



HAL
open science

Rate-Distortion Optimized Graph Coarsening and Partitioning for Light Field Coding

Mira Rizkallah, Thomas Maugey, Christine Guillemot

► **To cite this version:**

Mira Rizkallah, Thomas Maugey, Christine Guillemot. Rate-Distortion Optimized Graph Coarsening and Partitioning for Light Field Coding. *IEEE Transactions on Image Processing*, 2021, 30, pp.5518 - 5532. 10.1109/TIP.2021.3085203 . hal-03230325

HAL Id: hal-03230325

<https://hal.science/hal-03230325v1>

Submitted on 19 May 2021

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

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

Rate-Distortion Optimized Graph Coarsening and Partitioning for Light Field Coding

Mira Rizkallah Thomas Maugey Christine Guillemot
 Inria Rennes Bretagne Atlantique
 e-mail: firstname.lastname@inria.fr

Abstract—Graph-based transforms are powerful tools for signal representation and energy compaction. However, their use for high dimensional signals such as light fields poses obvious problems of complexity. To overcome this difficulty, one can consider local graph transforms defined on supports of limited dimension, which may however not allow us to fully exploit long-term signal correlation. In this paper, we present methods to optimize local graph supports in a rate distortion sense for efficient light field compression. A large graph support can be well adapted for compression efficiency, however at the expense of high complexity. In this case, we use graph reduction techniques to make the graph transform feasible. We also consider spectral clustering to reduce the dimension of the graph supports while controlling both rate and complexity. We derive the distortion and rate models which are then used to guide the graph optimization. We describe a complete light field coding scheme based on the proposed graph optimization tools. Experimental results show rate-distortion performance gains compared to the use of fixed graph support. The method also provides competitive results when compared against HEVC-based and the JPEG Pleno light field coding schemes. We also assess the method against a homography-based low rank approximation and a Fourier disparity layer based coding method.

Index Terms—Light fields, compression, graph transforms, super-rays, graph reduction, graph partitioning

I. INTRODUCTION

Light Fields represent light rays emitted by every point in a scene and along different orientations [1]–[3]. They constitute very large volumes of high-dimensional data that contains redundant information in both the spatial and angular dimensions, hence the need to design efficient compression schemes. The gain in popularity of light fields for a variety of applications has given an impulse to research in light field compression. An overview of recent work can be found in [4].

The correlation or redundancy within a light field follows structures that depend on scene geometry, and therefore may not be regular. Graphs and graph-based transforms are natural tools for processing the signal along irregular structures. In [5] and [6], we have shown that non separable and separable spatio-angular graph-based transforms could allow efficient energy compaction of light fields, and eventually high compression efficiency. More precisely, in [5] and [6], we construct local graphs on individual 4D super-rays with a fixed size that are therefore used as transform supports, instead of traditional blocks in Standard coders. The concept of super-ray has been introduced in [7] as an extension to light fields

of the concept of super-pixels. Super-rays group perceptually similar and corresponding pixels within and across several views, i.e. corresponding to the same set of 3D points of the imaged scene. The color signal lying on these local graphs is compacted into few coefficients by using a Graph Fourier transform. As per their construction, the graph supports may not be fully adapted to the content of the scene and do not guarantee an optimal rate-distortion performance. More precisely, in uniform regions, shrinking the support of the graph may reduce the capability of the transform to decorrelate the signal hence to compact the signal energy. On the other hand, some graph supports may be too large if they group signals with different statistical properties for example in textured or heterogeneous regions.

In this paper, we investigate methods to optimize the graph supports in a rate-distortion sense. This allows us to adjust the super-ray size to the scene content in a way that the computational complexity of the graph transform remains tractable. This complexity is directly related to the number of vertices in the graph. To well capture light field spatial and angular correlation, the 4D super-ray may need to be of large dimension. In that case, to keep the transform complexity tractable, we reduce the graph dimension, i.e. the number of vertices, without altering its basic properties, by using graph coarsening and reduction techniques [8]. The signal can also be reduced to lie on the corresponding smaller subspace. The graph dimensionality reduction is performed in order to have a number of vertices below a number corresponding to an acceptable complexity. However, if the signal approximation on the reduced graph gives a too coarse approximation of the input signal (i.e. corresponding to a distortion higher than a fixed amount), then the graph on the input super-ray is not reduced. In this case, its support, i.e. the underlying 4D super-ray, is partitioned into two sub-graphs using spectral clustering. Furthermore, a super-ray having already an acceptable number of vertices can be considered as too large when two regions with different statistical properties are coded together. This leads to a drop in performance. In this case, the underlying 4D super-ray, is also partitioned into two sub-graphs using spectral clustering.

We derive the distortion and rate models to guide these different graph reduction and partitioning steps, so that the resulting partition and graphs lead to the best rate-distortion performance for a given target complexity. We have assessed the rate-distortion gain obtained with the proposed algorithm in a complete light field coding scheme, in comparison with

standard non separable spatio-angular graph transforms in [5], and in comparison with several reference methods (HEVC-based coding [9] [10], JPEG-Pleno VM 1.1 and 2.1 (using both the WaSP and MuLE configurations) [11] [12], Homography low rank approximation (HLRA) in [13], Fourier disparity layers [14], the disparity prediction based codec of [15] and the translation assisted coder in [16]). Experimental results show substantial gains in rate distortion performance especially at low bitrate.

In summary, our contributions are as follows:

- We propose models to estimate the distortion of the reconstructed signals in a super-ray and the rate needed to code the graph supports and the transform coefficients. These models guide the reduction and partitioning steps.
- We develop a Rate-Distortion optimization method to find optimal graph supports using graph reduction and partitioning to cope with the computational complexity of the graph Fourier transform.
- We propose a full coding scheme that outperforms the previously proposed schemes based on graph spectral light field coding, and that also has competitive results compared to state of the art light field coders.

The rest of the paper is organized as follows. After a brief overview of state of the art light field compression methods and graph transforms in Section II, we introduce the graph coarsening and partitioning methods in Section IV. Section IV also presents the distortion and rate models that we derived for performing the rate-distortion optimization. We then describe the complete light field coding scheme in Section V. Experimental results are presented and discussed in Section VI.

II. RELATED WORK

A. Light field Compression

Research in light field compression has been very active in the past decade, with a variety of solutions which can be broadly classified into the following categories: the solutions that directly encode the lenslet images and those which first extract views from the raw data, after de-vignetting and demosaicing, and then encode the views by exploiting the spatio-angular correlation. A good overview can be found in [4]. Most of the solutions in the first category aim at adding extra prediction modes in existing standards to capture the spatio-angular correlation. This is for example the case in [17]–[22] where the authors use block-based self-similarity techniques, or macro-pixel directional prediction modes as in [23], as additional HEVC Intra prediction modes. Scalable extensions of HEVC have also been considered, e.g., as in [24] where a sparse set of micro-lens images is encoded in a base layer. The other micro-lens images are reconstructed at the decoder using disparity-based interpolation and inpainting. The reconstructed images are then used to predict the entire lenslet image and a prediction residue is transmitted yielding a multi-layer scheme. While the above solutions mostly rely on the HEVC standard, Chao et al. [25] instead apply a graph lifting transform on irregularly spaced color components of pixel in the raw data without demosaicing.

Our proposed approach falls in the second category of methods which compress views extracted from the raw light field data. Solutions encoding the set of views as a pseudo video sequence using standard inter-coding methods have first been investigated, e.g. using HEVC in [9], [26], the latest JEM coder [27], or multi-view HEVC based coding [28], [29] in a single layer, or following a hierarchical approach [30], [31] first coding a subset of views in a base layer using HEVC, which are then used to predict views in an enhancement layer.

While the above techniques significantly rely on block-based prediction mechanisms of standardized solution, methods departing from classical Inter-coding tools have also been proposed, as in [32] where the author exploits inter-view correlation by using homography and 2D warping to predict views, or as in [31] where the authors use a linear approximation computed with Matching Pursuit for disparity based view prediction. View synthesis techniques have also been considered in [31], [33], [34] for reconstructing the entire light field from a sparse set of views, using either depth image-based rendering techniques [34], [35], convolutional neural networks [33], or linear approximation computed with Matching Pursuit for disparity based view prediction [31]. The synthesized set of views are then used as predictors of the original light field views. An approach based on view synthesis has also been adopted in the context of JPEG Pleno [11].

Another direction of research for exploiting spatio-angular correlation has been in the design of sparse coding and transform methods, e.g. with learned dictionaries as in [36] or exploiting signal priors as sparsity in the 4D Fourier domain [37], [38], or sparsity in the shearlet transform domain [39]. Various models and transforms have also been proposed for light field de-correlation and compression. A global homography-based low rank approximation approach is introduced in [13] while, in [40], the authors describe a framework referred to as Steered Mixture-of-Experts (SMoE) where high-dimensional kernels are used to sparsely represent the plenoptic function. Local transforms applied either on 4D blocks using 4D-DCT [41], or graph-based transforms defined locally on super-rays in [42] have also been explored for light field compression.

B. Simplifying graph structures : sparsification and reduction

As graphs may grow in size, it becomes important to look for comprehensive ways of simplifying their structures while preserving some key properties. We can simplify the graphs by reducing the number of edges, also known as graph sparsification ([43]–[46]). As such, sparsification can yield computational benefits in particular whenever the number of edges is the main bottleneck, mainly relevant for machine learning techniques concerning graph-structured data. Alternatively, we may seek to reduce directly the number of vertices in the graph, by some form of vertex selection or re-combination scheme followed by re-wiring. This is known as graph reduction and form a key element in modern graph signal processing pipelines, especially with regards to graph partitioning ([47]–[49]). In machine learning, graph reduction is used to create multi-scale representations of graph signals ([50]–[52]), and as a layer in graph convolutional neural networks ([53]–[55]). Main benefits of graph reduction techniques

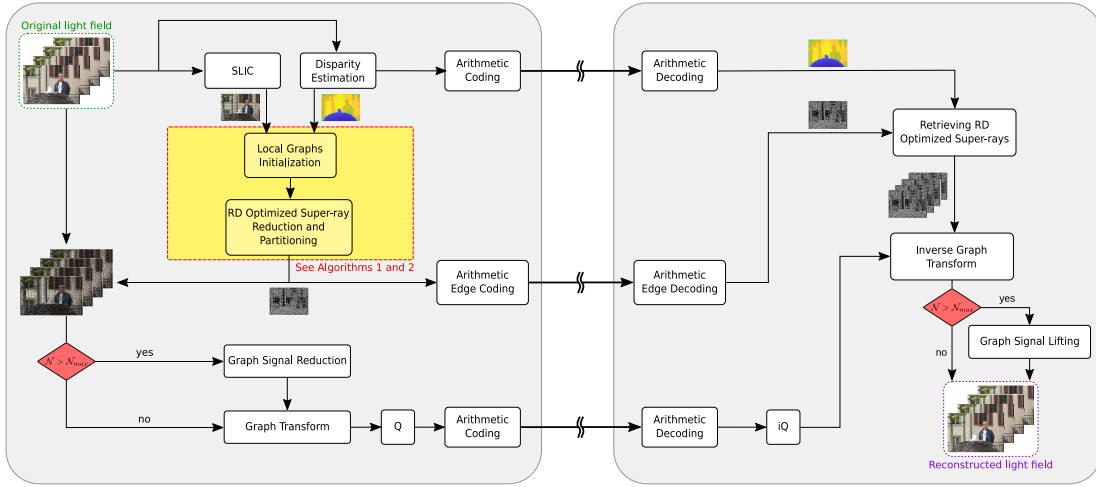


Fig. 1: Proposed light field coding scheme. The proposed algorithms in section IV-D for rate-distortion optimization of the Graph Transform supports correspond to the pipeline steps shown in the yellow box.

reside in their ability to deal with sparse graphs (N nodes with at most $N \log(N)$ edges), as well as accelerating the graph based algorithms whose complexity is mainly dependent of the number of nodes and edges in the graph, such as the graph Fourier transform.

In this paper, in order to reduce the complexity of geometry-aware non separable graph transforms in the 4D light field data, we propose to simplify the graphs. We rely on a specific kind of reduction, i.e. graph coarsening where at each level, reduced vertices are formed by contracting disjoint sets of connected nodes. We use the recently proposed method [8] based on local variation algorithms in order to approximate the original large graphs by a coarser graph at the expense of a restricted spectral approximation - an important property for compression.

C. Graph Fourier Transform

Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and a signal \mathbf{x} residing on this graph. Given an adjacency matrix \mathbf{A} where $A_{i,j} = 1$ if v_i and v_j are connected with an edge and 0 otherwise, we define a diagonal *degree matrix* \mathbf{D} , where $d_{i,i} = \sum_j A_{i,j}$. The *combinatorial graph Laplacian matrix* \mathbf{L} is $\mathbf{L} = \mathbf{D} - \mathbf{A}$ [56]. Because \mathbf{L} is symmetric positive semi-definite, it can be eigen-decomposed into:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \quad (1)$$

where $\mathbf{\Lambda}$ is a diagonal matrix containing real positive eigenvalues λ_k along the diagonal, and \mathbf{U} is composed of orthogonal eigenvectors \mathbf{u}_l as columns. Eigenvalues λ_l can be interpreted as *graph frequencies*, and eigenvectors \mathbf{U} interpreted as corresponding graph frequency components [56]. The set of eigenvectors in \mathbf{U} form the *graph Fourier transform* (GFT), which can be used to decompose a graph-signal \mathbf{x} residing on \mathcal{G} into its frequency components via $\boldsymbol{\alpha} = \mathbf{U}^T \mathbf{x}$, similar to known discrete transforms such as DCT. The benefit of such representation and transform is that it can be applied on irregular domains, and thus on more flexible supports that are shape-adaptive.

A signal \mathbf{x} is considered to be smooth on a graph if strongly connected vertices have similar signal values. This is usually quantified in terms of the Laplacian quadratic form (the total variation):

$$\mathcal{S}(\mathbf{x}) = \mathbf{x}^\top \mathbf{L} \mathbf{x} \quad (2)$$

In general, graph based coding methods as in [42] make use of the graph Fourier transforms to capture the main characteristics of the signals. The coefficients are thus encoded instead of the original signals. The smoother the signal is on a graph, the more its energy is concentrated in low frequencies, and the more it is easily compressible.

III. NOTATIONS AND SCHEME OVERVIEW

Let $L(u, \tau, x, y)$ be a light ray of the light field L , and (u, τ, x, y) its coordinates using the two plane parametrization, where (u, τ) and (x, y) are the angular (view) and spatial (pixel) coordinates respectively. A super-pixel SP_i denotes a group of rays within the same view (u_i, τ_i) and a super-ray SR extends that concept by grouping super-pixels across all views of the light field [7].

In the proposed coding scheme depicted in Figure 1, each super-ray is the support of a graph-based transform and is optimized in a rate-distortion sense in order to have a support that best captures spatio-angular correlation while keeping the graph transform computation tractable. First, the initial segmentation map of the Light field top-left view is computed with the well known SLIC algorithm [57]. We then estimate the disparity maps of all views with the method proposed in [58]. These maps are used in the proposed RD-optimized graph partitioning to segment the top-left view. The resulting segmentation map is coded along with the assigned disparity values per super-ray as additional side information. Non separable graph Fourier transforms are then applied separately on graph signals obtained after graph reduction or after partitioning. Transform coefficients are then quantized, encoded with arithmetic coding and sent to the decoder.

The decoder, having received the optimized segmentation of the top-left view, along with the necessary disparity information, is able to construct the 4D super-rays of the light

field i.e. the graph supports (with or without reduction). With the retrieval of the graphs and the quantized transform coefficients, the inverse quantization, graph lifting (if reduction was applied) and inverse graph Fourier transform are applied to recover the 4D light field.

In the next section, we describe the proposed R-D optimized graph coarsening and partitioning method, and in Section V we explain in detail the coding components of the entire scheme.

IV. R-D OPTIMIZED GRAPH COARSENING AND PARTITIONING

A. Problem formulation

In order to efficiently capture spatial and angular correlations in the 4D light field, we want to find local graphs $\{\mathcal{G}_k\}$ on individual 4 dimensional super-rays $\{SR_k\}$ relying on the geometrical information. Those graphs are further used as transform graph supports, instead of traditional blocks in standard coders. The color signals $\{\mathbf{x}_k\}$ residing on those graphs are transformed with the Graph Fourier Transform as detailed above. The complexity of the graph transform can be limited by restricting the size of the graph. However, limiting the graph size prevents us from capturing long term signal dependencies of the signal hence from compacting the signal energy. On the other hand, increasing the dimension of the graph enables us to capture long term dependencies but at the expense of a high graph transform complexity. In addition, if the graph is too large, it may group regions of the image with different statistical characteristics, leading to a low energy compaction of the graph Fourier transform. In that case, smaller super-rays that adhere well to object boundaries should be favored.

In this work, we aim at finding optimized local graph supports in a rate-distortion sense with a constraint on the graph transform complexity *i.e.* the maximum number of nodes in the local graphs, which limits the resolution of the visual signals that can be supported. First, we want the graph supports to group pixels both in spatial and angular domain in order to capture correlations in the 4 dimensions of the light field. We also want to have larger graphs in uniform regions to capture long term signal dependencies and smaller graphs in non uniform regions. In order to define those graphs, we are equipped with two graph dimensionality reduction tools namely graph coarsening and graph partitioning.

We therefore aim to split the global graph of a light field into several smaller sub-graphs $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_M\}$ with their respective signals $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ which can successfully represent the 4D Light field while respecting the constraint defined above. Each graph is $\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i, \mathbf{A}_i\}$ where \mathcal{V}_i are the vertices in the graph, with $|\mathcal{V}_i| < N_{max}$, \mathcal{E}_i are their edges, and \mathbf{A}_i is the adjacency matrix. \mathbf{x}_i is the signal defined on the i^{th} graph. The signals on each of those graphs can be independently processed, and transformed separately using the graph Fourier transform defined above. Our problem can be

formulated as the following rate-distortion minimization:

$$\begin{aligned} \min_{\tilde{\mathcal{G}} = \{\mathcal{G}_i\}} \quad & \mathcal{R}_C(\tilde{\mathcal{G}}) + \mathcal{R}_B(\tilde{\mathcal{G}}) \\ \text{subject to} \quad & \mathcal{D}(\mathcal{G}_i) < D_{max} \forall i \\ & N_i < N_{max} \forall i \end{aligned} \quad (3)$$

$\tilde{\mathcal{G}} = \{\mathcal{G}_i\}$ represents the set of *local graphs* capturing local color information and the color variation inside the 4D light field. $\mathcal{D}(\mathcal{G}_i)$ is the distortion between the original signal and the reconstructed one on the i^{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. Each of these terms possibly depend on the chosen partition of the graph and of the coding scheme envisioned. We propose one way of solving (3) where we initialize the optimization process with very large graph supports and then reduce them: either lower the complexity with graph coarsening when the signal is sufficiently smooth or reduce the graph dimension by splitting, when the content is either textured or heterogeneous. For both graph coarsening and splitting, we derive the distortion and rate models in order to guide the steps in a full R-D optimized graph partitioning algorithm for coding light fields. The two methods, *i.e.*, the graph coarsening and the graph partitioning into smaller sub-graphs are illustrated in the upper and lower branches respectively of Figure (2) for a given super-ray.

B. Graph reduction by coarsening

One way to reduce the dimension of a local graph \mathcal{G} , *i.e.* reducing the number of nodes, is to find a coarse graph that approximates it. Basically, having an initial graph $\mathcal{G}_0 = \{\mathcal{V}, \mathcal{E}, \mathbf{L}_0\}$ and a signal $\mathbf{x}_0 = \mathbf{x}$, in a typical reduction scheme, we proceed to the following recursive equations as detailed in [8]:

$$\mathbf{L}_\ell = \mathbf{P}_\ell^\mp \mathbf{L}_{\ell-1} \mathbf{P}_\ell^+ \quad \mathbf{x}_\ell = \mathbf{P}_\ell \mathbf{x}_{\ell-1} \quad (4)$$

where $\mathbf{P}_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ are matrices with more columns than rows, and ℓ is the level of reduction. Symbols $+$ and \mp denote the pseudo-inverse and the transpose pseudo-inverse respectively. The reduced signal \mathbf{x}_c can be lifted back to \mathbb{R}^N by recursion $\tilde{\mathbf{x}}_{\ell-1} = \mathbf{P}_\ell^+ \tilde{\mathbf{x}}_\ell$ with $\tilde{\mathbf{x}}_c = \mathbf{x}_c$. We can express the reduced quantities (the signal and the Laplacian), in a more compact form as:

$$\begin{aligned} \mathbf{x}_c &= \mathbf{P} \mathbf{x} \\ \mathbf{L}_c &= \mathbf{P}^\mp \mathbf{L}_0 \mathbf{P}^+ \\ \tilde{\mathbf{x}} &= \mathbf{\Pi} \mathbf{x}, \end{aligned} \quad (5)$$

with $\mathbf{P} = \mathbf{P}_c \dots \mathbf{P}_1$, $\mathbf{P}^+ = \mathbf{P}_1^+ \dots \mathbf{P}_c^+$ and $\mathbf{\Pi} = \mathbf{P}^+ \mathbf{P}$ and $\tilde{\mathbf{x}}$ is the lifted vector. The resulting graph is denoted by $\mathcal{G}_c = \{\mathcal{V}_c, \mathcal{E}_c, \mathbf{L}_c\}$. Even in the general setting, where the reduction matrix \mathbf{P} is arbitrary, we can note two main properties. The first is that \mathbf{L}_c is also positive semi-definite after reduction. Moreover, for every vector in $\mathbf{x} \in im(\mathbf{\Pi})$, we have $\mathbf{x}^\top \mathbf{L} \mathbf{x} = \mathbf{x}_c \mathbf{L}_c \mathbf{x}_c$.

Coarsening is a special case of graph reduction abiding to a set of constraints that render the graph transformation explainable. At each coarsening level, contraction sets are

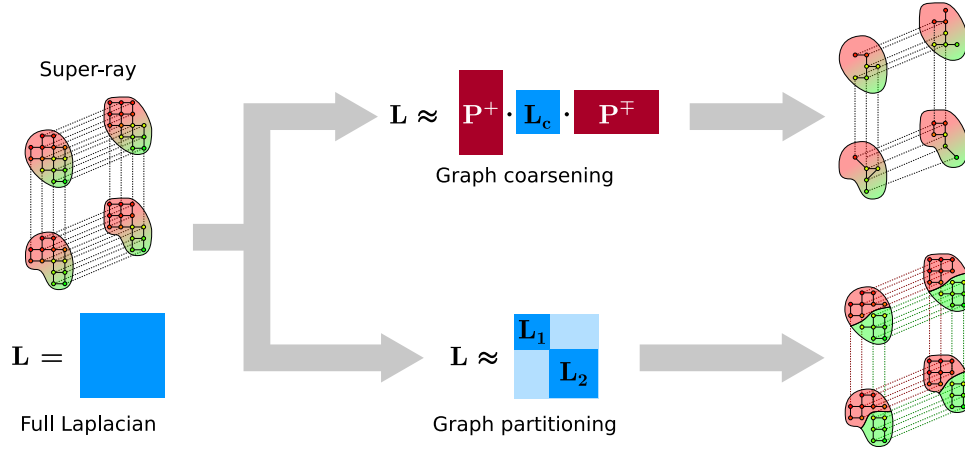


Fig. 2: Illustration for one super-ray of the graph dimensionality reduction methods, i.e., graph coarsening and graph partitioning, used in our Rate-Distortion optimization.

formed from the vertices. As such, every reduced variable at level (l) corresponds to a small set of adjacent vertices in the graph of level ($l - 1$) and coarsening amounts to a scaling operation. The set of adjacent vertices is denoted by $\mathcal{V}_{l-1}^{(r)}$ and is called a contraction set to produce one vertex in the reduced graph denoted by v_r^l . As each level of reduction, the matrix \mathbf{P}_l is a coarsening matrix if and only if it satisfies the following two conditions:

- 1) It is a surjective mapping of the vertex set. In other words, if $\mathbf{P}_l(r, i) \neq 0$ then $\mathbf{P}_l(r', i) = 0 \forall r' \neq r$
- 2) It is locality preserving, equivalently, the sub-graph of \mathcal{G}_{l-1} induced by the non-zero entries of $\mathbf{P}_l(r, :)$ is connected for each r .

This method of constructing the matrices \mathbf{P}_l has two interesting properties. First, with the previous definition of the coarsening matrix, the expensive pseudo-inverse computation can be substituted by simple transposition and re-scaling. More precisely, it can be shown that the pseudo-inverse of the matrix \mathbf{P}_l is given by $\mathbf{P}_l^+ = \mathbf{P}_l^\top \mathbf{D}^{-2}$ where \mathbf{D} is a diagonal matrix defined as $\mathbf{D}_{r,r} = \|\mathbf{p}_r^l\|_2$, with \mathbf{p}_r^l being the r^{th} row of \mathbf{P}_l . Moreover, if we constrain at each level all non zero entries of \mathbf{P}_l^+ to be equally valued, then the resulting coarsened matrix \mathbf{L}_l is also a graph laplacian matrix. In the case of laplacian consistent coarsening as defined before, it has been shown that for every node v_r in the reduced graph \mathcal{G}_l and node v_i in the graph at the previous level \mathcal{G}_{l-1} , the matrices $\mathbf{P}_l \in \mathbb{R}^{N_l \times N_{l-1}}$ and $\mathbf{P}_l^+ \in \mathbb{R}^{N_{l-1} \times N_l}$ are defined as:

$$[\mathbf{P}_l](r, i) = \begin{cases} \frac{1}{\|\mathcal{V}_{l-1}^{(r)}\|} & \text{if } v_i \in \mathcal{V}_{l-1}^{(r)} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$[\mathbf{P}_l^+](i, r) = \begin{cases} 1 & \text{if } v_i \in \mathcal{V}_{l-1}^{(r)} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

for all v_r in \mathcal{V}_l . The resulting laplacian after reduction can be further used for applying graph transform on the reduced signals.

The work in [8] shows that a graph can be reduced such that its fundamental structural properties are preserved namely its first eigenvectors and eigenvalues. The methods based on local variation algorithms [8] are tailored to provide the

following restricted spectral approximation guarantee:

Definition 1: restricted spectral approximation guarantee : Let \mathbf{R} be a k -dimensional subspace of \mathbb{R}^N . Matrices \mathbf{L}_c and \mathbf{L} are (\mathbf{R}, ϵ) -similar if there exists an $\epsilon \geq 0$ such that :

$$\|\mathbf{x} - \tilde{\mathbf{x}}\|_{\mathbf{L}} \leq \epsilon \|\mathbf{x}\|_{\mathbf{L}} \quad \forall \mathbf{x} \in \mathbf{R} \quad (8)$$

where $\tilde{\mathbf{x}} = \mathbf{P}^+ \mathbf{P} \mathbf{x}$.

where $\|\mathbf{a}\|_{\mathbf{L}} = \mathbf{a}^\top \mathbf{L} \mathbf{a}$. If we define \mathbf{R} to be the space spanned by the k first eigenvectors of the original laplacian \mathbf{U}_k , then ensuring that ϵ is the smallest possible implies that eigenvalues and eigenvectors of the original graph and its reduced counterpart are aligned. In that way, an appropriately constructed coarse graph aims to capture the global problem structure of the original graph, i.e. the low frequencies of the signals. In our experiments, we modify the multi-level coarsening algorithm of the paper [8] by setting the threshold $\epsilon' = \infty$, aiming at a fixed reduction ratio at each level, instead of the smallest ϵ .

1) *Distortion estimation*: Since the reduction procedure is independent in each local graph, then the whole distortion is the sum of the distortions on each of the graphs $\{\mathcal{G}_i\}$. The distortion of the lifted signal, w.r.t the original signal on a graph \mathcal{G}_i depends not only on the quality of the coarsening itself, but also on the signal statistical properties. In a typical compression scheme, first we would find the reduced counterpart of the original signal on the coarse graph, apply a graph transform, then send the transformed coefficients to the decoder side, with or without quantization. The decoder having received the transformed coefficients can perform the inverse graph transform then lift back the recovered reduced signal to the original space.

Consider one specific graph \mathcal{G}_i (i.e. one super-ray in the light field) with the signal \mathbf{x}_i (i.e. the luminance values) lying on its nodes. After reduction, the coarsened signal is $\mathbf{x}_{ci} = \mathbf{P} \mathbf{x}_i$ where c is the last coarsening level. A graph transform is applied on the reduced signal \mathbf{x}_{ci} by projecting it on the eigenvectors of the coarse graph: $\hat{\mathbf{x}}_{ci} = \mathbf{U}_{ci}^\top \mathbf{x}_{ci}$. Once the transformed coefficients are computed, they will be sent to the decoder side after quantization. The inverse transform is applied in the decoder to retrieve $\tilde{\mathbf{x}}_{ci} = \mathbf{U}_{ci} \hat{\mathbf{x}}_{ci}$ where $\hat{\mathbf{x}}_{ci}$

are the quantized coefficients. Finally, $\tilde{\mathbf{x}}_{ci}$ is lifted back to the original space by $\tilde{\mathbf{x}}_i = \mathbf{P}^+ \tilde{\mathbf{x}}_{ci}$. The distortion is computed as the mean squared error between the original signal \mathbf{x}_i and the lifted signal $\tilde{\mathbf{x}}_i$ as follows:

$$\mathcal{D}(\mathcal{G}_i) = \frac{1}{N_i} \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 = \frac{1}{N_i} \|\mathbf{x}_i - \mathbf{P}^+ \mathbf{U}_{ci} \hat{\mathbf{x}}_{ci}\|^2 \quad (9)$$

If no quantization then

$$\begin{aligned} \hat{\mathbf{x}}_{ci} &= \hat{\mathbf{x}}_{ci} \\ &= \mathbf{U}_{ci}^\top \mathbf{x}_{ci} \\ &= \mathbf{U}_{ci}^\top \mathbf{P} \mathbf{x}_i \end{aligned} \quad (10)$$

Incorporating the latter in (9) and since the graph transform is orthonormal, using only the signal in the spatial domain, we can compute the exact distortion due to the coarsening procedure as:

$$\mathcal{D}_0(\mathcal{G}_i) = \frac{1}{N_i} \|\mathbf{x}_i - \mathbf{P}^+ \mathbf{P} \mathbf{x}_i\|^2 = \frac{1}{N_i} \|\mathbf{x}_i - \mathbf{P} \mathbf{x}_i\|^2 \quad (11)$$

If quantization, then we have to account for the quantization errors.

$$\begin{aligned} \mathcal{D}_q(\mathcal{G}_i) &= \frac{1}{N_i} \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 \\ &= \frac{1}{N_i} \|\mathbf{x}_i - \mathbf{P}^+ \tilde{\mathbf{x}}_{ci}\|^2 \\ &= \frac{1}{N_i} \|\mathbf{x}_i - \mathbf{P}^+ \tilde{\mathbf{x}}_{ci} + \mathbf{P}^+ \mathbf{x}_{ci} - \mathbf{P}^+ \mathbf{x}_{ci}\|^2 \\ &= \frac{1}{N_i} \|\mathbf{x}_i - \mathbf{P}^+ \mathbf{x}_{ci} + \mathbf{P}^+ \mathbf{x}_{ci} - \mathbf{P}^+ \tilde{\mathbf{x}}_{ci}\|^2 \\ &= \frac{1}{N_i} \|\mathbf{x}_i - \mathbf{P}^+ \mathbf{P} \mathbf{x}_i + \mathbf{P}^+ (\mathbf{x}_{ci} - \tilde{\mathbf{x}}_{ci})\|^2 \end{aligned} \quad (12)$$

Using the inequalities of Frobenius norms, we get

$$\begin{aligned} \mathcal{D}_q(\mathcal{G}_i) &= \frac{1}{N_i} \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 \\ &\leq (\sqrt{\mathcal{D}_0(\mathcal{G}_i)} + \frac{1}{\sqrt{N_i}} \|\mathbf{P}^+ \|\|\mathbf{x}_{ci} - \tilde{\mathbf{x}}_{ci}\|)^2 \\ &\leq (\sqrt{\mathcal{D}_0(\mathcal{G}_i)} + \|\mathbf{P}^+\| \sqrt{\mathcal{D}_{G_{ci}}})^2 \end{aligned} \quad (13)$$

It is thus clear that the upper bound of the distortion is related to both the error due to the coarsening and lifting procedure and also to the quantization that altered the reduced signal. If we consider high bitrate compression and a uniform quantization with step size q , $\mathcal{D}_{G_{ci}}$ can be approximated by $\frac{q^2}{12}$, independent of the coarsening procedure.

2) *Rate approximation*: As per its construction, the matrix \mathbf{P} , is equivalent to an orthogonal projection when applied to the signal \mathbf{x} . Also, when the distortion on the lifted signal $\tilde{\mathbf{x}} = \mathbf{P}^+ \mathbf{P} \mathbf{x}$ is restricted to be lower than a certain threshold τ , then we are sure that ϵ in Equation (8) is also small. This induces that:

$$\mathbf{x}_c^\top \mathbf{L}_c \mathbf{x}_c \approx \mathbf{x}^\top \mathbf{L} \mathbf{x} \quad (14)$$

Moreover, the number of coefficients that we will be sending to represent the reduced signals is much lower than the initial number of coefficients on the original high dimensional graph. We can thus assume that the bitrate needed to send reduced signals is lower than the bitrate needed to code the original signals when the distortion on the lifted signals is constrained.

C. Graph spectral clustering

When the reduction is not feasible on the graph without degrading the quality of the signal to below a specified distortion D_{max} , we propose to use a second method to reduce the dimension of our graphs: the graph spectral clustering. In this case, the nodes in the graph are divided into two sets, and new local graphs are constructed between them as shown in figure 3.

In order to find the optimal clustering into two sets \mathcal{V}_A and \mathcal{V}_B , we rely on the normalized cut algorithm, which is known for favoring the smoothness inside partitions. More precisely, in the specific super-ray \mathcal{SR}_k , we first extract the spatial graph from the top left view. We create a new spatial graph keeping the same connectivity but with weights depending on the euclidean distances in the Y space (*i.e.* the luminance space) and in the disparity domains relying on the original light field color and disparity information. We define weights as

$$w_{i,j} = \begin{cases} \exp\left(-\frac{\delta_Y^2(i,j)}{\sigma(\delta_Y)}\right) \exp\left(-\frac{\delta_d^2(i,j)}{\sigma(\delta_d)}\right) & \text{if } a_{i,j} \neq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

where $a_{i,j}$ is the connectivity between the vertices i and j , $\sigma(\delta_Y)$ and $\sigma(\delta_d)$ are the standard deviations of the euclidean distance in the color and disparity domains respectively. The disparity information is here to guide the steps along with the color information, so that any two objects having the same color but different disparities in the scene can be well segmented, and so that trivial solutions are avoided. Note that this graph is only used for the normalized cut algorithm, and is thus not transmitted nor used for the graph Fourier transforms. At the output of the normalized cut procedure, we have the coloring of the nodes into two sets in the top-left view: \mathcal{V}_A^1 and \mathcal{V}_B^1 . Using the partial coloring of the vertices on this view, the nodes \mathcal{V} in the original local graph \mathcal{G} can be divided into two sets \mathcal{V}_A and \mathcal{V}_B , as follow.

We first compute a mask to define where the initial super-ray existed in the original segmentation. The median disparity values per set of nodes d_A and d_B are assigned to all the vertices in \mathcal{V}_A^1 and \mathcal{V}_B^1 respectively. We project the labels from the first view to the other views using the disparity assigned for each node by filling only the pixels where the mask is true. More precisely, the algorithm proceeds row by row. In the first row of views, we perform horizontal projections from the top-left to the other views next to it. For each other row of views, a vertical projection is first carried out from the top view then horizontal projections to the other views are performed. The pixels that end up outside the initial super-ray are discarded. The occluded pixels are assigned the label of the neighboring super-ray corresponding to the foreground objects (*i.e.* having the higher disparity). As for appearing pixels (the remaining blanks), they will be clustered with the neighboring background super-ray (*i.e.* having the lower disparity).

Once the vertices are assigned to two different sets \mathcal{V}_A and \mathcal{V}_B , we reconstruct local graphs connecting the nodes in both the spatial and angular domains of the light field. As such, instead of having a graph \mathcal{G} , we end up with two smaller graphs \mathcal{G}_A and \mathcal{G}_B .

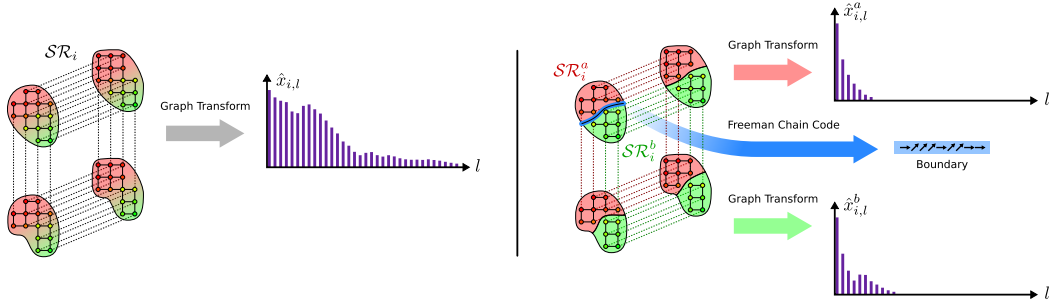


Fig. 3: Toy example of graph partitioning on a small super-ray defined on 4 light field views.

1) *Distortion estimation*: For a specific graph, since the graph Fourier transform is an orthonormal transform, the distortion is equivalent to :

$$\mathcal{D}(\mathcal{G}_i) = \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 = \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{i,q}\|^2 \quad (16)$$

where \mathbf{x}_i and $\tilde{\mathbf{x}}_i$ are the original signal and the decoded signal respectively, $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_{i,q}$ are the original and quantized signal graph Fourier coefficients. If we consider a uniform scalar quantizer with small quantization step sizes q for all coefficients, then

$$\mathcal{D}(\mathcal{G}_i) = q^2 \frac{N_i}{12} \quad (17)$$

with N_i being the number of nodes in the graph \mathcal{G}_i . The distortion is thus independent of the fact that the graph is partitioned or not into two sets, since at the end, the number of nodes is the same. The problem in Equation (3) is reduced to minimizing the rate terms with only one constraint: the maximum number of nodes.

2) *Rate of the transform coefficients*: We can evaluate the rate of the graph Fourier coefficients $\mathcal{R}_C(\mathcal{G}_i)$ in the i^{th} local graph using a modified normalized version of the approximation in [59] based on the total variation of the signal on a graph:

$$\begin{aligned} \mathcal{R}_C(\mathcal{G}_i) &= \mathcal{S}_{norm}(\mathbf{x}_i) = \frac{\mathbf{x}_i^\top \tilde{\mathbf{L}}_i \mathbf{x}_i}{\sum \tilde{d}_i} \\ &= \frac{\mathbf{x}_i^\top (\mathbf{L}_i + \gamma \mathbf{u}_0 \mathbf{u}_0^\top) \mathbf{x}_i}{\sum \tilde{d}_i} \\ &= \frac{\sum_l \lambda_l \hat{x}_{i,l}^2 + \gamma \hat{x}_{i,0}^2}{\sum_l \lambda_l + \gamma} \end{aligned} \quad (18)$$

The altered version of the laplacian *i.e* $\tilde{\mathbf{L}}_i$ is used in order to take into account the DC term. \mathbf{u}_0 is the first eigenvector of the Laplacian \mathbf{L}_i equally valued on all the nodes with $\frac{1}{\sqrt{N_i}}$. λ_l and $\hat{x}_{i,l}$ are the eigenvalues of the original laplacian and the transformed Fourier coefficients of the signal \mathbf{x}_i . Hence the aforementioned approximation is an eigenvalue weighted mean of the of sum of squared transform coefficients which depends on both the underlying graph \mathcal{G}_i and the statistical properties of the signal \mathbf{x}_i residing on its vertices. Such proxy assumes that the bitrate of the transform coefficients increases when the normalized smoothness of the signal on the graph decreases and when the DC coefficients are smaller. This approximation relates also to the energy compaction property of an orthonormal transform, where we aim at having most of the energy compacted in the low frequency coefficients. The less \mathcal{S}_{norm} is, the higher the energy is compacted in the

coefficients corresponding to the smallest eigenvalues *i.e* the low frequencies of the graph Fourier transform.

3) *Rate of the graph description*: When we partition a graph, then we have to send the information that describes this partitioning to the decoder side. In our method, we impose that each partition forms a connected component of the original graph. Thus a common way to code the coloring of the vertices (the cluster's membership) is to code the boundary between the partition. In order to approximate the coding rate of the boundary ζ between two partitions A and B , we use 8-directional differential freeman chaincodes [60] and estimate the coding rate of this boundary as the entropy computed as follows:

$$\mathcal{C}_\zeta(A, B) = -\#_l \sum_{k=1}^8 p_k \log_2(p_k), \quad (19)$$

where $\#_l$ is the number of chaincodes of the boundary ζ and $\{p_k\}$ are the probabilities of each of the 8 directions.

4) *Prediction of the cut decision*: For a given target distortion and a given super-ray \mathcal{SR} with graph \mathcal{G} where $N < N_{max}$, the rate performance of the compression scheme depends on a key parameter: the decision to partition the graph or not. To automatically predict the cut decision, we train a model represented by a function f of a set of input features:

$$\text{Decision}(\mathcal{SR}) = f(\mathcal{SR} \text{ features, target distortion}). \quad (20)$$

To generate training data labels, we compute several super-rays with different resolutions on different datasets. We then compute the entropy of the graph transform coefficients \mathcal{H}_B^q on each of them with different quantization step sizes (as such we can vary the distortion). Afterwards, we apply spectral clustering on each and extract the two partitions. We apply graph Fourier transforms on the resulting partitions and also compute the entropy of the transform coefficients \mathcal{H}_A^q and \mathcal{H}_B^q . We find the rate of the boundary between them as already detailed (in IV-C3). In fact, splitting a graph into two implies removing the connections on the borderline between them, hence adding a boundary. In order to have the best RD trade-off in a super-ray, at each iteration, we should partition if and only if this brings a decrease in the criterion in (3). In other words, the difference in bitrate can be found as:

$$\Delta \mathcal{R}(\mathcal{G}) = N \mathcal{H}_0^q - (N_A \mathcal{H}_A^q + N_B \mathcal{H}_B^q + \mathcal{C}_\zeta(A, B)) \quad (21)$$

Then, the labels can be extracted by taking the best decision (cut or not) corresponding to the lowest rate needed for coding as follows:

$$\text{Decision}(\mathcal{G}) = \begin{cases} 1 & \text{if } \Delta \mathcal{R}(\mathcal{G}) \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (22)$$

Feature space: The decision prediction is regarded as a classification problem with the following input features:

- An indicator of the statistical properties of the signal \mathbf{x}_i on the initial graph \mathcal{G}_i in the super-ray \mathcal{SR}_i without partitioning: the normalized modified total variation as defined in Equation (18) of the signal \mathbf{x}_i on the graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathbf{L}_i)$.
- An indicator of the statistical properties of the signal \mathbf{x}_i in the super-ray after partitioning: the normalized modified total variation of \mathbf{x}_i on a modified graph $\mathcal{G}_i^{cut} = (\mathcal{V}_i, \mathcal{E}_i, \mathbf{L}_i^{cut})$, where we reduce the weights on the edges between the two clusters *i.e.* partitions retrieved after applying spectral clustering.
- An indicator for the boundary rate: $\mathcal{C}_\zeta(A, B)$ as defined in Equation (19).
- The quantization step size corresponding to a target distortion.

Support Vector Machine: The model input features are all discrete variables. SVM are known to be robust for learning classification with discrete variables. We then chose it for learning our classifier. The defined above have been pre-processed with a whitening procedure to remove noise. A standard scaler has been applied to whiten each of the features, and has been saved for testing time.

D. RD optimization algorithm

Having set all the approximations detailed in the previous sections, finding the solution of the problem in Equation (3) is in general a combinatorial task, so we will solve it using traditional agglomerative approximations.

In order to initialize the graph supports and signals prior to the optimization procedure, we rely on a partitioning of the entire 4D light field into geometry-aware super-rays as already explained in [5]. We compute super-pixels in the top-left view using the well-known SLIC algorithm that consists of a k-means clustering, taking into account depth, spatial neighborhood and color information. Then, using the estimated disparity map, we project the segmentation labels to all the other views. Namely, having a segmentation map in the top left view and the corresponding disparity map, we compute the median disparity per super-pixel, and use it to project the segmentation mask to the other views.

We build a local graph per super-ray so we can jointly capture spatial and angular correlations between pixels in the light field. If we consider the luminance values in the whole light field and a segmentation map S , the k^{th} super-ray \mathcal{SR}_k is represented by a graph signal $\mathbf{x}_k \in \mathbb{R}_{N_k}$ defined on an undirected connected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ which consists of a finite set \mathcal{V} of vertices corresponding to the pixels at positions $\{u_l, v_l, x_l, y_l\}, l = 1 \dots N$ such that $S(u_l, v_l, x_l, y_l) = k$. A set \mathcal{E} of edges connect each pixel and its 4-nearest neighbors in the spatial domain (*i.e.* in each view), and to its corresponding pixels, found by disparity based projection (with the disparity being equal to the median disparity per super-ray), in the 4 nearest neighboring views.

At the output of the initialization, we have a undersegmentation with few super-rays corresponding to β non overlapping

Algorithm 1: Light Field Rate Distortion optimized reduction and partitioning Algorithm

Data: The set of graphs and signals for all initial super-rays, the disparity map and labels map of the light field, the maximum tolerated number of vertices, maximal distortion:

$$\{\mathcal{G}_i = \{\mathcal{V}^i, \mathbf{L}_i, \mathbf{x}_i\}, \mathbf{D}_{ref}, \mathbf{S}_{ref}, N_{max}, \mathcal{D}_{max}$$

Result: Optimized segmentation map (super-rays) and local graphs: $\mathbf{SM}_{opt}, \{\tilde{\mathcal{G}}_i\}$

Initialize \mathbf{S}_{opt} and $\tilde{\mathcal{G}} \leftarrow \emptyset$;

foreach Super-ray \mathcal{SR}_i **do**

$\mathcal{M}_i \leftarrow [S_{ref} = i]$;

$\mathbf{S}_{opt}^i, \tilde{\mathcal{G}}_i \leftarrow \text{Algo2}(\mathcal{G}_i, \mathbf{S}_{ref}, \mathbf{D}_{ref}, N_{max}, \mathcal{D}_{max})$;

Update \mathbf{S}_{opt} and $\tilde{\mathcal{G}}$;

$\mathbf{S}_{opt}[\mathcal{M}_i] \leftarrow \mathbf{S}_{opt}^i$;

$\tilde{\mathcal{G}} \leftarrow \tilde{\mathcal{G}} \cup \tilde{\mathcal{G}}_i$

end

subgraphs of the Light Field $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_\beta\}$. We then run the optimization in a parallel manner on all the local subgraphs. In this context, we propose a novel method to optimize the segmentation in a rate-distortion sense using graph reduction and partitioning on each of the super-rays. The final algorithm is detailed in Algorithm 1. At the output of this optimization, we have an approximate solution for the problem in Equation (3) with γ non overlapping optimized local graphs of the Light Field $\{\tilde{\mathcal{G}}_1, \tilde{\mathcal{G}}_2, \dots, \tilde{\mathcal{G}}_\gamma\}$.

V. CODING SCHEME

In this section, we detail the main building blocks of the scheme and describe the different coding components.

A. Coding of the disparity values and the segmentation map

The optimized segmentation map of the reference view is encoded using the arithmetic edge coder *EAC* proposed in [61]. The contours are first represented by differential chain codes and divided into segments. Then, to efficiently encode a sequence of symbols in a segment, *EAC* uses a linear regression model to estimate probabilities, which are subsequently used by the arithmetic coder. Disparity values (one disparity value per super-ray) are encoded using an arithmetic coder.

B. Coding of transformed coefficients

Even after optimizing the partitions, the energy compaction is not the same in all super-rays. This can be explained by the fact, that the segmentation may not well adhere to object boundaries due to disparity estimation errors, resulting in high frequencies. Also, the transform coefficients obtained for reduced signals do not have the same statistical properties as for the non-reduced signals. In order to optimize the coding performance, we optimize the quantization steps for each of the two groups of super-rays: the reduced and partitioned ones separately. First, for each group, we divide the set of super-rays into four classes, where each class is defined according to an energy compaction criterion.

Algorithm 2: Recursive Super-ray Rate Distortion optimized reduction and partitioning Algorithm

Data: The graph and signal for a super-ray, the disparity map and segmentation map of the first top-left view, the maximum tolerated number of vertices, the maximum tolerated distortion and the subspace dimension as in **definition 1**:

$$\{\mathcal{G}_i = \{\mathcal{V}^i, \mathbf{L}_i, \mathbf{x}_i\}\}, \mathbf{D}_{ref}, \mathbf{S}_{ref}, N_{max}, \mathcal{D}_{max}, k$$

Result: Optimized super-ray segmentation and local graphs: $\mathbf{S}_{opt}^i, \tilde{\mathcal{G}}_i$

```

if  $\mathcal{N}_i \geq N_{max}$  then
  Try Graph Coarsening on  $\mathcal{G}_i$ 
   $\{\mathcal{G}_c, \mathbf{L}_c, \mathbf{P}\} = \text{Coarsening}(\mathcal{G}_i, k)$ ;
  Find distortion  $\mathcal{D}(\mathcal{G}_i)$  as in (11);
  if  $\mathcal{D}(\mathcal{G}_i) \leq \mathcal{D}_{max}$  then
     $\mathbf{S}_{opt}^i \leftarrow \mathbf{S}_{ref}^i$ ;
     $\tilde{\mathcal{G}}_i \leftarrow \mathcal{G}_c$ ;
    Return  $\mathbf{S}_{opt}^i$  and  $\tilde{\mathcal{G}}_i$ ;
  end
end
 $\{\mathcal{G}_i^1, \mathcal{G}_i^2\} = \text{Spectralclustering}(\mathcal{G}_i)$ ;
Compute the features:  $\mathcal{C}_\zeta(A, B)$  (19),  $\mathcal{R}_C(\mathcal{G}_i)$ ,
 $\mathcal{R}_C(\mathcal{G}_i^{cut})$  (18) and  $q$  (17);
Whiten the Features with the standard scaler;
Predict the partitioning decision (yes/no) with SVM
with the previously computed whitened features ;
if ( $decision = false$ ) & ( $\mathcal{N}_i < N_{max}$ ) then
  Return  $\mathbf{S}_{ref}^i$  and  $\mathcal{G}_i$ 
else
   $\mathbf{S}_{opt}^i, \tilde{\mathcal{G}}_i^1 \leftarrow \text{Algo2}(\mathcal{G}_i^1, \mathbf{S}_{ref}, \mathbf{D}_{ref}, N_{max}, \mathcal{D}_{max})$ ;
   $\mathbf{S}_{opt}^i, \tilde{\mathcal{G}}_i^2 \leftarrow \text{Algo2}(\mathcal{G}_i^2, \mathbf{S}_{ref}, \mathbf{D}_{ref}, N_{max}, \mathcal{D}_{max})$ 
end
 $\mathbf{S}_{opt}^i \leftarrow \mathbf{S}_{opt}^{i1} \cup \mathbf{S}_{opt}^{i2}$ ;
 $\tilde{\mathcal{G}}_i \leftarrow (\tilde{\mathcal{G}}_i^1 \cup \tilde{\mathcal{G}}_i^2)$ ;
Return  $\mathbf{S}_{opt}^i$  and  $\tilde{\mathcal{G}}_i$ ;

```

Specifically, after applying the graph transforms on all super-rays, and using the natural scanning order (increasing order of eigenvalues), we assign a class number to each super-ray. For a class i , the high frequencies are defined as the last $\text{round}(N \times (4-i)/4)$ coefficients where N is the total number of coefficients. Each super-ray belongs to class i if it does not belong to class $i-1$ and the mean energy per high frequency coefficient is less than 1. Thus, we first find the super-rays in the first class then remove them from the search space before finding the other classes, and similarly for the following steps. We code a flag with an arithmetic coder to give the information of the class of super-rays to the decoder side. In class i , the last $\text{round}(N \times (4-i)/4)$ coefficients of each super-ray are discarded. The remaining low frequency coefficients are grouped into 32 uniform groups. We then find optimal quantization step sizes assuming a uniform quantization per group of coefficients as explained below. Using the optimized quantization steps, the coefficients are quantized then arithmetically encoded with Context Adaptive Binary Arithmetic Coder (CABAC).

1) *Finding the optimal quantization step sizes:* We choose a training dataset (in our experiments, we used *Fountain-Vincent2*). Then, for each kind of super-rays (original and reduced), a rate-distortion optimization is performed to retrieve the optimal quantization step sizes. More precisely, the rate cost for each group of coefficients quantized with a step size Q , is approximated by its entropy. The distortion is defined as the mean squared error between original and quantized coefficients. By varying the quantization step Q on each of the 32 groups of coefficients for all super-rays, we can compute overall rate-distortion costs for all possible cases. The optimal quantization steps for different output qualities are then retrieved after finding the convex envelope of all rate-distortion curves.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental setup

In order to assess the performance of the proposed RD optimization algorithm, we apply the proposed coding scheme on real light fields captured by plenoptic cameras from the EPFL light fields dataset [62] and Inria dataset [63] and on synthetic light fields from the HCI light field dataset [64]. To avoid the strong vignetting and distortion problems in the views at the periphery of the light field, we first consider the 9×9 central views. For generalization purposes, we also assess the performance of our coding scheme on 4 plenoptic images following the common test conditions defined in [10]: we coded original 13×13 views, and computed the YUV-PSNR as defined in [10]. The full set of real light fields considered for the experiments is: *Friends 1, Vespa, Stone Pillar Inside, Stone Pillars Outside, Bikes, Fountain Vincent 2* and *Danger de Mort* from the dataset in [62] and *Toys, Bench* and *Fruits* from the Inria Light Field dataset [63]. Synthetic light fields considered for the test are: *Greek, Sideboard* from the HCI Light Field dataset. Some of those are shown in Figure 4.

We use the method based on deep learning in [58] to estimate the disparity of the top-left views. Examples of the computed disparity maps are shown in the second row of Figure 4. The segmentation map of the reference view are computed with SLIC [57] and a sparse set of disparity values is then computed. The initial segmentation and disparity values both lead to the initial super-rays, i.e., the initial local graph supports as described in section IV-D. The number of initial super-rays are fixed to 500 in the experiments reported in this paper. Having a smaller number of super-rays implies a higher super-ray size hence a tremendous increase in the complexity due to partial eigen-decompositions needed for graph spectral clustering and coarsening. Experimentally, we have observed for our datasets that the use of 500 super-rays offered a good compromise between computational complexity and compression performance.

As for the maximum number of nodes allowed for the graph transforms, there is indeed a trade-off between the specified value of \mathcal{N}_{max} and the computational complexity of the proposed approach, but also between \mathcal{N}_{max} and the reconstruction quality of the lifted graph signals. Since the diagonalization of the laplacian has a complexity of $O(n^3)$,

it is important to keep it under a certain limit. Increasing the maximum number of nodes by $m < \mathcal{N}_{max}$ leads to a raise in the computational complexity of $O(m^3)$. On the other side, reducing \mathcal{N}_{max} leads to a high coarsening level; the more a graph is reduced, the more small-term high frequencies are discarded, and the less is the ability of the reduced graph to capture small-term spatial signal differences. This leads to a consequential drop of the reconstruction quality after lifting, and reduces the interest of performing a graph reduction. With all this in mind, in the experiments reported in this paper, \mathcal{N}_{max} has been fixed to 5000 nodes since this number provided us with interesting rate-distortion results and an acceptable computational complexity.



Fig. 4: Examples of real and synthetic light fields used in our experiments. 1st row: *Friends 1* and *Vespa* from EPFL’s dataset [62], *Toys* and *Fruits* from Inria’s dataset, *Sideboard* and *Greek* from HCI dataset [64]. 2nd row: corresponding disparity map estimated with the method in [58].

B. Analysis of the proposed RD optimized graph coarsening and partitioning algorithm

1) *Validation of the cut decision prediction model:* The cut decision prediction model has been trained using a large number of super-rays with different dimensions computed on a training set of light fields by applying the super-ray construction method explained in section IV-D. More precisely, we chose 3 light fields, one from each dataset for the training process : *Stone Pillar Inside* from EPFL light field dataset, *Fruits* from Inria light field dataset, and *Greek* from HCI dataset. Three initial segmentations are performed using different initial numbers of clusters (3000, 6000 and 9000 super-rays) leading to different super-ray sizes. Each segmentation is referred to as a layer. The model is then trained on all super-rays in all layers in the training set.

In order to validate the model, the learning is tuned by 10-fold cross-validation. We divide the super-rays used for training into 10 distinctive sets. Then in each learning operation, we take 1 set for test, and train on the other 9 sets. The percentage of partitioning decisions matching the model in Equation 3 are reported in Table I. The prediction score is computed as the mean of the prediction scores over the 10 learning operations and is around 94%. Also, while the complexity of computing the graph transforms and the diagonalization of the laplacian of a size n is $O(n^3)$, our model reduces the complexity to $O(n^2)$. Note that we also tried to modify the features by taking two different total variations, to capture and distinguish between the signal variations in both spatial and angular domain. With no clear improvement, we preferred to use the original set of features that are proven to be meaningful for the prediction.

2) *Optimized segmentation map as a function of the target distortion:* In this section, we show how the rate distortion optimized segmentation varies with respect to the target distortion of the reconstructed light field. We show in Figure 5

the optimized segmentation maps for the light field *Fruits*, obtained with our algorithm when we enable both the tools (i.e. graph coarsening, and graph spectral clustering).

If we look at the first row of the figure, it is clear that the graph reduction tool is very efficient for low bitrate and a high target distortion *i.e* when the target PSNR is low. More precisely, we tend to have bigger super-rays when the target PSNR is lower than 35 dB. The second row shows a binary map where black and white correspond to original or reduced super-rays respectively. Most of the super-rays are reduced to lower dimensions. The third row shows the PSNR of the reconstructed super-rays after the lifting procedure as expressed in Equation 11. The information detained by the coarsened signals on the reduced super-ray guarantees an acceptable reconstruction of the super-rays where the signals are mainly smooth. On the other hand, when we seek a higher quality, the information kept in the coarsened signals is not sufficient to reconstruct the high frequency details of the respective super-rays therefore the graph partitioning tool is used instead.

		EPFL JPEG Pleno				Inria		
train	test	Friends1	FV2	SPI	Vespa	Fruits	Bench	Toys
	Friends1		94.6	92.4	95.2	93.9	90.3	88.5
FV2		95.0	93.0	95.4	94.3	92.0	91.2	95.6
SPI		94.6	92.6	95.9	94.1	93.4	94.1	96.6
Vespa		94.4	92.0	95.0	93.8	90.0	87.1	93.8
Fruits		94.1	92.0	96.4	93.3	93.7	94.2	96.5
Bench		94.1	92.0	96.1	93.4	93.5	94.2	96.5
Toys		95.1	93.1	95.8	94.8	93.8	94.1	97.3

TABLE I: Percentage of right cut decisions (in %) after training and testing on different sets of super-rays corresponding to the different light fields from the Inria and EPFL dataset.

3) *Bitrate cost percentage analysis:* We evaluated for all the datasets used in our experiments, the percentage allocated for each of the different parts composing the final bit-streams. In a nutshell, we observe that at high bitrates, a dominant part of bits are allocated to encoding the graph transform coefficients with a percentage of $\approx 95\%$. The cost of both disparity values and segmentation maps is negligible in that case compared to the cost of the transform coefficients. On the other hand, for low bitrates, the segmentation map occupies most of the bitstream while only $\approx 30\%$ of the bits are allocated for transform coefficients. This can be explained by the fact that most of the super-rays are reduced, and the graph transform coefficients of the reduced super-rays are too coarsely quantized.

C. Comparative assessment of compression performance

In this section, we evaluate the compression performance of our method. We first compare the method used against previous graph based coding methods with a fixed super-ray size then analyse its performance against state of the art methods in light field coding.

1) *Comparison with previous graph transform based methods:* We first investigate the performance of our scheme with the two reduction tools: graph coarsening and graph partitioning and compare it to the performance of a traditional

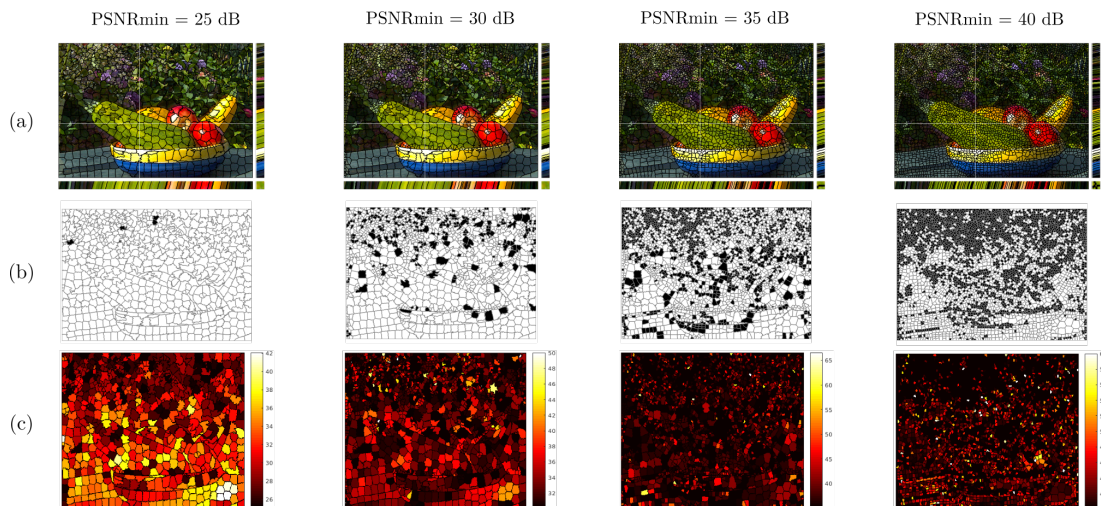


Fig. 5: Segmentation maps for different tolerated distortion levels (expressed in terms of PSNR (dB)) for the dataset *Fruits*. In the first row, the optimized segmentation maps obtained after applying the algorithm in for different values of D_{max} . The second row shows a binary mask where white regions correspond to coarsened super-rays. The last row shows the quality of the reconstructed super-rays after lifting.

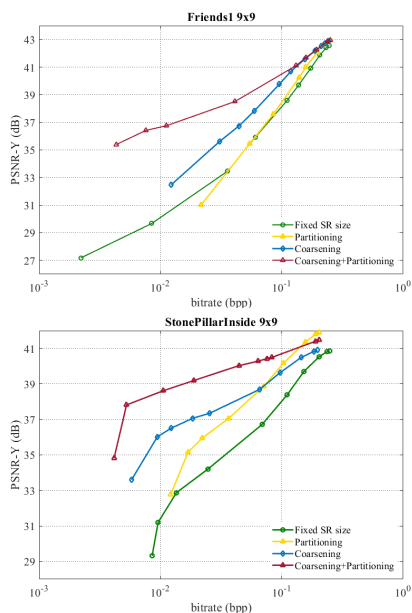


Fig. 6: PSNR-rate performance curves of the proposed compression scheme against graph based approaches with the JPEG Pleno dataset [62].

graph transform based scheme with a fixed size of super-rays proposed in [5]. For that, we study 3 different variants of the RD-optimized segmentation algorithm: (1) graph coarsening is enabled but no graph partitioning is applied unless the number of nodes is higher than the maximum allowed (5000 in our experiments). (2) graph coarsening is disabled and partitioning is necessarily applied if the number of nodes N higher than N_{max} , and predicted by the model $N < N_{max}$. (3) Both graph coarsening and graph partitioning are enabled. We compare the above variants to a simple scheme where we fix the super-ray size (denoted by *Fixed SR*).

Results are shown in Figure 6 for two light fields. Our method with both the functionalities enables, brings a lot of improvement to the rate distortion performance compared to the original graph transform based scheme. The main limita-

tion of the graph transform scheme in [5] is the complexity of the non separable graph transform which limits the resolution of the super-ray that can be supported. Our proposed method in this paper grants the possibility to have bigger super-rays for areas where the signal is smooth. Since the graph transform is applied on the reduced super-rays, the proposed scheme allows to exploit the long term signal dependencies in the spatial domain of the light field making the graph transform feasible.

The graph partitioning brings improvement for high bitrate, but not that much for low bitrate. Two reasons might explain this behavior. First, we are not capable of exploiting the long term signal dependencies since we are still limited in the maximum resolution. Moreover, we can partition the super-rays on regions where the signal is not smooth i.e texture. This leads to a better energy compaction on those regions, hence a lower bitrate needed to retrieve a very good reconstruction quality. On the other hand, the graph coarsening is very efficient at low bitrate, but not that much on high bitrate. The ability of the graph reduction tool to represent the original signals on lower dimensional spaces while preserving the main spectral characteristics of the underlying graphs, leads to substantial bitrate savings for a specified target distortion. Indeed, the number of coefficients to code is much less than original super-rays.

2) *Comparison with state of the art coders*: We assess the compression performance obtained with the our graph based coding method against different coding solutions proposed in the literature: direct encoding of the views as a pseudo-video sequence with HEVC based coding [9] [10], JPEG Pleno Verification models 1.1 and 2.1 (WaSP and MuLE) [12], the Homography Low Rank Approximation method (HLRA) in [13] and Fourier Disparity layers in [14], the light field translational coder [16] and the disparity prediction based codec of [15]. In the simulations, for HEVC-lozenge, the base QPs are set to 20, 26, 32, 38 and a GOP of 4 is used. The HEVC version used in the tests is HM-16.10. We used both versions of JPEG Pleno VM 1.1 and 2.1. In Figure 7, our

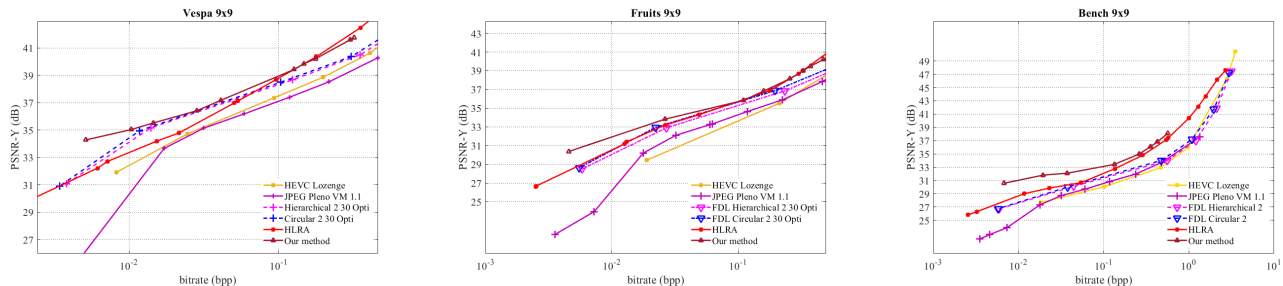


Fig. 7: PSNR-rate performance curves of the proposed light field coding method against FDL bases schemes [14], HLRA [13], HEVC-Lozenge [9] and JPEG Pleno VM 1.1 [11] with a set of light fields from Inria and EPFL datasets [63] [62].

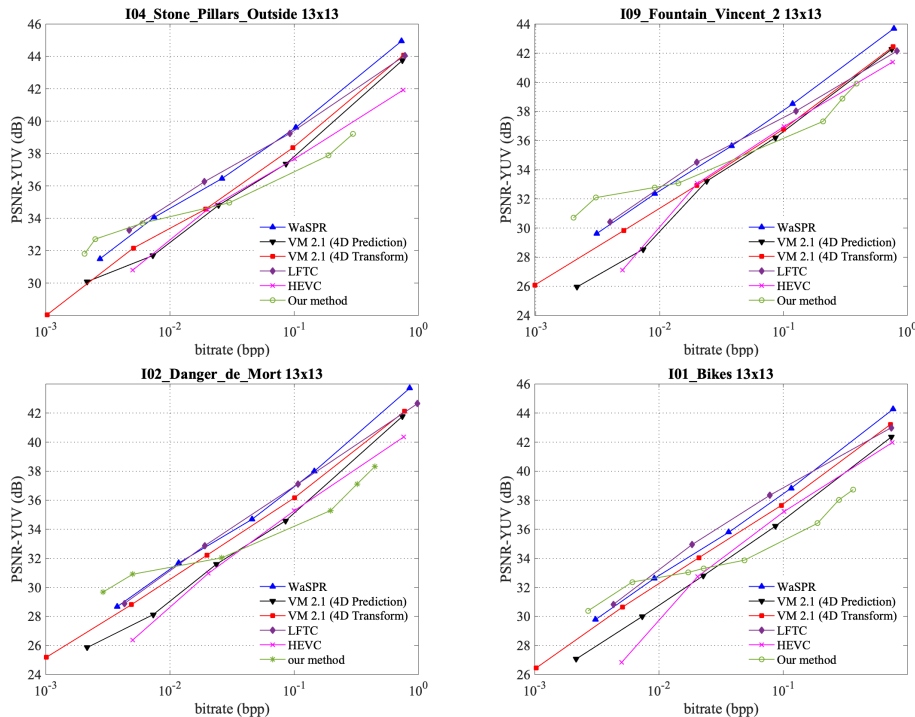


Fig. 8: Rate-distortion performance of the proposed method on plenoptic images. Comparisons are made against VM 2.1 (WaSP and MuLE) [12], WaSP-R [15], LFTC [16], and HEVC anchors [10], under the conditions specified in the CTC [10]

method with both functionalities enabled (graph coarsening and partitioning) is investigated against HEVC-Lozenge, JPEG Pleno verification model version 1.1, Fourier Disparity Layer (FDL) and HLRA for a set of dense light field datasets from the Inria Light field dataset and the EPFL light field dataset. Substantial gains in favor of our coding scheme are observed especially at low bitrates for most of the test light fields. The benefit of the graph coarsening reduction tool is clear. Graph coarsening not only preserves the small total variation on the coarsened graph, but also reduces the overall number of coefficients to code which brings a very good performance at low bitrates range.

D. Limitations of the proposed method

For synthetic light fields in HCI dataset such as *Greek* and *Sideboard* shown in Figure 9, our graph based coding scheme performs in general better than state of the art coding schemes such as the method in [15], the JPEG Pleno Verification model software version 2.1 and the coding method based on Fourier disparity layers [14] at low bitrates. It however comparatively

performs worse at high bitrates. This can be explained by the fact that while the baseline of light fields captured with plenoptic cameras is limited by the aperture size of the camera, synthetic light fields may have much larger disparities between views. In that case, super-ray projections, using one median disparity value per super-ray, are therefore less accurate when the super-ray is large, and are not able to fully compensate for inter-view disparities, especially for scenes with complex geometry such as *Greek*. Moreover, both synthetic light fields that we tested are free of imperfections such as noise. The real world Light fields imperfection degrade considerably the performance of classical coding schemes based on prediction while graph coarsening is not affected. The noise is mostly removed by the low rank model approximation that lies behind the graph coarsening technique.

Simulation results with real light fields under the common test conditions defined by JPEG-Pleno give performance curves (see Figure 8), with a similar behaviour, i.e., better performances at low bit rates but lower performances at higher bit rates, compared with reference methods. This can be

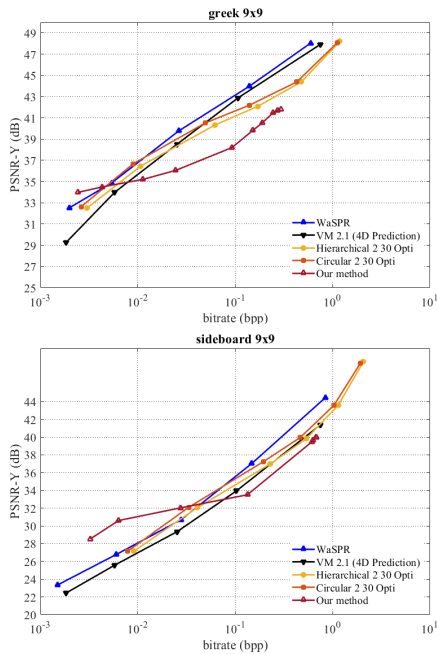


Fig. 9: PSNR-rate performance curves of the proposed graph based coding method on the HCI dataset [64] compared to recently published light field encoders in [15], [14] and [65] with the 4D prediction mode.

explained by the fact that the peripheral views of the 13×13 set of light field views captured by a Lytro camera suffer from vignetting effect, which degrades the performance of the depth estimation algorithm. Errors on depth maps induce a less accurate super-ray construction and a reduced de-correlation efficiency of the graph transform. As for the computational complexity, compared to HEVC anchor, the execution time of our coding method is 10 times higher for a single light field given a target quality. This is indeed dependent on the implementation and further optimization using fast graph Fourier transforms can be considered.

VII. CONCLUSION

In this paper, we have proposed a graph-transform based light field coding scheme using on a rate-distortion optimized graph coarsening and partitioning algorithm. The transform supports are based on super-rays constructed in a rate distortion optimized way. To overcome the complexity issue on big super-rays, a graph coarsening tool is applied and the graph transform is applied on reduced super-rays. The proposed algorithm also relies on another technique for dimensionality reduction i.e. graph clustering.

We have derived the distortion and rate models allowing us to optimize the graph supports in a rate-distortion sense. We hence propose a novel segmentation optimization method that aims at finding the optimized graph supports prior to graph transform and coding. This algorithm has been shown to bring substantial rate-distortion performance gains compared to a scheme based on the graph transform on fixed size super-rays and to state of the art light field coding schemes for light fields with small baselines.

ACKNOWLEDGMENT

The authors would like to thank Jinglei Shi for having provided the disparity maps used in our experiments. We would also like to thank Pekka Astola and Ioan Tabus for providing the results of the state of the art methods following the Common Test Conditions.

REFERENCES

- [1] R. Ng, "Light field photography," Ph.D. dissertation, Stanford University, 2006.
- [2] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 765–776, Jul. 2005.
- [3] T. Georgiev and A. Lumsdaine, "Focused plenoptic camera and rendering," *Journal of Electronic Imaging*, vol. 19, no. 2, Apr. 2010.
- [4] C. Conti, L. D. Soares, and P. Nunes, "Dense light field coding: A survey," *IEEE Access*, vol. 8, pp. 49 244–49 284, 2020.
- [5] M. Rizkallah, X. Su, T. Maugey, and C. Guillemot, "Geometry-aware graph transforms for light field compact representation," *IEEE Transactions on Image Processing*, vol. 29, pp. 602–616, 2020.
- [6] M. Rizkallah, T. Maugey, and C. Guillemot, "Prediction and sampling with local graph transforms for quasi-lossless light field compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 3282–3295, 2020.
- [7] M. Hog, N. Sabater, and C. Guillemot, "Super-rays for efficient light field processing," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1187–1199, Oct. 2017.
- [8] A. Loukas, "Graph reduction with spectral and cut guarantees," *Journal of Machine Learning Research*, vol. 20, no. 116, pp. 1–42, 2019.
- [9] M. Rizkallah, T. Maugey, C. Yaacoub, and C. Guillemot, "Impact of light field compression on focus stack and extended focus images," in *European Signal Processing Conf. (EUSIPCO)*, Aug. 2016, pp. 898–902.
- [10] I. J. S. JPEG, "Jpeg pleno light field coding common test conditions v3.3," Doc. N84049, 2019.
- [11] "ISO/IEC JTC 1/SC29/WG1 JPEG JPEG Pleno light field coding vm 1.1," Doc. N81052, 2018.
- [12] C. Perra, P. Astola, E. A. B. da Silva, H. Khanmohammad, C. Pagliari, P. Schelkens, and I. Tabus, "Performance analysis of JPEG Pleno light field coding," in *Applications of Digital Image Processing XLII*, A. G. Tescher and T. Ebrahimi, Eds., vol. 11137, International Society for Optics and Photonics. SPIE, 2019, pp. 402 – 413. [Online]. Available: <https://doi.org/10.1117/12.2528391>
- [13] X. Jiang, M. Le Pendu, R. Farrugia, and C. Guillemot, "Light field compression with homography-based low-rank approximation," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1132–1145, Oct. 2017.
- [14] E. Dib, M. Le Pendu, and C. Guillemot, "Light field compression using fourier disparity layers," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 3751–3755.
- [15] P. Astola and I. Tabus, "Coding of light fields using disparity-based sparse prediction," *IEEE Access*, vol. 7, pp. 176 820–176 837, 2019.
- [16] B. Hériard-Dubreuil, I. Viola, and T. Ebrahimi, "Light field compression using translation-assisted view estimation," in *2019 Picture Coding Symposium (PCS)*, 2019, pp. 1–5.
- [17] Y. Li, R. Olsson, and M. Sjöström, "Compression of unfocused plenoptic images using a displacement intra prediction," in *IEEE Int. Conf. Multimed. Expo Workshops (ICMEW)*. IEEE, 2016, pp. 1–4.
- [18] C. Conti, P. Nunes, and L. D. Soares, "HEVC-based light field image coding with bi-predicted self-similarity compensation," in *IEEE Int. Conf. Multimed. Expo Workshops (ICMEW)*, Jul. 2016.
- [19] R. Monteiro, L. Lucas, C. Conti, P. Nunes, N. Rodrigues, S. Faria, C. Pagliari, E. da Silva, and L. Soares, "Light field hevc-based image coding using locally linear embedding and self-similarity compensated prediction," in *IEEE Int. Conf. Multimed. Expo Workshops (ICMEW)*. IEEE, 2016, pp. 1–4.
- [20] C. Perra and P. Assuncao, "High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement," in *IEEE Int. Conf. Multimed. Expo Workshops (ICMEW)*, Jul. 2016.
- [21] R. J. S. Monteiro, P. J. L. Nunes, N. M. M. Rodrigues, and S. M. M. Faria, "Light field image coding using high-order intra block prediction," *J. on Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1120–1131, Oct. 2017.

- [22] D. Liu, P. An, R. Ma, W. Zhan, X. Huang, and A. A. Yahya, "Content-based light field image compression method with gaussian process regression," *IEEE Transactions on Multimedia*, 2019.
- [23] R. Zhong, S. Wang, B. Cornelis, Y. Zheng, J. Yuan, and A. Munteanu, "Efficient directional and 11-optimized intra-prediction for light field image compression," in *IEEE Int. Conf. Image Process. (ICIP)*, Sept. 2017, pp. 1172–1176.
- [24] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag, "Scalable coding of plenoptic images by using a sparse set and disparities," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 80–91, Jan. 2016.
- [25] Y.-H. Chao, G. Cheun, and A. Ortega, "Pre-demosaic light field image compression using graph lifting transform," in *IEEE Int. Conf. Image Process. (ICIP)*, Sept. 2017, pp. 3240–3244.
- [26] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression," in *IEEE Int. Conf. Multimed. Expo Workshops (ICMEW)*. IEEE, 2016, pp. 1–4.
- [27] C. Jia, Y. Yang, X. Zhang, X. Zhang, S. Wang, S. Wang, and S. Ma, "Optimized inter-view prediction based light field image compression with adaptive reconstruction," in *IEEE Int. Conf. on Image Processing, ICIP*, 2017.
- [28] W. Ahmad, R. Olsson, and M. Sjöström, "Interpreting plenoptic images as multiview sequences for improved compression," in *IEEE Int. Conf. Image Process. (ICIP)*, 2017.
- [29] W. Ahmad, M. Ghafoor, S. A. Tariq, A. Hassan, M. Sjöström, and R. Olsson, "Computationally efficient light field image compression using a multiview hevvc framework," *IEEE access*, vol. 7, pp. ss. 143 002–143 014, 2019.
- [30] L. Li, Z. Li, B. Li, D. Liu, and H. Li, "Pseudo-sequence-based 2-d hierarchical coding structure for light-field image compression," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1107–1119, Oct. 2017.
- [31] S. Zhao and Z. Chen, "Light field image coding via linear approximation prior," in *IEEE Int. Conf. on Image Processing, ICIP*, 2017.
- [32] S. Kundu, "Light field compression using homography and 2d warping," in *IEEE Int. Conf. Acoust., Speech Signal Process (ICASSP)*, Mar. 2012, pp. 1349–1352.
- [33] X. Jiang, M. L. Pendu, and C. Guillemot, "Light fields compression using depth image based view synthesis," in *Hot3D workshop held jointly with IEEE Int. Conf. Multimed. Expo, ICME*, Jul. 2017.
- [34] I. Tabus, P. Helin, and P. Astola, "Lossy compression of lenslet images from plenoptic cameras combining sparse predictive coding and jpeg 2000," in *IEEE Int. Conf. Image Process. (ICIP)*. IEEE, 2017, pp. 4567–4571.
- [35] T.-H. Tran, Y. Baroud, Z. Wang, S. Simon, and D. Taubman, "Light-field image compression based on variational disparity estimation and motion-compensated wavelet decomposition," in *IEEE International Conference on Image Processing (ICIP)*, Sept. 2017, pp. 3260–3264.
- [36] J. Chen, J. Hou, and L.-P. Chau, "Light field compression with disparity-guided sparse coding based on structural key views," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 314–324, Jan. 2018.
- [37] F. Hawary, C. Guillemot, D. Thoreau, and G. Boisson, "Scalable light field compression scheme using sparse reconstruction and restoration," in *IEEE International Conference on Image Processing (ICIP)*, Sept. 2017, pp. 3250–3254.
- [38] E. Dib, M. L. Pendu, and C. Guillemot, "Light field compression using fourier disparity layers," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 3751–3755.
- [39] W. Ahmad, S. Vagharshakyan, M. Sjöström, A. Gotchev, R. Bregovic, and R. Olsson, "Shearlet transform based prediction scheme for light field compression," in *International Data Compression Conference (DCC)*, Mar. 2018, p. 396.
- [40] R. Verhack, T. Sikora, G. Van Walleendael, and P. Lambert, "Steered mixture-of-experts for light field images and video: Representation and coding," *IEEE Transactions on Multimedia*, 2019.
- [41] M. B. de Carvalho, M. P. Pereira, G. Alves, E. A. B. da Silva, C. L. Pagliari, F. Pereira, and V. Testoni, "A 4d dct-based lenslet light field codec," in *IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 435–439.
- [42] M. Rizkallah, X. Su, T. Maugey, and C. Guillemot, "Graph-based transforms for predictive light field compression based on super-pixels," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP*, 2018.
- [43] J. Batson, D. A. Spielman, N. Srivastava, and S.-H. Teng, "Spectral sparsification of graphs: theory and algorithms," *Communications of the ACM*, vol. 56, no. 8, pp. 87–94, 2013.
- [44] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1913–1926, 2011.
- [45] D. R. Karger, "Random sampling in cut, flow, and network design problems," *Mathematics of Operations Research*, vol. 24, no. 2, pp. 383–413, 1999.
- [46] D. Peleg and A. A. Schäffer, "Graph spanners," *Journal of graph theory*, vol. 13, no. 1, pp. 99–116, 1989.
- [47] S. Wang and Z. Zhang, "Improving cur matrix decomposition and the nyström approximation via adaptive sampling," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2729–2769, 2013.
- [48] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors a multilevel approach," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 11, pp. 1944–1957, 2007.
- [49] D. Kushnir, M. Galun, and A. Brandt, "Fast multiscale clustering and manifold identification," *Pattern Recognition*, vol. 39, no. 10, pp. 1876–1891, 2006.
- [50] D. I. Shuman, M. J. Faraji, and P. Vanderghenst, "A multiscale pyramid transform for graph signals," *IEEE Transactions on Signal Processing*, vol. 64, no. 8, pp. 2119–2134, 2015.
- [51] M. Gavish, B. Nadler, and R. R. Coifman, "Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning," in *ICML*, 2010, pp. 367–374.
- [52] S. Lafon and A. B. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 9, pp. 1393–1403, 2006.
- [53] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3693–3702.
- [54] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.
- [55] J. Liang, S. Gurukur, and S. Parthasarathy, "Mile: A multi-level framework for scalable graph embedding," *arXiv preprint arXiv:1802.09612*, 2018.
- [56] 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.
- [57] R. Achanta, A. Shaji, K. K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [58] X. Jiang, J. Shi, and C. Guillemot, "A learning based depth estimation framework for 4d densely and sparsely sampled light fields," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2257–2261.
- [59] 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, 2014.
- [60] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, vol. EC-10, no. 2, pp. 260–268, June 1961.
- [61] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression," in *2012 19th IEEE International Conference on Image Processing*, Sept 2012, pp. 1541–1544.
- [62] "EPFL light field image dataset," <http://mmsgp.epfl.ch/EPFL-light-field-image-dataset>.
- [63] "INRIA Lytro image dataset," X. jiang, M. Le Pendu, R. Farugia, C. Guillemot, Light Field Compression with Homography-based Low Rank Approximation, special issue on Light Field Image Processing of the IEEE J. on Selected Topics in Signal Processing, IEEE J-STSP, vol. 11, No. 7, pp. 1132-1145, Oct. 2017; <http://clim.inria.fr/research/LowRank2/datasets/datasets.html>.
- [64] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4d light fields," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 19–34.
- [65] "Verification model software version 2.1 on JPEG Pleno light field coding," Doc. N83034, 2019.