On the Use of Graph Fourier Transform for Light-Field Compression

Vitor Rosa Meireles Elias and Wallace Alves Martins

***Abstract*—This work proposes the use and analyzes the viability of graph Fourier transform (GFT) for light-field compression. GFT is employed in place of discrete-cosine transform (DCT) in a simplified compression system based on high-efficiency video coding (HEVC). The effect on GFT efficiency of different implementations for prediction procedure is analyzed, as well as different methods for computing GFT given residual images. Results indicate that the prediction scheme is sensitive to the type of light field being compressed, and a preliminary method for selecting the best prediction scheme is explored. Moreover, considering multiple residual images when computing GFT, instead of only one central image, improves compression rate and makes compression more uniform across multiple views. GFT achieves reduction of up to 21.92% in number of transform coefficients when compared to DCT-based compression, while providing better or equal mean squared reconstruction error.**

***Index Terms*—Signal Processing on Graphs, Graph Fourier Transform, Light Field, Compression, High Efficiency Video Coding, Discrete-Cosine Transform, Prediction.**

I. Introduction

Light field imaging is a promising technology that opens a variety of new possibilities to entertainment industries, such as photography and cinema, by capturing 4D data from a scene [1]–[7]. Light field technology is based on the 5D plenoptic function $L(x, y, z, \theta, \phi)$, which describes the amount of light L, denominated radiance, along every position (x, y, z) in space and in any direction (θ, ϕ). Theoretically, if the plenoptic function for a region of interest is known, any image associated with that region can be recreated, from every perspective. This motivates the use of light field in entertainment industries, mainly photography and cinema [1]. Other application for light fields reside in medical imaging, such as microscopy [8] and brain imaging [9]. In practice, determining the plenoptic function is unfeasible, so light field cameras capture a 4D parametrization of the plenoptic function that consists of multiple photographs of a scene. This can be done moving a digital camera in a grid of various positions and taking photographs at each position, by using an array with multiple cameras, or by adding a microlens array in front of the camera sensor [3].

As light field data consists of multiple photographs, data size may increase drastically depending on the configuration of the light-field recording setup, making the manipulation of the resulting data a challenging task [10]–[15]. The "JPEG Pleno" initiative, conducted by the JPEG standardization committee, aims at providing solutions for framework and data manipulation considering several multiview image techniques, such as light field [6]. The delivery of a complete set of tools, including framework, coding, tests, and software, is set to 2018 [6], [16]. This requires in-depth research in order to develop and improve the various tools.

The use of graphs is specially relevant when dealing with an irregular domain or any domain that is not well represented by traditional time series [17]. In the current stage of the information era, the necessity of dealing with data from enormous networks, such as social networks, sensor networks, transport networks, among many others, increases daily. Given the non-ordered nature of these networks, using graphs as an underlying domain for the associated data becomes an interesting alternative to standard analyses [18]. Data from these networks become signals on graphs and, in order to manipulate these data, tools from classic digital signal processing (DSP) are adapted to signals on graphs, yielding the emerging field of digital signal processing on graphs (DSP_G) [17], [19]–[23].

Two important concepts that serve as basis for a signal processing framework for signals on graphs are the definitions of shift operator and frequency domain. As an emerging field, there are no consensus regarding the proper definitions of these concepts, giving rise to many researches addressing the approach that best fits each particular application [24]. One approach is based on the spectral graph theory [25], which uses the graph Laplacian $\mathcal{L}$ as shift operator and its eigenvectors as spectrum of the graph. This approach is usually restricted to undirected graphs, for which relations between two different elements are symmetrical, i.e., an edge from element i to element j has the same value as an edge from j to i. A second approach, valid for both directed and undirected graphs, uses the adjacency matrix of the graph $\mathbf{A}$ as shift operator [19], [26], [27]. In this case, the spectrum of the graph is defined as the eigenvalues of $\mathbf{A}$. This approach is the one adopted throughout this work, as it allows the use of more general classes of graphs.

This work is an extended version of the work presented in [28], where the application of *graph Fourier transform* (GFT) was proposed and studied as an alternative to the discrete-cosine transform (DCT) in the compression of light-field data. The objective of this work is to provide an improvement for light-field compression systems based on high-efficiency video coding (HEVC) [14], [29]. In HEVC, DCT and discrete-sine transform (DST) are used as block transforms, with the objective of mapping data into a frequency-related domain where quantization (and thus compression) is more efficient. This increase in efficiency is due to the energy compaction property

Mr. Vitor R. M. Elias and Prof. Wallace A. Martins are with the Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil (emails: vitor.elias@smt.ufrj.br, wallace.martins@smt.ufrj.br).

The authors are grateful for the financial support provided by CNPq, CAPES, and FAPERJ, Brazilian research councils.

Digital Object Identifier: 10.14209/jcis.2018.10

related to these trigonometric transforms when applied to images. It has been shown in [30], [31] that GFT is able to concentrate energy in fewer coefficients when compared to DCT, decreasing compression *distortion* when using the same number of coefficients. GFT usually depends on the original data and, thus, is not a fixed transform. Transmitting the transform basis from encoder to decoder is required, increasing transmission *rate*, and the impact of this task must be dealt with in order for GFT to be more efficient than DCT in the *rate-distortion* sense.

A. Scope and Contributions

This work begins by providing a review on both light field and DSP_G theories and an overview on how both these concepts are employed in this work. This includes: presentation of introductory concepts on both topics, motivation of the proposed approaches, notation, and database adopted throughout this work. The remaining part of this work focuses on analyzing the viability of using GFT in place of DCT under different analysis methods. We investigate forms of improving the performance of GFT by studying some of its parameters for which no consensus has been reached. The main contributions of this work are:

- Proposal and investigation of real applications for the developing field of DSP_G , given real and practical light field data.
- Performance comparison between GFT and traditional and broadly used DCT, analyzing viability of using GFT in the proposed application.
- Study of the effects of different settings for graph representation on GFT.

B. Outline

In Section II, background review on both light field and DSP_G is provided, including theory, applications, and motivation. Section III presents the proposed approach for using GFT light-field compression in an HEVC-based system. Section IV describes the entire methodology regarding database, definitions, and other concepts adopted throughout this work. Simulations and results are presented in Section V. Section VI presents a brief discussion of the results and future works. Section VII presents a conclusion for this work.

II. Light Field and DSP_G : A Review

This section reviews the main concepts related to both light fields and DSP_G . It begins by presenting light-field theory, focusing on recent implementations and how light-field data is generated. Then basic graph concepts and notations adopted in this work are presented, along with recent advances in the area.

A. Light field

Early notions of interpreting light as a field and conceiving a vector function to represent the amount of light present at (and passing through) points in space date back to the beginning of the 20th century. In 1936, Andrey Gershun introduced the term *light field* [32] and an early version of the function that would later be called the *plenoptic function*. In its standard interpretation, the 5D plenoptic function $L(x, y, z, \theta, \phi)$, which is a scalar field, describes light intensity that goes through a given point in space as a function of its position and the direction toward which the light ray is headed. Light intensity is denominated radiance and is given in $\text{W}/_{\text{sr}\cdot\text{m}^2}$ (watts per steradian per meter squared, i.e., power per solid angle per area). The function $L(\cdot)$ may be extended to higher dimensionality, for instance, by also considering time or wavelength. The idea of this function is to convey the complete information about a *scene*[1] associated with electromagnetic radiation. If $L(\cdot)$ is known, then every possible *view*[2] associated with a scene can be reconstructed by correctly arranging evaluations of the function for different points and directions in space, having several applications in imaging, photography, rendering, and other areas.

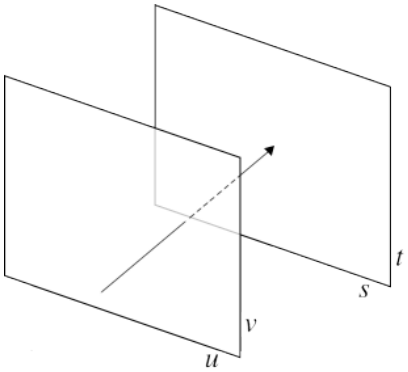

Figure 1: Planes *st* and *uv*, which serve as 4D parametrization for plenoptic function.

In practice, the plenoptic function is not available or obtainable in a feasible way. If free space is assumed, that is, the space associated with the region of interest is free of obstacles, the plenoptic function may be represented in lower dimensionality, considering a light ray sustains its radiance for different points along a given direction. The assumption of free space may be generalized to keeping the region of interest limited to the convex hull of any object. A straightforward parametrization of the plenoptic function in four dimensions is composed by two planes as shown in Figure 1. This representation of plenoptic function in four dimensions leads to current implementations of light-field-capturing devices. In devices used for capturing scenes and creating a light-field composition, the *uv* plane is taken as the *camera plane* and the *st* plane as the *focal plane*. That is, multiple light rays from the scene located at plane *st* travel along the space and hit a sensor region in plane *uv*, creating a view of the scene [1]. Common implementations are:

- array of cameras, with all cameras focused on the scene, creating a discrete version of plane *uv*;
- moving camera over a grid, capturing the scene at each point of the grid. It is actually similar to using an array of

[1]In this context, *scene* is a region of interest in space, usually containing an observable object.

[2]In the sense of a graphical projection of the scene onto a planar surface.

Figure 2: Example of light-field data, consisting of multiple views of a scene, captured by a moving camera.

cameras, but requiring a static scene. An example of light field captured by a moving camera is shown in Figure 2;
- microlens array inside a conventional digital single-lens reflex (DSLR) camera, where each microlens captures light from a different direction rendering different perspectives of the scene.

Light-field technology comes with several applications, most of them in the entertainment field. With light-field data captured by systems such as the aforementioned ones, features otherwise unfeasible become direct applications. For instance, synthetic aperture photography allows changing the focal point of a picture after it was taken. Light-field rendering allows the creation of novel views not previously captured. Light field displays may improve virtual-reality displays by using full light-field data rather than simple stereoscopic views. Light-field applications, however, are very data-intensive, since a single traditional image is now represented by a set of multiple images. Recent researches are dedicated to dealing with the high amount of data from light field [10], [11], [14].

B. Digital signal processing on graphs

Graphs are commonly defined as mathematical structures composed by two different sets: set $\mathcal{V} = \{v_0, v_1, \ldots, v_{N-1}\}$ composed of N vertices (also known as nodes) and set $\mathcal{E} = \{e_{00}, e_{01}, \ldots, e_{(N-1)(N-1)}\}$ of N^2 edges. Vertices are basic units and are interpreted as objects of a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, which can be used to model objects in diverse systems, e.g., points in $\mathbb{R}^2$, sensor locations in a network, social-network users, or chemical elements on a molecule, among many other applications. Edges e_{ij}, whose meaning and (possibly complex) value rely on the application of the graph, represent pairwise relations between vertices v_i and v_j, being equal to zero if there is no relation. The *neighborhood* of a vertex v_i is defined as the set of all vertices directly connected to v_i by a non-zero edge. These assumptions consider that there are no multiple edges between two vertices, but there are no restrictions to self-loops, which means a vertex can be directly related to itself. In this context, relation between elements does not have a fixed definition and depends on the application. If the relation between vertices v_i and v_j is the same as the relation between vertices v_j and v_i for every pair of vertices, i.e., $e_{ij} = e_{ji}, \ \forall i, j$, the graph is denominated *undirected graph*. Otherwise, if the direction of the edge is relevant and $e_{ij} \neq e_{ji}$ for some pair of vertices, the graph is denominated *directed graph*. An example for an undirected graph is shown in Figure 3, with $N = 4$ vertices. This graph is not fully connected, since many edges are equal to zero. Another form of representing the relations between vertices is the *adjacency matrix* $\mathbf{A} \in \mathbb{C}^{N \times N}$, whose element $[\mathbf{A}]_{ij} = e_{ij}$. The graph is undirected if, and only if, $\mathbf{A}$ is symmetric. Throughout the rest of this paper, graphs will be represented by pairs $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\}$.

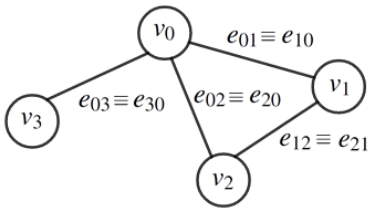

Figure 3: Example of undirected graph with $N = 4$.

Graphs are traditionally used as tools for data visualization and system modeling, whereas classical digital signal processing (DSP) is traditionally constructed around well-structured domains, such as time or space. Time domain is interesting for DSP as it holds properties that are particularly useful in the analysis of discrete-time signals. Consider a discrete-time finite-duration signal $s[n]$ as a function $s : \{0, 1, \ldots, N-1\} \rightarrow \mathbb{C}$ that maps instants $n \in \{0, 1, \ldots, N-1\}$ in time domain into the complex plane. Time domain is well-structured, as comparisons such as $n_1 < n_2$ and $n_1 = n_2$ are feasible for any two points n_1, n_2 within $\{0, 1, \ldots, N-1\}$, and it is a totally ordered domain. For many applications that emerge with recent advances and necessities in technology, treating signals associated with unstructured and more general domains is required. These applications are usually associated with networks, such as social, transport, sensor, and biological networks, for which representing the underlying domain with time or space would waste part of the information regarding connections among elements in the network. Graphs provide the suitable discrete domain for signals extracted from these types of network. Moreover, these applications are usually data-intensive, and graphs are a natural tool for representation of Big Data [20].

The concept of signals on graphs uses the set of N vertices $\mathcal{V}$ of a graph $\mathcal{G}$ as the domain of a dataset of N elements, equivalently to the use of N time instants $n \in \{n_0, n_1, \ldots, n_{N-1}\}$, as shown in Figure 4. The set of edges $\mathcal{E}$ of the graph is used to encode any relevant relationship between elements of the signal that could not be represented

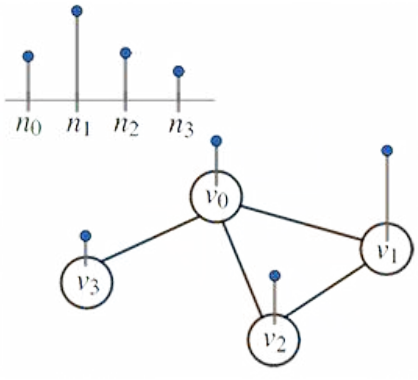

Figure 4: Relation between a signal represented in time domain and in graph domain.

in the time domain. A classic example is a sensor network that measures local temperature for N sensors distributed across several points of a country. Each location is represented by a vertex of the graph and the locally measured temperature is the signal on the vertex. Edges may be used to indicate distance between sensors, rendering an undirected graph. Another example is the measurement of user activity on a social network. Vertices would indicate each user account, for which an online-time is measured, and users are connected to each other via "following" tags, rendering a directed graph. For both cases, representing signals in time domain discards pieces of information that could be of paramount importance when processing these data.

The notation for signals on graph adopted throughout this paper is as follows: a graph signal given by $s : \mathcal{V} \to \mathbb{C}$ is referred to as a vector $\mathbf{s}$. The n-th entry of vector $\mathbf{s}$ is $s_n = s[v_n]$, with $v_n \in \mathcal{V}$.

Once graph domain and the definition of a signal over this domain are formally stated, one can build tools to process signals on graphs, which lead to two major approaches developed in the last years. The first approach is based on graph spectral theory [25] and on the graph Laplacian, being restricted to undirected graphs with non-negative edge values. This approach has received great attention and much effort was put into developing tools with these concepts [17]. Tools for DSP_G are mostly translations from already-consolidated classical DSP tools, which was mostly exploited by the second approach proposed by Sandryhaila and Moura [19], [20], [26], whose concepts are adopted and reviewed in the following definitions.

The first and most fundamental tool translated from classical DSP is the *unit-delay* or *unit-shift* operator, denoted as $\mathcal{T}^{-1}$, which consists of an essential block in filter design. In DSP, when a unit shift $\mathcal{T}^{-1}$ is applied to a length-N discrete-time signal $s[n]$, the signal is shifted in time resulting in a signal

$$\tilde{s}[n] = \mathcal{T}^{-1}\{s[n]\} = s[(n-1) \bmod N]. \tag{1}$$

The unit-shift operator $\mathcal{T}^{-1}$ is a linear transformation, implying that it can be associated with a matrix. Indeed, when

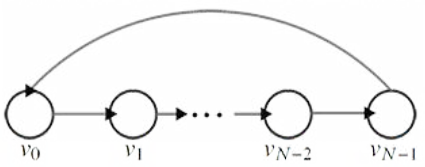

Figure 5: Cyclic graph: generalization of discrete-time domain.

using vector notation, one can rewrite Equation (1) as

$$\begin{bmatrix} \tilde{s}[0] \\ \tilde{s}[1] \\ \vdots \\ \tilde{s}[N-1] \end{bmatrix} = \underbrace{\begin{bmatrix} & & 1 \\ 1 & & \\ & \ddots & \\ & 1 & \end{bmatrix}}_{=\mathbf{C}} \cdot \begin{bmatrix} s[0] \\ s[1] \\ \vdots \\ s[N-1] \end{bmatrix}. \tag{2}$$

One can interpret the relation in Equation (2) within a graph framework. Indeed, consider the directed cyclic graph in Figure 5. Given all edges equal to 1, this graph can be interpreted as a graph generalization of the discrete-time domain, where each vertex v_n represents a time instant $n \in \{0, 1, \ldots, N-1\}$. The adjacency matrix of this graph is the *cyclic-shift matrix* $\mathbf{C}$ appearing in Equation (2).

One can bring these ideias to the graph domain by considering a graph $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\}$ as the underlying structure for a signal $\mathbf{s}$, and by identifying the *graph-shift* operator with the graph adjacency matrix $\mathbf{A}$. That is, a shifted signal $\tilde{\mathbf{s}}$ on a graph is given by

$$\tilde{\mathbf{s}} = \mathbf{A}\mathbf{s}. \tag{3}$$

This definition for graph shift means that shifting a signal on graph domain is equivalent to replacing each signal sample s_n by a linear combination, given by the n-th row of $\mathbf{A}$, of its neighborhood. This approach is not restricted to undirected graphs, allowing the use of directed graphs with complex-valued edges. A straightforward property of this definition is that it generalizes the unit-shift operator from classical DSP.

Given a formal definition for unit-shift in the graph domain, defining filters is the next natural step and it is performed by translating filtering concepts from classical DSP. In discrete-time domain, the output from a finite-duration impulse response (FIR) filter with length P is defined by the linear combination of its P most recent inputs, i.e.,

$$\begin{aligned} \tilde{s}[n] &= h_0 s[n] + h_1 s[n-1] + \cdots + h_{P-1} s[n-P+1], \\ &= \sum_{p=0}^{P-1} h_p \mathcal{T}^{-p}\{s[n]\}, \end{aligned} \tag{4}$$

where the time-invariant coefficients $h_0, h_1, \ldots, h_{P-1}$ define the impulse response of the filter and each term $s[n-p]$ results from shifting $s[n]$ with a shift operator $\mathcal{T}^{-p}$. For a signal with finite duration N, applying an FIR causal filter of length $P \leq N$, that is, $h_p = 0$ for $p < 0$ and $p \geq P$, induces the following circular convolution

$$\begin{bmatrix} \bar{s}[0] \\ \bar{s}[1] \\ \vdots \\ \bar{s}[N-1] \end{bmatrix} = \underbrace{\begin{bmatrix} h_0 & h_{N-1} & \cdots & h_1 \\ h_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & h_{N-1} \\ h_{N-1} & \cdots & h_1 & h_0 \end{bmatrix}}_{=\mathbf{H}(\mathbf{C})=\sum_{p=0}^{P-1} h_p \mathbf{C}^p} \begin{bmatrix} s[0] \\ s[1] \\ \vdots \\ s[N-1] \end{bmatrix}, \quad (5)$$

which shows that the filter is equivalent to a length-P polynomial over the cyclic-shift matrix $\mathbf{C}$. Analogously, the linear, *shift*-invariant *graph filter* is defined as a polynomial over the adjacency matrix $\mathbf{A}$, i.e.,

$$\mathbf{H}(\mathbf{A}) = \sum_{p=0}^{P-1} h_p \mathbf{A}^p. \quad (6)$$

Once signals, shift, and filters on graphs are defined, concepts of *spectral decomposition* and *Fourier transform* can be extended to graph domain. For a signal space $\mathcal{S}$, spectral decomposition of $\mathcal{S}$ is the identification of W filtering-invariant subspaces $\mathcal{S}_0, \ldots, \mathcal{S}_{W-1}$ of $\mathcal{S}$. Being invariant to filtering means that, for a signal $\mathbf{s}_w \in \mathcal{S}_w$, the output of filtering this signal is $\bar{\mathbf{s}}_w = \mathbf{H}(\mathbf{A})\mathbf{s}_w \in \mathcal{S}_w$. The spectral decomposition is uniquely determined for every signal $\mathbf{s} \in \mathcal{S}$ if, and only if:

- $\mathcal{S}_w \cap \mathcal{S}_r = \{0\},\ w \neq r$;
- $\dim(\mathcal{S}_0) + \cdots + \dim(\mathcal{S}_{W-1}) = \dim(\mathcal{S}) = N$;
- Each $\mathcal{S}_w$ is irreducible to smaller subspaces,

and, in this case,

$$\mathcal{S} = \mathcal{S}_0 \oplus \mathcal{S}_1 \oplus \cdots \oplus \mathcal{S}_{W-1}. \quad (7)$$

Given $\mathcal{S}$ as defined in Equation (7), satisfying the above conditions, any signal $\mathbf{s} \in \mathcal{S}$ is univocally represented as

$$\mathbf{s} = \mathbf{s}_0 + \ldots + \mathbf{s}_{W-1}. \quad (8)$$

The diagonalization of the adjacency matrix $\mathbf{A}$ leads to a spectral decomposition of the signal space $\mathcal{S}$ on the graph domain. Nonetheless, given the arbitrary nature of $\mathbf{A}$, as allowed in this DSP_G approach, it is not always diagonalizable. It is shown in [19] that the Jordan decomposition $\mathbf{A} = \mathbf{V}\mathbf{J}\mathbf{V}^{-1}$ is used to conduct spectral decomposition of $\mathcal{S}$ on graphs. $\mathbf{J}$ is the Jordan normal form and $\mathbf{V}$ is the matrix whose columns are the generalized eigenvectors of $\mathbf{A}$, which are the bases of the subspaces of $\mathcal{S}$. Hence, Equation (8) can be written as

$$\mathbf{s} = \mathbf{V}\hat{\mathbf{s}}, \quad (9)$$

where $\hat{\mathbf{s}}$ is the vector of coefficients that expand $\mathbf{s}$ into the subspaces of $\mathcal{S}$. The union of these subspaces is the *graph Fourier basis*. The *graph Fourier transform* (GFT), which provides the coefficients of the expansion of a signal over the *graph Fourier basis*, is defined as

$$\mathbf{F} = \mathbf{V}^{-1}, \quad (10)$$

such that $\hat{\mathbf{s}} = \mathbf{F}\mathbf{s}$. The *inverse graph Fourier transform* (IGFT) is given by

$$\mathbf{F}^{-1} = \mathbf{V}. \quad (11)$$

If the graph is undirected, $\mathbf{A}$ is a symmetric matrix and it is diagonalizable. The graph Fourier transform is then obtainable from the eigenvectors of $\mathbf{A}$. In this case, the eigenvectors are orthogonal and $\mathbf{V}^{-1} = \mathbf{V}^\text{T}$, which makes computation of the transform matrix $\mathbf{F}$ less intensive.

III. Proposed approach to light field compression

The application of HEVC-based methods for compression of light-field data has been intensively researched over the past years [14], [29], [33], [34]. HEVC presents a complex scheme composed by intra-frame and inter-frame prediction, motion estimation and compensation, transformation, quantization, coding, and other procedures, for which several configurations are available. These procedures are applied to *coding tree units*, which are blocks of up to 64×64 pixels into which video frames are divided. Notable procedures considered in this work are inter-frame prediction, transformation, and quantization, whose general concepts are explained below.

- **Inter-frame prediction**: When encoding a block of pixels of the current frame, the algorithm searches for a similar block, denominated *reference block*, from the previously encoded frame. Instead of encoding the raw values of pixels of the current block, the algorithm encodes only the difference between current and reference blocks. This difference is denominated *prediction residual*. The prediction procedure may be a complex process, using, for example, algorithms to estimate and compensate movement of blocks between different frames. Residual blocks should have less entropy than raw blocks, which makes compression in transformation, quantization, and coding stages more efficient. It must be noted that, in order to make inter-frame prediction possible, at least one frame that was previously encoded must have been encoded without inter-frame prediction. This frame is referred to as *intra frame*.
- **Transformation**: HEVC applies two-dimensional discrete-sine transform (DST) and, mostly, discrete-cosine transform (DCT) to residual blocks. Transformation is used to map data from residual blocks into a frequency-related domain, where energy concentration in lower frequencies can be exploited during compression. The output of transformation stage is a transform-coefficient block. Transform coefficients are real values that indicate how much each frequency component contributes to build the image in the original domain, in this case, the residual block.
- **Quantization**: Quantization maps coefficient values that may assume any value from a large, possibly continuous, set into a smaller set, allowing application of coding procedures otherwise unfeasible. The stronger the quantization, the fewer bits will be necessary to encode transform coefficients, thus reducing the associated *rate*.

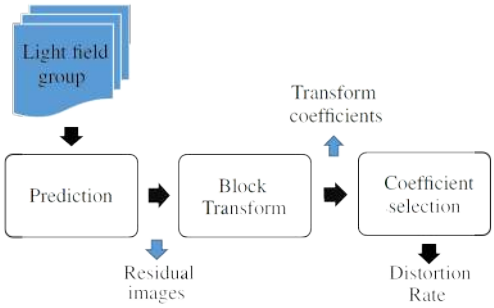

Figure 6: Block diagram describing the simplified compression process adopted throughout this work.

Quantization is a lossy process, i.e., information is permanently lost once coefficients are quantized. The loss of information is called *distortion*, for which several metrics are available. Compression processes must consider the trade-off between rate and distortion.

This work proposes and analyzes the viability of using GFT in place of DCT in HEVC-based light-field encoders, while exploiting the similarity among light-field images. The use of GFT within data compression context, and specifically image compression, is not new. The competence of GFT for concentrating information in few transform coefficients in a competitive manner when compared to other transforms is known and has been approached in other works [30], [31], [35]. Notwithstanding GFT inducing relatively high energy concentration, the transform and its inverse IGFT depend on the adjacency matrix $\mathbf{A}$, which has no fixed structure and depends on the application and on the data. The impact of storing or transmitting $\mathbf{A}$ or the transform matrix $\mathbf{F}$ must be considered during compression. The method proposed in this work aims at reducing the impact of the extra data related to graph structure by exploring the redundancy that exists among images near to each other in light fields.

IV. Methodology

In order to assess the performance of using GFT for light-field compression, a simplified compression process is defined, as presented in Figure 6, which is detailed in the next subsections. A database composed by 7 light fields is used. Three of them, namely *Humvee*, *Knights*, and *Tarot*, are obtained from the *Stanford Light Field Archive* [36] and some sample views are shown in Figure 7. These light fields are captured from real scenes using a moving camera on a rectangular grid with 16×16 positions, yielding 256 total images for each light field. The other four light fields are generated synthetically, obtained from the *HCI 4D Light Field Dataset* [37], [38]; sample views for *boxes*, *cotton*, *dino*, and *sideboard* are presented in Figure 8. For these light fields, views are captured over a grid of 9×9 positions, for a total of 81 images for each light field. Database information is summarized in Table I. Only the luminance component from these light fields is used throughout this work, despite the fact that RGB versions are depicted here.

Figure 7: Sample views from light fields captured from real scenes. *Humvee* (top), *Knights* (bottom left), and *Tarot* (bottom right).

Figure 8: Sample views from light fields captured from synthetic scenes. *Boxes* (top left), *Cotton* (top right), *Dino* (bottom left), and *Sideboard* (bottom right).

A. Prediction

The input of video codecs is a stream of frames ordered according to their time stamps. It is reasonable to assume that similarity between frames decays when two frames are selected further apart in time if compared to similarity between two consecutive frames. Thus, prediction for video streams can be implemented by selecting the frame that comes right before the current frame. It is worth noting that complex prediction schemes are not usually limited to only one frame.

Table I: Database information

Light field	Scene	View resolution [pixels]	Grid size
Humvee	Real	640×512	16×16
Knights	Real	1024×1024	16×16
Tarot	Real	1024×1024	16×16
Boxes	Synthetic	512×512	9×9
Cotton	Synthetic	512×512	9×9
Dino	Synthetic	512×512	9×9
Sideboard	Synthetic	512×512	9×9

For light fields, a prediction order is not straightforward. It is expected that views close to each other should be more similar. However, there is no consensus on how to determine the optimal selection of views or the boundaries for spatial neighborhood used for prediction in light fields. Considering the light field *humvee* as example, with a grid of 16×16 positions, three prediction schemes are considered in this work:

- **Rows**: Prediction is performed over each row with 1×16 images, independently from other rows. The first image from each line is assumed to be an *intra image*, i.e., no prediction is used when coding this image. For the remaining 15 images from each line, prediction residuals are calculated. A simple prediction scheme is adopted. The prediction image $\mathbf{I}_k^{\mathrm{p}}$ for the k-th image $\mathbf{I}_k$ in a light field row, where $k \in \{2, 3, \ldots, K\}$ and $K = 16$ in this example, is given by $\mathbf{I}_k^{\mathrm{p}} = \mathbf{I}_{k-1}$. That is, each image is assumed to be equal to the previous image in the line, given the high similarity among adjacent views in light fields. Finally, the residual image $\mathbf{R}_k$ is computed as the difference between current image and its prediction, i.e., $\mathbf{R}_k = \mathbf{I}_k - \mathbf{I}_k^{\mathrm{p}} = \mathbf{I}_k - \mathbf{I}_{k-1}$. A total of $K-1$ residual images are computed for each row.
- **Columns**: Prediction using columns is similar to prediction using rows. Columns with 16×1 images are treated independently, and the first image from each column is an *intra image*, whereas the remaining are *inter images*. Computation of residual images $\mathbf{R}_k$ is analogous to the one described for **rows**.
- **Blocks**: When using a block scheme to perform prediction, a 3×3 block of views is selected. The central view of the block is the *intra image* and the prediction image for every *inter image* is the central view. In other words, a block is composed by $K = 9$ views on a 3×3 grid. The central image $\mathbf{I}_c$ is *intra*-encoded, for some $c \in \{1, 2, \ldots, K\}$. Per group, $K - 1$ residual images are computed as $\mathbf{R}_k = \mathbf{I}_k - \mathbf{I}_c$, for $k \in \{1, 2, \ldots, K\}$ and $k \neq c$.

Given one of the prediction schemes described, the set of views selected for prediction procedure, i.e, views from a row, column, or block, will be referred to as *prediction group*.

B. Transformation

As stated, block transform is used to map data from residual image blocks into a frequency-related domain. This allows better compression of the data. HEVC uses DCT for residual blocks from size 4×4 up to 32×32, and DST for some cases of 4×4 blocks. In this work, GFT is used to transform blocks of size 32×32 and results are compared to those of DCT. If smaller blocks, such as 4×4 or 8×8, are used, it is expected that blocks at the same position for different residual images should have low correlation with each other, given the parallax between adjacent views. For large 32×32 blocks, the impact of parallax is reduced. High correlation among blocks in the same position from several views in a prediction group is beneficial for the proposed compression scheme, as will be further explained in this section.

Up to this point, images are treated as sets of pixels in 2D space. In order to make the use of GFT possible, the signal associated with a residual block must be represented as a signal on a graph, previously defined as a vector $\mathbf{s}$, such that the n-th entry s_n is a function of the vertex $v_n \in \mathcal{V}$. Let the signal associated with a pixel from an $M_1 \times M_2$ residual block be $r : \mathbb{I}^{M1 \times M2} \rightarrow \mathbb{R}$, where $\mathbb{I}^{M1 \times M2}$ represents the set of integer indexes for the positions of pixels on the $M_1 \times M_2$ block. That is, for each position on the $M_1 \times M_2$ block, a residual-related real value is assigned. The signal on graph is defined such that $s[M_1(m_2 - 1) + m_1] = r[(m_1, m2)]$, for $(m_1, m_2) \in \mathbb{I}^{M_1 \times M_2}$. That is, the graph signal $\mathbf{s}$ is defined as a column vector formed by stacking the columns of the residual block.

Let a residual block $\mathbf{B}_{k,t}$, $k \in \{1, \ldots, K\}$, $t \in \{1, \ldots, T\}$, be the $M_1 \times M_2$ block from the k-th residual image $\mathbf{R}_k$ (in a prediction group with $K-1$ residual images) that was divided into T blocks. The graph signal associated with this block is $\mathbf{s}_{k,t}$. The corresponding adjacency matrix is denoted by $\mathbf{A}_{k,t}$ and the GFT matrix by $\mathbf{F}_{k,t}$. Note that the transform matrix, and consequently its inverse, depend on the signal, unlike the DCT, which is the same for every $M_1 \times M_2$ block. The first consideration adopted in this work in order to reduce the impact of transmitting the transform matrix is to build a sparse adjacency matrix and transmit $\mathbf{A}_{k,t}$ instead of $\mathbf{F}_{k,t}$. The adjacency matrix $\mathbf{A}_{k,t}$ is built according to the nearest-neighbor (NN) image model [39], which is shown to offer an efficient image representation whilst providing a sparse and fixed graph structure. This model defines an image as a 2D nearest-neighbor graph. An NN graph is a graph for which a vertex v_i is connected to v_j if, and only if, the distance $d(v_i, v_j)$ is minimum among the distance between v_i and all other vertices. For a regular structure like an image, the minimum distance exists for more than one pixel, as depicted in Figure 9. Using NN image model implies that each vertex of the graph will have at most four non-zero edges, and pixels at the corner, border, or interior of the block have different number of edges. The model also assumes that an image is a 2D NN graph constructed as a Cartesian product of two 1D NN graphs. A 1D NN graph is a possibly-directed line graph similar to the one presented in Figure 5, apart from the loop edge. This generates a structure where multiple edges assume the same value, indicated by coefficients $a_0, \ldots, a_{M_1-2}$ and $b_0, \ldots, b_{M_2-2}$ in Figure 9. As a result, considering an $M_1 \times M_2$ residual block $\mathbf{B}_{k,t}$, the corresponding adjacency matrix $\mathbf{A}_{k,t} \in \mathbb{R}^{N \times N}$, $N = M_1 M_2$, has at most $(M_1-1)+(M_2-1)$ unique non-zero coefficients. For blocks of size 32×32, this means 62 unique non-zero coefficients out of 1024 entires of $\mathbf{A}_{k,t}$. The coefficients $a_0, \ldots, a_{M_1-2}$ and $b_0, \ldots, b_{M_2-2}$ are

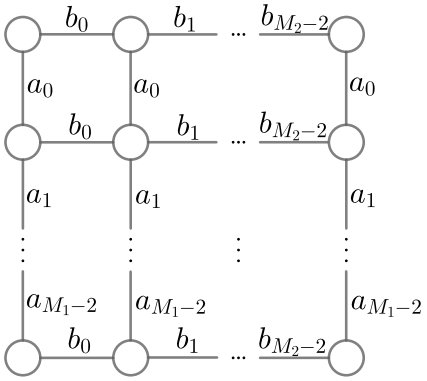

Figure 9: Relation edges according to the NN image model. Edges connect only pixels at minimum distance among all pixels.

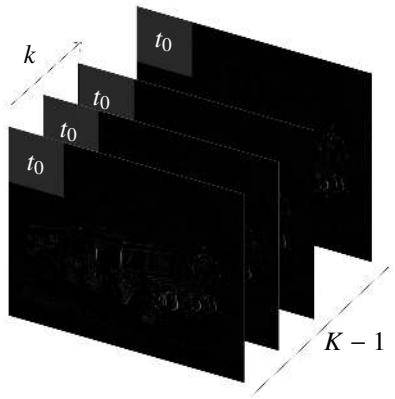

Figure 10: Representation of a block position t_0 for residual images from a prediction group.

defined so as to minimize the ℓ_2 distortion introduced by the shift operation, i.e., $\|\mathbf{A}_{k,t}\mathbf{s}_{k,t} - \mathbf{s}_{k,t}\|_2$. As described in [39], this minimization is solved as an overdetermined least-squares problem. This entire reasoning eventually implies that the adjacency matrix $\mathbf{A}_{k,t}$ is transmitted in place of the graph Fourier transform matrix $\mathbf{F}_{k,t}$. While this saves bandwidth, it adds complexity to the decoder, as the eigenvectors of $\mathbf{A}_{k,t}$ must be computed. Note that $\mathbf{A}_{k,t}$ is symmetric and, thus, diagonalizable. Finally, it is worth pointing out that other schemes rather than the NN image model could have been employed as well, which might induce different performances; however, the NN model proved viable, as corroborated by the results achieved in this work (see Section V).

The second consideration employed to reduce the impact of $\mathbf{A}_{k,t}$, besides forcing sparsity and fixed structure via NN image model, is to exploit the redundancy among the many views in the light field in order to avoid transmitting $\mathbf{A}_{k,t}$ with every single block. Considering that every view is equally divided into T blocks, only one $\mathbf{A}_{t_0}$ is transmitted for a given block position t_0 across the entire prediction group. Figure 10 shows an example of block position t_0 across views from a prediction group. This consideration assumes that blocks in the same position are highly correlated among several residual images. In this work, two similar methods for computing matrix $\mathbf{A}_{t_0}$ are considered. The first is using only adjacency matrices associated with one of the $K-1$ residual images $\mathbf{R}_k$. For *rows* or *columns* prediction schemes, using the central residual image (for example $\mathbf{R}_8$ when $K = 16$) is an intuitive choice, since other views are symmetrically similar to it. For *blocks* prediction scheme, there is no defined choice. The second method is to use multiple residual images $\mathbf{R}_k$ and compute the coefficients of $\mathbf{A}_{t_0}$ by minimizing $\sum_{k=k_1}^{k_2} \|\mathbf{A}_{k,t_0}\mathbf{s}_{k,t_0} - \mathbf{s}_{k,t_0}\|_2$, $1 \leq k_1 < k_2 \leq K-1$. That is, the distortion introduced by the shift operator is minimized jointly for multiple, possibly all, residual images in a prediction group. For both methods, using an adjacency matrix which is not specifically computed for a given block may degrade the efficiency of the GFT, but the impact of transmitting the matrix is slightly reduced.

Once the adjacency matrix is computed, the GFT matrix for each block position is given by $\mathbf{F}_t$, whose columns are the eigenvectors of $\mathbf{A}_t$ —the reader should keep in mind that the index k can now be dropped from $\mathbf{A}_{k,t}$ and $\mathbf{F}_{k,t}$ since it is assumed that adjacency and transform matrices do not depend on the residual image, given that only one matrix is considered for a given block position across the entire prediction group. The transform coefficients for each block from residual images in the prediction group are computed as $\hat{\mathbf{s}}_{k,t} = \mathbf{F}_t\mathbf{s}_{k,t}$, where $\mathbf{s}_{k,t}$ is the graph signal corresponding to each block.

C. Coefficient selection

A heuristic technique is adopted to assess the performance of GFT against DCT for light-field compression when employed in an HEVC-based compression system. The IGFT is given by the transpose of $\mathbf{F}_t$, since eigenvectors from $\mathbf{A}_t$ are orthogonal. If IGFT is applied to transform coefficients $\hat{\mathbf{s}}_{k,t}$, the signal $\mathbf{s}_{k,t}$ is perfectly recovered. In practical applications, compression occurs when transform coefficients are quantized, resulting also in loss of information. In this work, a simplified compression process is conducted by setting Q smallest transform coefficients to zero, resulting in compressed transform coefficients $\hat{\mathbf{s}}_{k,t}^{Q}$. When IGFT is applied to these coefficients, the signal $\mathbf{s}_{k,t}^{Q}$, which is recovered by inverse transform, is an approximation of the original signal $\mathbf{s}_{k,t}$. A compressed version $\mathbf{B}_{k,t}^{\text{GFT}}$ of the original block $\mathbf{B}_{k,t}$ can be constructed from the signal recovered. For the case of DCT, the 2D DCT is applied directly to block $\mathbf{B}_{k,t}$ and by setting the smallest coefficients from the transform block to zero, a compressed block $\mathbf{B}_{k,t}^{\text{DCT}}$ is recovered via inverse discrete-cosine transform (IDCT).

V. Simulations and results

Simulations were conducted in order to compare GFT against DCT when employed in the proposed compression system. The basic concept underlying all simulations presented in the next subsections is to set GFT coefficients to zero as much as possible while still recovering blocks with less distortion when compared to a specific DCT compression. The number of compressed DCT coefficients is fixed at $Q^{\text{DCT}} = 924$, i.e., only the 100 largest out of 1024 coefficients are kept and DCT is fixed at approximately 10:1 compression ratio. Distortion

Table II: Simulation results for transform-setup analysis

	Central residual		Part of group		Entire group	
Light field	**Reduction [%]**	**Standard deviation of Q**	**Reduction [%]**	**Standard deviation of Q**	**Reduction [%]**	**Standard deviation of Q**
Humvee	8.97	6.97	9.65	4.63	8.63	1.82
Knights	13.40	11.04	16.67	8.57	17.53	1.93
Tarot	-3.91	3.50	-0.65	1.96	-0.29	0.83
Boxes	0.22	4.56	6.57	2.45	7.76	1.42
Cotton	5.90	3.05	6.28	1.94	6.07	1.00
Dino	21.22	5.14	21.92	3.61	21.18	1.92
Sideboard	-3.89	2.67	-2.29	1.23	-2.04	0.86

D^{DCT} is evaluated for DCT. For each residual image, the simulation searches for the largest number of compressed GFT coefficients Q^{GFT} for which the corresponding distortion D^{GFT} is still smaller or equal to D^{DCT}. It is important to note that both Q^{DCT} and Q^{GFT} are set for an entire residual image and, thus, every block in each residual image will be represented by the same number of coefficients. The figure of merit used to characterize distortion is the *mean squared error* (MSE) between compressed and original residual images. For some simulations, the *structural similarity* (SSIM) index [40] is also considered as figure of merit for distortion. While MSE represents an indication of absolute error between images, the SSIM index provides information related to changes in structural information between images.

Different simulation setups are considered given the options described in Section IV. Three prediction methods were proposed, namely: *rows*, *columns*, and *blocks*. Moreover, two methods for building the adjacency matrix are considered. The first uses only one reference residual image, whereas the second uses multiple residual images when computing the coefficients of $\mathbf{A}_t$. The effects of these different setups are analyzed in this section. The database presented in Section IV and detailed in Table I is used.

A. Transform-setup analysis

As presented in Section IV-B, the coefficients of $\mathbf{A}_t$ may be computed either for a single reference residual image or jointly for multiple residual images. For this simulation, using the *rows* prediction scheme, three setups are considered for transform computation:

- Using only one central residual image as reference. The 8-th residual image $\mathbf{R}_8$ for real light fields, where $K = 16$ images per line, and the 5-th residual image $\mathbf{R}_5$ for synthetic light fields, with $K = 9$ images per line;
- Using part of the prediction group. Residual images from $\mathbf{R}_5$ to $\mathbf{R}_{10}$ for real light fields and from $\mathbf{R}_3$ to $\mathbf{R}_6$ for synthetic light fields;
- Using all residual images from the prediction group.

Table II shows the results obtained for simulations considering these three setups. Results show the *reduction* in number of coefficients used by GFT when compared to DCT for the entire light field, so that GFT is still able to yield better or equal distortion for every residual image. Reduction values for the total number of coefficients (#) for each light field are computed as

$$\text{Reduction} = \frac{\#\text{ DCT coefficients} - \#\text{ GFT coefficients}}{\#\text{ DCT coefficients}}. \quad (12)$$

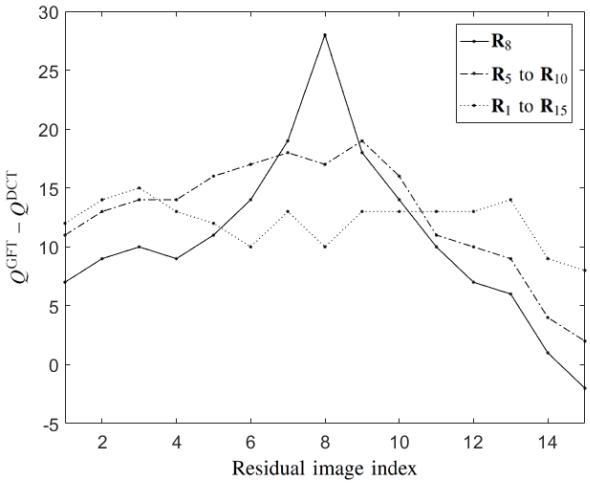

Figure 11: Number of compressed coefficients Q according to residual image position for the three proposed methods for computing $\mathbf{A}_t$.

It is worth highlighting that the number of coefficients associated with the adjacency matrices is included in # GFT coefficients and, thus, the impact of transmitting $\mathbf{A}_t$ is considered. GFT shows slight improvement over DCT for most cases, yielding up to 21.92% of reduction in number of coefficients. The analysis shows that using multiple residual images when building $\mathbf{A}_t$ improved the results for all cases when compared to results obtained using only one residual image as reference. This result can be observed in Table II by considering each light field independently, which is represented by each row. For each light field, an increasing trend in the reduction value can be noted when going from "Central residual" to "Entire group" sections, with few exceptions, indicating the overall improvement when using multiple residual images.

A relevant analysis given different transform setups is to observe the standard deviation of the number of coefficients used by the GFT across the residual images. The standard deviation of Q^{GFT} is estimated for each light field, using the number of compressed GFT coefficients Q^{GFT} from each residual image as sample for the standard deviation estimator. From Table II, it is notable that using the entire prediction group reduces the standard deviation of Q^{GFT}. When GFT is built using only the central residual image, its efficiency is high for the central residual image, but decays as residual images get further apart form the central reference. This is expected, since correlation is reduced and the impact of using a single

Table III: Results for simulation using SSIM index and $\mathbf{A}_t$ computed from all residual images

	Humvee	**Knights**	**Tarot**	**Boxes**	**Cotton**	**Dino**	**Sideboard**
Reduction [%]	4.22	10.56	1.15	2.38	-5.36	10.74	-1.14
Standard deviation of Q	1.50	1.83	0.99	1.06	1.15	1.73	1.33

Table IV: Simulation results for prediction-setup analysis

	Rows		**Columns**		**Blocks**	
Light field	**Reduction [%]**	**Standard deviation of Q**	**Reduction [%]**	**Standard deviation of Q**	**Reduction [%]**	**Standard deviation of Q**
Humvee	8.63	1.82	-2.80	4.06	3.15	5.52
Knights	17.53	1.93	11.50	2.24	16.09	1.86
Tarot	-0.29	0.83	-8.42	0.81	-5.55	3.07
Boxes	7.76	1.42	11.00	1.18	7.38	1.81
Cotton	6.07	1.00	6.10	0.90	6.18	3.00
Dino	21.18	1.92	15.22	1.24	15.53	5.00
Sideboard	-2.04	0.86	2.10	1.84	-0.25	2.72

transform matrix is increased, requiring more coefficients. Constructing the transform while considering multiple images reduces the efficiency decay across the prediction group. This effect is depicted in Figure 11, where the difference $\Delta Q = Q^{\text{GFT}} - Q^{\text{DCT}}$ in number of compressed coefficients for one row of the *humvee* light field is presented. In this case, the coefficients of $\mathbf{A}_t$ are not considered. The three proposed transform setups are considered. The peak for ΔQ at $\mathbf{R}_8$ is notable when this residual image is the only one used for transform computation. When using all residual images, this effect is no longer present, allowing for a more uniform compression across all images.

This simulation was replicated using SSIM as metric when searching for Q^{GFT}. Only the transform setup based on all residual images for the construction of $\mathbf{A}_t$ was used, considering it achieved the best results in the previous simulation. Results are presented in Table III. Values for reduction in number of coefficients are lower than the ones obtained when using MSE, but GFT is still competitive when compared to DCT. Moreover, small values for standard deviation are achieved, as expected.

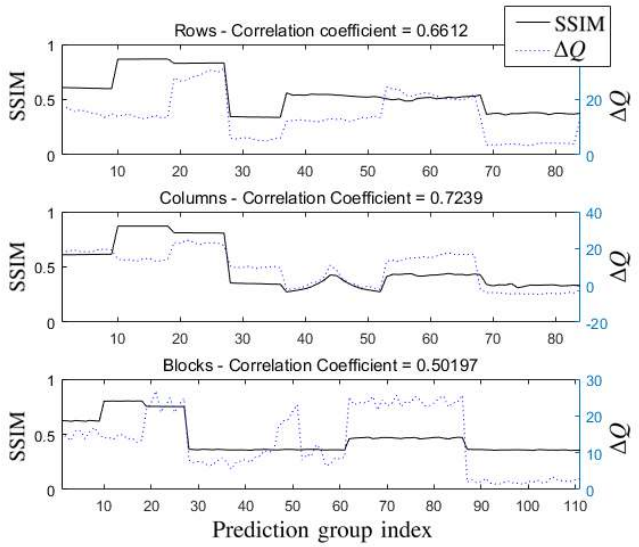

Figure 12: Analysis of the correlation between average similarity in a prediction group and the resulting efficiency of using that group for light-field compression.

B. Prediction-setup analysis

In this simulation, the three proposd prediction methods, namely *rows*, *columns*, and *blocks*, are tested. The transform matrix is built using all residual images from each group when computing the matrix coefficients. Results are shown in Table IV. For real light fields, using the *rows* prediction scheme yields the best results, followed by *blocks*, which increases the standard deviation of Q^{GFT} across residual images. For synthetic light fields, the discrepancy in results among different methods is reduced and the efficiency of *columns* prediction scheme slightly increases.

These results indicate that different prediction methods may be better suitable for some specific type of light field. Video encoders usually work with several possible configurations for each processing stage. This opens the possibility of searching for the best prediction method when compressing light field images in a more complex system. An analysis of how the similarity between images in a prediction group affects the compression efficiency in that group was conducted. For each light field, prediction groups based on the three proposed methods were constructed. For each group, the SSIM index is computed for every pairwise combination of residual images in that group and the average SSIM index value is computed. That is, for each prediction group, the corresponding average structural similarity is computed. Figure 12 shows the average SSIM results for every group for all light fields in the available database, along with the average reduction in number of coefficients used per group. This simulation is conducted for the three prediction methods. Results indicate high correlation between intra-group similarity and compression efficiency. In a more complex compression system, similarity could be used as a metric for the selection of the best prediction method.

C. Transform coding gain

The transform coding gain is a criterion commonly used in order to assess the effectiveness of a transform, by comparing the transform quantization against direct quantization in the original domain [41]. For orthogonal block transforms, which is the case for both GFT and DCT, and assuming high-rate, i.e.,

Table V: Transform coding gain for GFT and DCT computed for each light field, considering independent transforms $\mathbf{A}_t$

	Humvee	Knights	Tarot	Boxes	Cotton	Dino	Sideboard
G_{GFT}	9.24	28.13	19.20	3.40	2.05	5.04	5.18
G_{DCT}	5.12	27.53	21.47	2.91	1.90	4.07	4.24

Table VI: Transform coding gain for GFT and DCT computed for each light field, considering all $\mathbf{A}_t$ as a single transform

	Humvee	Knights	Tarot	Boxes	Cotton	Dino	Sideboard
G_{GFT}	5.45	8.04	7.58	1.91	1.66	2.59	2.25
G_{DCT}	5.24	8.06	9.79	1.98	1.80	2.67	2.40

every coefficient contributes equally to distortion after optimal bit allocation, the transform coding gain is given by

$$G_{\text{T}} = \frac{\frac{1}{N}\sum_i \sigma_i^2}{\sqrt[N]{\prod_i \sigma_i^2}}, \tag{13}$$

where σ_i^2 is the variance of the i-th transform coefficient across all blocks. The transform coding gain is the ratio between arithmetic and geometric means of coefficient variances. When estimating the transform coding gain from data from a light field, it must be considered that GFT is not a single transform, since it was defined as a data-dependent transform. In order to compute the transform coding gain G_{GFT} associated with GFT, blocks are treated according to their position t, for which a single $\mathbf{A}_t$ is defined. That is, given a prediction group, an independent $G_{\text{GFT},t}$ is computed for each block position, since the transform $\mathbf{F}_t$ is restricted to that block position in that prediction group. For each light field, the final gain G_{GFT} is given by the average gain across all block positions for all prediction groups. For DCT, the transform coding gain G_{DCT} is computed in the same way as G_{GFT} to make comparison possible. Results for the estimation of transform coding gain, using *rows* prediction method and using all residual images for computing $\mathbf{A}_t$, are presented in Table V. Transform coding gain shows better efficiency for GFT when compared to DCT for all light fields but one (*Tarot*). For some block positions from *Humvee* light field, $G_{\text{GFT},t}$ could not be computed due to zero variance encountered in some coefficients, resulting in zero geometric mean. For *Humvee*, the dynamic range of $G_{\text{GFT},t}$ was set to 10 by limiting the maximum value.

Table VI shows results for transform coding gain if the entire light-field data is considered at once, i.e., assuming that GFT is a unique data-independent transform. As expected, these results are worse for GFT.

VI. Discussion and Future Work

The simulations show mixed results in the comparison between GFT and DCT for light-field compression in an HEVC-based system. When comparing the reduction in number of coefficients, GFT is rather promising, being capable of reducing the number of transform coefficients by up to 21.18% in some cases, while keeping equal or better distortion when compared to DCT. Transform coding gain was used in order to provide an insight of how well transform coefficients may be coded, but results may be biased due to GFT not being a data-independent transform. The compression system employed is a simplified model based on HEVC, therefore several possibly relevant optimization procedures were not considered. Employing GFT in a more complex system is required so as to allow practical operation analysis. Moreover, several possible methods for implementing GFT were proposed, but some analyses were restricted to a single setup using *rows* prediction scheme and computing $\mathbf{A}_t$ from all residual images. A broader analysis could offer better understanding of GFT behavior.

VII. Conclusions

This work proposed and analyzed the use of GFT for light-field compression in an HEVC-based compression system. The comparison of the proposed method against the traditionally used DCT shows that GFT greatly reduces the number of coefficients required to represent light-field residual images while providing smaller distortion when compared to DCT. Different methods for constructing and employing GFT were tested for real and synthetic light fields. Using multiple images when computing the coefficients of the adjacency matrix provides a more uniform compression across residual images within a prediction group and improves reduction when compared to computing coefficients from a single reference residual image. An analysis of different prediction methods was conducted, as well as an analysis of how similarity between images within a prediction group affects the performance. When estimated from coefficients from the entire light field, transform coding gain favors DCT. Given the fact that GFT is data-dependent and, thus, not a fixed transform, a transform coding gain analysis regarding blocks for which GFT is unique was conducted. This analysis yields better transform coding gain for GFT. The compression system adopted may be improved in order to allow the comparison with practical coding systems.

References

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, New Orleans, LA, Aug 1996, pp. 31–42, doi: 10.1145/237170.237199.

[2] R. Ng, *Digital Light Field Photography*. PhD thesis, Stanford University, Stanford, CA, USA, 2006. AAI3219345.

[3] M. Levoy, "Light fields and computational imaging," *Computer*, vol. 39, pp. 46–55, Aug 2006, doi: 10.1109/MC.2006.270.

[4] D. Lanman and D. Luebke, "Near-eye light field displays," *ACM Transactions on Graphics*, vol. 32, pp. 220:1–220:10, Nov 2013, doi: 10.1145/2508363.2508366.

[5] L.-Y. Wei, C.-K. Liang, G. Myhre, C. Pitts, and K. Akeley, "Improving light field camera sample design with irregularity and aberration," *ACM Transactions on Graphics*, vol. 34, pp. 152:1–152:11, Jul 2015, doi: 10.1145/2766885.

[6] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens, "JPEG Pleno: toward an efficient representation of visual reality," *IEEE Multimedia*, vol. 23, pp. 14–20, Oct.-Dec 2016, doi: 10.1109/MMUL.2016.64.

[7] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field image processing: An overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, pp. 926–954, Oct 2017, doi: 10.1109/jstsp.2017.2747126.

[8] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, "Light field microscopy," *ACM Transactions on Graphics*, vol. 25, pp. 924–934, Jul 2006, doi: 10.1145/1141911.1141976.

[9] N. C. Pégard, H.-Y. Liu, N. Antipa, M. Gerlock, H. Adesnik, and L. Waller, "Compressive light-field microscopy for 3D neural activity recording," *Optica*, vol. 3, pp. 517–524, May 2016, doi: 10.1364/OPTICA.3.000517.

[10] M. Magnor and B. Girod, "Data compression for light-field rendering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 338–343, Apr 2000, doi: 10.1109/76.836278.

[11] T. Sakamoto, K. Kodama, and T. Hamamoto, "A study on efficient compression of multi-focus images for dense light-Field reconstruction," *2012 IEEE Visual Communications and Image Processing (VCIP)*, San Diego, CA, Nov 2012, doi: 10.1109/VCIP.2012.6410759.

[12] C. Perra, "On the coding of plenoptic raw images," in *22nd Telecommunications Forum (TELFOR)*, IEEE, Belgrade, Nov 2014, doi: 10.1109/telfor.2014.7034539.

[13] A. Vieira, H. Duarte, C. Perra, L. Tavora, and P. Assuncao, "Data formats for high efficiency coding of lytro-illum light fields," in *IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Orleans, Nov 2015, doi: 10.1109/ipta.2015.7367195.

[14] C. Perra and P. Assuncao, "High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement," in *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, Seattle, WA, Jul 2016, pp. 1–4, doi: 10.1109/ICMEW.2016.7574671.

[15] E. Cornwell, L. Li, Z. Li, and Y. Sun, "An efficient compression scheme for the multi-camera light field image," in *19th International Workshop on Multimedia Signal Processing (MMSP)*, Luton, Oct 2017, doi: 10.1109/mmsp.2017.8122243.

[16] "ISO/IEC JTC 1/SC 29 Programme of Work." https://www.itscj.ipsj.or.jp/sc29/29w42901.htm. Accessed: 2018-02-26.

[17] 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013, doi: 10.1109/MSP.2012.2235192.

[18] I. Jablonski, "Graph signal processing in applications to sensor networks, smart grids, and smart cities," *IEEE Sensors Journal*, vol. 17, pp. 7659–7666, Dec 2017, doi: 10.1109/jsen.2017.2733767.

[19] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, pp. 1644–1656, Apr 2013, doi: 10.1109/TSP.2013.2238935.

[20] A. Sandryhaila and J. M. Moura, "Big data analysis with signal processing on graphs: representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, pp. 80–90, Sept 2014, doi: 10.1109/MSP.2014.2329213.

[21] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 65, pp. 3462–3477, Jul 2017, doi: 10.1109/tsp.2017.2690388.

[22] O. Teke and P. P. Vaidyanathan, "Extending classical multirate signal processing theory to graphs—part I: Fundamentals," *IEEE Transactions on Signal Processing*, vol. 65, pp. 409–422, Jan 2017, doi: 10.1109/tsp.2016.2617833.

[23] O. Teke and P. P. Vaidyanathan, "Extending classical multirate signal processing theory to graphs—part II: M-channel filter banks," *IEEE Transactions on Signal Processing*, vol. 65, pp. 423–437, Jan 2017, doi: 10.1109/tsp.2016.2620111.

[24] A. Gavili and X.-P. Zhang, "On the shift operator, graph frequency, and optimal filtering in graph signal processing," *IEEE Transactions on Signal Processing*, vol. 65, pp. 6303–6318, Dec 2017, doi: 10.1109/tsp.2017.2752689.

[25] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997, doi: 10.1090/cbms/092.

[26] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: graph Fourier transform," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancoucer, BC, May 2013, vol. 62, pp. 6167–6170, doi: 10.1109/ICASSP.2013.6638850.

[27] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, pp. 3042–3054, Jun 2014, doi: 10.1109/MSP.2014.2329213.

[28] V. R. M. Elias and W. A. Martins, "Graph Fourier transform for light field compression," in *XXXV Simpósio Brasileiro de Telecomunições e Processamento de Sinais (SBrT)*, São Pedro, São Paulo, Sept 2017.

[29] 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 International Conference on Multimedia & Expo Workshops (ICMEW)*, Seatle, WA, Jul 2016, pp. 1–4, doi: 10.1109/ICMEW.2016.7574670.

[30] 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, pp. 419–433, Jan 2015, doi: 10.1109/tip.2014.2378055.

[31] G. Shen, W.-S. Kim, S. K. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *28th Picture Coding Symposium*, Nagoya, Dec 2010, doi: 10.1109/pcs.2010.5702565.

[32] A. Gershun, "The light field," *Journal of Mathematics and Physics*, vol. 18, no. 1-4, pp. 51–151, 1939, doi: 10.1002/sapm193918151.

[33] Y. Li, R. Olsson, and M. Sjostrom, "Compression of unfocused plenoptic images using a displacement intra prediction," in *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, Seatle, WA, Jul 2016, pp. 1–4, doi: 10.1109/ICMEW.2016.7574673.

[34] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, and F. Dufaux, "Integral images compression scheme based on view extraction," in *23rd European Signal Processing Conference (EUSIPCO)*, Nice, Aug 2015, doi: 10.1109/eusipco.2015.7362353.

[35] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, pp. 129–150, Mar 2011, doi: 10.1016/j.acha.2010.04.005.

[36] "The (new) Stanford light field archive." http://lightfield.stanford.edu/lfs.html. Accessed: 2018-02-20.

[37] "4D Light Field Benchmark." http://hci-lightfield.iwr.uni-heidelberg.de. Accessed: 2018-02-20.

[38] 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 (ACCV)*, Taipei, 2016, pp. 19–34, doi: 10.1007/978-3-319-54187-7_2.

[39] A. Sandryhaila and J. M. F. Moura, "Nearest-neighbor image model," in *19th IEEE International Conference on Image Processing*, Orlando, FL, Sept 2012, no. 3, pp. 2521–2524, doi: 10.1109/ICIP.2012.6467411.

[40] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, Apr 2004, doi: 10.1109/tip.2003.819861.

[41] T. Wiegand, "Source coding: Part I of fundamentals of source and video coding," *Foundations and Trends® in Signal Processing*, vol. 4, no. 1-2, pp. 1–222, 2010, doi: 10.1561/2000000010.

Vitor R. M. Elias was born in Brazil in 1990. He received the B.Sc in Electronics and Computer Engineering (cum laude) degree from the Federal University of Rio de Janeiro (UFRJ) in 2013. In 2015, he received the M.Sc. degree in Electrical Engineering from the COPPE/UFRJ, where he is pursuing the Ph.D. degree at the Program of Electrical Engineering since 2016. He is currently enrolled at the Signals, Multimedia, and Telecommunications Lab (SMT) and his experience and research interests include the areas of image and video compression, machine learning, and digital signal processing on graphs, with applications to image, video, and biomedical signals processing.

Wallace A. Martins was born in Brazil in 1983. He received the Electronics Engineer degree from the Federal University of Rio de Janeiro (UFRJ) in 2007, the M.Sc. and D.Sc. degrees in Electrical Engineering also from UFRJ in 2009 and 2011, respectively. He was a Research Visitor at University of Notre Dame (USA, 2008), at Université Lille 1 (France, 2016), and at Universidad de Alcalá (Spain, 2018). From 2010 to 2013 he was an Associate Professor of the Federal Center for Technological Education Celso Suckow da Fonseca (CEFET/RJ). Since 2013 he has been with the Department of Electronics and Computer Engineering (DEL/Poli) and Electrical Engineering Program (PEE/COPPE) at UFRJ, where he is presently an Associate Professor. His research interests are in the fields of digital signal processing, digital communications, visible light communications, massive MIMO systems, underwater communications, microphone/sensor array processing, adaptive signal processing, and digital signal processing on graphs. Dr. Martins received the Best Student Paper Award from EURASIP at EUSIPCO-2009, Glasgow, Scotland, and the 2011 Best Brazilian D.Sc. Dissertation Award from Capes.