. 2013.

(J1) T. Maugey, B. Pesquet-Popescu *Side information estimation and new symmetric schemes for multi-view distributed video coding*, Journal of Visual Communication and Image Representation (Special issue: Resource-Aware Adaptive Video Streaming), Vol. 19, Issue 8, Pages 589-599, Dec. 2008.

International conferences papers

(**C48**) K. Gu, T. Maugey, S. Knorr, C. Guillemot, *Omni-NeRF: Neural Radiance Field from 360° image captures,* IEEE ICME, Jul 2022, Taipei, Taiwan

(**C47**) R. Kaafarani, M. Blestel, T. Maugey, M. Ropert, A. Roumy, *Evaluation Of Bitrate Ladders For Versatile Video Coder*, IEEE VCIP, Dec 2021, Munich, Germany

(**C46**) A. Marie, N. Mahmoudian Bidgoli, T. Maugey, A. Roumy, *Rate-distortion optimized motion estimation for on-the-sphere compression of 360 videos*, IEEE ICASSP, Jun 2021, Toronto, Canada

(**C45**) F. Hawary, T. Maugey , C. Guillemot, *Sphere mapping for feature extraction from 360 fish-eye captures* IEEE International Workshop on Multimedia Signal Processing (MMSP), Sep 2020, Tempere, Finland. pp.1-6

(**C44**) N. Mahmoudian Bidgoli, T. Maugey , A. Roumy, *Intra-coding of 360-degree images on the sphere* Picture Coding Symposium (PCS), Ningbo, China, Nov. 2019

(**C43**) N. Mahmoudian Bidgoli, T. Maugey, A. Roumy, F. Nasiri and F. Payan, *A geometry-aware compression of 3D mesh texture with random access* Picture Coding Symposium (PCS), Ningbo, China, Nov. 2019

(**C42**) P. Garus, J. Jung, T. Maugey and C. Guillemot, *Bypassing Depth Maps Transmission For Immersive Video Coding* Picture Coding Symposium (PCS), Ningbo, China, Nov. 2019

(**C41**) N. Mahmoudian Bidgoli, T. Maugey , A. Roumy, E*valuation framework for 360-degrees visual content compression with user-dependent transmission* IEEE ICIP, Tapei, Taiwan, Sep. 2019

(**C40**) T. Maugey , L. Guillo, C. Le Cam, *FTV360: a Multiview 360-degree Video Dataset with Calibration Parameters*, ACM Multimedia Systems Conference, Amherst, MA, US, June 2019.

(**C39**) F. Nasiri, N. Mahmoudian-Bigdoli, F. Payan, T. Maugey ., *A geometry-aware framework for compressing 3D mesh textures*, IEEE ICASSP, Brighton, UK, May. 2019. *cited in IEEE MMTC Review Letter of April 2019*

(**C38**) M. Rizkallah, T. Maugey, C. Guillemot . *Graph-based Spatio-angular Prediction for Quasi-Lossless Compression of Light Fields*, Data Compression Conference, Cliff Lodge, Snowbird, UT, US , Mar. 2019.

(**C37**) E. Dupraz, T. Maugey, A. Roumy and M. Kieffer. *Rate-Distortion Performance of Sequential Massive Random Access to Gaussian Sources with Memory*, Data Compression Conference, Snowbird, Utah, US, Mar. 2018.

(**C36**) M. Rizkallah, F. De Simone, T. Maugey, C. Guillemot, P. Frossard, *Rate Distortion Optimized Graph Partitioning for Omnidirectional Image Coding* EUSIPCO, Athens, Greece, Sept. 2018. *Best Student Paper*

(**C35**) X.Su, M. Rizkallah, T. Maugey, C. Guillemot, *Rate-Distortion Optimized Super-Ray Merging for Light Field Compression* EUSIPCO, Athens, Greece, Sept. 2018.

(**C34**) T. Maugey, O. Le Meur, Z. Liu, *Saliency-based navigation in omnidirectional image*, IEEE MMSP, London, UK, Oct. 2017.

(**C33**) N. Mahmoudian Bidgoli, T. Maugey, A. Roumy *Correlation Model Selection for interactive video communication*, ICIP, Beijing, China, Sep. 2017

(**C32**) Xin Su, M. Rizkallah, T. Maugey, C. Guillemot *Graph-based light fields representation and coding using geometry information*, ICIP, Beijing, China, Sep., 2017.

(**C31**) M. Rizkallah, T. Maugey, C. Yaacoub and C. Guillemot *Impact of Light Field Compression on Focus Stack and Extended Focus Images*, EUSIPCO, Budapest, Hungary, Aug. 2016

(**C30**) X. Su, T. Maugey and C. Guillemot, *Graph-based representation for multiview images with complex camera configurations*, IEEE ICIP, Pheonix Arizona, Sep. 2016

(**C29**) T. Maugey, P. frossard and C. Guillemot, *Guided inpainting with cluster-based auxiliary information*, IEEE ICIP, Quebec, Canada, Sep., 2015

(**C28**) A. Roumy and T. Maugey, *Universal lossless coding with random user access: the cost of interactivity*, IEEE ICIP, Quebec, Canada, Sep., 2015 (Top 10% papers)

(**C27**) L. Toni, T. Maugey, and P. Frossard, *Packet Scheduling in MultiCamera Capture Systems*, VICIP, Malta, Dec., 2014

(**C26**) A. De Abreu, N. Thomos, T. Maugey, L. Toni and P. Frossard, *Multiview Video Representations for Quality-Scalable Navigation*, VICIP, Malta, Dec., 2014

(**C25**) T. Maugey, G. Petrazzuoli, P. Frossard, M. Cagnazzo and B. Pesquet-Popescu *Key view selection in distributed multiview coding*, VICIP, Malta, Dec., 2014

(**C24**) T. Maugey, Y.H. Chao, A. Gadde, A. Ortega and P. Frossard *Luminance coding in graph-based representation of multiview images*, IEEE ICIP, Paris, France, Oct., 2014

(**C23**) Y. Gao, G. Cheung, T. Maugey, P. Frossard and J. Liang *3D Geometry Representation using Multiview Coding of Image Tiles*, IEEE ICASSP, Florence, Italy, May, 2014

(**C22**) G. Petrazzuoli, T. Maugey, M. Cagnazzo and B. Pesquet-Popescu *A Distributed Video Coding System for Multi View Video Plus Depth*, IEEE Asilomar CSSC, Pacific Grove, CA, USA, Nov, 2013 - *Invited paper*

(**C21**) T, Maugey, A. Ortega, P. Frossard *Graph-Based vs Depth-Based Data Representation for Multiview Images*, IEEE Asilomar CSSC, Pacific Grove, CA, USA, Nov, 2013

(**C20**) T, Maugey, A. Ortega, P. Frossard *Multiview image coding using graph-based approach*, IEEE IVMSP, Seoul, Korea, June, 2013

(**C19**) T, Maugey, A. Ortega, P. Frossard *Graph-based representation and coding of multiview geometry*, IEEE ICASSP, Vancouver, May, 2013

(**C18**) I. Daribo, T, Maugey, G. Cheung, P. Frossard *RD optimized auxiliary information for inpainting-based view synthesis*, 3DTV Conference Zurich, Switzerland, Oct., 2012

(**C17**) T. Maugey, P. Frossard, G. Cheung *Consistent view synthesis in interactive multiview imaging* In international Packet Video Workshop, Orlando, USA, Sep 2012

(**C16**) L. Toni, T. Maugey, P. Frossard *Correlation-Aware Packet Scheduling for Multi-Camera Streaming* In IEEE Int. Conf. on Image Processing (ICIP), Munich, Germany, May 2012

(**C15**) T. Maugey, P. Frossard *Interactive multiview video system with low decoding complexity* In IEEE Int. Conf. on Image Processing (ICIP), Bruxelles, Belgium, Sep. 2011

(**C14**) V. Davidoiu, T. Maugey, B. Pesquet-Popescu, P. Frossard *Rate distortion analysis in a disparity compensated scheme* In IEEE Int. Conf. on Speech and Signal Processing (ICASSP), Prague, Czech Republic, May 2011

(**C13**) T. Maugey, C. Yaacoub, J. Farah, B. Pesquet-Popescu *Side information enhancement using an adaptative hash-based genetic algorithm in a Wyner-Ziv context* In IEEE Int. Workshop on Multimedia Signal Processing (MMSP), Saint-Malo, Oct 2010

(**C12**) M. Trocan, T. Maugey, E. Tramel, J. Fowler, B. Pesquet-Popescu *CS-reconstruction of multiview images using bootstrap-like disparity compensation* In IEEE Int. Workshop on Multimedia Signal Processing (MMSP), Saint-Malo, Oct 2010

(**C11**) G. Petrazzuoli, T. Maugey, M. Cagnazzo, B. Pesquet-Popescu *Side inforrmation refinement for long duration GOPs in DVC* In IEEE International Workshop on Multimedia Signal Processing (MMSP), Saint-Malo, Oct 2010

(**C10**) M. Trocan, T. Maugey, E. Tramel, J. Fowler, B. Pesquet-Popescu *Compressed-sensing of multiview images using disparity compensation* In Proc. IEEE Int. Conf. on Image Processing (ICIP), Sep 2010, Hong-Kong,

(**C9**) M. Trocan, T. Maugey, J. Fowler, B. Pesquet-Popescu *Disparity-compensated compressed-sensing reconstruction of multiview images* In Proc. IEEE Int. Conf. on Multimedia and Expo (ICME), Aug 2010, Singapore *reviewed in IEEE Communications Society Multimedia Communications Technical Committee (MMTC) Review Letters (R-Letters) in Aug. 2011*

(**C8**) T. Maugey, J. Gauthier, B. Pesquet-Popescu, C. Guillemot *Using an exponential power model for Wyner Ziv video coding*, In IEEE Int. Conf. on Speech and Signal Processing (ICASSP), Mar. 2010. Dallas, Texas, USA.

(**C7**) M. Cagnazzo, W. Miled, T. Maugey, B. Pesquet-Popescu *Image interpolation with edge-preserving differential motion refinement* Proc. IEEE Int. Conf. on Image Processing (ICIP), Nov. 2009. Cairo, Egypt

(**C6**) T. Maugey, W. Miled, M. Cagnazzo, B. Pesquet-Popescu *Fusion Schemes for Multiview Distributed Video Coding* In European Signal Processing Conference (EUSIPCO), August 2009. Glasgow, Scotland

(**C5**) W. Miled, T. Maugey, M. Cagnazzo, B. Pesquet-Popescu. *Image Interpolation with Dense Disparity Estimation in Multiview Distributed Video Coding* Int. Conf. on Distributed Smart Cameras (ICDSC), September 2009. Como, Italy.

(**C4**) M. Cagnazzo, T. Maugey, B. Pesquet-Popescu *A Differential Motion Estimation Method for Image Interpolation in Distributed Video Coding*, IEEE Int. Conf. on Speech and Signal Processing (ICASSP), Taipei, Taiwan, 18-22 April 2009

(**C3**) T. Maugey, W. Miled, B. Pesquet-Popescu *Dense Disparity Estimation in a Multi-view Distributed Video Coding System*, in IEEE Int. Conf. on Speech and Signal Processing (ICASSP), Taipei, Ta?wan, 18-22 April 2009

(**C2**) C. Dikici, T. Maugey, M. A. Agostini and O. Crave *Efficient Frame Interpolation for Wyner-Ziv Video Coding*, SPIE Electronical Imaging, Visual Communications and Image Processing conference (VCIP), San Jose, USA, 18-22 January 2009

(**C1**) T. Maugey, T. André, B. Pesquet-Popescu, J. Farah, *Analysis of Error Propagation Due to Frame Losses in a Distributed Video Coding System*, In European Signal Processing Conference (EUSIPCO), Lausanne, August 2008.

National conferences papers

(**CN7**) T. Maugey, C. Le Cam, L. Guillo, *Télévision à point de vue libre et système de capture à plusieurs caméra omnidirectionnelles* In Colloque GRETSI - Traitement du Signal et des Images, Juan-les-Pins, France, Sep. 2017

(**CN6**) A. Crinière, A. Roumy, T. Maugey, M. Kieffer, Jean Dumoulin, *Sélection optimale de capteurs de référence pour le stockage de données spatialement corrélées* In Colloque GRETSI - Traitement du Signal et des Images, Juan-les-Pins, France, Sep. 2017

(**CN5**) A. Roumy, T. Maugey *Compression et interactivité : étude de la navigation au récepteur* In Colloque GRETSI - Traitement du Signal et des Images, Lyon, France, Sep. 2015

(**CN4**) T. Maugey, P. Frossard *Nouvelle représentation de données pour les applications interactives de navigation vidéo* In Colloque GRETSI - Traitement du Signal et des Images, Brest, France, Sep. 2013

(**CN3**) T. Maugey, P. Frossard *Codage vidéo multi-vue pour une vision interactive au récepteur* In Colloque GRETSI - Traitement du Signal et des Images, Bordeaux, France, Sep. 2011

(**CN2**) T. Maugey, W. Miled, M. Cagnazzo, B. Pesquet-Popescu *Méthodes denses d'interpolation de mouvement pour le codage vidéo distribué monovue et multivue* Colloque GRETSI - Traitement du Signal et des Images, September 2009. Dijon, France

(**CN1**) J. Gauthier, T. Maugey, B. Pesquet-Popescu, C. Guillemot *Amélioration du Mod?le statistique de bruit pour le codage vidéo distribué* Colloque GRETSI - Traitement du Signal et des Images, September 2009. Dijon, France.

Thesis

(**T2**) T. Maugey *Distributed Video Coding of Mutliview sequences* PhD thesis

(**T1**) T. Maugey *Distributed Multiview Video Coding* Master thesis

Bibliography

[1] Y.-W. Huang, J. An, H. Huang, X. Li, S.-T. Hsiang, K. Zhang, H. Gao, J. Ma, and O. Chubach, "Block partitioning structure in the vvc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3818–3833, 2021.

[2] T. Dumas, A. Roumy, and C. Guillemot, "Context-adaptive neural network-based prediction for image compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 679–693, 2019.

[3] M. G. Blanch, S. Blasi, A. F. Smeaton, N. E. O'Connor, and M. Mrak, "Attention-based neural networks for chroma intra prediction in video coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 366–377, 2020.

[4] T. Cover and J. Thomas, *Elements of information theory*. John Wiley & Sons, 1999.

[5] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[6] F. W. Wheeler and W. A. Pearlman, "Spiht image compression without lists," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, vol. 4. IEEE, 2000, pp. 2047–2050.

[7] R. Dony *et al.*, "Karhunen-loeve transform," *The transform and data compression handbook*, vol. 1, no. 1-34, p. 29, 2001.

[8] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[9] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[10] L. Wang, L. Wang, Y. Luo, and M. Liu, "Point-cloud compression using data independent method—a 3d discrete cosine transform approach," in *2017 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2017, pp. 1–6.

[11] M. B. de Carvalho, M. P. Pereira, G. Alves, E. A. da Silva, C. L. Pagliari, F. Pereira, and V. Testoni, "A 4d dct-based lenslet light field codec," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 435–439.

[12] S. Viraktamath and G. V. Attimarad, "Impact of quantization matrix on the performance of jpeg," *International Journal of Future Generation Communication and Networking*, vol. 4, no. 3, pp. 107–118, 2011.

[13] M. Budagavi, A. Fuldseth, and G. Bjøntegaard, "Hevc transform and quantization," in *High Efficiency Video Coding (HEVC)*. Springer, 2014, pp. 141–169.

[14] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, 1987.

[15] A. Moffat, "Huffman coding," *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, pp. 1–35, 2019.

[16] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the h. 264/avc video compression standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 620–636, 2003.

[17] V. Sze and M. Budagavi, "High throughput cabac entropy coding in hevc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1778–1791, 2012.

[18] F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, *Emerging technologies for 3D video: creation, coding, transmission and rendering*. John Wiley & Sons, 2013.

[19] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li *et al.*, "Emerging mpeg standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2018.

[20] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "An overview of ongoing point cloud compression standardization activities: Video-based (v-pcc) and geometry-based (g-pcc)," *APSIPA Transactions on Signal and Information Processing*, vol. 9, 2020.

[21] L. Velho and J. Sossai Jr, "Projective texture atlas construction for 3d photography," *The Visual Computer*, vol. 23, no. 9, pp. 621–629, 2007.

[22] A. Maglo, G. Lavoué, F. Dupont, and C. Hudelot, "3d mesh compression: Survey, comparisons, and emerging trends," *ACM Comput. Surv.*, vol. 47, no. 3, Feb. 2015. [Online]. Available: https://doi.org/10.1145/2693443

[23] C. Portaneri, P. Alliez, M. Hemmer, L. Birklein, and E. Schoemer, "Cost-driven framework for progressive compression of textured meshes," in *Proceedings of the 10th ACM Multimedia Systems Conference*, 2019, pp. 175–188.

[24] F. Payan and M. Antonini, "Multiresolution 3d mesh compression," in *Proceedings. International Conference on Image Processing*, vol. 2. IEEE, 2002, pp. II–II.

[25] ——, "Wavelet-based compression of 3d mesh sequences," in *ACIDCA-ICMI'2005*, 2005.

[26] C. M. Mendes, K. Apaza-Agüero, L. Silva, and O. R. P. Bellon, "Data-driven progressive compression of colored 3d mesh," in *2015 IEEE International Conference on Image Processing (ICIP)*, Sep. 2015, pp. 2490–2494.

[27] A. Maglo, I. Grimstead, and C. Hudelot, "Pomar: Compression of progressive oriented meshes accessible randomly," *Computers and Graphics*, vol. 37, no. 6, pp. 743 – 752, 2013, shape Modeling International (SMI) Conference 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0097849313000794

[28] M. Quach, J. Pang, D. Tian, G. Valenzise, and F. Dufaux, "Survey on deep learning-based point cloud compression," *Frontiers in Signal Processing*, 2022.

[29] D. Meagher, "Geometric modeling using octree encoding," *Computer graphics and image processing*, vol. 19, no. 2, pp. 129–147, 1982.

[30] W. Zhu, Y. Xu, L. Li, and Z. Li, "Lossless point cloud geometry compression via binary tree partition and intra prediction," in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2017, pp. 1–6.

[31] O. Devillers and P.-M. Gandoin, "Geometric compression for interactive transmission," in *Proceedings Visualization 2000. VIS 2000 (Cat. No. 00CH37145)*. IEEE, 2000, pp. 319–326.

[32] T. Golla and R. Klein, "Real-time point cloud compression," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 5087–5092.

[33] B. Kathariya, L. Li, Z. Li, J. Alvarez, and J. Chen, "Scalable point cloud geometry coding with binary tree embedded quadtree," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, July 2018, pp. 1–6.

[34] K. Zhang, W. Zhu, and Y. Xu, "Hierarchical segmentation based point cloud attribute compression," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 3131–3135.

[35] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3d point cloud sequences," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1765–1778, April 2016.

[36] C. Cao, M. Preda, and T. Zaharia, "3d point cloud compression: A survey," in *The 24th International Conference on 3D Web Technology*, ser. Web3D '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–9. [Online]. Available: https://doi.org/10.1145/3329714.3338130

[37] F. Hawary, T. Maugey, and C. Guillemot, "Sphere mapping for feature extraction from 360-degree fish-eye captures," in *IEEE International Workshop on Multimedia Signal Processing*. IEEE, 2020.

[38] J. P. Snyder, *Flattening the earth: two thousand years of map projections*. University of Chicago Press, 1997.

[39] E. Kuzyakov and D. Pio, "Next-generation video encoding techniques for 360 video and vr," *Facebook,[Online]*, 2016.

[40] C.-W. Fu, L. Wan, T.-T. Wong, and C.-S. Leung, "The rhombic dodecahedron map: An efficient scheme for encoding panoramic video," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 634–644, 2009.

[41] M. Yu, H. Lakshman, and B. Girod, "A framework to evaluate omnidirectional video coding schemes," in *2015 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2015, pp. 31–36.

[42] R. Ng, "Light field photography," Ph.D. dissertation, Stanford University, 2006.

[43] T. Georgiev, G. Chunev, and A. Lumsdaine, "Superresolution with the focused plenoptic camera," in *Computational Imaging IX*, vol. 7873. International Society for Optics and Photonics, 2011, p. 78730X.

[44] S. Wanner and B. Goldluecke, "Globally consistent depth labeling of 4d light fields," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 41–48.

[45] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 673–680.

[46] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Depth estimation with occlusion modeling using light-field cameras," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2170–2181, 2016.

[47] X. Sun, Z. Xu, N. Meng, E. Y. Lam, and H. K.-H. So, "Data-driven light field depth estimation using deep convolutional neural networks," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 367–374.

[48] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 193, 2016.

[49] J. Shi, X. Jiang, and C. Guillemot, "A framework for learning depth from a flexible subset of dense and sparse light field views," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5867–5880, 2019.

[50] E. Ekmekcioglu, M. Mrak, S. Worrall, and A. Kondoz, "Utilisation of edge adaptive upsampling in compression of depth map videos for enhanced free-viewpoint rendering," in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 733–736.

[51] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map coding with distortion estimation of rendered view," in *Visual Information Processing and Communication*, vol. 7543. SPIE, 2010, pp. 82–91.

[52] X. Jiang, M. Le Pendu, R. A. Farrugia, and C. Guillemot, "Light field compression with homography-based low-rank approximation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1132–1145, 2017.

[53] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, P. T. Kovács, and V. K. Adhikarla, "Subjective evaluation of super multi-view compressed contents on high-end light-field 3d displays," *Signal Processing: Image Communication*, vol. 39, pp. 369–385, 2015.

[54] S. T. Barnard and M. A. Fischler, "Computational stereo," SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER, Tech. Rep., 1982.

[55] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[56] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[57] S. Khamis, S. Fanello, C. Rhemann, A. Kowdle, J. Valentin, and S. Izadi, "Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 573–590.

[58] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *The journal of machine learning research*, vol. 17, no. 1, pp. 2287–2318, 2016.

[59] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Hevc-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 601–605.

[60] S. Rossi and L. Toni, "Navigation-aware adaptive streaming strategies for omnidirectional video," in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2017, pp. 1–6.

[61] M. Hosseini and V. Swaminathan, "Adaptive 360 vr video streaming: Divide and conquer," in *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2016, pp. 107–110.

[62] H. Shum and S. B. Kang, "Review of image-based rendering techniques," in *Visual Communications and Image Processing 2000*, vol. 4067. International Society for Optics and Photonics, 2000, pp. 2–13.

[63] C. Fehn, "Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv," in *Stereoscopic Displays and Virtual Reality Systems XI*, vol. 5291. International Society for Optics and Photonics, 2004, pp. 93–104.

[64] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3d video," in *Applications of Digital Image Processing XXXII*, vol. 7443. SPIE, 2009, pp. 233–243.

[65] C. Guillemot and O. Le Meur, "Image inpainting: Overview and recent advances," *IEEE signal processing magazine*, vol. 31, no. 1, pp. 127–144, 2013.

[66] V. Jantet, C. Guillemot, and L. Morin, "Object-based layered depth images for improved virtual view synthesis in rate-constrained context," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 125–128.

[67] E. Dupraz, T. Maugey, A. Roumy, and M. Kieffer, "Rate-distortion performance of sequential massive random access to gaussian sources with memory," in *2018 Data Compression Conference*, March 2018, pp. 406–406.

[68] M. Wien, *High Efficiency Video Coding: Coding Tools and Specification*. Springer, 2015.

[69] J. Lou, H. Cai, and J. Li, "A real-time interactive multi-view video system," Singapore, 2005, pp. 161–170.

[70] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–644, July 2003.

[71] H. Kimata, M. Kitahara, K. Kamikura, and Y. Yashima, "Free-viewpoint video communication using multi-view video coding," *NTT Technical Review*, vol. 2, no. 8, pp. 21–26, Aug. 2004.

[72] S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, and Y. Yashima, "View scalable multiview video coding using 3-d warping with depth map," vol. 17, no. 11, pp. 1485–1495, Nov. 2007.

[73] Y. Liu, Q. Huang, S. Ma, D. Zhao, and W. Gao, "Rd-optimized interactive streaming of multiview video with multiple encodings," *Journal of Visual Communication and Image Representation*, vol. 21, no. 5-6, pp. 523–532, 2010.

[74] G. Cheung, A. Ortega, and N. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," *IEEE Transactions on Image Processing*, vol. 3, no. 3, pp. 744–761, Mar. 2011.

[75] S. C. Draper and E. Martinian, "Compound conditional source coding, Slepian-Wolf list decoding, and applications to media coding," in *IEEE International Symposium on Information Theory*, 2007.

[76] A. Wyner, "Recent Result in the Shannon theory," *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 2–10, 1974.

[77] V. Stankovic, A.D.Liveris, Z. Xiong, and C. Georghiades, "On code design for the Slepian-Wolf problem and lossless multiterminal networks," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1495 –1507, april 2006.

[78] C. Guillemot and A. Roumy, "Chapter 6 - Toward constructive Slepian-Wolf coding schemes," in *Distributed Source Coding: theory, algorithms and applications*, P. L. Dragotti and M. Gastpar, Eds. Boston: Academic Press, 2009, pp. 131 – 156. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780123744852000111

[79] F. Dufaux, W. Gao, S. Tubaro, and A. Vetro, "Distributed video coding: trends and perspectives," *EURASIP Journal on Image and Video Processing*, vol. 2009, pp. 1–13, 2010.

[80] M. Grangetto, E. Magli, and G. Olmo, "Distributed arithmetic coding for the slepian–wolf problem," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2245–2257, 2009.

[81] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *EURASIP Signal Processing*, vol. 86, no. 11, pp. 3123–3130, 2006.

[82] D. Varodayan, "Implementation of Rate-Adaptive LDPC Accumulate Codes for Distributed Source Coding." [Online]. Available: http://ivms.stanford.edu/~varodayan/ldpca.html

[83] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, pp. 440–442, 2002.

[84] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, Aug. 2014. [Online]. Available: https://epfl-lts2.github.io/gspbox-html/

[85] J. Pfaff, A. Filippov, S. Liu, X. Zhao, J. Chen, S. De-Luxán-Hernández, T. Wiegand, V. Rufitskiy, A. K. Ramasubramonian, and G. Van der Auwera, "Intra prediction and mode coding in vvc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3834–3847, 2021.

[86] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li *et al.*, "Emerging mpeg standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2018.

[87] Y. Ye, J. M. Boyce, and P. Hanhart, "Omnidirectional 360° video coding technology in responses to the joint call for proposals on video compression with capability beyond hevc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1241–1252, 2019.

[88] M. Wien, J. M. Boyce, T. Stockhammer, and W.-H. Peng, "Standardization status of immersive video coding," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 5–17, 2019.

[89] P. Schelkens, P. Astola, E. A. Da Silva, C. Pagliari, C. Perra, I. Tabus, and O. Watanabe, "Jpeg pleno light field coding technologies," in *Applications of Digital Image Processing XLII*, vol. 11137. International Society for Optics and Photonics, 2019, p. 111371G.

[90] W. Hu, J. Pang, X. Liu, D. Tian, C.-W. Lin, and A. Vetro, "Graph signal processing for geometric data and beyond: Theory and applications," *IEEE Transactions on Multimedia*, 2021.

[91] 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.

[92] G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng, "Graph spectral image processing," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 907–930, May 2018.

[93] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press, 2014.

[94] R. Schnabel and R. Klein, "Octree-based point-cloud compression." *Spbg*, vol. 6, pp. 111–120, 2006.

[95] C. Zhang, D. Florencio, and C. Loop, "Point cloud attribute compression with graph transform," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 2066–2070.

[96] R. L. de Queiroz and P. A. Chou, "Transform coding for point clouds using a gaussian process model," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3507–3517, July 2017.

[97] R. A. Cohen, D. Tian, and A. Vetro, "Attribute compression for sparse point clouds using graph transforms," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1374–1378.

[98] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Neighbors do help: Deeply exploiting local structures of point clouds," *arXiv preprint arXiv:1712.06760*, vol. 1, no. 2, 2017.

[99] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 206–215.

[100] 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.

[101] K. M. Gorski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann, "Healpix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere," *The Astrophysical Journal*, vol. 622, no. 2, p. 759, 2005.

[102] M. Maitre and M. N. Do, "Depth and depth–color coding using shape-adaptive wavelets," *Journal of Visual Communication and Image Representation*, vol. 21, no. 5-6, pp. 513–522, 2010.

[103] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 725–743, 2000.

[104] G. Fracastoro, F. Verdoja, M. Grangetto, and E. Magli, "Superpixel-driven graph transform for image compression," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 2631–2635.

[105] 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, 2000.

[106] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[107] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool, "Seeds: Superpixels extracted via energy-driven sampling," *International Journal of Computer Vision*, vol. 111, no. 3, pp. 298–314, 2015.

[108] M. Hog, N. Sabater, and C. Guillemot, "Superrays for efficient light field processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1187–1199, Oct 2017.

[109] 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.

[110] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.

[111] S. A. Khayam, "The discrete cosine transform (dct): theory and application," *Michigan State University*, vol. 114, pp. 1–31, 2003.

[112] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel, "Coupled quasi-harmonic bases," in *Computer Graphics Forum*, vol. 32, no. 2pt4. Wiley Online Library, 2013, pp. 439–448.

[113] T. Sikora and B. Makai, "Shape-adaptive dct for generic coding of video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 1, pp. 59–62, 1995.

[114] A. Loukas, "Graph reduction with spectral and cut guarantees." *Journal of Machine Learning Research*, vol. 20, no. 116, pp. 1–42, 2019.

[115] [Online]. Available: https://www.domo.com/learn/data-never-sleeps-8

[116] J. Ascenso, P. Akyazi, F. Pereira, and T. Ebrahimi, "Learning-based image coding: early solutions reviewing and subjective quality evaluation," in *Optics, Photonics and Digital Technologies for Imaging Applications VI*, vol. 11353. International Society for Optics and Photonics, 2020, p. 113530S.

[117] J. Chen, M. Karczewicz, Y.-W. Huang, K. Choi, J.-R. Ohm, and G. J. Sullivan, "The joint exploration model (JEM) for video compression with capability beyond HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[118] Z. Shi, X. Sun, and F. Wu, "Photo album compression for cloud storage using local features," *IEEE Journal on emerging and selected topics in circuits and systems*, vol. 4, no. 1, pp. 17–28, 2014.

[119] X. Zhang, Y. Zhang, W. Lin, S. Ma, and W. Gao, "An inter-image redundancy measure for image set compression," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2015, pp. 1274–1277.

[120] L. Sha, W. Wu, and B. Li, "Novel image set compression algorithm using rate-distortion optimized multiple reference image selection," *IEEE Access*, vol. 6, pp. 66 903–66 913, 2018.

[121] X. Liu, G. Cheung, C.-W. Lin, D. Zhao, and W. Gao, "Prior-based quantization bin matching for cloud storage of JPEG images," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3222–3235, 2018.

[122] M. G. Blanch, M. Mrak, A. F. Smeaton, and N. E. O'Connor, "End-to-end conditional gan-based architectures for image colourisation," in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2019, pp. 1–6.

[123] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool, "Generative adversarial networks for extreme learned image compression," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 221–231.