Light Fields Imaging Coding

João Pedro de Carvalho Barreira Garrote

Instituto Superior Técnico

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Fernando Manuel Bernardo Pereira

Prof. João Miguel Duarte Ascenso

Prof. Catarina Isabel Carvalheiro Brites

Examination Committee:

Chairperson: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino

Supervisor: Prof. João Miguel Duarte Ascenso

Member of the Committee: Prof. Caroline Conti

November 2017

Acknowledgements

My first words go to my family, especially my parents and my sister. I cannot think of enough ways to express my gratitude, for without their love and endless support it would be impossible to come as far as I have.

I would also like to thank Professor Fernando Pereira, Professor João Ascenso and Professor Catarina Brites for being amazing teachers: their availability, guidance and patience were essential. Moreover, seeing their devotion and dedication has shown me I could not have had better advisors.

For my friends, who always helped me and took some of their time to cheer me up, a sincere thank you.

I would like to make a special reference to every researcher, colleague and teacher whose work and knowledge contributed towards the making of this Thesis.

To everyone who contributed in the development of this Thesis a huge and sincere thank you, your help was of great importance.

Resumo

Ao longo dos últimos anos, aplicações e sistemas multimédia mostraram um crescimento notável, muito devido a novas tecnologias de processamento de sinal e de comunicação para vários tipos de dados (áudio, vídeo, etc.). Em particular, as tecnologias visuais evoluíram de forma a serem mais eficientes levando a novas aplicações e serviços. Recentemente, novos tipos de sensores e ecrãs emergiram, necessitando de formatos de representação mais ricos e desta forma mais adequados para representar o mundo de uma forma mais imersiva. Os *light fields* emergiram como um formato de representação visual 3D bastante promissor, permitindo uma representação mais rica e mais realista de uma cena visual. Uma das formas populares de aquisição de *light fields* pode ser considerada como uma evolução da fotografia digital tradicional; a diferença reside numa matriz de micro-lentes colocadas entre a lente principal da câmara e o fotosensor permitindo capturar tanto a intensidade como a direção dos raios de luz. Este tipo de aquisição resulta em grandes quantidades de dados que necessitam de um espaço de armazenamento e largura de banda muito maiores em comparação com uma imagem 2D convencional de resolução espacial semelhante. Portanto, a compressão de dados *light field* é essencial para a transmissão e armazenamento deste tipo de formato. Este tipo de representação oferece novas possibilidades de interação com o conteúdo, tais como a refocagem (alteração do campo de profundidade) e visualização de diferentes perspetivas da cena visual, após a aquisição.

O objetivo desta Tese de Mestrado é desenvolver uma solução eficiente de codificação de imagens *light field*, explorando as características inerentes a esse tipo de representação de dados visuais. Com este objetivo, o inicio desta tese consiste numa revisão e analise das soluções de codificação de imagens *light field* mais relevantes encontradas na literatura. Muitas das soluções para a compressão de imagens *light field* são baseadas nas normas de codificação de vídeo disponíveis e não oferecem escalabilidade de vistas. Desta forma, foi desenvolvido um codificador baseado na transformada *wavelet* de forma a oferecer escalabilidade de vistas, qualidade e espacial, visando oferecer uma solução adaptável a diferentes tipos de dispositivos de reprodução e a diferentes taxas de transmissão.

O codificador proposto explora a correlação entre vistas (Inter) com ferramentas de estimação e compensação de disparidade bem como um codificador baseado na norma JPEG 2000. Os resultados obtidos permitem concluir que há uma melhoria no desempenho em relação ao codificador Intra (JPEG 2000 puro) quando se usa o codificador proposto, obtendo ganhos de desempenho constantes para um conjunto de imagens *light field* usadas para avaliação.

Palavras-Chave: light field; array de micro-lentes; estimação e compensação de disparidade; compressão de imagem; transformada wavelet; JPEG 2000.

Abstract

Over the last years, multimedia and system applications have shown a remarkable growth, mainly due to newer signal processing and communication technologies for several media types (audio, video, etc). In particular, visual related technologies have evolved to be more and more efficient leading to new multimedia applications and services. More recently, new types of sensors and displays have emerged requiring richer representations formats, more suitable to provide a more immersive experience of the world. Light fields have emerged as one of the most promising 3D representation formats, enabling a richer and more faithful representation of a visual scene. One of the most popular light field acquisition methods can be considered as an evolution of traditional digital photography; the key difference is the placement of an array of micro-lenses between the camera main lens and the photosensor which allows to capture both the radiance and the direction of the light rays. This type of acquisition results in large amounts of data which require a larger storage space and transmission bandwidth compared with a conventional 2D image of similar spatial resolution. Therefore, light field data compression has a critical role in the transmission and storage of this type of format. This type of representation format offers new possibilities to interact with the visual content, namely, refocusing (depth field change) and visualization of different perspectives of the visual scene, after acquisition is performed.

The objective of this Master Thesis is to develop an efficient lenslet light field image coding solution, exploiting the characteristics of this type of visual representation. Considering this objective, it starts by reviewing and analyzing the most relevant light field imaging coding solutions found in the literature. Many of the solutions available on the literature for light field compression are based on video coding standards and do not offer view scalability. Therefore, a wavelet-based encoder was designed and implemented to offer view, quality and spatial scalability that can meet the demand of different types of display and data transmission rates.

The proposed coding solution exploits the Inter view correlation with disparity estimation and compensation tools and reuses a JPEG 2000 image codec. The results obtained show that an improvement regarding the Intra codec (only JPEG 2000) when the proposed Inter scheme is used, achieving constant performance gains for the set of light field images selected for evaluation.

Keywords: light field; microlens array; disparity estimation and compensation; image compression; wavelet transform; JPEG 2000.

Table of Contents

List of Figures

List of Tables

List of Acronyms

1D	One Dimensional
1D-DWT	One Dimensional Discrete Wavelet Transform
2D	Two Dimensional
2D-DWT	Two Dimensional Discrete Wavelet Transform
3D	Three Dimensional
3D-DWT	Three Dimensional Discrete Wavelet Transform
4D	Four Dimensional
5D	Five Dimensional
7D	Seven Dimensional
AVC	Advanced Video Coding
AMVP	Advanced Motion Vector Estimation
BD	Bjøntegaard Delta
BD-Rate	Bjøntegaard Delta Rate
BD-PSNR	Bjøntegaard Delta PSNR
BI-SS	Bi-Predicted Self-Similarity
CCD	Charge-Coupled Device
CMOS	Complementary Metal-Oxide Semiconductor
CFA	Color Filter Array
CU	Coding Unit
CB	Code-Block
CDF	Cohen-Daubechies-Feauveau
CDQP	Compression-Driven Quantization Parameter
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
DoG	Difference of Gaussians
DCLFC	Disparity Compensated Light Field Coding
EPI	Epipolar Images
EBCOT	Embedded Block Coding with Optimized Truncation
GRBG	Green Red Blue Green
HEVC	High Efficiency Video Coding
ICME	International Conference on Multimedia & Expo
JPEG	Joint Photographic Experts Group
KLT	Karhunen-Loeve Transform
LCD	Liquid Crystal Display
LIDAR	Light Detection and Ranging

LLF	Lenslet Light Field
LLE	Local Linear Embedding
LSB	Least Significant Bitplanes
MSB	Most Significant Bitplanes
MVC	Multiview Video Coding
MPEG	Moving Picture Experts Group
PU	Prediction Unit
QDQP	Quality-Driven Quantization Parameter
PSNR	Peak-Signal-to-Noise-Ratio
RGB	Red Green Blue
RD	Rate-Distortion
RDO	Rate-Distortion-Optimization
RANSAC	Random Sample Consensus
SA	Sub-Aperture
SS	Self-Similarity
SA-DWT	Shape-Adaptive Discrete Wavelet Transform
SPIHT	Set Partitioning in Hierarchical Trees
SIFT	Scale Invariant Feature Transform
SD	Secure Digital
TSP	Travelling Salesman Problem
VAC	Vergence-Accommodation-Conflict
VP	Viewpoint

Chapter 1

1. Introduction

The purpose of this Chapter is to introduce the lenslet light field imaging coding topic as well as the Thesis objectives and structure. This chapter is structured in: i) context and motivation; ii) objectives; iii) Thesis structure.

1.1. Context and Motivation

Enormous progress has been achieved in the way images are captured, stored, distributed and displayed; moreover, recent technological breakthroughs in communication, image acquisition and coding technologies made possible several applications, from Instagram to YouTube, which are nowadays rather popular. Digital photography received a growing attention and a steady evolution over the last decade, with cameras capable of increased spatial resolution and wide dynamic range; offering nowadays an unprecedent level of image quality. More recently, new acquisition and processing technologies allow for the capture of new and richer visual data, resulting in several 3D visual representation formats, such as light fields, point-clouds or holography. In the last years, there were major developments in sensors/cameras and displays that allow for a better acquisition and replication of the visual world. These new devices, such as Lytro and Raytrix cameras, which are two of the most notable lenslet light field cameras, acquire more information about the light, namely information about its direction, and allow a more immersive and natural way to experience visual content.

In this context, the plenoptic function is considered the foundation for all the representation formats and therefore plays a key role. This 7D equation describes every possible view, from every position, at every moment in time and at any wavelength, thus can represent every photograph, every movie, everything everyone has ever seen. Light fields result from a dimensionality reduction of the 7D plenoptic function to 4D and describe the amount of light radiance in a scene as well as the directionality of each light ray. Light fields can be captured by either an array of cameras or by a special designed camera that uses an array of microlenses in front of the photosensor to sample the amount of light coming from different directions, also called lenslet light field camera. Figure 1 left) shows a microlens array, detailing each microlens and Figure 1 right) shows an example of a lenslet light field image. Nowadays, some lenslet light field cameras are available in the market, e.g. Lytro, and Raytrix, making light fields imaging a technology available to end users.

Figure 1: *left) Example of a microlens array [1]; right) Example of a lenslet light field image [2].*

With this richer representation, the visual data can be manipulated a posteriori by the users, allowing to control the focus, scene perspective or even stereoscopy image creation, which is not possible with conventional imaging formats. Figure 2 left) shows an example of post-capture refocus at different depths. These functionalities can be applied in different contexts such as photography, cinema, immersive communications and medical imaging; moreover, with the acquisition of different views is possible to build a virtual model to be used in both virtual reality and augmented reality applications, as shown in Figure 2 right).


Figure 2: *left) Example of post-capture refocusing [3]; right) Example of augmented reality viewing, blending the real world with a virtual model, namely a submarine, constructed with light fields [4].*

The growing availability of light field technology has brought many new challenges, such as efficient image storage, privacy, advanced manipulation, extraction of semantic information among others which drove the scientific community to improve existing solutions and creating new ones. In the context of this Thesis the attention will be especially devoted to efficient compression solutions which are essential in applications which require the transmission or storage of large amount of light field data, especially over constrained bandwidth networks. However, since light field imaging considers additional dimensions, not only the intensity but also the direction of the incoming light rays, the amount of data associated is higher and thus the need for a coding solution with high compression efficiency is even more critical.

Other important aspect of such systems, is that it should be easy to obtain a 2D visual representation from light field compressed data to support applications for which this limited form of representation is enough, and therefore assuring backward scalability. For example, it should be not necessary to decode the entire light field if the user just wants a coarse visualization of a light field from a fixed perspective and with single point of focus. Other applications may require just a subset of the views (e.g. horizontal views) of the light field and therefore it should be possible by partially decoding, to obtain that representation.

1.2. Objectives

In this context, the main objective of this work is to design, implement and assess an efficient lenslet light field imaging coding solution based on the best available technologies including also the existing coding standards. This objective is also being pursued by the JPEG standardization group in its JPEG PLENO call for proposals [5]. To reach this objective, the work developed under this Thesis' context was organized in the following tasks:

- Reviewing the most relevant lenslet light field imaging coding solution available in the literature; these coding solutions are based from well-known image standard codecs, such as JPEG or JPEG 2000, to video coding standards such as HEVC or other non-standard solutions considering the unique characteristics of the lenslet light field camera data.

- Implementation of a novel lenslet light field imaging coding solution based on the reviewed literature. In this case, a scalable solution based on the wavelet transform was designed to obtain multiple layers, notably offering view scalability, i.e. only a subset of the compressed views (a view corresponds to light coming from a single direction) is decoded to obtain a coarse light field representation. Also, the proposed solution provides other types of scalability such as resolution or quality scalability since it is still required by many applications. To maintain a high compression efficiency, the correlation between views of the light field is exploited with a disparity prediction step.
- Evaluation of the novel lenslet light field coding solution, namely to experimentally justify some of the design choices and compare the performance of the proposed solution under different configurations.

1.3. Thesis Structure

To achieve the proposed objectives in a logical and clear manner this Thesis is organized as follows: i) Chapter 1 provides some context about the light field topic and ends by defining the major objectives of this Thesis; ii) Chapter 2, begins by describing the main concepts behind the 3D visual representation formats, in particular the lenslet light field data format and ends with an extensive review of relevant light field imaging coding solutions which correspond to the state-of-the-art in this field; iii) Chapter 3, begins by introducing the foundations of the proposed solution (wavelet transform), presents the overall coding architecture and ends with a detailed description of the main techniques; iv) Chapter 4, will assess the adopted coding solution performance; v) Chapter 5, details the main conclusions and ends with some suggestions for future work.

Chapter 2

2. Lenslet Light Fields: Basics and Main Coding Solutions

The main purpose of this chapter is to review the basic concepts behind the main 3D representation models, moving after to reviewing the basics on lenslet acquisition and rendering techniques while considering the available lenslet cameras in the market. Finally, the chapter ends with the reviewing of the most relevant coding solutions available for light field images.

2.1. Basic Concepts

The concept of lenslet light fields, the main topic of this Thesis, is related to the general concept of light fields. In fact, a lenslet light field image is a 4D, or 5D if time is also considered, light field representation format. Figure 3 left) shows a 4D light field plus time representation which may be acquired with a single lenslet light field camera. To better understand 3D representation technologies in general, it is useful to start with the so-called plenoptic function [6]. The plenoptic function is a powerful representation model from where all 3D representation formats may be obtained. As many of the terms used in this Thesis are not usual terms, and some of them are used sometimes with slightly different meanings, it seemed useful to start this chapter by defining some basic concepts to ensure the conceptual coherence of the work and avoid misleading the reader.

The **Plenoptic Function** is a light model which measures the intensity of light seen from any viewpoint position in the 3D space, any angular viewing direction, for each wavelength and over time. This is a very powerful model which dimensions are detailed [6] now:

- **Wavelength** – by restricting the representation to non-coherent light, it is possible to use the Fourier representation to describe a propagating wave as an infinite sum of random-phase sinusoids within the visible spectrum, each with different energy [7]. Mathematically, this is represented as the power density for each wavelength ($\boldsymbol{\lambda}$). Because the human visual system only has three different types of cones, there is no need to have all the wavelength spectrum represented in detail but instead three bandwidth components are enough, this means red, green and blue [8].
- **Space** – the electromagnetic wave at a point ($\boldsymbol{x, y, z}$) in space can be decomposed by a sum of wavefronts coming from all different directions; each direction may be described by two angles, usually represented by ($\boldsymbol{\theta, \phi}$) [8].
- **Time** – if the energy of each electromagnetic wave varies in time, one can add a temporal dimension, finally leading to the function $\boldsymbol{P(x, y, z, \theta, \phi, t, \lambda)}$ as shown in Figure 3 right).

The entity measured by the plenoptic function is called **radiance,** which is physically the amount of light travelling along a ray. In this Thesis, this entity is denoted by *P*, but it may be denoted by *L* in some references [9] [10]; it is measured in *W* (watts) per steradian (sr) per meter squared (m^2) [11]. To acquire/sample such a complex light description function, one can imagine placing an idealized eye at every possible (*x*,*y*,*z*) location while recording the intensity of the light rays passing through the center of the pupil at every possible angle (θ, ϕ), for every wavelength (λ), at every time (t) [8]. Naturally, since

it is not possible to simultaneously represent a scene from every possible point of view, for every wavelength, at every moment in time, the trend is to simplify the plenoptic function representation by lowering its dimensionality. In this way, the huge amount of data contained in the plenoptic function can be reduced. In this context, several 3D representation formats have emerged, notably Light Fields, Point Clouds, Meshes and Light Fields plus Depth.

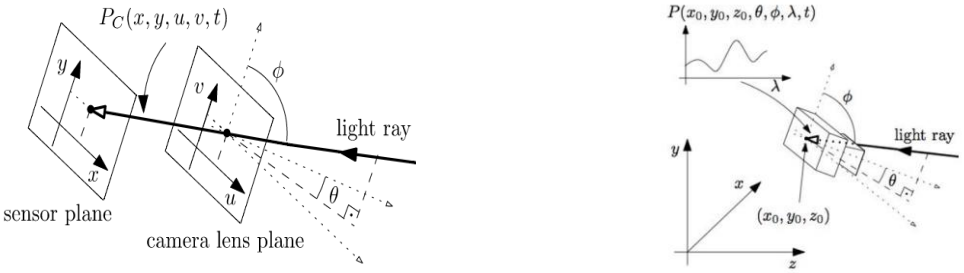


Figure 3: *left) 4D light field plus time with a single lenslet light field camera [6]; right) Visualization of the plenoptic function [6].*

A **Light Field** is a 3D representation format where the scene's light radiance is measured by projecting it into a high number and high density set of 2D sensors, either scattered in space, e.g. arrays of cameras, or located in a single camera, e.g. lenslet (Lytro-type) cameras [12], generating so-called views; in practice, the views show the same scene from different perspectives. This model involves both horizontal and vertical parallax support and eventually a high angular resolution between the views. The captured light density should be so high that the usual vergence-accommodation conflict (VAC) typical of low view density solutions, e.g. stereo pairs, should be negligible, allowing to give the user the impression of experiencing a real visual scene. The most important light field parameters, notably for a light-field camera array, are the number of cameras, the resolution of each camera and their arrangement; for a lenslet light field camera these parameters are the sensor resolution and the number and characteristics of the microlenses in the array. Although depth information is implicitly present in the acquired data, it is not directly represented as only radiance/texture/color is acquired; however, depth may be extracted from the texture using appropriate complex algorithms; naturally, the quality of the depth extraction depends on the number of views available and its spatial resolution. Such a powerful representation format enables rendering 3D photorealistic scenes, novel viewpoint rendering, synthetic aperture photography and post-capture image refocusing [9]. All types of light field formats defined below may or may not consider the temporal dimension.

A **5D Light Field** is a light field captured by an **arbitrary arrangement of cameras**; as the cameras are scattered in a 3D space without any regular arrangement, three coordinates are necessary to define the camera position (e.g. ***x***,***y***,***z***), and two other coordinates are necessary to define the acquisition position within each camera, thus resulting into a 5D function, **P(*x*,*y*,*z*,*u*,*v*)**. If lenslet light fields cameras are used instead of conventional cameras, the 5D function is still enough to describe the radiance in the scene as lenslet light fields camera can be understood as many small resolution conventional cameras.

A **4D Light Field is** a light field that is captured by a 2D array of cameras or a lenslet light field camera. The **2D array of cameras** are arranged in a regular way, e.g. a linear or circular arc arrangement; this also includes the case of one or many lenslet light field cameras. In a 2D array of cameras, the camera position (*x*,*y*,*z*) is replaced by the index ***K***,***l*** of the camera position in the 2D array, and the ray direction may be replaced by the spatial position **(*u*,*v*)**. In this case, each camera records a 2D slice, (*u*,*v*) of a 4D Light Field, **P(*k*,*l*,*u*,*v*)**; by concatenating these slices, the Light Field is obtained

[9]. Figure 4 shows examples of acquired data with a 2D array of conventional cameras and a lenslet light field camera. In a **lenslet light field camera**, a microlens array is placed between the photosensor and the main lens [13]; in this case, the light rays coming from a given object with various incident angles are first refracted and then captured by a 2D image sensor, thus allowing to capture of so-called *angular data*; this means the captured radiance is organized in the rectangular 2D sensor with a spatial position **(*x*,*y*)** and an angular direction (θ, ϕ); the angular direction may be replaced by the coordinates **(*u*,*v*),** finally yielding a 4D function **P(*x*,*y*,*u*,*v*).** The light intensity distribution corresponding to a lenslet image, associated to a 2D array of micro-images, is depicted in Figure 4 right) [14] where each micro-image corresponds to a single microlens.

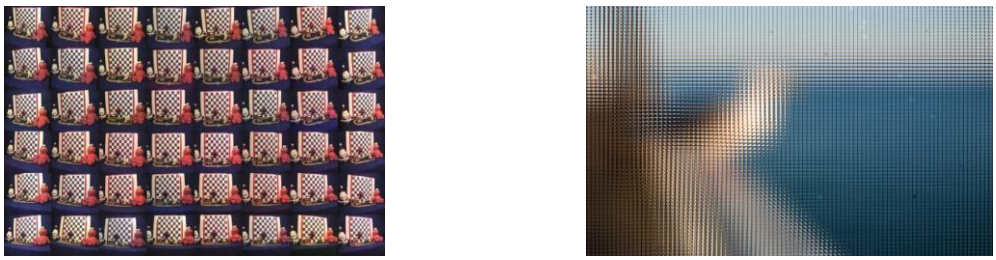


Figure 4*: left) Example data from a 2D array of cameras [8]; right) Example data from a lenslet light field camera [15].*

A **3D Light Field** is a light field captured using a **1D array of cameras** in a linear or a circular arc arrangement, resulting into a 1D array of images. Usually, the camera position (*x*,*y*,*z*) is replaced by the index ***k*** of the camera in the array and two other coordinates are necessary to define the ray position within each camera **(*u*,*v*)**. This enables only horizontal or vertical parallax; if enough cameras are available, new views can be synthetized with good quality. In summary, light fields are largely characterized by angular information as the light associated to each scene position is acquired from different angles.

Point Cloud is a representation format consisting in a set of 3D points defined by a corresponding 3D position in space, (*x*,*y*,*z*), each characterized by the radiance for each direction (direction dependent texture), over time. However, the radiance associated to the point clouds is often simplified with a single color for each point position and a normal direction to characterize the surface reflection. Simplified point clouds have been used for many years in Computer Graphics. More recently, point clouds are directly acquired from the real world with 3D sensors such as LIDAR devices, Time of Flight range cameras, and structured light projectors working in the visible or infrared wavelengths [6]. These sensors provide depth information to be used in combination with acquired color data to define color enable point clouds. If time is considered, the point cloud is named dynamic (static, otherwise). Important point cloud parameters are the number of points and its density as they have a major impact on the final rendered image quality. Most often, a point cloud is used to represent a 3D surface, e.g. to model buildings, terrains and objects in videogames, as can be seen in Figure 5 left).

Mesh is a representation format where the 3D world objects are represented by the vertices and connections of a 3D mesh, with each mesh's face associated to some texture. Meshes are often obtained from point clouds by adding connectivity to the points of the point cloud; the connectivity may be added using either polygon or triangle mesh facets, as shown in the example in Figure 5 mid). Color may be added to the mesh, which defined the scene geometry, e.g. by interpolating the colors available from the vertices; this is particularly easy, if a single color is available for each point cloud position.

Meshes are interaction friendly, easy to edit and rather good for synthetic data, meaning they are particularly appropriate for applications like gaming, virtual reality and augmented reality [6].

Finally, **Light Field plus Depth** is a representation format where a light field is complemented with depth information for each pixel, representing the distance from the captured object to the camera. The availability of depth information allows synthesizing texture for any number of view positions between the acquired views, thus reducing the requirements in terms of the number of cameras and coding rate. Naturally, there is a minimum number of texture views that need to be available to guarantee a minimum quality for the synthesized views as this quality typically decreases when the distance between available views increases. The spatial arrangements for this 3D representation format are the same as for the light fields format with the difference that now also a depth sensor (and not only a color camera) is associated to each array position. Figure 5 right) shows an example of a texture image and a depth map pair.

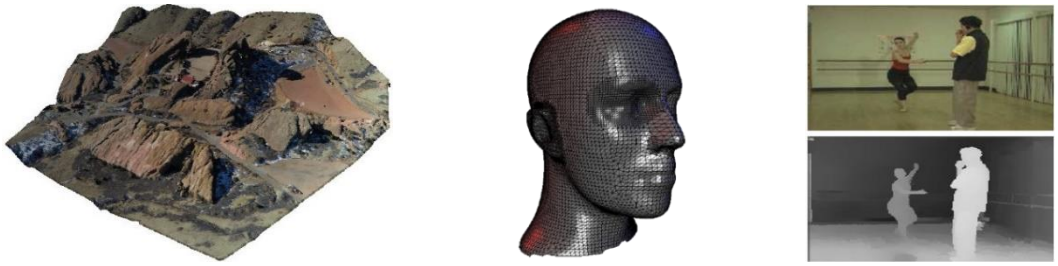


Figure 5: *left) Point cloud representation [16]; mid) Mesh representation [17]; right) Light field plus depth representation for a single view [18].*

Despite being alternative solutions for similar problems, these formats do not compete too much. As they sample the plenoptic function in different ways, they have different characteristics and thus advantages and disadvantages, which makes them particularly appropriate for different applications and their required functionalities [6]. To resume and organize most of the formats mentioned above, Figure 6 shows a simple diagram where the relation between the various formats representing the data associated with the plenoptic function is highlighted.

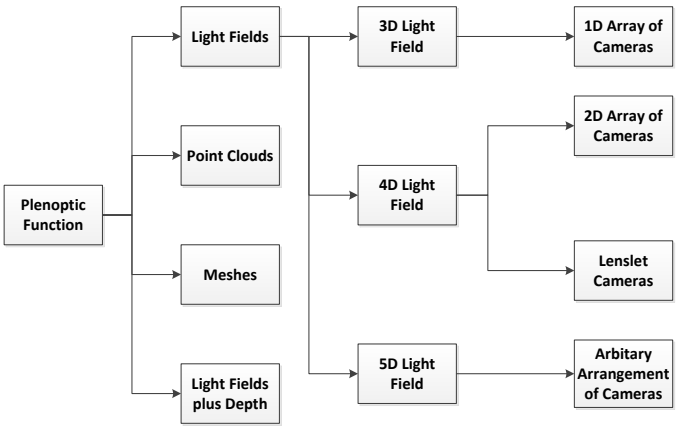


Figure 6: *Relationship between the plenoptic function and the main 3D representation formats.*

2.2. Lenslet Light Fields Cameras: A Review

This section will detail the basic principles behind the acquisition of 4D Light Field with a lenslet light field camera. It is structured in the following way: i) description of the acquisition process; ii) enumeration of the tools used in processing the raw lenslet light field data into a light field image to be coded; iii) review of the most relevant lenslet light field cameras in the market. Sometimes the architecture of a

conventional camera will be used as a comparison, highlighting the differences and similarities between the two types of cameras. In fact, it is possible to obtain a lenslet light field camera by just adding a new optical element to a traditional digital camera as described by Ng. et al. in [19].

2.2.1. Lenslet Images: Acquisition

A lenslet light field camera, such as those manufactured by Lytro and Raytrix [20] [21], captures a new kind of image data/modality by also including the so-called *angular information* that was impossible to capture with conventional cameras. To enable this new acquisition model, a microlens array is introduced in the traditional digital camera architecture. The main elements of a lenslet light field camera are depicted in Figure 7 left).

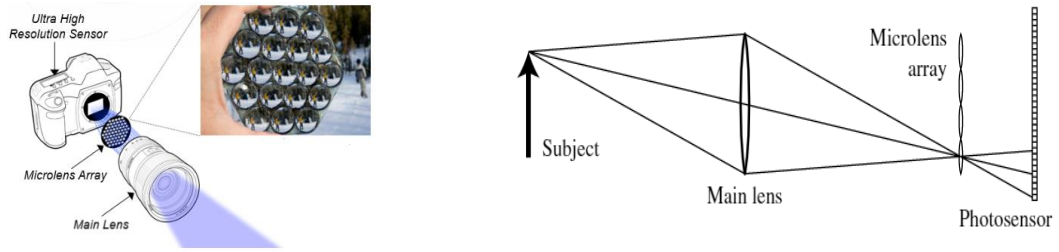


Figure 7: *left) Principal elements of a lenslet light field camera, detailing the microlens array [22]; right) Main lens focuses the subject onto the microlens array, which separates the converging rays into an image on the photosensor behind it [19].*

In summary, the main elements are the following:

- **Main lens** – It has the purpose to focus the light rays from the outside scene into the inside of a camera, more precisely into the microlens array; the optical process is illustrated in Figure 7 right). Two fundamental parameters are the focal length and the aperture size of the main lens.
- **Microlens array** – Usually composed by thousands of tiny lenses, spherical or aspherical; these lenses may be shaped like squares, spheres or hexagons and are arranged in a rectangular, hexagonal or custom grid [23] as shown in Figure 8 left) and mid). Important parameters are the number of microlenses and their resolution (ability to resolve detail). Figure 8 right) shows the effect the microlens array produces in the captured image, each tiny circle corresponds to a singular microlens; usually, each one of these tiny images is called a *micro-image*.

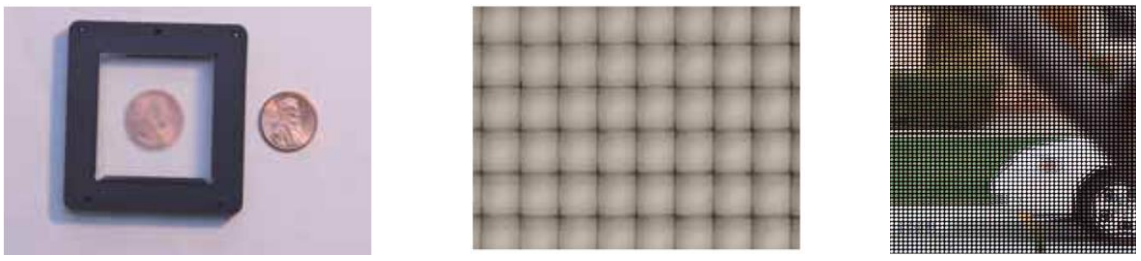


Figure 8: *left) Microlens array with 296 × 296 lenslets, each one 125 microns wide [24]; mid) Zoom of a microlens array, showing the squared lenses [24]; right) Lenslet light field image where each tiny circle corresponds to the image formed under each microlens [25].*

- **Photosensor** – Either a charge-coupled device (CCD) or a CMOS image sensor, which measures the number of arriving photons and converts them into electrons [26]; nowadays, most used photosensors have its pixels masked by a color filter array (CFA), with the most popular being the Bayer pattern filter [27]. The resolution, i.e. the number of pixels and their size, is the most important parameter of a photosensor.

The way these components are assembled in a light field camera is very similar to the traditional digital camera's architecture. Figure 9 shows the main differences between those architectures; notice that the elements on the image are not at scale, as in reality, the photosensor is very close to the microlens array and the size of the microlenses in the image is too big. Details concerning the different lenslet light field cameras architecture can be found in Appendix A.

Figure 9: *left) In conventional digital cameras, the light goes directly from the main lens to the photosensor [4]; right) In a lenslet light field camera, the microlens array is placed between the main lens and the photosensor [5].*

Due to imperfections in the optical elements, some artifacts may appear in the captured image, degrading its quality; this is true for any type of architecture. Usually, these artifacts can be mitigated with some additional optical elements or computational effort (detailed in the next section). The main relevant phenomena are:

- The **spherical aberration**, where the light rays refracted through a lens, with a convex side and a flat side, converge at different depth. Rays passing through the periphery of the spherical interface refract too strongly, converging at a depth closer than the rays which go through the center. As result, these light rays are blurred, thus reducing the contrast and resolution of the micro-image. Some optical solutions are to hold down the aperture or to add lens elements to reduce the artifacts; for further details, see [19]. Figure 10 left) shows the described effect.
- The **vignetting**, where the light rays coming from more obliquus angles carry less energy than those coming from the optical axis, resulting in a darkening of the areas with less energy. Figure 10 mid) shows a crop of a lenslet image where the vignetting effect has not yet been corrected.
- Finally, there is the **chromatic distortion** which is a dispersion effect that happens when the lens is unable to focus all wavelengths at the same convergence point. Figure 10 right) shows the different wavelength rays converging at different depths [28].

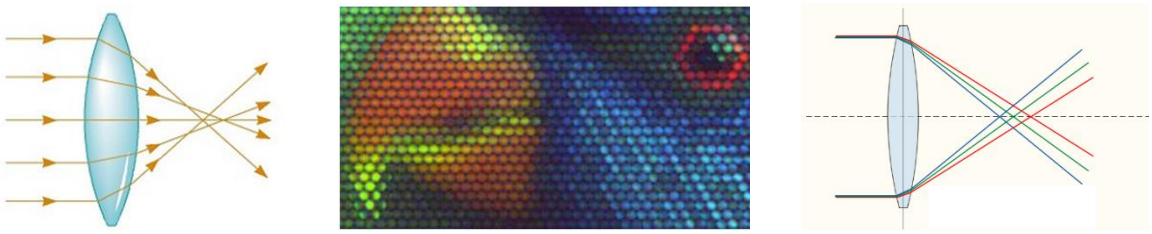


Figure 10: *left) Illustration of spherical aberration phenomenon, detailing the rays converging at different depths [29]; mid) Crop of a raw lenslet light field image without vignetting correction [30]; right) Illustration of chromatic distortion [31].*

Dansereau developed a camera model that precisely describes a real physical camera by considering the lens distortion and projection effects through the microlens array described in [30], which allows obtaining better looking images. At last, the raw lenslet image is obtained as a grid of micro-images, where each one captures light arriving from different angles as shown in Figure 11 left). In summary, the final product of the capturing process with a lenslet light field camera is a raw lenslet

image which cannot, naturally, be visualized in a regular 2D display; for this to happen, some rendering solution has to be applied, resulting for example in Figure 11 right), as it will be detailed in the following sections [30].

Figure 11: *left) Raw lenslet light field image after demosaicing [32]; right) Rendered view from a raw lenslet light field image [32].*

2.2.2. Lenslet Images: Acquisition to Coding Processing Architecture

The raw light field data acquired with a lenslet light field camera, either using an unfocused or focused architecture (details about each architecture can be found on Appendix A), must be processed to obtain a RGB or YCrCb light field image, so that it can be properly coded and stored or transmitted [30]. The processing chain presented below is largely based on the processing chain adopted in the Light Field Toolbox software, developed by D. Dansereau [33], which is commonly used by the research community. Figure 12 shows the basic processing chain from acquisition to coding; while the rectangles correspond to the processing modules, the circles correspond to data inputs/outputs, the dashed line rectangles correspond to procedures performed only once while the pointy line rectangles correspond to the optional processing modules. Details for each module are available in [30] [33]; naturally, there are variations of this architecture, e.g. [34] [35]. Note that the format .LFR is associated to the light field data acquired with a Lytro Illum camera.

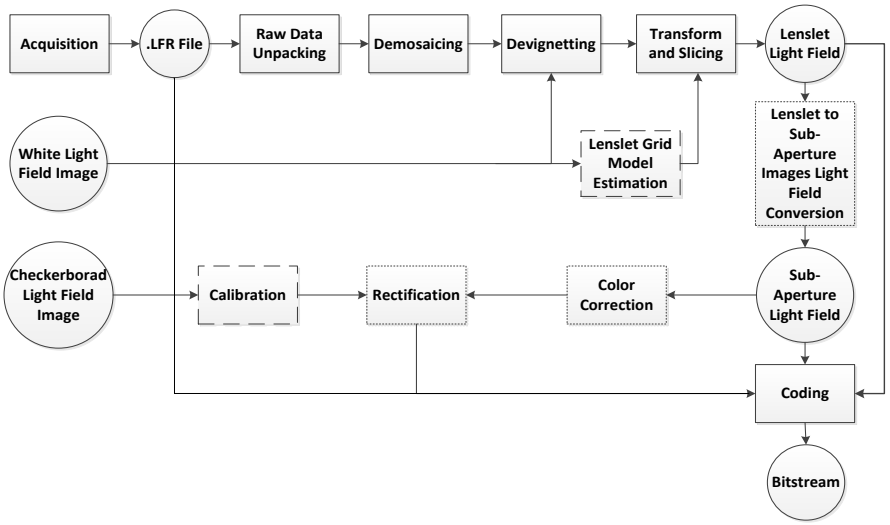


Figure 12: *Architecture from acquisition to coding [30] [33] [36].*

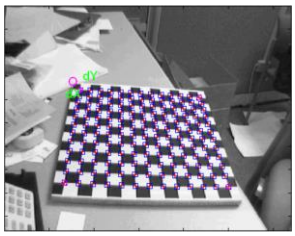
The raw data can be coded directly, but usually the processing chain, including the raw data unpacking, demosaicing, devignetting and transform and slicing, is used with the objective of converting the hexagonally arranged, raw data into a rectangularly aligned lenslet light field (output from the Transform and Slicing module), as shown in Figure 12. At this stage, the lenslet light field may: i) go straight to coding; ii) be converted into a friendlier representation format, notably a sub-aperture (SA) images light field, and after coding; iii) same as ii) but with color correction and rectification before the

coding [33]. Figure 12 illustrates the mentioned possibilities. In the processing chain, each module processes, extracts or manipulates the data from the raw lenslet image and associated metadata to make it suitable for coding as will be discussed in Section 2.4. The lenslet grid model's and calibration's parameters play an important role in the processing chain; they are calculated once and need to be known before the process starts, as they serve as input to some of the other modules. In the following, all the processing modules will be briefly described, starting with the modules which are run only once, detailing the objective and the input/output of each one [33] [37].

- **Lenslet grid model estimation** – Once per camera, a set of white images, as the one shown in Figure 13 left), are analyzed with the objective of estimating a lenslet grid model, one per white image. The lenslet grid model describes the rotation, spacing and offset of the lenslet images on the photosensor; grid parameters are estimated by traversing the lenslet image centers, finding the mean horizontal and vertical spacings and offsets, and performing line fits to estimate rotation [33]. To find the lenslet image centers, the vignetting effect is exploited as the brightest spot in each white image approximates its center, see Figure 13 right). A white image is an image acquired using a setup requiring a white diffuser i.e., a device that scatters the light in a way called soft lightning (it's called this way, when the light source is large (in size) relatively to the subject); this effect may be obtained by making the light to reflect diffusely from a white surface or by using a translucent material [38] [39]. Each camera comes preloaded with a unique set of white images, taken with different zoom and focus settings; the images can also be stored in the external SD Card [33] [40].

Figure 13: *left) Average white raw image used in calibration [35]; right) White image overlaid with grid estimation, where the red dots represent the lenslet centers [30].*

- **Calibration** – In this module, the objective is to estimate several parameters used mainly for compensating the camera optical aberrations. By analyzing different checkerboard light field images, as those shown in Figure 14, the calibration parameters are obtained by identifying first the corners locations, as shown in Figure 14 left), followed by an initialization of the pose and intrinsic parameters and, finally, an optimization of these same parameters. The checkerboard light field image, whose dimensions should be known, must not be color corrected nor rectified (which would invalidate the results); to fully exploit its advantages, it should be smaller and denser than the checkerboard presented in Figure 14 right)**.** Note that the calibration's procedures for a focused lenslet light field camera are slightly different; see details in [41]. The calibration parameters (i.e. focal length, zoom length, exposure length and so on) can be stored jointly with the raw lenslet image. In Lytro's cameras, the file is in LFX format if calibration parameters are stored with the raw lenslet image or in LFR format if the parameters are not included [37].



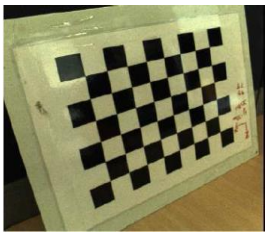
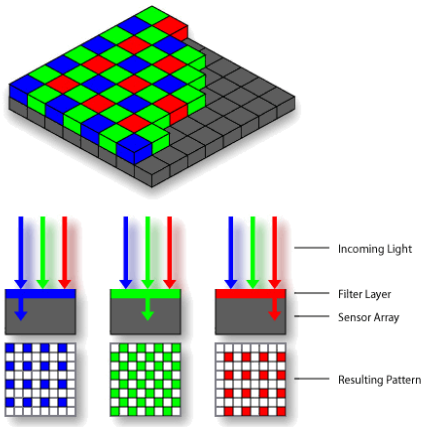


Figure 14: *left) Example checkerboard edges identification for calibration with MATLAB Toolbox [42]; right) Example checkerboard [33].*

The light field processing chain proceeds as follows:

- **Demosaicing** – This module has the task to recover a full RGB lenslet light field image from the raw lenslet image, which has been obtained with a Bayer-pattern filter, as the one shown in Figure 15 left). Demosaicing a raw Bayer pattern image requires an underlying image model to guide the decisions for the reconstruction of the missing color channels; as only one-color channel is sampled for each pixel, the nearby samples need to be used to reconstruct the other two channels [27]. Figure 15 right) shows how the Bayer filter samples the colors in a lenslet light field image.



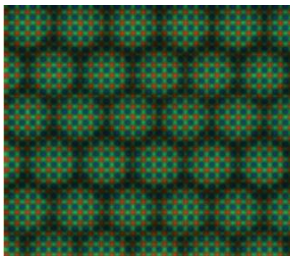


Figure 15: *left) Bayer filter, detailing the captured pattern [43]; right) Bayer Pattern as seen in a lenslet light field image [44].*

In the software made available by Danserau [30], a conventional linear demosaicing technique is applied over the raw image. Yet this method yields undesirable effects, such as blurring caused by averaging neighboring color values across edges, for the pixels near the lenslet edges, which are therefore neglected in the calibration process. A raw lenslet image before demosaicing is shown in Figure 16.

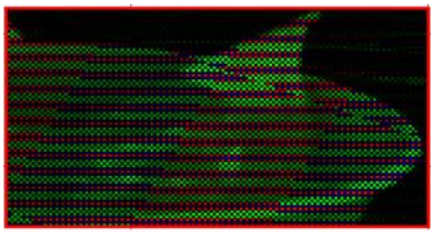



Figure 16: *left) Raw lenslet light field image, before demosaicing [45]; right) Zoom in of the red rectangle in the previous image [45].*

- **Devignetting** – The objective is here to remove the optical defects caused by the vignetting effect, which has been described in Section 2.2.1. The RGB lenslet light field (LLF) image is divided by its corresponding white image to compensate the reduced intensity near the micro-images edges. The appropriate white image is selected based on the serial number, zoom and

focus setting, using for that purpose the lenslet light field image and associated metadata. The output is an image where the vignetting effect is unnoticeable.

- **Transform and slicing** – The objective of this module is to have all micro-images' centers falling on pixels' centers and to produce an orthogonal (and e.g. not hexagonal) lenslet image grid [13]; to find the micro-images' centers, the input LLF image is resampled, rotated, scaled and sliced so that an orthogonal grid is achieved [30]. In the Lytro Illum camera, the grid is hexagonally packed which slightly complicates this processing stage, as it needs to be reshaped to square pixels. The output of this process is an aligned light field. Figure 17 shows the sequence of processing: in green, the photosensor's pixels and, in blue, the micro-images.

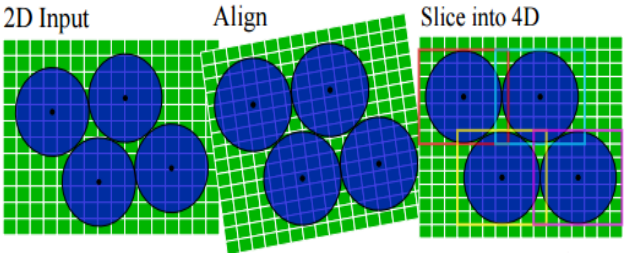


Figure 17: *Stages in transform and slicing module [46].*

- **Coding** – As illustrated in Figure 12, at this stage, four approaches may be taken to code the LLF data:
  1. **Raw light field coding** – The raw light field data can be directly coded without any pre-processing performed.
  2. **Lenslet light field coding** – The lenslet data is coded as a set of rectangular micro-images.
  3. **SA images light field coding** – The lenslet data is arranged as a set of SA images which are after coded.
  4. **SA images light field coding after color correction and rectification** – The lenslet data is arranged as a set of SA images which are color corrected and rectified before coding.
- **Lenslet to SA images light field conversion** – In this module, the (aligned) lenslet light field is re-arranged to obtain a different data structure, more convenient for the following processing tools. In the MATLAB Light Field Toolbox [40], the lenslet image is converted to a stack of SA images, each one depicting a different scene perspective [33], depicted in Figure 18 left). This stack has the dimensions 15 × 15 × 434 × 625 × 4 where 15 × 15 represents the number of views, 434 × 625 represents the resolution of each view and 4 corresponds to the RGB and weighting image components.

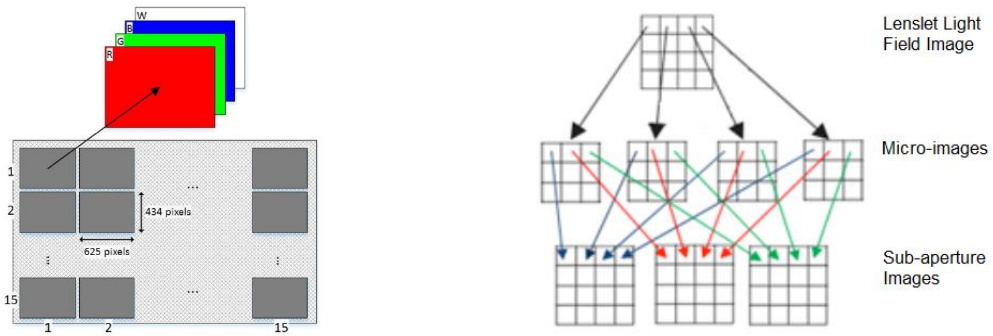


Figure 18: *left) Set of SA images [5]; right) Illustration of the SA images creation process [47].*

A SA image is obtained by taking the same pixel from each micro-image and putting them together, as shown in Figure 18 right), this way an image resembling a traditional 2D image taken from a digital camera is achieved [19].

- **Color Correction** – The objective of this module is to apply color correction to each one of the 2D images in the SA image stack. Exploiting the available light field metadata, this module applies color balancing, RGB color correction and gamma correction. This is an optional procedure that can be performed when processing the lenslet image for the first time or later [33].
- **Rectification** – This module intends to reverse the effects of lens distortion in each one of the 2D images that form the stack of SA images; to do so, it is necessary to select an ideal intrinsic matrix (describing the lenslet light field camera's physical parameters) for the input, color corrected or not, 2D image. It is advisable to use the same grid model for the rectification and calibration processes.

2.2.3. Lenslet Cameras

Nowadays, there are two main manufacturers of lenslet light field cameras, Raytrix [48] and Lytro [20]. Despite using similar technologies and designs, the products from these two manufacturers have unique characteristics and purposes. Lytro has developed two generations of light field cameras; Lytro Illum is the second-generation camera and improves the photosensor's resolution, the rendered 2D image's resolution as well as the design of the much smaller first-generation camera, depicted in Figure 19 left). As the Lytro cameras are photography oriented, they allow the user to ***a posteriori*** adjust the aperture (changing the focus), to change slightly the perspective, to create cinematic animations and to retrieve a depth map. The Raytrix R42 Series camera is more oriented towards industrial purposes. It provides refocus capabilities, depth map generation, 3D models formation and can be used for applications such as plant growth analysis, fluid tracking, and microscopy. This short review will cover the most representative model of each manufacturer, notably the Lytro Illum [49] (depicted in Figure 19 mid)) and Raytrix R42 Series (depicted in Figure 19 right)). Only the specs relevant for this Thesis will be detailed.

Figure 19: *left) Lytro first generation camera [50]; mid) Lytro illum model [51]; right) Raytrix R42 Series model [52].*

Illum is the latest lenslet light field camera developed by Lytro and it is based on the so-called unfocused architecture presented in the previous section. This camera includes a LCD display/touch screen for user interaction, contrary to the Raytrix camera. Its main specs are the following [53]:

- **Sensor -** CMOS with 40 Megaray resolution (number of light rays captured per picture) leading to a 2D rendered image resolution up to 4 Megapixels.
- **Microlens array** - Microlenses with an aperture of f/2.0 organized in an hexangular grid [39].

While Raytrix has a large variety of camera models adopting the focused architecture presented in the previous section [41], the R42 Series design is selected here as it is the one with higher

photosensor's resolution. Note that the Raytrix R42 Series model has also video capabilities; however, since this addresses image compression, the video issues will not be detailed. Its main specs are the following:

- **Sensor -** CMOS with 41.5 Megaray resolution leading to a 2D rendered image resolution up to 10 Megapixels; the pixels are 1.12 µm wide and square shaped.
- **Microlens array -** Spherical microlenses made from polymer on glass, each one with an aperture of f/2.8. The microlenses have different focal lengths, as shown in Figure 20 left), meaning they are focused on different planes, acquiring images like those in Figure 20 right) where the micro-images are not all focused at the same depth plane. This extends the depth-of-field of the camera (multi-focus plenoptic camera) [41].

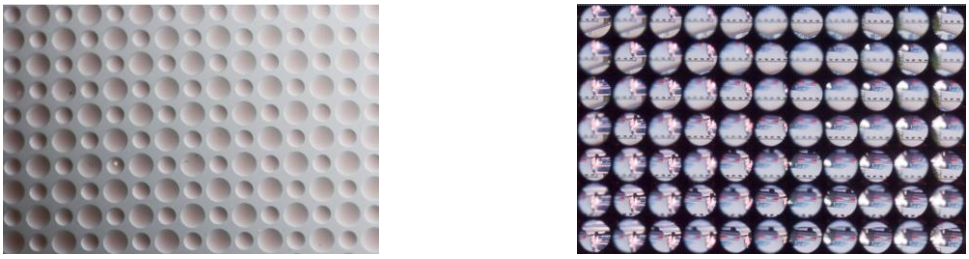


Figure 20: *left) Variable focal length microlens [22]; right) Raw image acquired with variable focal length microlenses [22].*

While both cameras capture visual data extra dimensions using a microlens array, they have different microlens array characteristics, notably the array positioning and the microlenses shapes, for Raytrix R42, the microlenses have different focal lengths, thus allowing to construct a focused image at any depth of focus, and improving the digital refocusing [22]. For example, Raytrix R42 allows having a better 2D rendered resolution despite its photosensor resolution being almost the same as Illum; this is possible because the Raytrix camera uses different focal length microlenses and is based on the focused camera architecture. Naturally, each one of these cameras comes equipped with appropriated software to manipulate the data, notably to render the content in multiple perspectives and focal planes.

2.3. Lenslet Light Fields Cameras: Rendering

In principle, displaying a full light field image may be done by reverting the acquisition process thus recreating a real light field as shown in Figure 21 left). However, if the visualization has to happen in other types of displays, e.g. a regular 2D display, the rich data acquired by the lenslet light field camera needs to be processed so that some extracted/rendered information can be displayed in the proper format while providing the user with the best quality of experience. The most relevant types of displays to consider are: i) a single 2D image for a conventional 2D display; ii) a pair of 2D images/views for a stereoscopic display; iii) multiple 2D images/views for an autostereoscopic display. The several possibilities mentioned here are listed in Figure 21 right) [36] [30], while distinguishing multiple perspective and multiple focal point rendering for 2D displays. The main target of this section is to briefly review some of the most relevant rendering methods for 2D conventional displays and even more briefly, for stereo and multi-view displays. Thanks to the extra dimensions in the acquired light field data, it is possible to *a posteriori* perform digital refocusing and move slightly the observer perspective. For example, the Lytro cameras provide software which allow the users to exploit these two functionalities, providing to the users with post-capture digital refocusing, creation

of live pictures (digital refocus plus slight perspective changes) and creation of 2D and 3D animations [54].

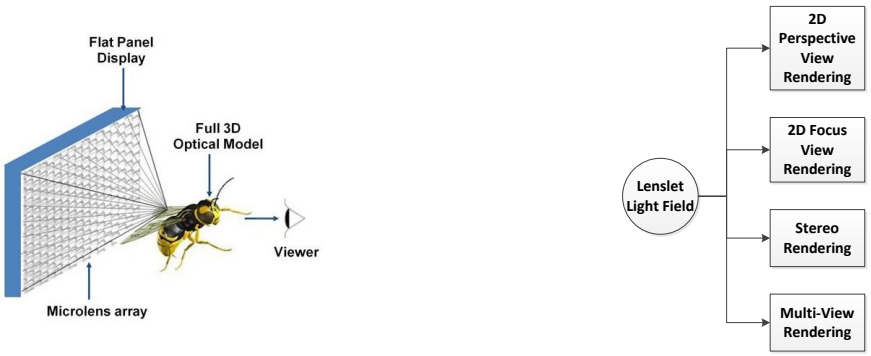


Figure 21: *left) Reverse process of light field acquisition made with a microlens array [55]; right) Several rendering possibilities from a lenslet light field.*

The first rendering solutions for the mentioned functionalities missed a good 2D image rendering spatial resolution; this resulted in several subsequent new architectures and processing methods to improve the rendered image resolution. The way data is extracted and processed to render a 2D image is the key to achieve higher rendering resolution and quality; in practice, the rendered images are the images seen by the users and, thus, the rendering process has a strong impact on the final user experience. For this reason, some of the most relevant rendering methods in the literature and posterior improvements for lenslet light fields are reviewed in the following. The methods here presented may be organized according to the type of data they use, notably only texture or texture plus depth. The texture based rendering methods process the data considering the properties of the lenses, geometry of the camera and distance from the camera to the objects in the scene. The texture plus depth based rendering methods exploit the availability of depth information to design more complex and better performing rendering solutions, explicitly exploiting the availability of some geometry information [55]. To estimate depth information, there are several methods available in the literature, notably disparity matching based methods and epipolar images (EPI) based methods. In the later, the pixel depth is estimated through the slope of its corresponding depth line in the EPI. It is important to highlight that depth estimates from any source are usually noisy, implying that global smoothing schemes may have to be employed to improve the final depth result [56].

2D Perspective View Rendering

This functionality allows the user to slightly change the perspective, as depicted in Figure 22. Naturally, as the lenslet light field cameras have a very narrow baseline between microlenses, both the maximum horizontal and vertical parallaxes are relatively small.

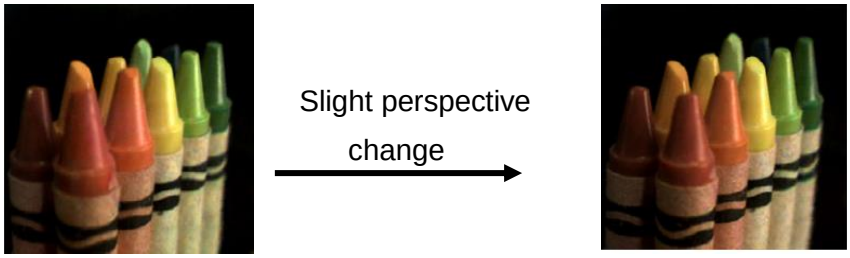


Figure 22: *Slight change in perspective for both horizontal and vertical parallaxes [19].*

Among the most relevant methods to render multiple perspective views from a lenslet light field, it is possible to highlight:

- The most **basic approach** (texture-based) to extract a 2D view from a lenslet light field image consists in creating the 2D image by selecting the pixel in each micro-image corresponding to the chosen perspective, see Figure 23 up left); each 2D image obtained by this method corresponds to a different perspective view and is called a SA *image* [55] [57]. The number of views available is equal to the resolution of each micro-image and the resolution of the extracted view (the SA views) is equal to the number of micro-images. This approach may involve some depth artifacts in the rendered 2D image and stereo limitations due to the limited number of views. The rendering software in the first camera proposed by Ng *et al*. [19] used this rendering method; unfortunately, the images rendered with this algorithm achieve a very low resolution (in his work, the rendered 2D image resolution was only 300 x 300 pixels [58]).
- The **view selective blending** (texture-based) method targets to improve the rendered 2D image resolution [55] [57]. To do so, it uses the basic extraction method described above, it upsamples each view to increase its resolution and stacks the selected views to finally blended them all together by averaging all the superimposed pixels; in this way, a higher rendering 2D image resolution is achieved. Figure 23 up right) shows the process of selecting views for the blending process and Figure 23 down) shows a representation of the blending process.

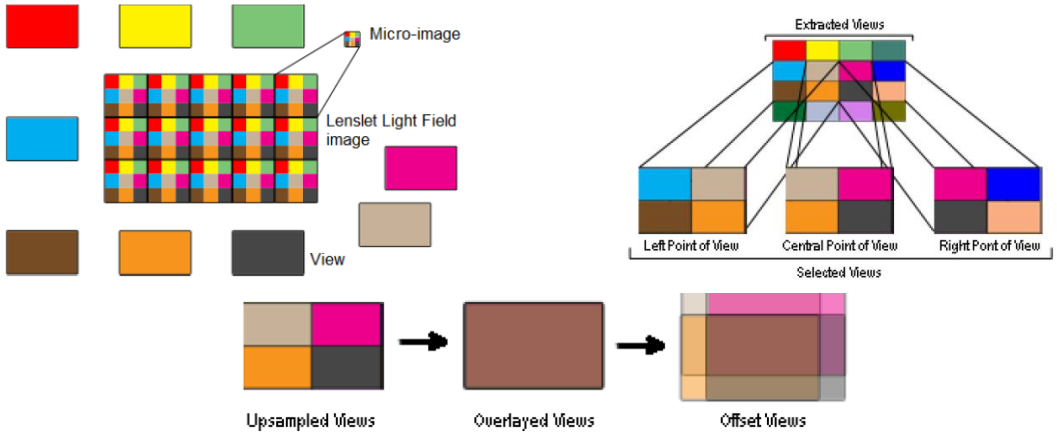


Figure 23: *up): left) Basic extraction method, each colored rectangle corresponds to an individual 2D view [55]; right) Illustration of the selection process in the view selective blending rendering method [55]; down) Overlay and average of the offset views [55].*

- The **single-sized patch method** (texture-based) has been proposed by T. Georgiev and A. Lumsdaine [58] and is based on the characteristics of the focused lenslet light field camera [55] [57]. In the unfocused lenslet light field camera, each spatial point is sampled by a single micro-lens, and so rendering involves integrating pixels in each micro-image, resulting in one pixel extracted per micro-image. When rendering data acquired with a focused lenslet light field camera, the integration must be performed across micro-images, instead of within micro-images [58], allowing the extraction of more than one pixel per micro-image. Consequently, this rendering approach consists in extracting a square area (the so-called *patch*) of pixels from each micro-image, which is after combined with other patches to form the final rendered 2D image. By changing the patch position inside the micro-image, a different perspective is selected; naturally, choosing incorrectly the patch size can lead to noticeable artifacts, as illustrated in Figure 24.

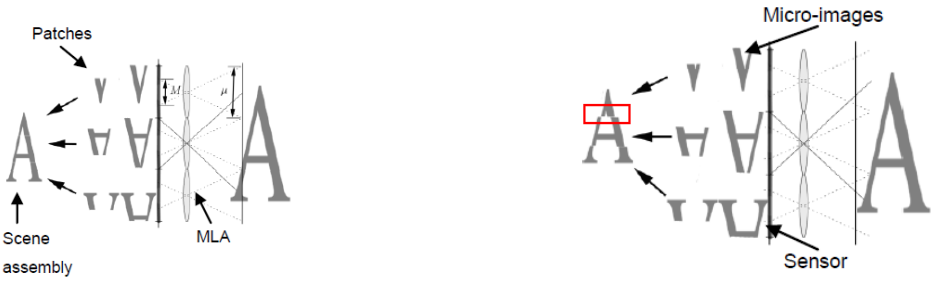


Figure 24: *Patch based 2D images formation: left) Good patch size selection [55]; right) Too small patch size resulting into artifacts [55].*

- The **single-sized patch blending method** (texture-based), also proposed by T. Georgiev and A. Lumsdaine [59], works in a similar way to the previous method; however, instead of ignoring the pixels outside the patch borders, they are now kept to be blended later [57] [55]. The blending is done by taking a weighted average, where each pixel inside each micro-image will have a different weight. The weighting may be defined by a Gaussian distribution centered on the patch, meaning that pixels further away from the patch center have a lighter weight and vice-versa, as shown in Figure 25.

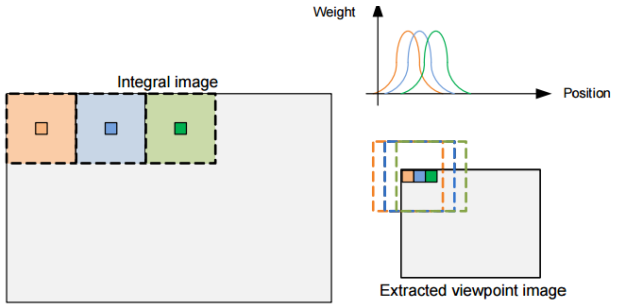


Figure 25: *Illustration of the pixels weighting in the blending process in the single-sized patch blending method [57].*

- The **disparity map method** (texture plus depth based) has been proposed by T. Georgiev and A. Lumsdaine [58] and targets to obtain a full focused and artifacts free rendered image by having the patch size adaptively determined by the depth data [57] [55]. To do so, it is necessary to determine the depth for the objects within each micro-image. As the patches may differ in size depending on the depth, some of them may need some degree of magnification when assembling them, to match the size of the remaining ones.
- The **depth blending method** (texture plus depth based) has been proposed by T. Georgiev and A. Lumsdaine [27] and consists in a combination of the blending and disparity map methods above, targeting to produce less rendering artifacts in comparison with the previous solutions [57] [55].
- The **3D model based rendering method** (texture plus depth based) creates 2D perspective views using a 3D model which is built using both texture and depth information. Figure 26 left) shows an example of a perspective view rendered from a 3D model, while Figure 26 mid) show the corresponding depth map and Figure 26 right) shows a total focus rendered image derived from the created 3D model [60].

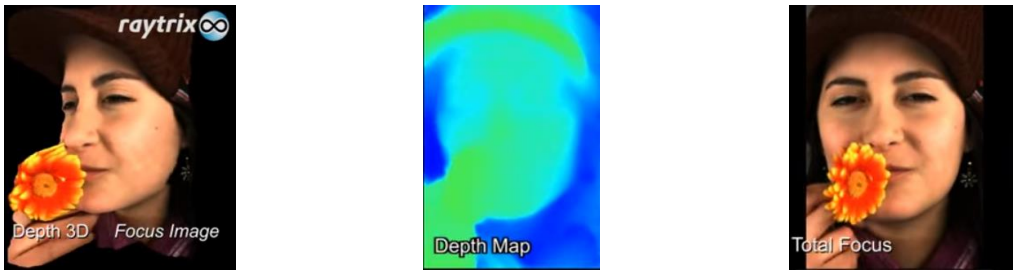


Figure 26: *Data acquired by the Raytrix R11 camera: left) 3D data model [60]; mid) Depth map [60]; right) Total focus rendered image [60].*

2D Focus View Rendering

This functionality allows the user to focus the rendered image at different depths after taken the picture, as shown in Figure 27. It may also provide the capability to sharpen several depth planes at once, resembling the process of regulating a camera's aperture [55].

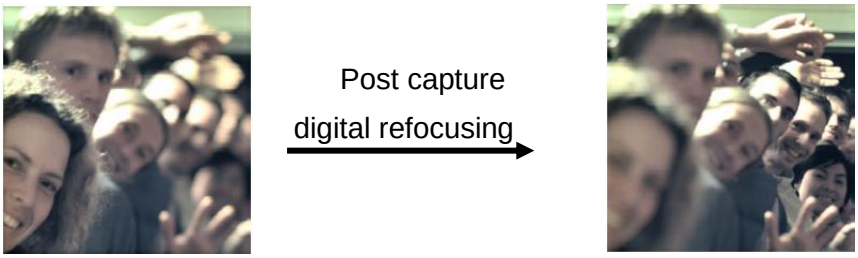


Figure 27*: Example of digital refocusing [19];*

Among the most relevant methods to create views with different focal points from a lenslet light field, it is possible to highlight:

- The **basic approach** uses the SA images, extracted as described in the basic approach for 2D perspective view rendering, to perform digital refocusing which corresponds to shifting and adding the various SA images [19].
- In [61] Ng has introduced a new method for digital refocusing based on a Fourier representation, which is based on slicing rather than integration/adding. The solution obtained is significantly faster when compared to basic approach [61].
- The **super-resolution method** (texture-based) performs digital refocusing in the same way as described for the basic approach but using now the view selective blending method described for 2D perspective view rendering, thus obtaining higher resolution refocused images. To control the rendered focal point, a specific drift value may be applied to each view in the stack (by increasing its distance to the central point) before the blending process.
- The **single-sized patch method** (texture-based) works as described for 2D perspective view rendering. The size of the patch dictates the plane in focus, allowing to perform digital refocusing.
- The **single-sized patch blending method** (texture-based) performs digital refocusing as the single-sized patch approach while obtaining better 2D focus images resolution. The reason behind the resolution improvement is detailed in the single-sized patch blending method for 2D perspective view rendering.
- The **disparity map method** (texture plus depth based) is the first method to deliver a full focused image because the extracted viewpoint image is full-focused; as each patch size is adapted to the depth of the objects in the micro-image [55]. Choosing to focus at a specific depth plane corresponds to defocus the other depth planes.

- Finally, the **depth blending method** (texture plus depth based) consists in a combination of the disparity map method with the blending method.

Stereo/Multiview Rendering

When comparing the rendering needs for stereoscopic and multi-view autostereoscopic displays with a conventional 2D displays, the main difference regards the number of views that needs to be rendered with a specific baseline. Thus, to perform rendering for a stereoscopic display, one should simply extract two views with a baseline corresponding to the distance between the human's eyes; naturally, to perform rendering for autostereoscopic displays, a higher number of views should be extracted, each one corresponding to a different perspective, to be displayed with the target to give the user an immersive and realistic experience.

2.4. Main Lenslet Light Field Coding Solutions Review

The acquired lenslet light field image carries a huge amount of data, thus asking for efficient coding solutions for ensuring lower amounts of storage, transmission rates and, if possible, applications with real-time requirements, e.g. streaming, where both compression efficiency and complexity are critical issues. The possible coding solutions for lenslet light field image can be grouped into four categories:

1. Standard compliant coding solutions, e.g. JPEG, HEVC Intra.
2. Standard compliant coding solutions applied after some data re-organization.
3. Extended standard coding solutions where additional tools are included to improve the standard compression performance for lenslet light field images.
4. Non-standard based coding solutions, i.e. not fitting in any of the previous groups.

While the light field coding domain is rather recent, there are already several solutions proposed in the literature. In the following, the most relevant coding solutions for each category defined above will be reviewed. The choice of the solutions to review was made based on their relevance, provided detail, technical novelty, performance and year of publication.

2.4.1. Standard Compliant Coding Solutions

Objective

The direct application of standard compliant coding solutions to lenslet light field images is the most straightforward and simple coding approach. Naturally, these coding solutions cannot exploit all available redundancy, e.g. the redundancy between the various micro-images, but they allow benefiting from the standard ecosystem as standard bitstreams and decoders are used. As so, it is interesting to review the performance of these coding solutions as they establish benchmarkings for the upcoming improvements as well as provide an indication on the minimum performance. This review is largely based on the study available in [62].

Architecture and walkthrough

The standard compliant coding solutions suitable for lenslet light field image coding can be divided in still image coding standards, namely JPEG and JPEG 2000, and video coding standards used in the Intra coding mode, namely HEVC Intra and H.264/AVC Intra. Even though they do not fully exploit the lenslet light field images redundancies, they can be directly applied to the demosaiced light field image and benefit from standard compliant coding [62]. The most relevant

available standard coding solutions to be applied for images are:

1. **JPEG** – This standard coding method is based on the discrete cosine transform (DCT), where spatial redundancy is exploited, followed by the quantization of the DCT coefficients, where the irrelevancy is exploited, and finally entropy coding where the statistical redundancy is exploited. It is a very simple and rather old codec but still largely used nowadays. The block diagram for the JPEG encoding architecture is depicted in Figure 28 left).
2. **JPEG 2000** – This standard coding method is based on the discrete wavelet transform (DWT). It was developed to improve not only the compression performance regarding JPEG but also to add new features, such as scalability; it is capable of lossy and lossless compression. The encoder starts by transforming the initial image from RGB color space to YCrCb or YUV color space, then the image is subjected to the wavelet transform, which can be reversible or irreversible, and finally the transform coefficients are quantized and entropy codified. The block diagram for the JPEG 2000 encoding architecture is depicted in Figure 28 right).

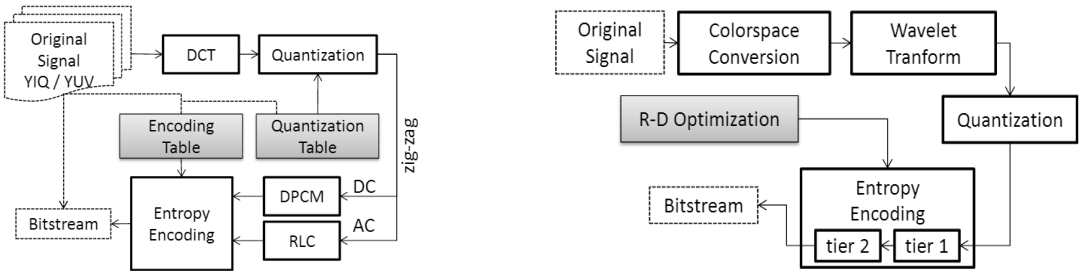


Figure 28: *left) JPEG encoding architecture [63]; right) JPEG 2000 encoding architecture [63].*

3. **H.264/AVC Intra –** This standard coding solution corresponds to set of Intra coding tools included in the H.264/AVC video coding standard. It begins by splitting the input image into macroblocks, which residue is Integer DCT coded after appropriate Intra prediction. In general, the macroblocks may be coded in Intra or Inter mode depending on the type of redundancy exploited. In the Intra coding mode, the macroblock may be predicted using already decoded neighboring samples, thus exploiting the spatial redundancy; this is different from previous coding standards which set the prediction signal to zero, meaning that the macroblock is coded without any reference to previously coded information (excluding DC coefficients). Finally, the quantized DCT coefficients are entropy coded. The block diagram for the H.264/AVC Intra encoding architecture is depicted in Figure 29 left).
4. **HEVC Intra –** This standard coding method follows the classic block-based hybrid video coding approach as H.264/AVC. The basic coding algorithm is a hybrid of inter-picture prediction to exploit the temporal statistical dependencies, intra-picture prediction to exploit the spatial statistical dependencies, and transform coding to further exploit the spatial statistical dependencies in the prediction residue. The main technical novelties regarding H.264/AVC for Intra coding are [64] [65]: i) HEVC Intra coding structure is no longer based on the macroblock concept but instead in a quadtree scheme with varying size blocks. The largest coding unit (CU) can be of size 16 x 16, 32 x 32, or 64 x 64 luminance pixels and each CU can be recursively divided into four equally sized CUs; the size of each CU is defined as 2*N* x 2*N* with $N \in \{4, 8, 16, 32\}$, if the maximum hierarchical CU depth of four is applied [65]; ii) for Intra prediction, the prediction units (PU) may take the size of 2*N* x 2*N*. HEVC increases the number of H.264/AVC Intra prediction (IP) modes to 35 (DC, planar and 33 angular IP modes), for each PU size [65].

The block diagram for the HEVC Intra encoding architecture is depicted in Figure 29 right).

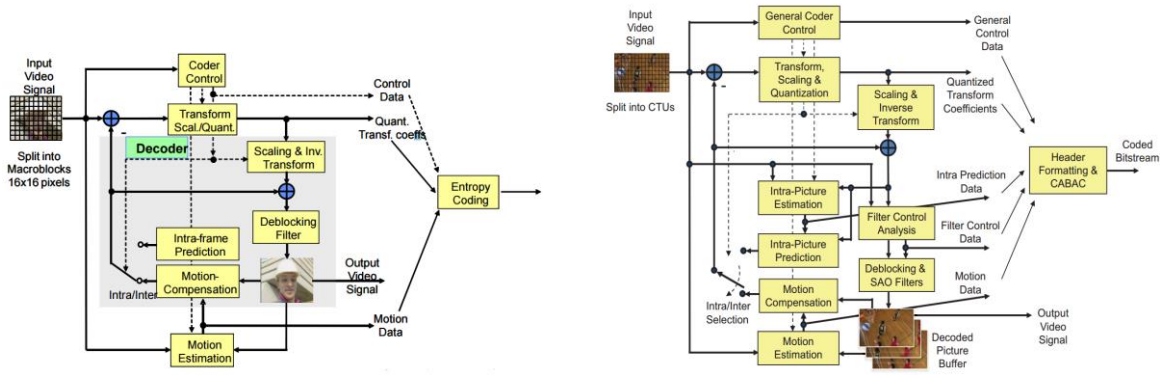


Figure 29: *left) H.264/AVC architecture implementation [64]; right) HEVC Intra architecture implementation [66].*

Performance Evaluation

In [62], the performance assessment of the standard compliant coding solutions was made with 6 light field images selected from the MMSPG-EPFL Light Field Image Dataset. Each image was acquired with a Lytro Illum camera, where the photosensor was overlaid with a Bayer filter aligned in the "GRBG" format with a resolution of 7728 x 5368 samples; the demosaicing produces an RGB image with the same resolution. The performance was evaluated for: i) multi-focus rendered images, with three focal points rendered for each light field image; ii) multi-perspectives rendered images, with nine perspectives rendered for each light field image. The MATLAB Light Field Toolbox [33] was used for the rendering where the SA images were taken as the rendered perspective view, without further resolution improvement while the multi-focus images were rendered by controlling a single parameter, designated *focal shift*, which defines the focal distance; more details related to the test conditions and assessment methodology are available in [62].

For evaluating the compression performance of each coding solution, the rendered perspective-views/focus-view images from the decompressed light field image were compared with the same perspective-views/focus-view image rendered from the original light field image (without compression); the distortion was measured using the PSNR as quality metric. To evaluate only the coding impact, the rendered solution applied for both the compressed and uncompressed light fields was the same, this means the MATLAB Toolbox [33].

Figure 30 reports the RD performance where the average Y PSNR is shown as a function of the average rate for the rendered views, in this case in bit-per-pixel. Regarding the multi-perspectives RD performance, the following conclusions were taken: i) for a given light field image, considering a given coding solution, the curves for each perspective view are quite similar; this behavior can be seen in Figure 30 left) for the JPEG coding solution; ii) the distortion values for the various views are very close, both for the higher and lower bitrates; however, the quality differences are larger for intermediate bitrates, also shown in Figure 30 left); iii) generally, the perspective views farther from the center, such as the views 1, 7, 3 and 9 from Figure 30 mid), have lower RD performance. According to [62], this may be related to the *vignetting* effect, since views near the edges are darker and usually noisier. As expected, HEVC Intra is clearly the most efficient solution, as seen in Figure 30 right), although the specific type of content, more or less high frequencies, may naturally affect the RD performance.

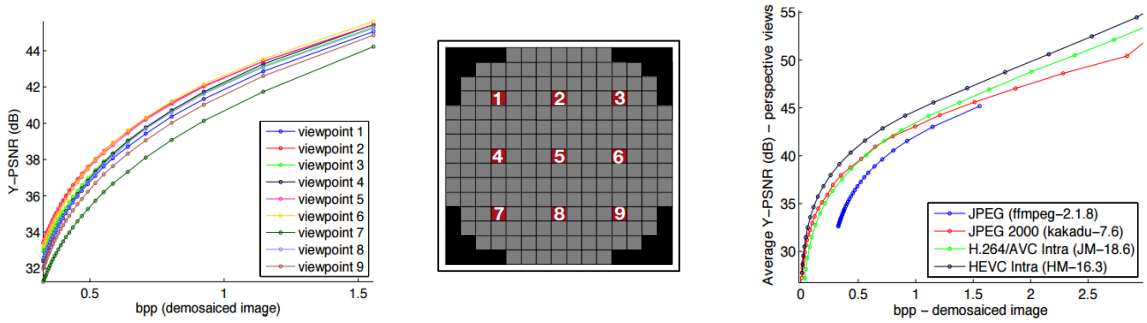


Figure 30: *left) JPEG RD performance for the 9 selected views of the Bikes light field image [62]; mid) Positions in the set of SA images selected for the multiple perspective rendering [62]; right) Average RD performance comparison for the Desktop light field image [62].*

Concerning the multi-focus case, the following conclusions were taken: i) the RD performance is significantly affected by changing the specific focus point; ii) the focus views do not have all the same relative RD performance, most likely due to the nature of the image content; iii) the PSNR achieved for a specific rate for all coding solutions is higher than for the multi-perspectives case; this may be seen by comparing Figure 30 left) with Figure 31 left). In summary, the results establish the superiority of HEVC Intra, followed by JPEG 2000, H.264/AVC and JPEG; however, sometimes JPEG may achieve better RD performance than H.264/AVC Intra, notably for the higher rates/qualities, as shown in Figure 31 right).

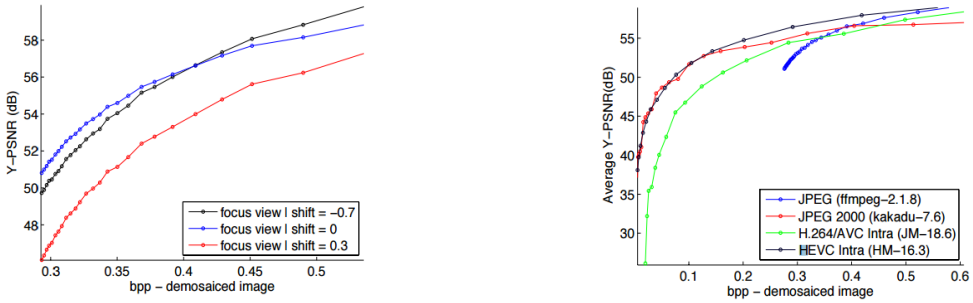


Figure 31: *left) JPEG RD performance for three focus views of the Desktop light field image [62]; right) Average RD performance comparison for the Friends 1 light field image [62].*

2.4.2. Standard Compliant Coding Solutions after Data Re-organization

This section describes two methods which seek to improve the standard compliant coding solutions' efficiency by re-organizing the lenslet light field data before coding; the main target of this re-organization is to better exploit the redundancy in the data, notably between the SA images. The first method is a simple re-organization of the data which takes the set of SA images as a sequence of video frames [67]. The second method codes the set of SA images using some appropriate 2D spatial prediction structure [68].

2.4.2.1. Simple Data Reorganization

Objective

While the previous section described the basic approach for light field coding, which corresponds to the straight, compliant usage of standard coding solutions, the first natural improvement possible is accomplished by just reorganizing the lenslet light field data and using again standard coding solutions. The data format, this means the way the data is organized and scanned, has in fact a tremendous

impact on the obtained compression performance. After the data is reorganized as a sequence resembling video, standard video coding solutions such as HEVC and MVC, can be applied.

Architecture and Walkthrough

The simplest way to improve the standard coding solutions' compression performance is by re-organizing the captured data as a 'video sequence' such that the resulting "temporally" adjacent views, the new 'video frames', have the greatest correlation possible. A limitation of the direct use of standard coding solutions even after data re-organization, e.g. transforming the lenslet image into the stack of SA images, is that they cannot make full use of the existing redundancies notably between views, as available coding standards are only able to exploit the inter-view redundancy for linear, horizontal camera arrangements; the method now presented is the first step in a long way to deal with this limitation. The following alternative procedures are based on the use of an HEVC codec, this means a video coding solution, as a light field coding solution.

The first step of the data re-organization consists in extracting the SA images from the raw lenslet light field image, as shown in Figure 32; this procedure has been explained in some detail in Section 2.3.

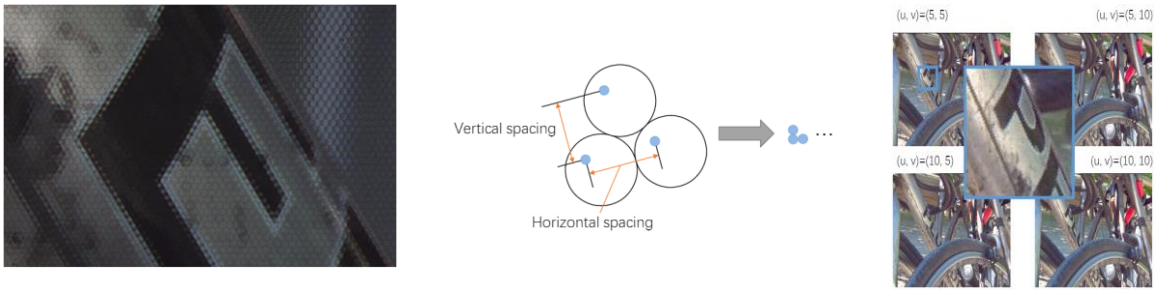


Figure 32: *left) Example of a raw lenslet light field image [68]; mid) Process used in extracting a single perspective view from the raw image, the small blue circles represent the pixels that together form a perspective view [68]; right) Example of several views extracted from a raw lenslet light field image [68].*

After, the SA views are organized as a pseudo 'video sequence' with a scanning order which may vary while targeting to exploit the most the redundancy between the successive views. In [67], two possible scanning orders are proposed to create the 'video sequence':

- **Raster scanning** – The 'video sequence' is built by scanning the views from left to right and top-down, as shown in Figure 33 left).
- **Spiral scanning** – The 'video sequence' is built by scanning the views from the central view outwards, following a spiral order as shown in Figure 33 right).

Figure 33: *left) Raster scanning order [67]; right) Spiral scanning order [67].*

By observing Figure 33, it is still possible to further exploit the dependency between views as some views are coded without exploiting the redundancy with some neighboring views. Due to the nature of the lenslet light field image data, once the SA views are organized in a 'temporal sequence', the available redundancy between SA images can be better exploited for compression efficiency purposes.

Performance Evaluation

In [67], the possible compression efficiency gains achieved by scanning the SA views in different scanning orders are studied. For test purposes, a set of 12 light field images were used; the images were capture with a Lytro Illum camera, each raw image with a total resolution of 7728 × 5368, in GRGB Bayer format. The images are demosaiced into a YUV image with 9375 × 6510 luminance resolution using the Light Field Toolbox software [33]. For the compression performance evaluation, five data formats were tested [67]:

1. **Light field** – Lenslet light field image obtained by the Light Field Toolbox software where the image is organized in micro-images.
2. **All-views** – SA views are extracted from the light field image and placed side-by-side;
3. **Light field filled** – Similar to the light field format, but with the black corners of each micro-image filled by extending the left neighbor pixels.
4. **Raster** – SA images raster scanned as described above.
5. **Spiral** – SA images spiral scanned as described above.

While the first three data formats are encoded using the HEVC Still Image Profile, the last two formats are encoded as a video sequence using the following HEVC configurations: i) All Intra; ii) Low Delay B; iii) Low Delay P; iv) Random Access. Figure 34 shows the RD performance results, in terms of YUV PSNR versus bitrate, with the lower rates region magnified. For the very low rates, the authors [67] claim that "all data formats using video data formats achieve quite reasonable or even good PSNR"; for these rates, all data formats using video coding produced similar results, due to high quantization parameters.

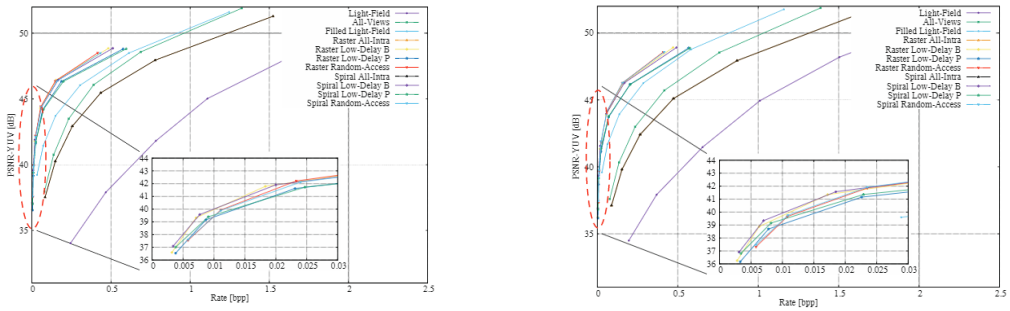


Figure 34: *RD performance: left) Bottles light field image [67]; right) People light field image [67].*

From Figure 34 it is clear that all data formats that used video coding offered much better performance than still image coding. In conclusion, different data formats coding configurations have strong influence on compression efficiency. Arranging the light field data as a 'video sequence', before coding, allows achieving better RD performance when using the HEVC standard in various configurations. Despite not providing the best efficiency, the Still Image Format of HEVC allows fast access to individual views. Regarding the two proposed scanning orders, they produce very similar results, implying that both can be used without expecting significant differences.

2.4.2.2. Data Reorganization with Improved Coding Order and Prediction Scheme

Objective

As mentioned above, the simple scanning of the SA images as a temporal video sequence does not fully exploit the correlation between views as some neighboring views are not used to create better predictions. In [68], a new standard-based coding solution is presented based on the previous solution, this means based on some scanning of the SA images, but also using a more efficient prediction structure similar to those used in the Multi View Coding (MVC) standard. In this way, the view similarities should be better exploited and so a better compression performance should be achieved.

Architecture and Walkthrough

The coding solution proposed here is similar to the solution in Section 2.4.2.1. on the way the different views are extracted and scanned. While the HEVC video coding standard is used, the way the inter-view prediction is created resembles MVC predictions, as inter-view predictions are made although created as temporal predictions. Concerning the proposed coding scheme, the main difference to MVC is that MVC does not easily handle hundreds of views (for a single time instant). The two main tools that contribute to RD performance improvements compared with the previous solutions are the prediction structure and the positioning of this structure:

- **Prediction structure** – Considers the lenslet light field image structure and it does so by considering two important characteristics: i) adjacent SA views, both horizontally and vertically, exhibit higher similarity with each other; ii) central views present higher similarity with each other when compared to the views near the borders. Each view is assigned to a prediction layer and views at a higher layer are coded after the views at lower layer, as such, the views in the lower layers can be predicted from the former decoded views; each view selects a prediction view/views from top, bottom, left and right directions, always from the closer selected view must be at a lower layer.
- **Structure positioning** – The views are processed in the following way: i) the central view is first coded as an I-frame; ii) after, the remaining views are coded as P- or B-frames in a symmetrical 2D hierarchical structure.

An example of the coding order and prediction scheme is illustrated in Figure 35 left) for a structure with 9 x 9 views [68]. The similarities between the proposed inter-view prediction scheme and the MVC inter-view prediction scheme are rather obvious, as can be seen by comparing Figure 35 left) with Figure 35 right).

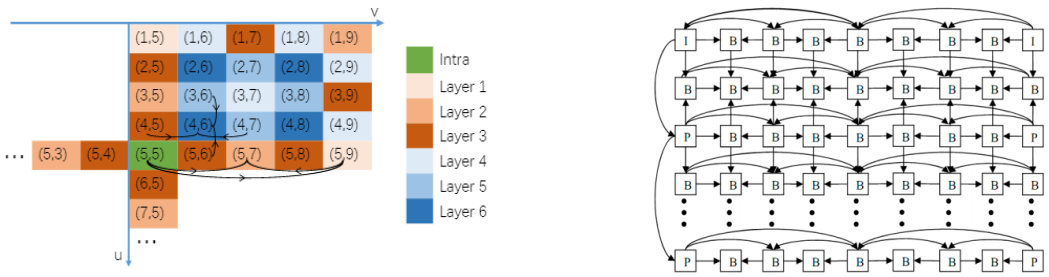


***Figure 35**: left) Example coding order and prediction scheme where the arrows illustrate prediction relations [68]; right) Inter-view prediction scheme from MVC [69].*

Performance Evaluation

The reported performance assessment [68] uses twelve lenslet light field images taken with a Lytro Illum camera, with the same resolution as those used with the previous coding method. The pseudo video sequence is encoded with the HEVC reference software (HM) as well as with the reference software for the next video coding standard under development in MPEG (JEM). For performance comparison, five different coding solutions are used, namely JPEG, HEVC Intra, HM-equal-QP, HM and JEM. JPEG and HEVC Intra are applied directly to the raw lenslet image; HM-equal-QP, HM and JEM are applied to the pseudo sequence of views. The prediction structure and coding order used for HM and JEM coding corresponds to the one presented above. Figure 36 clearly shows that the HM and JEM based coding solutions improve the RD performance, achieving significant coding gains compared to the other tested coding solutions; this means that is possible to improve the coding solutions' performance by re-arranging the data and efficiently exploiting its redundancies by using an improved prediction scheme appropriately positioned.

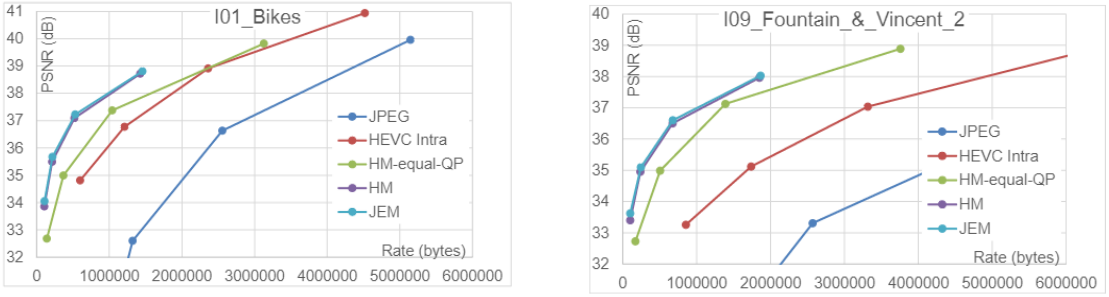


Figure 36: *RD performance for various coding solutions in: left) Bikes light field image [68]; right) Fountain and Vincent light field image [68].*

2.4.3. Extensions Based on Standard Coding Solutions

This section details two coding solutions which extend the HEVC standard with additional prediction tools, aiming to better exploit the additional redundancy present in light field images. These tools exploit additional similarities between image areas to reduce the prediction error. Despite both solutions having the same main objective and similar architectures, the new prediction modes are conceptually different, and so the proposals will be presented separately. As both solutions have a baseline and an improved extension, both will be described, always starting with the baseline solution [70] [71].

2.4.3.1. HEVC-based Bi-Predicted Self-Similarity Compensation

Objective

To better exploit the correlation properties of the lenslet light field data, new prediction tools are necessary towards achieving the best possible compression, notably by extending the available, state-of-the-art HEVC standard. The self-similarity (SS) concept detailed in [72] allows to design a new prediction mode, known as *SS prediction mode*, able to provide higher compression efficiency by exploiting the special arrangement of the lenslet light field image (this means the set of micro-images). The baseline prediction solution is an Intra prediction tool based on block matching; the best match between the current coding block (CB) and an already coded and reconstructed area of the image is signaled by a shift vector, known as the SS vector, this SS vector is conceptually similar to a motion vector but acting within the same image. While the SS baseline compensation prediction mode already

brings compression performance improvements in comparison with compliant standard solutions [70], further improvements were obtained, namely through the so-called *Bi-Predicted SS Compensation* (BI-SS) [71]; as the latter prediction mode is the most advanced, it will be more detailed. The objective is to improve the SS baseline prediction solution performance by including one more prediction mode.

Architecture and Walkthrough

The extended HEVC coding solution presented here [70] may be applied to the lenslet light field image without requiring any precise knowledge on the light field image structure such as the micro-images size or the number of pixels behind each microlens. As shown in Figure 37 left), the HEVC integration of the novel SS prediction modes requires the inclusion of three new modules in a HEVC Intra encoder, namely, the SS Estimation, the SS Compensation and the Reference Memory modules [70] which are briefly described here:

- **SS Estimation** – The main objective of this module is to create novel predictions for the block under coding by exploiting the SS concept. With the Bi-SS mode, two candidate predictors for the current CB are searched over the previously coded and reconstructed blocks in the relevant SS reference area, using a block-based matching, as illustrated in Figure 37 right). The first predictor is obtained using the baseline SS estimation process; the second candidate predictor is obtained as a linear combination between the first predictor and a second prediction candidate block.
- **SS Compensation** – The main objective of this module is to perform the SS compensation by creating the selected prediction block for the block under coding to which a prediction residue may eventually be added. For the SS compensation, the prediction block is obtained as an average between the two predictor blocks selected. The relative position between the block under coding and its predictors is defined using SS vectors.
- **Reference Memory** – The main objective of this module is to manage the reference list construction and signaling, notably including the SS reference area as prediction area for the HEVC Intra-coded frames.

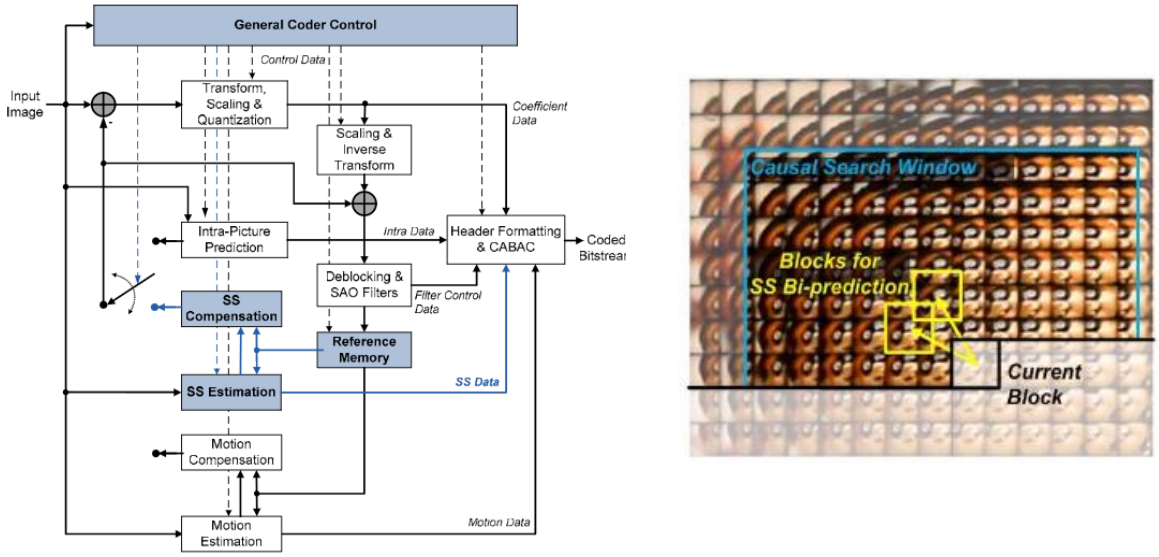


Figure 37: *left) Encoder architecture with the new modules required for SS based prediction [70]; right) Illustration of the chosen prediction blocks in the search window to perform the Bi-SS estimation process [70].*

It is also important to stress that the novel BI-SS enabled coding solution works both for single images and temporal sequences of images, this means light field videos. The additional SS-based coding modes are Intra coding modes and are added to the panoply of HEVC standard coding modes; the encoder should choose the best among SS-based and regular HEVC Intra prediction modes using a rate-distortion optimization (RDO) process [70].

Main Tools

The main tools, SS estimation and SS compensation, will be here presented together for each of the three novel SS-based prediction modes. To allow the new Intra prediction modes to be used together with the regular HEVC Intra coding modes, the I-slice definition was amended; the new syntax elements are almost the same as for the definition of HEVC P-slice coding modes as a vector and its residue are involved. Also, the entropy coding process was adapted from the Inter P-slices [72].

A. Baseline SS Prediction Mode (SS)

The baseline **SS prediction mode** has an impact on the SS estimation and SS compensation tools; the ***self-similarity estimation*** process adapts the search range depending on the depth of the current CU in the recursive splitting process, and proceeds as follows [72]:

1. The splitting recursion starts with the maximum CU size (depth zero) in inward direction.
2. Until the maximum depth is not reached for the first time, the search area is restricted to the previous coded CU (with maximum size) in raster scan order, as shown in Figure 38 left).
3. When a CU is coded for the first time in the maximum depth, the recursion process goes in the outward direction and a new area is added to the search area; this can be seen in Figure 38 right), and corresponds to the previously coded CU, in Z-scan order, for the same depth of current CU.

Figure 38*: Recursion in: left) Inward direction [70]; right) Outward direction, corresponds to the Z-scan order indicated [70].*

In the ***self-similarity compensation*** process, for each prediction unit (PU) partition pattern, one or more SS vectors are derived. In addition, an adapted scheme based on Advanced Motion Vector Estimation (AMVP) is used to select, in a rate-distortion optimization (RDO) sense, a self-similarity prediction vector by checking only the availability of spatially neighboring PUs [72]. As defined in the HEVC standard [73], AMVP enables the construction of a candidate list for each motion vector; this list includes motion vectors of neighboring blocks with the same frame reference index as well as a temporally predicted motion vector.

B. Baseline SS-Skip Mode (SS-SKIP)

The baseline **SS-Skip mode** does not relies on the same **SS estimation** process as the described above for the baseline SS prediction mode. The **SS compensation** process is different as it involves evaluating a Modified-Merge technique to infer a self-similarity vector to the current PU with all the available spatially neighboring candidates stored in a candidate list [71]. This allows sending only an

index to indicate the selected prediction candidate as the motion parameters of the chosen candidate are copied to the current CU [72]. This Modified-Merge scheme sets the limits of the SS vector candidates to only the spatially neighboring PUs, thus implying that the merge candidates from the co-located blocks cannot be used, as they do not exist in the SS reference area; moreover, the position of the vector is limited so that it points inside the limited area defined in the SS estimation process, which depends on the depth of the current CU [72]. This guarantees that the CU identified by the selected SS vector is already available as a SS reference.

C. SS Extended Mode also Bi-Prediction SS Mode

The **BI-SS prediction mode** is based on the same SS estimation and compensation processes as described for the previous prediction modes but with some improvements. Now, the **SS estimation** process considers two prediction candidates to build the final prediction for the current block. The first candidate predictor is given by the most similar block, inside the search window, thus better matching the current block; the second candidate predictor is chosen by jointly searching for the best linear combination between the first candidate prediction block and a second candidate prediction block, using the same search window [70]. The linear combination is made using ½ weights for both prediction blocks using the algorithm proposed in [74]. In the **SS compensation** module, the two selected prediction blocks are then used to obtain the final bi-prediction block which corresponds to the average of the two selected prediction blocks [70].

Performance Evaluation

For performance assessment purposes, twelve light field images were used, corresponding to the images provided for the ICME 2016 Grand Challenge on Light Field Coding [36]. Three benchmarks solutions were adopted and compared with the BI-SS-based coding solution which includes the three SS-based prediction modes defined above: i) JPEG represented by the viewpoints images rendered from the JPEG decoded image [36] ; ii) HEVC, where the test images were encoded with HEVC using the Main Still Picture profile; iii) Uni-SS, where the test images were encoded using HEVC extended only with the baseline SS prediction modes, meaning that the Bi-SS prediction mode was not used. Four compression ratios were considered: 10, 20, 40 and 100; to ensure a fair subjective quality comparison with the anchor JPEG results, the target coding bits were considered to be the maximum allowed to encode the test images with the BI-SS solution [70]. Table 1 shows the performance assessment in terms of the Bjøntegaard Delta (BD) Rate (BD-Rate) and PSNR (BD-PSNR) for two quality metrics, notably the mean YUV PSNR and the mean Y PSNR for all rendered viewpoint images. By observing Table 1, it is possible to conclude that the proposed Bi-SS based coding solution outperforms all the other alternative coding solutions tested. Comparing the Bi-SS and JPEG RD performances, BD-PSNR gains up to 8.21 dB (maximum), which corresponds to BD-Rate saving of 80.46%, were achieved; in average, the Bi-SS solution has BD-Rate savings of 70.81% and BD-PSNR gains of 5.55 dB. Comparing with HEVC, BD-PSNR gains up to 1.82 dB (maximum), which corresponds to BD-Rate saving of 44.88%, were observed; in average, the Bi-SS solution has BD-Rate savings of -36.36% and BD-PSNR gains of 0.99 dB. Even when compared to baseline SS-based coding solution, there are BD-PSNR gains up to 0.67 dB (maximum), corresponding to BD-rate savings up 39.14% [70]. The mentioned results were achieved taking the mean Y as metric, for the mean YUV the conclusions are the same; Table 1 also shows the average gains in PSNR and BR (bitrate) for each tested solution.

Table 1: *BD-Rate and BD-PSNR performance regarding the Bi-SS solution for two quality metrics: left) Mean YUV PSNR [70]; right) Mean Y PSNR [70].*

Image ID	*JPEG*		*HEVC*		*LFC (Uni-SS)*	
	PSNR [dB]	BR [%]	PSNR [dB]	BR [%]	PSNR [dB]	BR [%]
I01	5.13	-66.57	1.06	-29.89	0.38	-12.70
I02	5.48	-70.56	0.68	-20.82	0.25	-8.60
I03	4.40	-58.06	0.23	-6.83	0.09	-2.91
I04	5.18	-62.25	0.25	-8.93	0.05	-1.76
I05	4.22	-71.75	0.69	-35.28	0.29	-19.53
I06	5.49	-79.12	1.52	-59.99	0.63	-42.55
I07	4.47	-68.64	0.37	-15.75	0.12	-5.79
I08	4.26	-72.60	0.73	-37.62	0.46	-29.04
I09	5.88	-78.74	1.54	-45.26	0.34	-13.50
I10	4.00	-72.66	0.15	-11.69	0.05	-5.33
I11	6.19	-85.78	1.74	-66.22	0.53	-34.10
I12	7.15	-81.36	1.61	-53.07	0.42	-22.19
Average	**5.15**	**-72.34**	**0.88**	**-32.61**	**0.30**	**-16.50**

Image ID	*JPEG*		*HEVC*		*LFC (Uni-SS)*	
	PSNR [dB]	BR [%]	PSNR [dB]	BR [%]	PSNR [dB]	BR [%]
I01	5.65	-65.67	1.21	-29.73	0.45	-13.08
I02	5.90	-69.22	0.79	-21.12	0.27	-8.06
I03	4.71	-55.18	0.25	-6.56	0.09	-2.49
I04	5.30	-59.83	0.32	-9.39	0.07	-2.10
I05	4.71	-70.43	0.76	-34.26	0.30	-18.21
I06	5.79	-77.41	1.71	-56.86	0.67	-39.14
I07	4.56	-68.32	0.37	-15.13	0.11	-4.97
I08	4.31	-69.83	0.80	-37.13	0.47	-27.78
I09	6.70	-77.11	1.82	-44.88	0.37	-12.66
I10	4.14	-71.12	0.17	-13.36	0.07	-7.23
I11	6.57	-85.17	1.88	-66.97	0.56	-34.17
I12	8.21	-80.46	1.80	-52.96	0.43	-20.89
Average	**5.55**	**-70.81**	**0.99**	**-32.36**	**0.32**	**-15.90**

2.4.3.2. HEVC-based Local Linear Embedding and Self-Similarity Compensation Prediction

Objective

In [71] a new prediction solution is proposed to exploit the non-local spatial redundancies in lenslet light fields by simultaneously using Local Linear Embedding (LLE) and Self-Similarity (SS) prediction modes; the solution is referred as a HEVC+LLE+SS solution [75], as also the regular HEVC prediction modes are simultaneously used. Both the new prediction modes improve the HEVC standard compression efficiency by exploiting further data correlations such as non-local spatial similarities, meaning spatial similarities not associated to the immediate spatial neighbors of the block under coding. Because the two novel approaches are conceptually different, there are situations in terms of coding rates, content types and optical setups, where one mode outperforms the other and vice-versa. Thus, to fulfill the objective to improve the overall HEVC compression efficiency both novel prediction modes are integrated in the same extended HEVC standard solution. The Self-Similarity concept has been already presented in Section 2.4.3.1. The LLE baseline solution [75] aims to provide better predictions by estimating a block through some of its best correlated neighboring patches.

Architecture and Walkthrough

The HEVC+LLE+SS coding architecture is similar to the one adopted for the SS-based coding solution presented in Section 2.4.3.1. Basically, they both extend the HEVC architecture, in this case by including two additional prediction modes based on the SS and LLE concepts. The SS prediction mode has already been defined in Section 2.4.3.1. Briefly, it is a prediction tool based on block matching applied to a defined window in the already decoded are of the image where the best match between the current CU and an already coded area of the image is signaled with a SS vector. Note that the SS prediction window is the same prediction window used for the novel LLE estimation. The LLE based prediction mode, detailed below, allows to estimate the current CB through a linear combination of the *k*-nearest neighbor patches, which must be located on a previously coded and decoded area of the image.

As both the LLE and SS baseline solutions have been already tested and have had their compression efficiency proven [75] [72], the main challenge lies here in the integration of both prediction modes in the HEVC Intra standard scheme. In this case, the LLE and SS modes compete, in terms of RD cost, with the HEVC's regular tools to exploit the spatial redundancy available in the light field image; the mode with the lowest coding cost should be selected to code the coding block

(CB). While the LLE mode is implemented as explained below, the SS mode corresponds to the baseline SS prediction mode detailed in Section 2.4.3.1. The SS and LLE prediction modes are different as LLE is based on implicit predictors while SS is based on explicit predictors. This means that while for the LLE case the decoder needs to repeat almost the same steps performed at the encoder, except for determining the optimal *k* value, for the SS case the decoder just applies the prediction vectors computed and sent by the encoder. In this context, it is expected that these prediction modes complement each other, thus improving the overall compression performance. [71].

Main Tool

As stated in [75], the **Local Linear Embedding prediction** mode is fundamentally an optimization with specific and appropriate constraints, able to map the high dimensional nonlinear light field data into a coordinate system of lower dimensionality. The basic idea is to predict the current coding block by using a linear combination of *k*-nearest neighbor (*k-NN*) patches (this means blocks), with these patches being fetched from a previously coded and reconstructed/decoded area of the image. Taking Figure 39 left) as reference, the LLE baseline prediction estimation proceeds as follows [75]:

- **Search Window Definition** – Define a causal search window, *W,* before the coding block for searching the *k-NN* patches that present the lowest matching error while using a template *C*.
- **Template Definition** – The template *C* in the already decoded area and located around the coding block is to be approximated by a linear combination of the *k* best template patches retrieved from the search process. To find the linear coefficients, LLE solves a least-squares optimization problem where the sum of the weighting coefficients must be equal to one; this ensures that the approximation of each data point lies in the linear subspace spanned by its nearest neighbors.
- **Prediction Block Estimation** – The block, *P,* to be coded is estimated with the same set of linear coefficients estimated for the template patches but now with the purpose of combining the square blocks associated to each previously identified *k*-nearest neighbor (*NN)* template patch.

A formal description of the LLE prediction mode may be found in [75]. The LLE-based prediction tool has been incorporated in the HEVC standard reference software by replacing some of the regular Intra prediction modes. More precisely, eight directional Intra prediction modes have been substituted, illustrated in Figure 39 right) by the dashed lines and bold numbers; the substituted modes are evenly spaced so that no prediction direction in particular is neglected. As using all the eight available *NN* patches may not be the most efficient option, the value of *k* selected is informed to the decoder by signaling one of the replaced directional modes as each one corresponds to a specific value of *k* [71]. The value of *k* is selected using RD optimization, implying that the encoder tests the eight possible values of *k* (1,…,8), and chooses the one producing the best block prediction result to be explicitly signaled to the decoder [71].

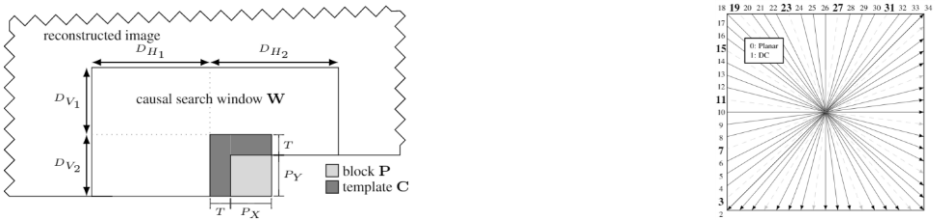


Figure 39: *left) LLE-based prediction components [75]; right) Set of 35 HEVC prediction modes, with the LLE modes represented [75].*

Performance Evaluation

The HEVC+LLE+SS coding solution has been assessed and compared with the JPEG and HEVC standards, as well as, with the baseline SS solution, denoted here as HEVC+SS. The solutions relying on HEVC used the reference software version HM-14.0. The adopted coding conditions were those defined for the ICME 2016 Grand Challenge [36]. Because no bitrate control algorithm was developed, the compression performance was assessed for several values of QPs where the resulting rate values closer to the target compression factors were chosen for the final results. To compare the several coding solutions, the Bjøntegaard Delta metrics (BD-Rate and BD-PSNR) were used. The decoded/reconstructed light field image was converted to the stack of associated SA images, this means a set of 2D low resolution RGB images. At this stage, each light field coding solution is compared with the reference coding solution using the PSNR-YUV average for all SA images. The results in Table 2, indicate the superiority of the HEVC+LLE+SS solution in terms of compression performance when compared to the other tested solutions, notably: i) average 71.44% bitrate savings and 4.73 dB of PSNR gains compared to JPEG; ii) average 16.87% bitrate savings and 0.33 dB of PSNR gains compared to HEVC+SS; iii) average 31.87% bitrate savings and 0.89 dB of PSNR gains compared to HEVC.

Table 2: *BD-Rate and BD-PSNR results for HEVC+LLE+SS solution compared with several alternative solutions.*

Benchmark	JPEG		HEVC		HEVC+SS	
BD	Rate	PSNR	Rate	PSNR	Rate	PSNR
I01	-63.97	4.76	-26.98	0.96	-9.56	0.28
I02	-69.78	5.14	-19.5	0.62	-7.25	0.21
I03	-57.93	4.15	-8.16	0.27	-4.39	0.14
I04	-61.96	4.61	-8.20	0.23	-1.01	0.03
I05	-71.46	3.69	-31.59	0.66	-16.72	0.26
I06	-78.63	5.03	-60.15	1.56	-42.63	0.66
I07	-67.11	4.19	-16.93	0.42	-8.84	0.19
I08	-69.89	3.82	-37.77	0.71	-30.15	0.48
I09	-79.49	5.50	-47.13	1.63	-17.52	0.44
I10	-70.09	3.56	-5.60	0.11	0.40	0.03
I11	-86.61	5.87	-68.34	1.86	-40.68	0.72
I12	-80.41	6.42	-52.05	1.60	-24.03	0.46
Average	**-71.44**	**4.73**	**-31.87**	**0.89**	**-16.87**	**0.33**

2.4.4. Non-standard Based Coding Solutions

This chapter purpose is to provide some knowledge on solutions which are not based in the previous presented standards. Two solutions will be detailed, both based on the wavelet transform.

2.4.4.1. 3D-DWT-Based Solution

Besides the many lenslet light field standard based coding solutions and extensions, there are also coding solutions which: i) perform the light field coding aided by depth information [76], targeting to improve the compression performance; ii) perform the light field coding using different coding architectures from the usual hybrid coding architecture adopted in all available video coding standards, e.g. based on wavelets. As depth data is not necessarily always available for all lenslet light fields, the depth-based coding solutions will not be considered here. Naturally, the objective of the non-standard based coding solutions is still to exploit the inherent correlations in the lenslet light field image although in a different, non-standard way.

Some of the few non-standard based coding solutions available in the literature have as cornerstone different transforms, notably the discrete wavelet transform (DWT) [77], discrete cosine transform (DCT) [78] and Karhunen-Loeve transform (KLT) [79]. These coding solutions use these transforms alone or even combined [80] [81]. In this section, two non-standard based coding solutions are presented, notably based on 3D-DWT [82] and 2D-DWT [78]; the performance assessment will be done by comparing these two coding approaches [82].

Objective

The main objective of the proposed coding solutions remains to find a better way to exploit the overall lenslet light field correlation, notably both the correlation within the SA images and the correlation across the SA images. Since no time is involved (as static images are the target of this Thesis), transforms are a very common and efficient way to exploit these spatial correlations. In the following, two different non-standard based coding solutions are briefly presented: i) a 2D-DWT based coding solution, which makes use of a 2D-DWT followed by a 3D-DCT [78]; and ii) a 3D-DWT based coding solution, which makes use of a 1D-WDT followed by a 2D-WDT [82].

Architecture and Walkthrough

Two coding approaches are presented below, one employing both DWT and DCT techniques and the other relying solely on a DWT based scheme. Each solution will be detailed separately, highlighting the main tools in which they are based; finally, they will be compared in terms of compression performance.

D. 2D-DWT Based Coding Solution

This coding approach is based on the use of a 2D-DWT followed by a 3D-DCT according to the architecture presented in Figure 40 left). The coding process proceeds as follows [78]:

1. **Preprocessing** – First, the SA or viewpoint (VP) images are created by picking a pixel in a certain position/perspective from each micro-image in the lenslet light field. The result is a set of SA images with the number of views corresponding to the number of pixels in each micro-image.
2. **2D-DWT on Viewpoint Images** – Next, each image is decomposed using a 2D-DWT transform with a certain number of decomposition levels, thus providing a set of bands for each viewpoint image from the lower frequencies up to the higher frequencies.
3. **3D-DCT on Lowest Bands** – As the lowest band of each viewpoint corresponds to a low resolution image and these images are correlated along the viewpoints, the lowest frequency bands are assembled together and an $8 \times 8 \times 8$ 3D-DCT transform is applied as shown in Figure 40 right).
4. **Quantization then Huffman Coding** – The coefficients resulting from the 3D-DCT are first quantized and then entropy encoded with a Huffman encoder. Figure 40 right), shows the scanning order used for the quantized 3D-DCT coefficients to create the symbols to be entropy coded.
5. **Quantization and Arithmetic Coding of Remaining Bands –** The other 2D-DWT bands are first quantized and then arithmetic encoded as not much correlation exist across the higher frequencies of the various viewpoint images.

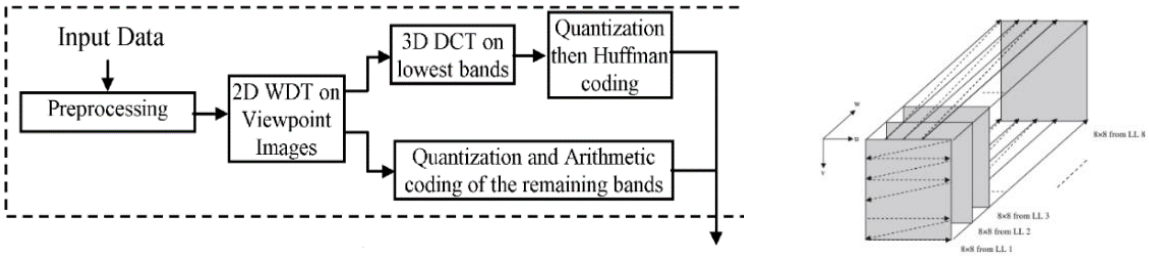


Figure 40: *left) 2D-DWT based encoding architecture [82]; right) Scanning order for the 8 × 8 × 8 coefficients in each 3D-DCT block [82].*

E. 3D-DWT Based Coding Solution

This coding approach is solely based on the use of the wavelet transform, applied more than once, thus achieving various frequency decomposition levels [82]. The 3D-DWT solution, illustrated in Figure 41, proceeds as follows:

1. **Viewpoint Images Extraction** – Extraction of the VP or SA images as described above, followed by a DC level shift to have a zero-averaged signal.
2. **1D-DWT over same pixel along the VP images** – Application of a 1D-DWT over the whole sequence of VP images for a specific pixel position; for *N* VP images (where *N* is the number of pixels in each micro-image), this process results in *N*/2 low-frequency and *N*/2 high-frequency bands.
3. **Recursive 1D-DWT Application** - Repetition of Step 2 on the obtained low-frequency bands until only two low frequencies bands are left.
4. **2D-DWT Application** – Next, a forward 2-levels 2D-DWT is applied on the last two low-frequency bands to exploit the correlation between these bands for all pixels of the VP images.
5. **Quantization and Arithmetic Coding** – Finally, quantization of all the bands, notably the 2D-DWT lower bands and the 1D-DWT higher bands is performed, followed by arithmetic coding.

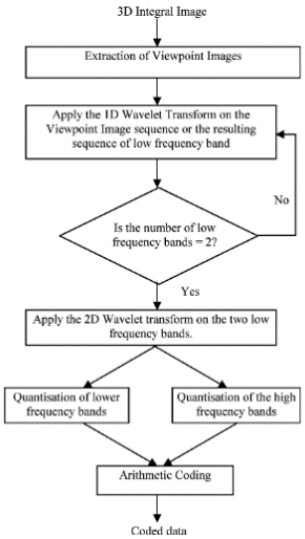


Figure 41: *3D-DWT based encoding fluxogram [82].*

Main Tools

A. 2D-DWT Based Coding Solution

The main innovation of this solution is that it is based on the successive application of a 2D-DWT and a 3D-DCT to exploit the light field image correlations; thus, these two techniques will be briefly detailed here [82]:

1. **2D-DWT** – The 2D-DWT transform is performed by applying two separate 1D-DWT transforms, notably along the columns and the rows of the VP image data; the transform is applied by using of a low-pass and a high-pass filter. The 2D-DWT application results into four frequency bands, which are each one quarter of the original size; Figure 42 left) shows the result of applying the 2D-DWT transform once while in Figure 42 right) the transform has been applied twice. The resulting lowest frequency band (LL) corresponds to a coarse scale approximation of the original VP image, basically a low resolution 2D image; the other bands (HL, LH, HH) contain detail in the horizontal, vertical and diagonal orientations, basically high frequency content. In this coding solution, the 2-levels decomposition was performed with a Cohen-Daubechies-Feauveau (CDF) 9/7 filter bank [78].

Figure 42: *Illustration of the 2D-DWT application on the VP image: left) 1-level [78]; right) 2-levels [82].*

2. **3D-DCT** – The application of the 3D-DCT has the objective of exploiting the correlation within and between the lowest frequency bands represented in Figure 43 left), each corresponding to a different low resolution VP image. The 3D-DCT is applied to an 8 × 8 × 8 volume, which means blocks of 8 × 8 pixels from 8 successive low resolution VP images as illustrated in Figure 43 right). The criterion to scan order the VP images to define the 8 VP images sets for the 3D-DCT should maximize the correlation between successive VP images.

Figure 43*: left) Illustration of the 2-levels 2D-DWT application on the VP image sequence [78]; right) Assembly process for the 3D-DCT, which consists in grouping the lower frequency bands into 8 × 8 × 8 blocks [78];*

B. 3D-DWT Based Coding Solution

This coding solution is based only on the DWT technique, namely a 1D-DWT and a 2D-DWT, and thus they will be briefly described here; as also the quantization and entropy coding steps show technical improvements, they are described too in the following:

a) **1D-DWT –** The VP images are ordered/organized using a raster-scan order and, after, all pixels in the VP images, with the same coordinates within the image, are arranged into a vector; the 1D-DWT transform is applied to all these vectors, thus exploiting the correlation along the VP images for each pixel position. The 1D-DWT is after recursively applied as depicted in Figure 44 left), for 5-levels of inter-view decomposition. The number of levels depends on the number of VP images and thus on the resolution of each micro-image.

b) **2D-DWT** – A 2-levels 2D-DWT is applied to the collection of the last two low-frequency bands for all positions in the VP images.

c) **Quantization and Entropy Coding** – At the end of the successive wavelet transforms, all the coefficients are quantized. The two lower frequency bands are quantized using a deadzone scalar quantizer while the remaining higher frequencies are quantized using a uniform scalar quantizer. The deadzone scalar quantizer enables to block the smallest coefficients, thus enabling longer runs of zeros enhancing the compression performance; as a high rate penalty is paid every time a run of zeros is broken. The scanning order used for all bands, prior to arithmetic coding, is illustrated in Figure 44 right) for the first high-frequency bands, which corresponds to the 28 high frequency bands shown in Figure 44 left).

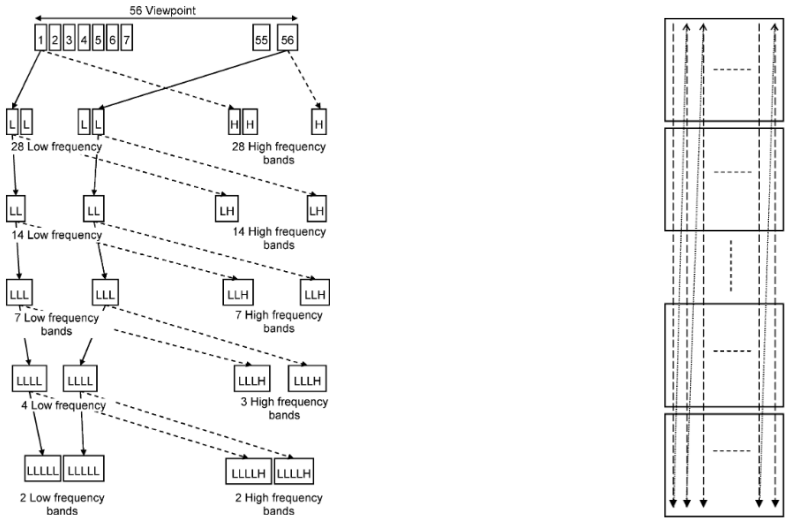


Figure 44: *left) Result from the recursive application of the 1D-DWT [82]; right) Scanning pattern for the sequence of bands [82].*

Performance Evaluation

In [82], both coding solutions described above have been tested and compared. Several images were used for the performance assessment, in this case captured using a lenslet light field camera system like the one depicted in Figure 45 left), which is able to capture lenslet light field images similar to those presented before in this Thesis. The RD performance was measured in terms of PSNR versus the number of bits per pixel (bpp). Figure 45 right) presents the RD performance using the average PSNR and average bpp for the data resulting from the set of selected light field images; these results show that the proposed 3D-DWT coding solution performs better than the previously available 2D-DWT coding solution.

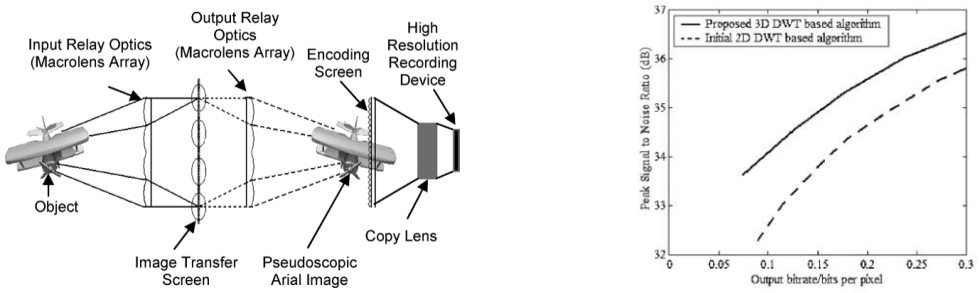


Figure 45: *left) Two-tier integral imaging system [82]; right) RD Performance for the 3D-DWT and the 2D-DWT coding solutions [82].*

2.4.4.2. Disparity-Compensated Lifting for Wavelet Compression of Light Fields

Objective

In [83], a disparity-compensated lifting technique is used in the context of a discrete wavelet transform (DWT) to exploit the correlation among the multiple SA views of a light field. The authors designed this solution for the compression of light fields, captured either by a 2D array of cameras or by cameras with an arbitrary spatial placement, e.g. distributed randomly on a hemisphere surrounding the object, aiming to provide the following features:

a) **Scalability** – View, spatial resolution and quality scalabilities are provided; this allows the implementation of light field systems able to adapt to different transmission bandwidths, display devices, storage capacities and computational resources [83].

b) **Improved Compression Performance** – The inter-view and spatial correlation is exploited by using the disparity-compensated lifting technique and the DWT (aided by the geometry model or 2D shape).

Architecture and Walkthrough

The encoder architecture proposed in [83] is shown in Figure 46. Since the 2D shape coding and 3D geometry model modules will be less relevant for the solution to be proposed in this Thesis, a shorter description is provided for these modules.

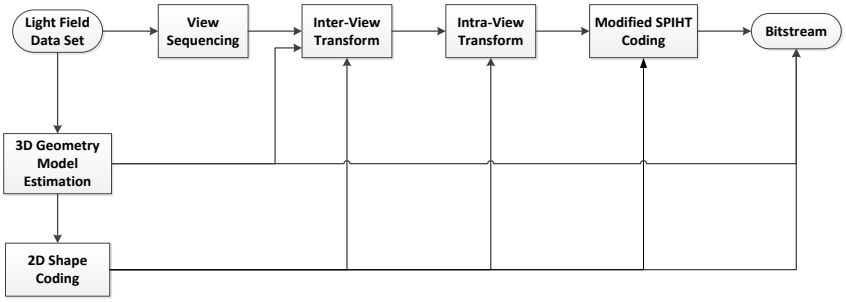


Figure 46: *Encoder architecture for the solution proposed in [83].*

The encoder walkthrough proceeds as follows:

- **3D Geometry Model Estimation** – The objective is to estimate a 3D geometry model from the light field data, from which disparity values can be derived and after used to perform disparity compensation. Thus, the 3D geometry model plays the role of a compact representation of the objects in the visual scene. The encoder transmits the geometry model to the decoder so that both encoder and decoder may derive the same disparity values when the inter-view transform is performed.

- **2D Shape Estimation** – The objective is to improve the compression performance and reconstruction quality at object boundaries, since it is assumed that the light field of interest captures a single object floating in space (background) and thus with sharp object discontinuities. The shape is obtained by segmentation and after exploited in the inter-view and intra-view transforms as well as in the SPIHT coding module. With this information, the disparity compensation in the inter-view transform module is improved at object boundaries; in the intra-view transform module, a shape-adaptive DWT (SA-DWT) is used while SPIHT was modified to disregard background pixels.
- **View Sequencing** – The objective is to arrange the light field data in a structure, namely a sequence of views, which is more appropriate to apply the inter-view transform. The algorithm orders the views in a sequence which should minimize the distance/difference between neighboring camera views.
- **Inter-View Transform** – The objective is to exploit the correlation among the multiple views both along the vertical and horizontal directions. This is performed by using a wavelet transform which is implemented by means of a lifting procedure, in this case a disparity-compensated lifting step. This means that disparity compensation is included when the inter-view transform is applied across the views in the light field data set to increase the de-correlation along the views.
- **Intra-View Transform** – The objective is to exploit the spatial correlation within all the frequency bands resulting from the inter-view transform, especially in the low-pass bands. A 2D-DWT transform or a SA-DWT, if 2D-shape is exploited, is applied to each frequency band to efficiently decorrelate the data inside the bands. The biorthogonal Cohen-Daubechies-Feauveau 9/7 was selected as the intra-view transform in [83]. Figure 47 shows the architecture for Haar DWT transform with 2-levels, where some low-frequency bands are further decomposed by a similar module, this allowing to obtain a coarser representation of the input images.

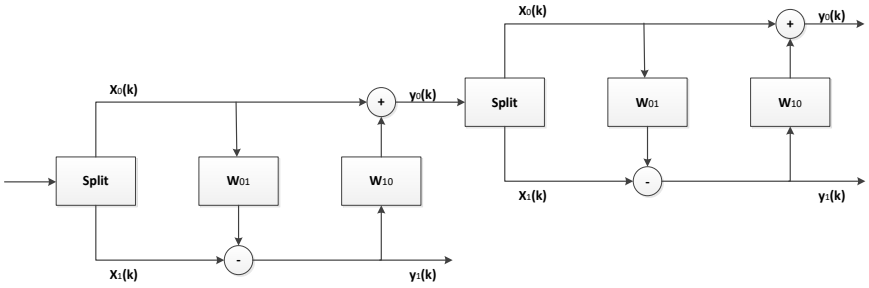


Figure 47: *2-level Haar wavelet transform architecture.*

- **Modified-SPITH Coding** – The objective is to entropy code the coefficients resulting from the intra-view transform. The SPITH algorithm, which exploits the inherent similarities across bands, was chosen in [83] due to its computational simplicity and compression efficiency. The coding algorithm was modified to operate in a block-wise manner, to disregard the zero-trees containing only background pixels, and to process each band separately.

Main Tools

The framework proposed in [83] mainly targets light field images for individual objects and thus there are extra background pixels and discontinuities at object boundaries. Due to their relevance for the objectives of this Thesis, the following main tools are described next with more detail:

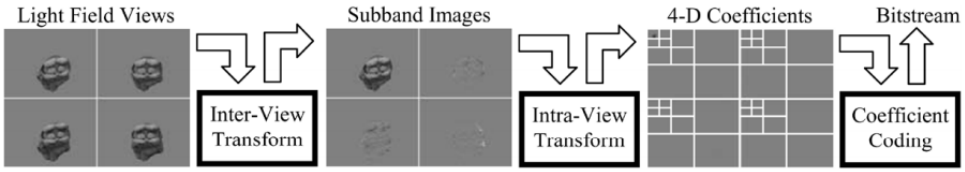


Figure 48: *Example coding walkthrough, taking as input four views, represented as a 2D-array of views.*

1. **View Sequencing** – Before applying the inter-view transform, the SA views are organized as a sequence of views, with a specific scanning order which should be compression efficiency friendly. In the case of a structured light field, as the example shown in Figure 48, the light field is captured by a 2D array of cameras, as such the data sets can be easily represented as a 2D array of views; in this case the columns and the rows naturally form the view sequence, thus the inter-view transform is applied horizontally and vertically. However, for unstructured light fields where cameras are not positioned on a regular grid, all views are rearranged such that neighboring views have high correlation and thus a wavelet transform can be applied to decorrelate the views more efficiently. Examples of both types of light fields used are depicted in Figure 49. Before applying view sequencing the views may be grouped into one or more clusters, by the *k*-means algorithm [84]. The view sequencing problem is cast as the travelling salesman problem (TSP), in which the objective is to find the cheapest closed path visiting a defined set of nodes. In this case, each node corresponds to a view and the cost of a path connecting two views is defined as the Euclidian distance between their corresponding camera centers. The TSP solution starts from a specific view and proceeds to visit every view exactly once, before returning to the initial view. If the views are grouped into clusters the view sequencing algorithm is applied independently to each cluster, hence reducing the complexity of TSP significantly.

Figure 49: *Example of the acquisition system of a: left) Unstructured light field [85]; right) Structured light field [85].*

2. **Inter-View Transform** – In [83], this corresponds to a discrete wavelet transform (DWT) implemented through a lifting step. For structured light fields where the views are organized in a grid structure, the inter-view transform is carried out by applying a 1D wavelet transform horizontally and vertically across the 2D array. For the unstructured case, a linear sequence of views is obtained using the previous step, this means the TSP algorithm. This inter-view transform includes disparity compensation in the lifting steps of the wavelet transform, i.e. a warping procedure is included to obtain an estimate for each view from another view. In [83], an explicit 3D geometry model is used to obtain the disparity values between any pair of views in the light field. The disparity is exploited in the DWT update and prediction steps, as shown in Figure 50 left); these steps are part of the wavelet transform, when a lifting structure is used. In practice, any type of DWT can be used as all of them can be factorized into lifting steps. To perform the disparity-compensated lifting, two warping

functions are needed, namely $w_{01}^{(k)}$ and $w_{10}^{(k)}$. The first function warps an even view (denoted as $x_0(k)$) or a low-pass band image (denoted as $y_0(k)$), for a decomposition higher than level 1, while the second function warps an odd view (denoted as $x_1(k)$) or a high-pass frequency band (denoted as $y_1(k)$). This procedure is detailed below for the Haar transform, as this is the simplest one to understand and implement:

I. **Split** – The objective is to divide the input sequence into two signals. The input signal must be a sequence with a maximum length of 2^j because the Haar DWT requires two images to generate two bands. The split process divides the data into two sequences, the even and odd views, each with length 2^{j-1}. The **Merge** step, used to perform the inverse wavelet transform, is the exact opposite of the split step.

II. **Warping (w_{01})** – The objective is to warp the even view in such way it resembles the odd view, this is accomplished by considering disparity information and corresponds to the predict step. After, the difference between the odd view and its prediction, a high-pass band $y_1(k)$ is computed and stored. The high-pass band is computed as:

$$y_1(k) = x_1(k) - w_{01}\big(x_0(k)\big) \tag{1}$$

III. **Inverse Warping (w_{10})** – The objective is to obtain a smoother version of the signal. This corresponds to obtaining a low-frequency band, $y_0(k)$. For this, it is necessary to warp the residual obtained in the previous step to have it aligned with the even view. This warped residual is added to the input even view, according to:

$$y_0(k) = x_0(k) + \frac{1}{2} w_{10}(y_1(k)) \tag{2}$$

3. **Intra-View Transform** – The objective of the 2D-DWT is to exploit the spatial correlation among neighboring pixels within each band image. The transform is applied separately to the X- and Y-axis of each band image. In the proposed solution, the biorthogonal Cohen-Daubechies-Feauveau 9/7 transform is employed. This 2D-DWT follows the same principles as explained in Section 2.4.4.1.
4. **Modified-SPIHT Coding** – SPIHT is an image entropy coding algorithm which objective is to exploit the redundancies across bands in a wavelet image decomposition. This method codes the most important transform coefficients first. The SPIHT encoder has been modified to regroup the DWT coefficients in each frequency band, as depicted in Figure 50 right), into single blocks and encode them separately, thus generating the corresponding bitstream. Although the coefficients are coded together within each block, the intra-view wavelet transform is applied to the entire image. The two main advantages of the block-wise SPIHT encoder are: i) lower memory requirements; ii) easy random access to any part of the image without having to decode it entirely.

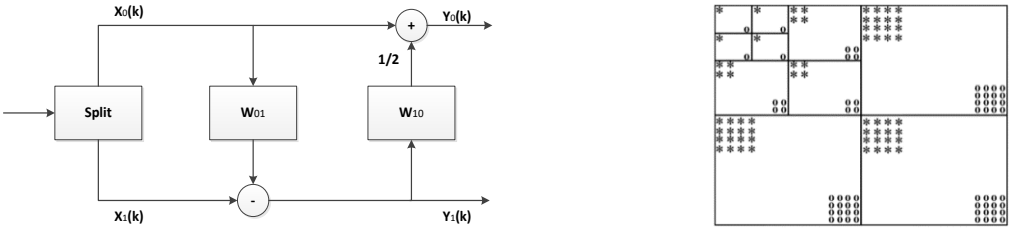


Figure 50: *left) Haar DWT architecture using a lifting scheme structure [86]**;** right) Process of data organization prior to block-wise SPIHT coding: the coefficients represented by * are grouped into one block, and those represented by 0 are grouped into another [1].*

The codec proposed in [83] allows obtaining a less rich representation of a light field depending on how much information has been coded. Due to the usage of the DWT, view, spatial and quality scalabilities are offered; the first results from the inter-view transform applied to a sequence of views; the second results from the intra-view transform applied to each band obtained from the inter-view transform. As SPIHT was designed to support quality scalability, the most important coefficients are sent first.

Performance Evaluation

Two types of light field data sets were used to evaluate the proposed light field codec with disparity-compensated lifting. The two data sets had the following characteristics:

1. **Regular** light field data sets: consists of two data sets named *Garfield* and *Penguin* where each data set has a hemispherical view arrangement with eight latitudes each containing 32 views, each with a resolution of 288×384 pixels. The data sets were structured as a regular 2D array with 8×32 views, each view with 288×384 pixels.
2. **Unstructured** light field data sets: consists of two data sets named *Bust and Buda.* With the data captured by cameras disposed in an irregular grid, for example having a denser sampling of views in a particular area of the scene; the data set includes 339 views for *Bust* (a real-world object), each with a 480×768 resolution, o and 281 views for *Buda* (a computer-generated object), each with a 512×512 resolution.

For the regular light fields, an approximate geometry model with 2048 triangles was reconstructed from the views using the method described in [87]. For the unstructured light fields, the geometry model and the camera parameters were estimated from the camera views using the method in [88]. Since several tools has been proposed, the performance results are detailed separately to better understand the performance impact of each tool.

View sequencing

The proposed method for organizing the views (TSP) has been compared with the heuristic method proposed [83], both using a 2-level inter-view transform. Regarding the TSP method, two alternatives were tested: i) using a single cluster, which means all data was organized together; and ii) using five clusters, which means the data was organized into five different groups before TSP application. As can be seen in Figure 51 mid), the multiple-cluster case introduces a slight RD performance degradation compared to the one-cluster case; this is because the correlation across the clusters is not exploited for the multi-cluster case. Despite the RD performance degradation, it is still advantageous to choose the multi-cluster case due to the reduced view sequencing complexity and more efficient data access.

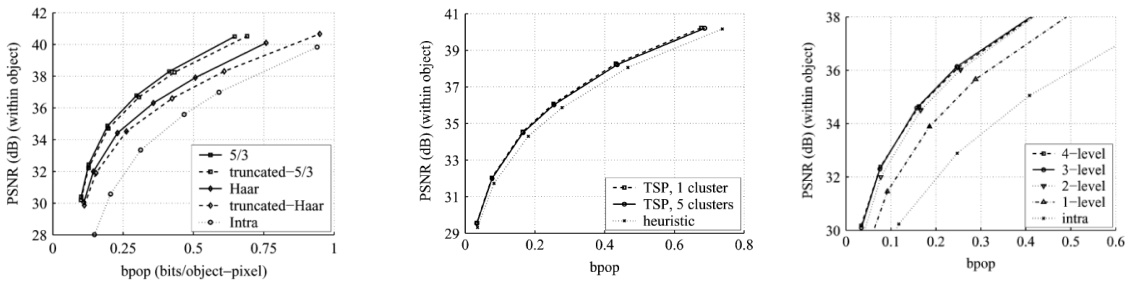


Figure 51: *left) Rate-PSNR performance for the Garfield data set for various 1-level, wavelet transforms [83]; mid) Rate-PSNR performance for the Bust data set for various view sequencing methods [83] ; right) Rate-PNSR performance for the Bust data set for various levels of 5/3 wavelet decomposition [83].*

Inter-View transform

Four different inter-view wavelet types were compared to the plain intra-coding scheme, where no inter-view transform is applied, namely: i) Haar; ii) 5/3; iii) truncated-Haar; iv) truncated-5/3. The truncated transforms have their coefficient number reduced by keeping the coefficients with larger amplitude. For the inter-view transform with 1-level (inter-view transform applied horizontally and vertically), the results presented in Figure 51 left), obtained for the luminance component, allow deriving the following conclusions:

- The inter-view transform (independently of the type) always improves the compression performance compared to the plain intra-view transform.
- The truncated filters behave poorly when compared to their non-truncated counterparts; the authors claim that this is mostly due to the ghosting artifacts in the low-pass band image, resulting from the occasional failure of disparity-compensation.
- The 5/3 wavelet transform performs better than the Haar wavelet transform in terms of rate-distortion at the cost of increased complexity. The performance improvement is due to the bidirectional predict and update steps. The best transform was the 5/3 non-truncated DWT.

Multiple inter-view transform levels

The 5/3 DWT was evaluated using different levels of decomposition, notably one to 4-levels using only the luminance component. The experimental results obtained are shown in Figure 51 right) and allow deriving the following conclusions:

- There is a RD performance improvement when the 2-level decomposition is applied comparing to the 1-level decomposition.
- Beyond the 2-level decomposition, there are no main RD performance advantages since the RD performance gains are smaller.
- Above the 3-level decomposition, there is a clear RD performance degradation; this can be explained by observing that the neighboring views at 4-level decomposition are too far apart to allow for an efficient compression.

Comparison with existing techniques

For the *Garfield* and *Penguin* data sets, the proposed coding solution was compared with the shape-adaptive DCT codec proposed in [89] (SA-DCT), and the texture map codec proposed in [90]. The texture map codec estimates an approximate geometry model from the light field and warps after the global views onto a global texture map reference frame. The performance assessment was performed with the following parameters:

1) The images were represented in the YCrCb format, with the chrominance components down-sampled by a factor of 2 in each direction.
2) The inter-view transform was the 1-level 5/3 DWT. The intra-view transform applied was a 4-level DWT for the luminance and a 3-level DWT for the chrominance components.
3) The *bit-per-object-pixel* (bpop) results were obtained by dividing the total bitstream size for the three-color channels by the number of object pixels (corresponding to the luminance resolutions); the reconstruction PSNR was computed using only the luminance component.

From this experiment, it was concluded that the proposed coding solution outperforms the SA-DCT solution, as shown in Figure 52 left) and mid). The decoded light field does not show any blocking artifacts (as common in DCT based coding solutions) and provides several types of scalability. When compared to the texture map codec, the proposed solution felt behind only for the very low bitrate region due to the extra overhead associated to the shape data; for the remaining bit-rates, the proposed coding solution outperforms significantly the texture map solution. For the higher bit-rates, a gain of more than 6 dB was achieved.

For the *Buddha* and *Bust* data sets, the proposed coding solution was compared with the SA-DCT codec using the following conditions: i) Only the luminance component was coded; ii) Using a 2-level 5/3 DWT as the inter-view transform and a 5-level 5/3 DWT as the intra-view transform.

From this experiment, it was concluded that the proposed coding solution achieves better RD performance as shown in Figure 52 right). More precisely, the proposed codec outperforms the SA-DCT codec by 1.5–2 dB and around 2 dB for *Bust* and *Buda*, respectively (naturally for the same *bpop*).

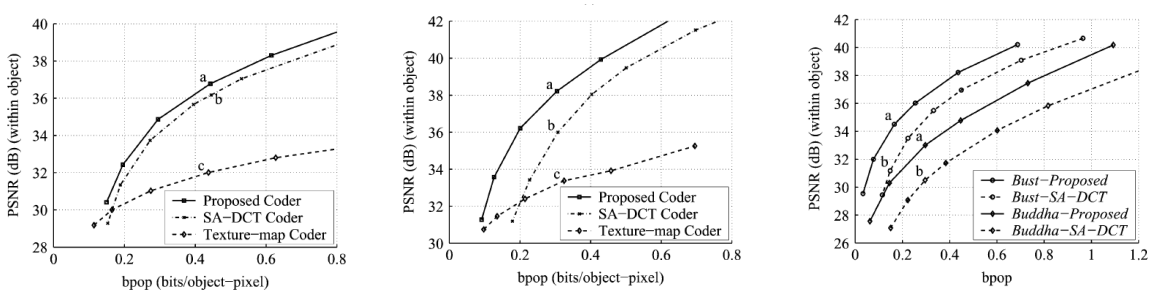


Figure 52: *Rate-PNSR performance comparison for the proposed, SA-DCT and texture-map codecs for: left) Garfield data set [83] ; mid) Penguin data set [83]; right) Bust and Buddha data sets [83].*

2.4.5. Light Field Coding Solutions Overview

Several different approaches for light field data coding have been studied, each offering different features and levels of compression efficiency. This section highlights the advantages and disadvantages of each light field coding category:

- **Standard Compliant Coding Solutions** – These coding solutions cannot exploit all available correlation, but benefit from the standard ecosystem as standard bitstreams and decoders can be used. From the reviewed literature, the standard solution with the best RD performance was HEVC Intra, followed by JPEG 2000, H.264/AVC and JPEG.
- **Standard Compliant Coding Solutions after Data Re-organization** – It was shown that the HEVC Intra RD performance was largely improved when using sub-aperture images organized as a video sequence or using some inter-prediction scheme. Thus, it is important to notice that these solutions are still not considering all the correlation of the light field data.
- **Extensions Based on Standard Coding Solutions** – These solutions can exploit the characteristics of the light field data as they can be applied to the raw image, to the lenslet image and to the SA image array. By including new tools into a standard solution architecture, it is possible to implement techniques specially designed for the light field data compression while still benefiting from the standard ecosystem.
- **Non-standard Based Coding Solutions** – In this last category there are many different types of architectures that were proposed. As these solutions are not based on standards they may not have

backward compatibility. In principle, these solutions may achieve high RD performance gains as they are developed solely for the compression of light field data. However, some of them lack maturity compared to the solutions of the previous categories.

Chapter 3

3. Proposing a Lenslet Light Field Coding Framework

In this chapter, a novel lenslet light field image coding solution is presented which specifically considers the unique characteristics of this type of data. Although the data acquired by a lenslet light field camera is very rich and thus 'heavy' as it captures the light from different directions, there is a significant amount of redundancy which may be exploited for coding purposes. The proposed coding solution is designed to compress light field images by first structuring the light field data into a set of SA images, each one from a slightly different perspective (i.e. capturing a different light direction). The different SA images are, in fact, highly correlated, similarly to the successive images of a video sequence. The proposed lenslet light field coding framework aims to provide two main features:

1. **Scalability**, notably view scalability, reconstruction quality scalability, and possibly spatial resolution scalability (for each view); all these scalability types result from the use of two discrete wavelet transforms applied in different ways.
2. **Compression efficiency**, by exploiting the correlation between and within the multiple SA images. This can be achieved with a lifting based wavelet transform coupled with efficient disparity compensation techniques.

The proposed coding framework may be applied with several different configurations, e.g. different decomposition levels, to provide different trade-offs in terms of the granularity of the scalability, and compression efficiency.

3.1. Discrete Wavelet Transform Basics

The discrete wavelet transform is used in the JPEG 2000 standard and exhibits notorious advantages when compared to the discrete cosine transform (DCT) used in the previous JPEG standard; namely the absence of "blocking" effects and the scalability it naturally provides [91]. The wavelet transform can be implemented using two different architectures: i) a **filter bank**; ii) a **lifting scheme,** also known as second generation wavelet transform.

Traditionally, the discrete wavelet transform is implemented using a series of filters [92], where the input signal is decomposed by passing through two filters: i) a low-pass filter, which strips the signal from its higher frequencies resulting in the so-called approximation coefficients, which are usually an averaged representation of the input signal; ii) a high-pass filter, which removes the low frequencies of the signal, yielding as output the details (high frequency) coefficients and once added to the approximate coefficients allow to recover the original input signal. This decomposition can be successively repeated to increase the frequency resolution, i.e. the approximation coefficients can be further decomposed with high and low-pass filters.

The alternative is to use a lifting scheme to implement the discrete wavelet transform as shown in Figure 53. This more recent way of implementing the wavelet transform has the advantages of

decreasing the amount of computation and offering a fast in-place calculation of the wavelet transforms, i.e. an implementation that does not require auxiliary memory [93].

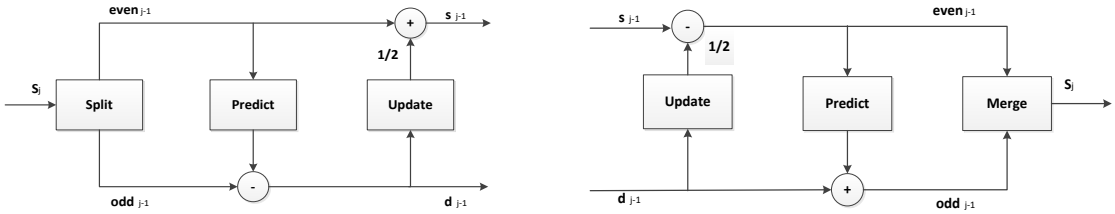


Figure 53: *Lifting structure for: left) Forward Haar wavelet transform [94]; right) Inverse Haar wavelet transform.*

The lifting scheme can be implemented for any wavelet transform family. The steps involved in implementing a Haar wavelet transform [94] with the lifting scheme are displayed in Figure 53. The steps involved are called Split, Merge, Predict, and Update (also called lifting step) and are described here for the Haar transform[1]:

1. **Split** – The objective of this step is to divide the input sequence of length 2^j samples into two sequences, even and odd, each with the same length 2^{j-1}; this is necessary because the Haar transform is performed by taking, as input, two samples at each time.
2. **Predict** – The objective of this step is to predict the odd samples with the even samples, and to store the difference between the odd element and its even "predictor" as $d_{-1} = odd_{j-1} - P(even_{j-1})$. The result is denoted as the **high-frequency band** and corresponds to the details in the input signal, d_{j-1}.
3. **Update** –The objective of this step is to restore the correct running average of the signal s_{j-1}, i.e. the coarser signal should have the same average as the original signal, therefore providing a smoother input for the next level of wavelet transform decomposition. The even sample is replaced with an average of the two input samples, considering the differences just computed and not the odd samples, $s_{j-1} = even_{j-1} + \frac{1}{2}U(d_{j-1})$. The result is denoted as the **low-frequency band** and corresponds to a coarser representation of the input signal, s_j.

The inverse lifting scheme, shown in Figure 53 right), allows recovering the input signal. This inverse structure is trivial to obtain as the inverse transform is applied by reversing the lifting step order and flipping the sign of the arithmetic operations [95]. Note that the scheme is reversible as long as the update and predict steps are invertible. The inverse lifting scheme is implemented by the following steps:

1. **Update** – The objective is to retrieve the even sample, $even'_{j-1}$ by subtracting from the low-frequency band the updated value as $even'_{j-1} = s_{j-1} - \frac{1}{2}U(d_{j-1})$.
2. **Predict** – The objective is here to retrieve the odd sample, odd'_{j-1}; this is done by adding the details to the prediction value as $odd'_{j-1} = P\left(even_{j-1}\right) + d_{j-1}$.
3. **Merge** – The objective of this step is to combine the even and odd samples, finally obtaining the original input signal.

The above steps are used to implement a 1-level Haar wavelet transform. To further decompose the signal, it is necessary to feed the low-frequency band (or in some cases high-frequency band)

[1] The Haar transform was invented by Alfréd Haar and it is the simplest of the wavelet transforms. For every pair of input samples, it relies on averaging and differencing the values to exploit the correlation between samples; resulting in a low-frequency band (average) and a high-frequency band (differences).

coefficients, obtained from the previous decomposition, to the same Haar wavelet transform as shown in Figure 54 using the low-frequency bands.

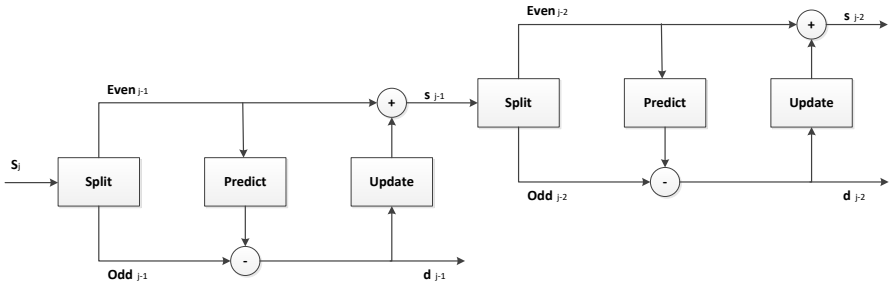


Figure 54: *2-level Haar lifting based wavelet transform applied to the first level low-frequency bands.*

3.2. Disparity Compensated Light Field Coding Architecture

The proposed coding solution estimates and compensates the disparity between the SA images of the 4D light field using a lifting-based wavelet decomposition scheme and, therefore, can exploit the correlation between the samples capturing the light for different directions. This coding solution is inspired in [83], namely on the way the disparity compensation is integrated into the lifting-based wavelet transform. The proposed coding solution is able to provide **view scalability**, which means that from the same bitstream, it is possible to extract some partial information and decode it so that a subset of the views of the entire light-field is obtained. When more information is decoded, more views can be obtained at the decoder side, eventually up to the entire number of views in the light field. This type of scalability is possible when a wavelet transform is applied across the entire set of views, treating each SA image as samples for the lifting based transform, thus obtaining low-frequency and high-frequency bands and, therefore, obtaining view scalability. Low-frequency bands may represent some views that when combined with the information from the high-frequency bands allow reconstructing the remaining views (for some decomposition level).

The proposed coding architecture, shown in Figure 55, aims to reuse as much as possible available coding tools, now organized to build an efficient and scalable lenslet light field coding solution. For example, the previously introduced Light Field Toolbox [33] is used to pre-process the lenslet light field image data, i.e. to obtain the SA images in the RGB color space, which are then converted to the YCrCb color space. Some architectural modules are based on the JPEG 2000 codec (highlighted in red in Figure 55, such as the 2D-DWT Intra-View Transform, the Uniform Scalar Quantization and finally, the EBCOT Coding. The Disparity Compensated Inter-View DWT corresponds to the main novelty of this solution, where the redundant content between SA images is exploited while targeting offering view scalability.

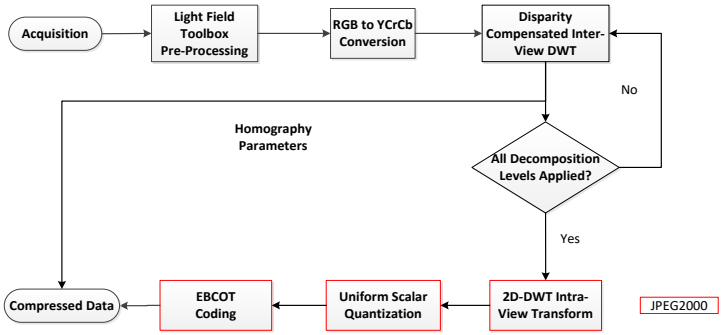
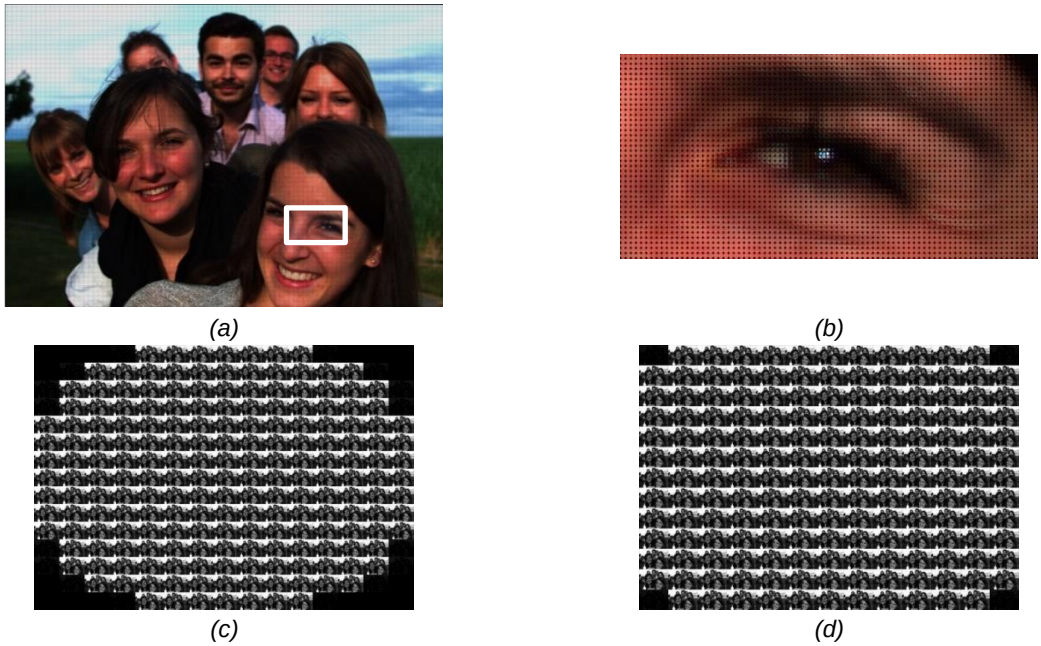


Figure 55: *Architecture of the disparity compensated light field encoder.*

A brief description of each module is presented next:

1. **Light Field Toolbox Pre-Processing:** The objective of this module is to convert the light field obtained directly from the sensor into a more suitable representation format. First, the so-called lenslet image is created from the raw sensor data by applying demosaicing, devignetting, clipping, and some color processing. Then, the lenslet image, formed by thousands of micro-images, is converted into an array of SA images, each representing a different perspective view, as shown in Figure 56. This module uses the available Light Field Toolbox v0.4 software [33]. While the original light field is composed by 225 SA images ($15 \times 15 = 225$), it was decided to discard both the first and last row and column of SA images, resulting into 169 SA images ($13 \times 13 = 169$), to avoid using SA images without enough quality, notably some black images that are obtained in the corners due to the vignetting effect (see Figure 56). This strategy has been also adopted by JPEG in the JPEG PLENO Call for Proposals [5]. This cropping process corresponds to extract just 13×13 pixels from each microlens, instead of the available 15×15 pixels.



(a) *(b)* *(c)* *(d)*

Figure 56: *Friends_1 light field image structured as: a) Lenslet image; b) Zoom of the white square, showing the micro-image under each microlens; c) 15×15 SA images, just Y component, obtained after processing with the Light Field Toolbox and RGB to YCrCb conversion; d) 13×13 SA images, just Y component, used for coding.*

2. **RGB to YCrCb Conversion:** The objective of this module is to improve the compression efficiency by converting the RGB data into YCrCb data which is a more compression friendly format as different sampling density may be used for the chrominances and luminance. Thus, at this stage, the SA images are converted from the RGB to the YCrCb color space.
3. **Disparity Compensated Inter-View DWT:** An inter-view wavelet transform is chosen to decorrelate the various SA images and compact their energy into a small number of bands. As the content captured by a lenslet light field camera exhibits a lot of similarities, it is proposed to enhance the lifting based wavelet transform performance by using disparity information in a similar way to the use of motion information in video compression; with this purpose in mind, a wavelet transform with disparity compensated lifting is adopted [83].The overall objective of the designed transform is to obtain low-frequency and high-frequency bands in such a way that the low-frequency band corresponds to a smoothed representation of a view and the high-frequency band corresponds to high frequency

information necessary to obtain the other view. The wavelet transform with disparity compensated lifting is applied to an array of SA images with size *N* and its frequency decomposition capabilities lead to *N*/2 low-frequency bands and *N*/2 high-frequency bands, as shown in Figure 57. To further exploit the correlation between the low-frequency or high-frequency bands, the wavelet transform can be used again in a second decomposition level, using now as input the low-frequency or high-frequency bands as already illustrated in Figure 54 for the low-frequency bands. With the application of 1-level decomposition transform, two scalability layers are available, the first associated to the low-frequency bands and the second associated to the high-frequency bands which correspond to details; for each decomposition level added, one more scalability layer becomes available.

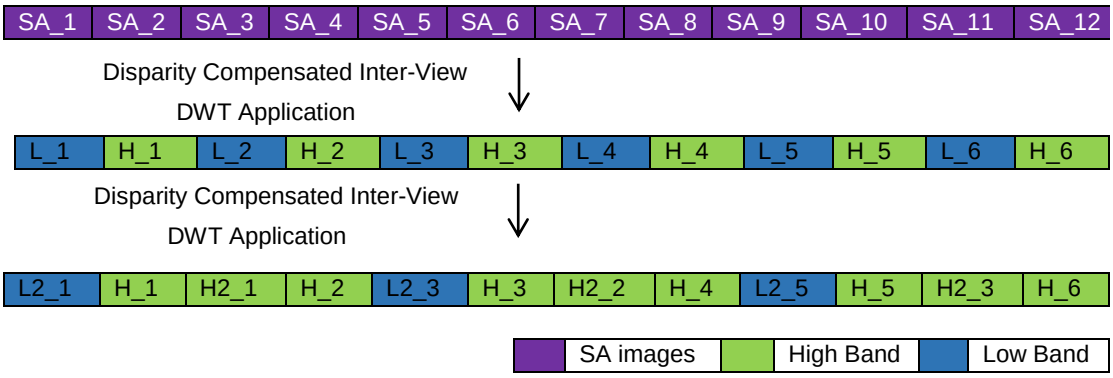


Figure 57: *Example of applying 1-level and 2-level disparity compensated inter-view DWT.*

The disparity compensated inter-view DWT exploits the correlation between the 4D array of SA images. However, the inter-view DWT transform can only receive as input a horizontal row or a vertical column of neighboring SA images. Thus, the transform can be applied first horizontally (row of horizontal neighboring SA images) and then vertically (column of neighboring SA images), to exploit both the correlation in the horizontal and vertical dimensions. Naturally, the opposite is also possible, this means applying the transform vertically and then horizontally. In Figure 58 left), the structure of the bands obtained from applying the inter-view transform horizontally with 1-level of decomposition is shown; moreover Figure 58 right) shows the structure resulting from applying the inter-view transform horizontally and then vertically to the low-frequency bands (2-level decomposition). In this figure, the following naming methodology is followed:

i. L_r and H_r are the bands obtained from a 1-level inter–view transform applied horizontally.

ii. L_rL_c and L_rH_c are the bands obtained from a 1-level inter–view transform applied vertically to low-frequency bands.

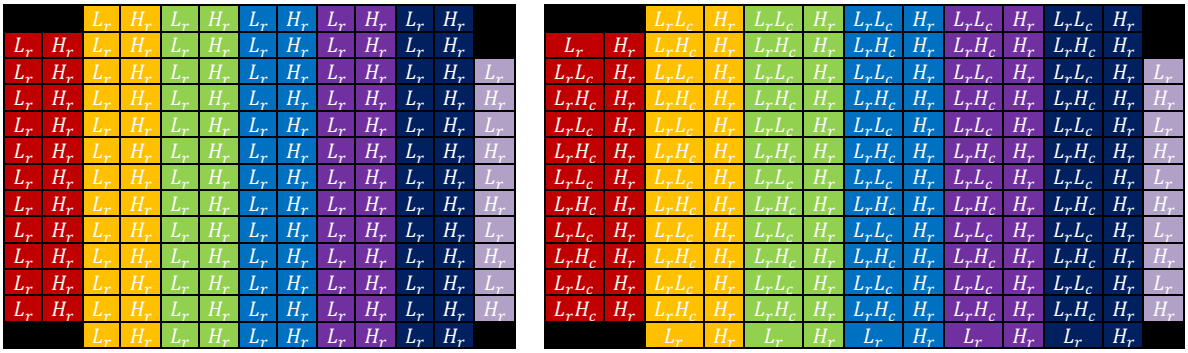


Figure 58: *Output of the Inter-view DWT transform with: left) 1-level decomposition: transform applied to the rows of the SA images; right) 2-level decomposition: transform applied to the rows of SA images array and after to the columns of low-frequency bands.*

Note that the disparity compensated inter-view transform is not applied to the black SA at the corners as there is not enough information available and such SA images are directly forwarded to the Intra-View 2D DWT module to exploit any available SA internal redundancy.

4. **Intra-View 2D-DWT:** The objective here is to exploit the spatial redundancy within each SA image or high-frequency/low-frequency band. The 2D-DWT transform with six decomposition levels, as proposed in the OPENJPEG software [96], has been adopted for application to all the frequency bands resulting from the inter-view transform. This process consists basically in applying a 1D-DWT along the X-axis (spatially horizontally) and, after, again along the Y-axis (spatially vertically) to each image/band. In the JPEG 2000 standard, typically four to eight decomposition levels are used [97]. The result of a 1-level 2D wavelet decomposition is four filtered and subsampled images, also known as bands. The application of the DWT JPEG 2000 transform will lead to the decomposition shown in Figure 59 up). JPEG 2000 can use two wavelet transforms, notably the Cohen–Daubechies–Feauveau (CDF) 9/7 irreversible wavelet transform and a rounded version of the CDF 5/3 wavelet transform. This last one uses integer coefficients and can be used for lossless coding [98]. The 2D-DWT enables resolution scalability as the SA images can be decoded at full resolution or only at a fraction resolution of it [83].For example, just decoding the LL3 band allows obtaining an image with a rather small resolution.

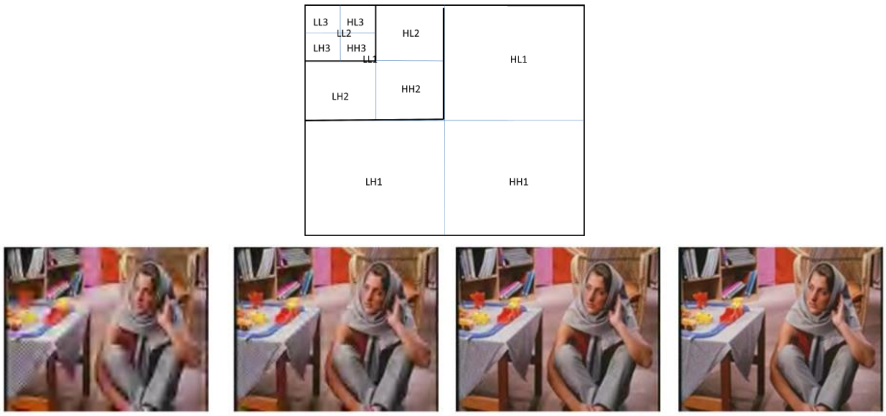


Figure 59: *up) 2D-DWT decomposition details [99] ; down) Example of quality scalability, the more bits received the better is the quality [99].*

5. **Quantization:** The objective is here to reduce the accuracy of the DWT coefficients to obtain higher compression. To reuse existing tools as much as possible, the quantization is performed using uniform scalar quantization with a dead-zone, which is one of the available JPEG 2000 quantization methods [100]. In JPEG 2000 Part 1, the deadzone size is twice larger than the quantizer step size while in JPEG 2000 Part 2, the deadzone size can be adjusted for each band. This quantization method allows also to progressively transmit the coefficients (quality or SNR scalability) by progressively sending the most significant bitplanes (MSB) and then advancing to the least significant bit (LSB) bitplanes; Figure 59 down) shows an example demonstrating the quality improvements when receiving more bits. All the bands obtained after applying the Intra-view 2D-DWT are quantized using this method [100].
6. **EBCOT:** EBCOT is the last step of the coding chain and has the objective to exploit the statistical redundancy (entropy coding) of the band coefficients. First, each band is divided into small rectangular blocks, referred to as codeblocks, and each codeblock is independently encoded with EBCOT. All codeblocks from low-frequency to high-frequency are scanned together from top to

bottom and left to right. Note that each band is independently coded from the other bands. EBCOT performs multiple-pass coding of the codeblock bitplanes obtained in the previous step. Three passes are used, notably significance propagation, magnitude refinement and cleanup; more details about each pass are available in [100]. For JPEG 2000 to be compression efficient, a context-based adaptive binary arithmetic coding method is used which exploits the correlation among bitplanes.

The code-stream organization of the proposed solution is compliant with the JPEG 2000 standard. For example, the first level of view scalability can be obtained with a standard JPEG 2000 decoder. Naturally, for the other layers, it is necessary to apply the inverse inter-view wavelet transform, but still, the high-frequency bands are just treated as images and a fully compliant JPEG 2000 bitstream is obtained. Therefore, it is useful to briefly describe the JPEG 2000 code-stream organization using Figure 60 as reference. The input image (YUV or RGB) is divided spatially into tiles (when using just one tile, the input image is processed as a whole). After the application of the wavelet transform, the coefficients are split into smaller units called precincts which correspond to rectangular blocks within a quantized band. Before being arranged into a stream, the precincts are split into codeblocks which are separately quantized and entropy coded, producing an elementary embedded bit-stream, which is further split into packets; finally, the packets from a band are collected into layers. Packets are a key feature from JPEG 2000 to provide quality scalability.

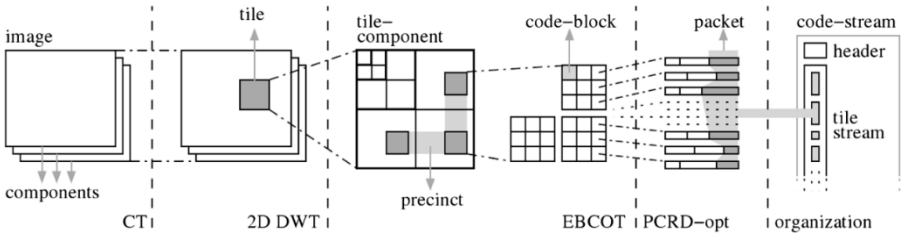


Figure 60: *JPEG 2000 framework detailing the coding units used [100].*

3.3. Inter-View Disparity Compensated Wavelet Transform

The inter-view disparity compensated wavelet transform proposed in this Thesis is the main novelty of the proposed coding solution. This transform was designed with a lifting structure to allow including disparity estimation and compensation techniques in the prediction and updating steps and, therefore, to exploit the inter-view correlation among the SA images of the light field. A simplified architecture of the forward transform is shown in Figure 61 which just includes an additional step (disparity estimation) and the predict and update steps that are designed to exploit the disparity between the SA images.

1. **Split**: The input, *I,* is a sequence of *N* SA images that are split into: a) *even* SA images and b) *odd* SA images. The next steps take place with an *even* and an o*dd* SA image, i.e. with just two SA images.
2. **Disparity estimation:** Compute a global view perspective geometric transformation matrix that describes the disparity from the *even* to the *odd* SA images and referred as w_{01}. The disparity from the *odd* to the *even* SA images is given by the inverse of the transformation matrix w_{01} and it is referred as w_{10}.
3. **Predict:** From the *even* SA image, the *odd* SA image is predicted using the geometric transformation matrix computed in the previous step, thus implementing a warping operation (or disparity compensation) that is referred as $w_{01}(even)$. At this stage, the **high-frequency** band can be computed as $d = odd - w_{01}(even)$.

4. **Update:** A warped version of the high-frequency band, d, referred as $w_{10}(d)$ is now used to update the even SA image. Thus, after the update step, the **low-frequency** band is computed as $s = even + \frac{1}{2}w_{10}(d)$.

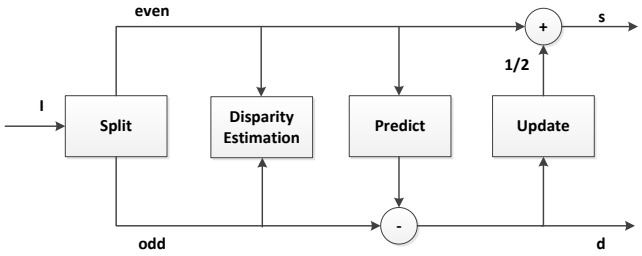


Figure 61: *Architecture for the 1-level Haar wavelet transform.*

The case previously described corresponds to a Haar DWT with 1-level of decomposition which performs disparity-compensation using a perspective geometric transform. A more detailed architecture detailing all main steps is shown in Figure 62.

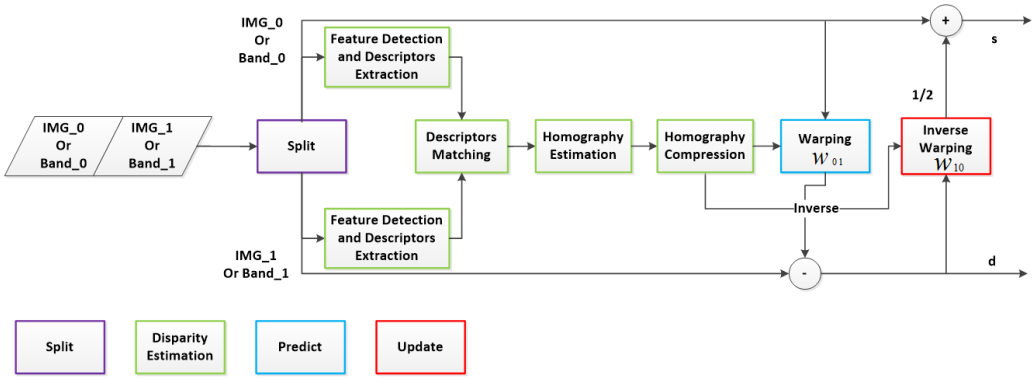


Figure 62: *Inter-view DWT applied to two SA images (IMG_0, IMG_1) or low-frequency/high-frequency bands (Band_0, Band_1); highlighting the relationship with the modules from Figure 61.*

As the light field data corresponds to a 13 × 13 array of SA images and the Haar DWT is applied to two images at a time, it is necessary to gather an even number of SA images to apply this transform. Moreover, this transform can be applied to a row or column of SA images in the first decomposition level or low-frequency/high-frequency bands (in the rows or columns) in the higher decomposition levels. In more detail, the Inter-View Disparity-Compensated DWT consists in the following steps, see Figure 62:

1. **Split:** Divides the input set of SA images (or bands) into two different, complementary sets where the even SA images are grouped into one set and the odd SA images (or bands) into another set.
2. **Feature Detection and Descriptor Extraction:** The objective of this module is to detect distinctive features in the images associated to keypoints or blobs, and extract after descriptors for those positions which represent the features in some space that is invariant to common deformations such as translation, scaling, rotation, perspective changes and partially invariant to illumination changes. In this case, it is proposed to use the popular SIFT (Scale Invariant Feature Transform) descriptor [101]. The SIFT descriptor uses spatial histograms of the image gradients to characterize the patch surrounding the detected keypoint, for any given SA image. The main steps of this module are described below [101]:

   i) **SIFT Feature Detection** – The feature detection or keypoint extraction can be divided in the following steps [102]: i) **Scale-space extrema detection:** first, the Gaussian scale-space representation of the image is obtained, which corresponds to a pyramid of images at different resolutions (scales) obtained by progressively smoothing and downsampling the input image.

Then, adjacent images in the pyramid are subtracted to obtain the Difference of Gaussians (DoG) from which maxima and minima can be found, see illustration in Figure 63. To identify potential interesting points invariant to scale and orientation, the DoG pyramid is searched over all scales and image locations; ii) **Keypoint localization:** In the previous step, potential keypoints are found, which need to be refined to get more accurate results. Based on measures of stability, undesirable feature points are removed, e.g. keypoints with low contrast and unstable [103]; iii) **Orientation assignment:** The last step is to compute the best orientation for each detected keypoint, in order to achieve invariance to image rotation, based on a histogram of local gradient directions at the specified scale.

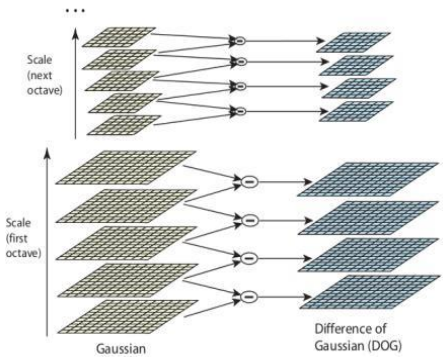


Figure 63: *Illustration of the Difference of Gaussians process for each scale [101].*

ii) **SIFT Descriptor Extraction** – At this stage, each keypoint is characterized with location, scale and orientation and, therefore, it is possible to compute a descriptor that characterizes the image region (patches) surrounding it. The descriptor extraction step can be divided into [102]: i) Normalization where the patch is appropriately rotated and scaled; ii) Keypoint description where the local image gradients are measured at the selected patch scale with the objective of computing the orientation histogram, as shown in Figure 64. Both the gradient magnitude and orientation at each position of the patch are considered and the gradient magnitudes are weighted by a Gaussian window.

Figure 64: *left) Gradient magnitude and orientation at each image sample in a region around the keypoint location; these are weighted by a Gaussian window, represented by the overlaid circle [102]; right) Orientation histograms summarizing the contents over 4x4 sub-regions [102] .*

3. **Descriptors Matching:** The objective here is to match individually a set of descriptors from one SA image with the descriptors from another SA image, obtaining a set of one-to-one correspondences. A simple approach has been followed consisting in taking each descriptor in the first set and matching it with all the descriptors in the second set, using some distance metric, e.g. the Euclidean distance. Then, a ratio test was applied. This test compares the ratio of distances between the two top matches for a given keypoint. If the ratio is above the threshold of 0.7, the match is rejected. The objective of this test is to increase the reliability of the matching procedure, thus avoiding some wrong matches between keypoints [102].

4. **Homography Estimation:** In this context, a homography is a geometric transformation which establishes a relationship between corresponding positions of two different, but somehow related, images; these correspondences were obtained in the previous step. Thus, at this stage, the objective of this module is to estimate the transformation between one SA image (or a low-frequency band) and the other. Several formulations for this transformation are possible, such as [104]: i) Affine Transform; ii) Perspective Transform; iii) Bilinear Transform; iv) Polynomial Transform. Considering the lenslet light field data characteristics, the most adequate transform for modelling the data seems to be the perspective transform since it is able to model complex geometry relationships between different perspectives of the objects in the visual scene and may support data acquired with different types of light-field cameras, e.g. cameras with different types of lenticular arrays. The perspective transform is defined by a 9-parameter (3×3) matrix which is able to describe the displacements that the objects in the visual scene suffer when the perspective changes. When applied to the case of SA images, this transform should describe well the disparity between the SA images (mainly determined by the characteristics of the microlens array). To estimate the perspective transform parameters, RANSAC, an iterative method to estimate parameters that is robust even when there are some wrong matches (outliers), is used. To avoid that outliers reduce the accuracy of the estimated transformation matrix, the method attempts to identify inliers, which is data that can be explained by a set of model parameters (typically estimated with a standard regression method) and, therefore, not considering outliers (erroneous correspondences) in the estimation.
5. **Homography Parameters Compression:** The parameters of the perspective transform (matrix values) are expressed with 8 bytes (64 bits), assuming double floating point precision. Since this precision may require a significant amount of rate, as these parameters have to be transmitted to the decoder, it is important to propose a quantization technique to compress this type of data. Note also that, each time the inter-view wavelet transform is applied, a different perspective transform matrix is used and, therefore, since there are many pairs of SA images (bands) for which a transform is applied, the number of matrices may be rather high. The technique proposed is implemented as described here:
    1. **Average matrix computation:** the first step is to compute an average matrix, $\mathrm{H}_{\mathrm{avg}}$; this is done using as input all perspective matrices for a given decomposition level, for each matrix parameter all values are summed and divided by the number of matrices.
    2. **Min and Max matrix computation:** after the minimum and maximum value of each matrix parameter is computed again using all matrices, thus obtaining $\mathrm{H}_{\mathrm{min}}$ and $\mathrm{H}_{\mathrm{max}}$. These last two values represent the dynamic range of the perspective parameters and, therefore, may be used to adjust the quantization step efficiently.
    3. **Residual matrix computation**: subtract the original matrix $\mathrm{H}_{\mathrm{org}}$ to be coded by the average matrix $\mathrm{H}_{\mathrm{avg}}$, as in (3); the purpose is to obtain residual parameter values close to zero.

        $$\mathrm{H}_{\mathrm{res}} = \mathrm{H}_{\mathrm{org}} - \mathrm{H}_{\mathrm{avg}} \quad (3)$$

    4. **Normalization**: the $\mathrm{H}_{\mathrm{res}}$ residual parameters are now normalized to the interval [0;1] using (4) and considering the auxiliary matrices computed in step 2; the purpose is to have a fixed dynamic range and, therefore, a more compression friendly representation of the perspective matrix, the normalized matrix $\mathrm{H}_{\mathrm{norm}}$.

$$\mathrm{H}_{\mathrm{norm}} = \frac{\mathrm{H}_{\mathrm{res}} - \mathrm{H}_{\mathrm{min}}}{\mathrm{H}_{\mathrm{max}} - \mathrm{H}_{\mathrm{min}}} \quad (4)$$

5. **Quantization**: to reduce the parameters precision, a quantization process is applied thus obtaining the quantized matrix $\mathrm{H}_{\mathrm{quant}}$ to be transmitted to the decoder. To use *N* bits (24 bits) for each parameter, it is necessary to apply (5) and (6). The value of *N* was obtained by testing different precisions values and selecting the one which resulted in a better RD performance.

$$\mathrm{H}_{\mathrm{quant}} = \mathrm{round}(\mathrm{H}_{\mathrm{norm}} * 2^{\mathrm{N}}) \quad (5)$$

$$round(x) = \begin{cases} 1 \; if \; x \geq 0{,}5 \\ 0 \; if \; x < 0{,}5 \end{cases} \quad (6)$$

The mentioned steps must be applied to each matrix obtained for a given decomposition level; the auxiliary matrices $\mathrm{H}_{\mathrm{avg}}, \mathrm{H}_{\mathrm{min}}$ and $\mathrm{H}_{\mathrm{max}}$ are also subjected to the same process of quantization, i.e. the step 5 mentioned above. The $\mathrm{H}_{\mathrm{quant}}$ matrix is transmitted to the decoder without any entropy coding, just using a fixed number of bits per parameter. Using this approach, where 24 bits are used to represent each homography parameter, it is possible to reduce the number of bits from the original 64 to 24, thus implying a compression factor of $\frac{64}{24} = 2{,}666(6)$) times.

The decoding process is straightforward after the auxiliary matrices $\mathrm{H}_{\mathrm{avg}}, \mathrm{H}_{\mathrm{min}}$ and $\mathrm{H}_{\mathrm{max}}$ are inverse quantized. The remaining matrices are obtained as follows:

a. **Inverse Quantization**: to obtain the approximate parameters, the quantization process is reversed, with *N* bits (24 bits), using (7):

$$\mathrm{H}_{\mathrm{quant}} = \frac{\mathrm{H}_{\mathrm{norm}}}{2^{\mathrm{N}}} \quad (7)$$

b. **De-normalization**: Parameter value dynamic range is recovered using (8):

$$\mathrm{H}_{\mathrm{de-norm}} = \mathrm{H}_{\mathrm{quant}} \times (\mathrm{H}_{\mathrm{max}} - \mathrm{H}_{\mathrm{min}}) + \mathrm{H}_{\mathrm{min}} \quad (8)$$

c. **Homography Matrix Computation**: the decoded residual matrix is obtained using (9):

$$\mathrm{H}_{\mathrm{dec}} = \mathrm{H}_{\mathrm{de-norm}} + \mathrm{H}_{\mathrm{avg}} \quad (9)$$

6. **Warping or Disparity Compensation (w_{01}):** The objective is to transform/warp an input even SA image in such a way that it becomes similar to the odd SA image, which in this case corresponds to a slightly different perspective, in practice performing disparity compensation. This warping process is performed by using the decoded transformation matrix, $\mathrm{H}_{\mathrm{dec}}$. Considering [*x*,*y*] the coordinates of a sample in the warped image and [*u*,*v*] the coordinates of the corresponding sample in the input image, the SA image prediction is computed by multiplying each sample position in the input image by the transformation matrix (a_{xy} coefficients, with $x, y \in [1; 3]$) and, finally, w is used to represent $[x, y]$ as a homogeneous vector $[xw, yw, w]$, as shown in (10). Homogeneous coordinates are a system of coordinates used in projective geometry with the advantage of being able to easily represent projective transformations as a matrix [104]. Computing the difference between an odd view and the warped even view results in the high-frequency band.

$$[xw, yw, w] = [u, v, 1] \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (10)$$

7. **Inverse Warping (w_{10}):** As the scene disparity involved in this kind of data mostly corresponds to translations, because it mostly due to the spatial separation between the microlenses and only a little due to optical defects in the microlens, the transformation matrix from a reference view into another

view can be inverted, and thus an inverse transformation matrix can be obtained. This is also a requirement from the disparity compensated wavelet transform which can only be applied when the warpings w_{01}.and w_{10}.are symmetric as otherwise the process may end up adding a residual to the odd view that is not aligned (prediction step), thus creating ghost artifacts. Therefore, the inverse warping is accomplished in the same way as the warping but now using the inverted transformation matrix as in (11). By computing a weighted sum between the even view and the warped high-frequency band, a low-frequency band (smoothed) SA image is obtained.

$$\mathrm{H_{dec}}^{-1} = \frac{1}{|\mathrm{H_{dec}}|}\begin{bmatrix} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{13} & a_{12} \\ a_{33} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{23} & a_{21} \\ a_{33} & a_{31} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{13} & a_{11} \\ a_{23} & a_{21} \end{vmatrix} \\ \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{12} & a_{11} \\ a_{32} & a_{31} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{bmatrix} \tag{11}$$

The inverse lifting scheme for the inverse disparity compensated wavelet transform to be performed at the decoder follows the scheme presented in Figure 65. As the homography matrix parameters computed in the homography estimation step are transmitted to the decoder, the inverse transform only needs to perform the predict and update steps in the reverse order flipping the signal in arithmetic operations, thus resulting in the original signal. As long as the warping is performed using the same homography matrix, and the bands suffer no compression the decoded even samples will the same as the original samples.

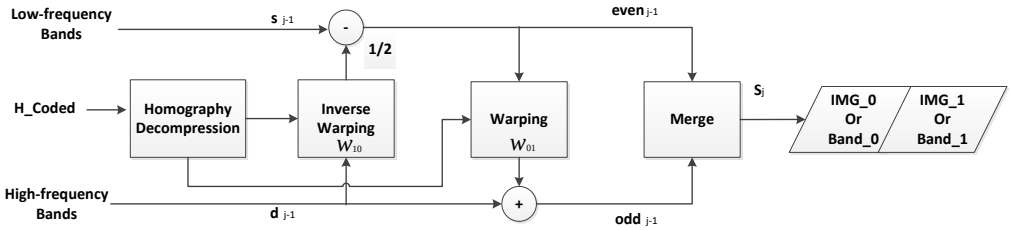


Figure 65: *Inverse inter-view DWT scheme.*

Regarding the software implementation, the inter-view DWT, with the several decomposition levels applied to different light field dimensions was implemented in this Thesis' context; as well as, the compression/decompression process used with homography parameters. The OpenCV software was used to implement some image processing tools, such as SIFT and homography computation and OPENJPEG software was used to implement the JPEG 2000 codec.

Chapter 4

4. Disparity Compensated Light Field Coding: Performance Assessment

The objective of this chapter is to assess the rate-distortion (RD) performance of the proposed coding solution, denominated as Disparity Compensated Light Field Coding (DCLFC). The performance will be compared against some relevant alternative coding solutions, notably still image coding standards, such as JPEG and JPEG 2000, and three coding solutions which were provided to JPEG as responses to the Call for Proposals on Light Field Coding in July 2017 [105] [106] [107]. This chapter will address the following items:

i) Performance of the proposed coding solution when applied only to horizontally or vertically neighboring views, which implies exploring the SA images correlation only in one dimension (1D).
ii) Performance of the proposed coding solution when applied to neighboring views, both horizontally and vertically, which implies exploring the SA images correlation in two dimensions (2D).
iii) Performance of the proposed coding solution against some relevant benchmarking solutions.
iv) Advantages of providing a quality scalable stream.
v) Additionally, the performance study of applying the inter-transform to both low-frequency and high-frequency bands can be found in Appendix B; moreover, the advantages of computing the homography parameters using original SA images or low-frequency bands can be found in Appendix C.

4.1. Test Material, Coding Conditions and Benchmarks

This section describes the adopted test material, the coding conditions and the benchmarks.

A. Test Material

To evaluate the RD performance of the DCLFC solution, five lenslet light field images have been selected from the MMSPG-EPFL Light Field Dataset [108]; this dataset has also been selected as the test set for the Light Field Compression Grand Challenge organized at ICME 2016 [36] and for the JPEG Pleno Call for Proposals on Light Field Coding [109]. The set of selected images is: i) *Bikes*; ii) *Danger_de_Mort*; iii) *Stone_Pillars_Outside*; iv) *Fountain_&_Vincent_*2; v) *Friends_1.* To simplify the text, from now on, the names will just be *Bikes*, *Danger*, *Stone*, *Fountain* and *Friends*. Figure 66 shows thumbnails of the selected light fields. The images were chosen by their content, aiming to have a diversified dataset, with both high and low frequency content and objects at different depths. All the test images were acquired using a Lytro Illum camera and, thus, are available in the LFR (light field raw) format. The software used to extract the light field data from the LFR file for further processing was the Light Field Toolbox, made available by D. Dansereau [33].

(a) *(b)* *(c)*

(d) *(e)*

***Figure 66**: Light field test images: (a) Bikes; (b) Danger; (c) Fountain; (d) Friends; (e) Stone.*

After demosaicing, color correction and rectification, the light field data is obtained as RGB images with an 8-bit unsigned integer representation. The light field output structure is a matrix of 15×15 SA images; however, for compression purposes, only 13×13 SA images, each with a spatial resolution of 625×434 pixels, are considered as the remaining border SA images have low quality, especially due to the vignetting effect. Because this is usually enough, the performance assessment will be made only for the luminance (Y) component of the SA images.

B. Benchmarks

All benchmarks consider the same set of 13×13 SA images, each with a resolution of 625×434 pixels. When coding with JPEG and JPEG 2000, the input SA images will be organized in two ways:

i) **SA images individually coded** – Each SA image is individually encoded and, thus, the RD points are defined by the total rate spent with the 13×13 SA images and the PSNR computed as the average of the PSNR for all these SA images.

ii) **SA images coded as a single "super image"** – Instead of encoding the 13×13 SA images one-by-one, the SA images are first arranged in a single "super image" which is then encoded all at once. The RD points are defined by the rate spent in the "super image" and the PSNR is computed as the average PSNR of all SA images extracted from the decoded "super image" (as for the previous case, the PSNR compares the decoded SA images with the original ones).

For JPEG, it is expected to have a very similar RD performance when coding the SA images individually and as a "super image"; as JPEG coding is DCT based, both the transform and quantization are applied to 8×8 samples blocks and, thus, there should be no substantial difference between the two situations. However, JPEG 2000 does not use the DCT, instead it relies on the DWT which is applied to the full input image, if using just one tile. As the 2D-DWT will be applied to different inputs and the x- and y-axis correlations to exploit different, the "super image" case may allow exploiting a bit more the redundancy and so might slightly improve the RD performance.

For the three solutions proposed to JPEG as responses to the Call for Proposals, follows bellow a brief description:

- **JPEG Pleno Proposal by Univ. of Science and Technology of China (labelled Pleno1)** [105] – The Pleno1 proposed codec is a lenslet light field image coding solution based on the division on the

SA into two sets, one coded with the HEVC standard and the other using linear view synthesis from the first set. The coding process begins by dividing the SA images stack into two groups, namely the S_A and S_B sets, in this case using an alternate checkerboard pattern. The S_A set is converted into a pseudo-sequence to be coded with HEVC. After each SA image in the S_B set is estimated as a weighted sum of the decoded S_A SA images using a set of coefficients that are quantized and coded. The residuals of the set S_B may be arranged again as a pseudo-sequence to be coded but the results for this proposal do not yet include any residual coding. The block diagrams representation of this codec is shown in Figure 67 up). In summary, the coded bitstream consists on a HEVC stream for the set S_A, and the transformation coefficients for the set S_B. This solution may deliver three layers of scalability: i) the first layer corresponds to the coded set S_A, which consists on a light field image sparsely sampled in the angular dimension; ii) the second layer correspond to adding the synthetized set S_B which offers angular resolution scalability also called view scalability; iii) the third layer (not used in the results) corresponds to adding the set S_B residues which offers quality scalability for these images.

- **JPEG Pleno Proposal by Univ. Tampere, Finland (labelled Pleno2)** [106] – The Pleno2 proposed codec performs the lossy compression of lenslet images by combining sparse predictive coding and JPEG 2000 to the stack of SA images. The first step is to code the entire lenslet image with JPEG 2000 at a fraction of the available bitrate; additionally, some selected views may be also coded with JPEG 2000 but with a higher PSNR. To improve the reconstruction, the geometry of the scene is computed using a depth estimation algorithm which also provides the segmentation of the scene into regions; after, the displacements of the regions in the various SA images regarding the central view are estimated. The pixels of one arbitrary region in one view are predicted using a sparse predictor, having as regressors the pixels from the corresponding pixels in nine neighbor views. The optimal predictors are found and their structure and parameters are transmitted to the decoder. The views are processed in a spiral sequence, starting from the central view. The block diagrams representation of the Pleno2 codec is shown in Figure 67 down). The bitstream consists on the full lenslet image coded with JPEG 2000, the depth map, the displacements for each region in each dependent view and, finally, the sparse predictors for each region and each view. Random access can be configured by selecting a set of views to be reference views, i.e. to be coded in a backward compatible way, e.g. using JPEG 2000.

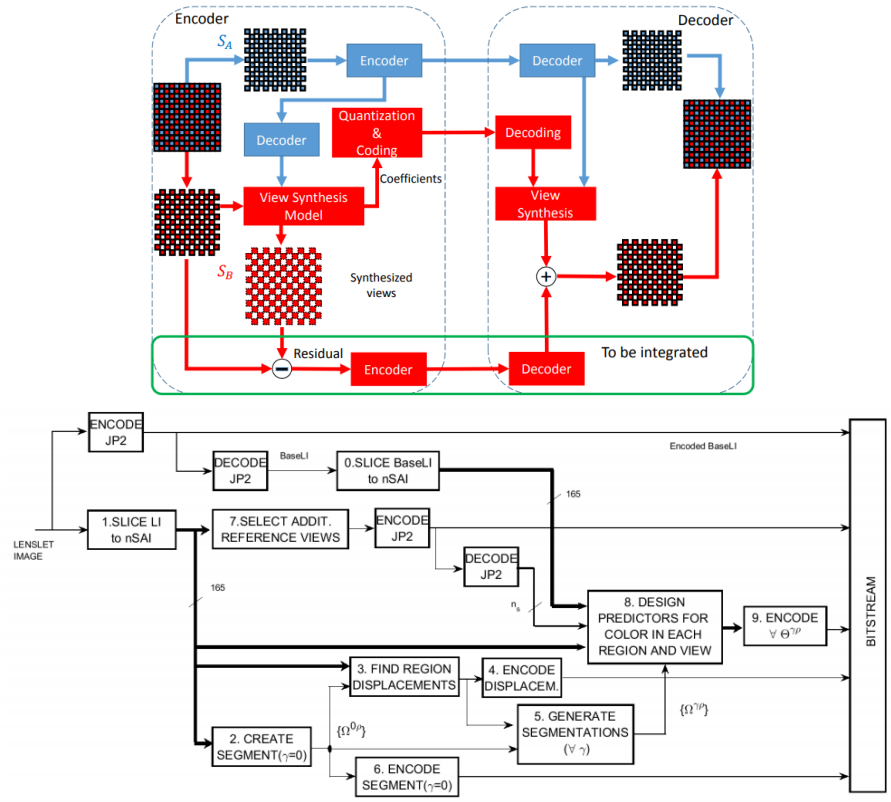


Figure 67: *The block diagrams of the codec: up) Pleno1 [105]; down) Pleno2 [106].*

- **JPEG Pleno Proposal by Ostendo, USA (labelled Pleno3)** [107] – The Pleno3 proposed codec makes use of both camera and scene metadata, along with the light field images, to compute the minimum number of reference images required to faithfully reconstruct the full light field. Then, for each reference image, disparity maps are obtained from the computed depth. If a higher quality reconstruction is needed, more reference images and disparity maps are selected for coding. Both the reference images and the reference disparity maps are compressed with HEVC to allow view synthesis at decoder. The block diagrams representation of the Pleno3 codec is shown in Figure 68. This solution provides scalability in terms of quality, computational complexity and views.

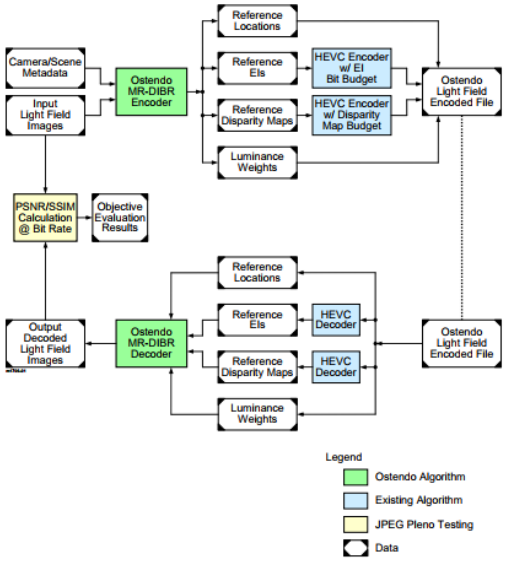


Figure 68: *The block diagrams of the Pleno3 codec [107].*

C. Quantization Parameters

Two quantization parameters may be used for the compression with JPEG 2000 using the adopted software. While the parameter "r", designated in the following as *compression-driven quantization parameter* (CDQP), controls the size of the output compressed image, the second parameter "q", designated in the following as *quality-driven quantization parameter* (QDQP), controls the quality of the output image. The quantization parameters values used for each coding solution are:

i) **JPEG** – The software used for JPEG encoding was the OpenCV software, version 2.4.13 [110]; each RD point was obtained using a so-called *quality parameter* which is in the interval [0; 100] (the higher, the better the quality) with the following values: 1, 15, 30, 50, 70 and 90.

ii) **JPEG 2000** – The software used for JPEG 2000 encoding was the OPENJPEG software version 2.1.2 [111]; each RD point was obtained by setting QDQP with the following values: 32, 36, 40 and 45.

iii) **DCLFC** – In the context of this Thesis, the software for the inter-view disparity compensated wavelet transform based on a lifting scheme was developed; moreover, the OPENJPEG software, version 2.1.2 was integrated in the coding solution to implement the JPEG 2000 codec, and also some image processing tools were integrated using the OpenCV software, version 2.4.13; each RD point was obtained by controlling the JPEG 2000 component of the designed DCLFC coding solution by setting the CDQP, with the following values: 65, 24, 16, 8, 6 and 4.

iv) **JPEG Pleno proposals** – The JPEG Pleno proposals used the rate points defined in the JPEG Pleno Call for Proposals, notably 0.75, 0.1, 0.02 and 0.005 bpp [109].

D. Coding Parameters Selection

During the performance assessment, it was observed that due to differences in terms of content between the low-frequency and high-frequency bands, the CDQP, which controls the size of the compressed stream, needed to be controlled by using different values for different bands. Otherwise, if the same CDQP is applied to all bands, the RD performance would be strongly penalized. In this context, the procedures to determine the appropriate compression-driven quantization parameters are described:

i) **JPEG 2000 coded SA images** – Some SA images are directly encoded with JPEG 2000, which means that the inter-transform scheme is not applied to them. For these images, the compression-driven quantization parameter (CDQP_SA) is given by (12), and the *CDQP* takes the values 65, 24, 16, 8, 6 and 4 has already stated. The objective is to preserve the SA image quality by applying a small *CDQP_SA* value.

$$CDQP\_SA = \frac{CDQP}{2} \tag{12}$$

ii) **Bands** – The inter-transform output consists on low-frequency and high-frequency bands, which are then encoded with JPEG 2000. While the low-frequency bands are an averaged representation of the two input SA images, the high-frequency bands consist mostly on a rather low energy (black) image as their content mainly represents the differences between neighboring SA images or bands. The objective is to spend more rate on a low-frequency band than on a high-frequency band. As the CDQP controls the output compressed size, it is appropriate to differentiate this parameter for the low-frequency and high-frequency bands as increasing this parameter for the high-frequency bands

allows spending less rate without degrading significantly the final reconstruction quality, thus resulting in a better RD performance. In summary, the CDQP value for each band is selected in the following way:

a. **Low-frequency bands**: The rate spent (in Bytes) is computed using (13) and the compression-driven quantization parameter for the low-frequency band, CDQP_LB, is given by (14). The Input_Image_Size corresponds to the size of the input SA images or low-frequency/high-frequency bands, i.e. $625 \times 434 \times 1\ (Bytes)$ for the used test material and for a SA image, CDPQ corresponds to one of the initially defined CDQP values for the DCLFC solution and, lastly, *CDQP_LB/HB_Ratio* corresponds to the ratio between the CDQP for the low-frequency bands and the high-frequency bands; the following values have been tested for *CDQP_LB/HB_Ratio*: 1, 2, 4, 6, 8, 10,12, 14, 16, 18, 20.

$$Bitrate\_LB = \left(\frac{2 \times Input\_Image\_Size}{CDQP}\right) / \left(1 + \frac{1}{CDQP\_LB/HB\_Ratio}\right) \quad (13)$$

$$CDQP\_LB = \frac{Input\_Image\_Size}{Bitrate\_LB} \quad (14)$$

b. **High-frequency bands**: The compression-driven quantization parameter for the high-frequency bands, *CDQP_HB* is given by (15). For each value of CDQP, all the corresponding *CDQP_LB/HB_Ratio* values were tested with the objective of finding the best (*CDQP*, *CDQP_LB/HB_Ratio*) pair in terms of RD performance. These tests have shown that the *CDQP_LB/HB_Ratio* parameter should not be constant along the rate, as maintaining the same *CDQP_LB/HB_Ratio* value for rates higher than 0.4 bpp would reduce the reconstruction quality, thus degrading the overall RD performance. In summary, the first three RD points are obtained using a given *CDQP_LB/HB_Ratio_1* and the last three RD points using another *CDQP_LB/HB_Ratio_2*, fulfilling the condition *CDQP_LB/HB_Ratio_1* > *CDQP_LB/HB_Ratio_2.*

$$CDQP\_HB = CDQP\_LB \times CDQP\_LB/HB\_Ratio \quad (15)$$

Another interesting characteristic to note is that the content of the high-frequency bands significantly changes for a number of wavelet decomposition levels higher than two, i.e. the high-frequency bands exhibit only more and more fine details; thus, it is proposed to use a lower *CDQP_HB* value for the decompositions with a higher number of levels using the rules defined in Table 3.

Table 3: *CDQP_HB dependency with the number of decomposition levels.*

Number of Decomposition Levels	CDQP_HB computation
1	$CDQP\_HB\_1$ = (15)
2	$CDQP\_HB\_2 = \frac{CDQP\_HB\_1}{2}$
3,4	$CDQP\_HB\_3 = \frac{CDQP\_HB\_1}{3}$

For better understanding, Table 4 includes an example with the various CDQP values depending on the type of image or band to encode; each set of parameters defines one of four target RD points. In this example, two distinct CDQP_LB/HB_Ratio values were chosen: i) CDQP_LB/HB_Ratio_1 = 12; ii) CDQP_LB/HB_Ratio_2 = 6. To compute the CDQP value for a SA image, CDQP_SA, (12) was used; for a low-frequency band, CDQP_LB, (13) and (14) were used and, finally, for a high-frequency band, CDQP_HB, (15) and Table 3 rules were used.

Table 4: *Examples of various CDQP values for a selected CDQP value.*

CDQP	**CDQP_LB/HB_Ratio**	**Decomposition level**	***CDQP*_HB**	***CDQP*_LB**	***CDQP*_SA**
65	12	1	422,496	35,208	32,5
		2	211,248		
		3 and 4	140,832		
24	12	1	156	13	12
		2	78		
		3 and 4	52		
6	6	1	21	3,5	3
		2	10,5		
		3 and 4	7		
4	6	1	14	2,33(3)	2
		2	7		
		3 and 4	4,66(7)		

4.2. Performance Assessment Methodology

This section describes the methodology adopted to evaluate the RD performance of the proposed coding solution; the assessment framework is displayed in Figure 69.

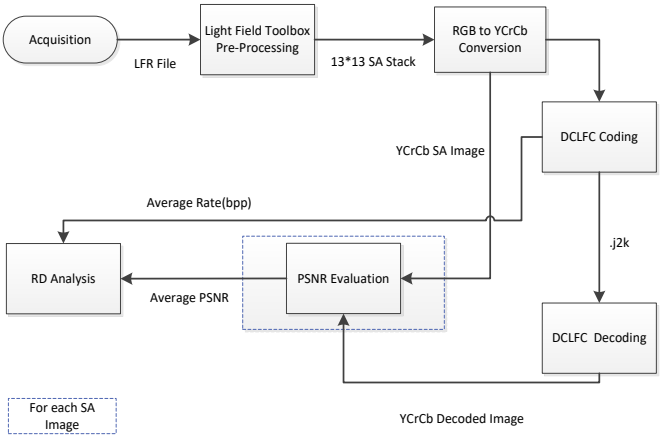


Figure 69: *Processing flow for RD performance assessment.*

The quality is evaluated by computing the PSNR between the original and the corresponding decoded SA images, following (16) and (17), where MAX_I is the maximum valid value that a sample can take, in this case 255 as 8-bit samples are used, *i* and *j* are the dimensions along the x- and y-axis, and I_1, I_2 are the input and output SA images. The rate is measured in bit-per-pixel (bpp) as shown in (18), where $Size_i$ is the number of bits spent in each compressed image and *Homography_rate* is the total rate spent in transmitting the homography parameters. To obtain the bit-per-pixel rate, it is necessary to divide the total rate by the number of coded SA images (169) and by the resolution of each one (625×434). Neither JPEG nor JPEG 2000 need homography parameters, thus, the Homography_rate in (18) is equal to zero and the $\sum_{i=0}^{168} Size_i$ is equal to the total number of bits spent to code the SA images, either individually or as "super image".

$$MSE = \frac{1}{i \times j} \sum (I_1 - I_2)^2 \tag{16}$$

$$PSNR\_SA = 10 \times log_{10} \times \left(\frac{MAX_I^2}{MSE} \right) \tag{17}$$

$$rate\_bpp = \frac{\sum_{i=0}^{168} Size_i + Homography\_rate}{169 \times 625 \times 434} \tag{18}$$

When two coding solutions have a similar RD performance, it becomes harder to identify clearly which one is the best. In such cases, the Bjøntegaard Delta metrics [112] are very useful as they are able to express the relative gain between two coding solutions by measuring the average difference between two RD curves. This measurement may be done in terms of rate savings between the two coding solutions (BD-Rate) or in terms of quality gains (BD-PSNR). A negative BD-Rate means that there are rate savings for the studied coding solution regarding an alternative coding solution and vice-versa; in the same context, a positive BD-PSNR implies an increase in quality between the two solutions. For computing the BD-PSNR and BD-Rate values, the following CDQP values were adopted: 65, 16, 8, 6. As the metric needs four points to compute reliable results, the values chosen target to provide a PSNR in the 27 to 42 dB range which is considered appropriate.

For better reading of the next sections, the naming convention is summarized in Table 5.

Table 5: *Naming convention for various parameters and configurations.*

Variable/label	Parameter/Configuration
QDQP	Quality-driven quantization parameter for JPEG 2000 which controls the output image quality (PSNR).
CDQP	Compression-driven parameter for JPEG 2000 which controls the output image size (rate).
DCLFC H_x&y or DCLFC V_x&y or DCLFC Hx_Vy_x&y	*x* is the CDQP_LB/HB_Ratio used in the first three RD points and *y* is the CDQP_LB/HB_Ratio used in the last 3 RD points
DCLFC Hx	DCLFC applied only horizontally with *x* -levels of inter-transform decomposition.
DCLFC Vy	DCLFC applied only vertically with *y* -levels of inter-transform decomposition.
DCLFC Hx_Vy	DCLFC applied, first horizontally and then vertically with -*x* levels of inter-transform decomposition (horizontal) and *y* -levels of decomposition (vertical).
DCLFC (L+H)	DCLFC applied to both the low-frequency and high-frequency bands.
(Original)	Homography parameters computed using the originals SA images.
(QS)	Compression using quality layers, generating a quality scalable (QS) stream.

4.3. DCLFC Performance Assessment

The proposed light field coding solution may be applied in several different configurations, each offering different trade-offs in terms of compression performance, random access and view scalability. This section intends to study the performance impact of the most relevant DCLFC configurations, notably in terms of RD performance. First, the situation where the redundancy between SA images is exploited just along one dimension, horizontal or vertical, is studied; after, the situation where the SA images decorrelation is performed in the two dimensions, both horizontally and vertically, is studied. This study will overall assess the impact on the RD performance of the number of dimensions where the redundancy is explored, as well as, the impact of the number of decomposition levels employed in the inter-transform. After, the benchmarking with the alternative solutions and, lastly, the RD performance when compressing each image or band with several quality layers leading to a scalable stream will be studied, already using a stable, best performing coding solution in terms of number of dimensions and decomposition levels.

4.3.1. DCLFC over a Single Light Field Dimension

This section presents the performance results and their analysis for the situation where the proposed DCLFC solution is applied to SA images which are neighbors in the vertical or horizontal dimensions. Moreover, also the impact of the number of decomposition levels is analyzed, in practice determining

how many decomposition levels should be used before the RD performance starts decreasing. Two main advantages come from the use of multiple decomposition levels when applying the inter-transform along each light field dimension:

i) **Compression performance** – In principle, the more levels of decomposition are used, the more the correlation between low-frequency and high-frequency bands should be explored. Thus, it is expected to achieve improvements in compression performance while the low-frequency and high-frequency bands still exhibit high correlation between themselves; when this correlation is too low, the compression performance will start to decrease.
ii) **View scalability** – View scalability is achieved by using the wavelet based inter-transform. The view scalable layers are naturally provided by the wavelet decomposition, e.g. when applying the inter-transform with one decomposition level, half the SA images are replaced by low-frequency bands and the other half by high-frequency bands, which means that a first scalable layer is created with a number of views equal to number of low-frequency bands and a second layer of views is obtained upon decoding the information for the high-frequency bands, thus finally decoding the remaining views.

4.3.1.1. DCLFC over the Horizontal Dimension

The first case concerns the application of the proposed coding solution to horizontally neighboring SA images. results for four configurations are presented in Figure 70, namely: i) **DCLFC H1**, DCLFC encoding using only 1-level inter-transform; ii) **DCLFC H2**, as DCLFC H1 but now using 2-level inter-transform; iii) **DCLFC H3**, as DCLFC H1 and DCLFC H2 but now using 3-level inter-transform, and finally a benchmark, iv) **JPEG 2000_Super_Image**, JPEG 2000 encoding each SA image, which means there is no inter-prediction between views, using as input the SA images arranged in a “super image”; the redundancy is exploited mostly within each SA image and thus no relevant inter-view redundancy exploitation is performed.

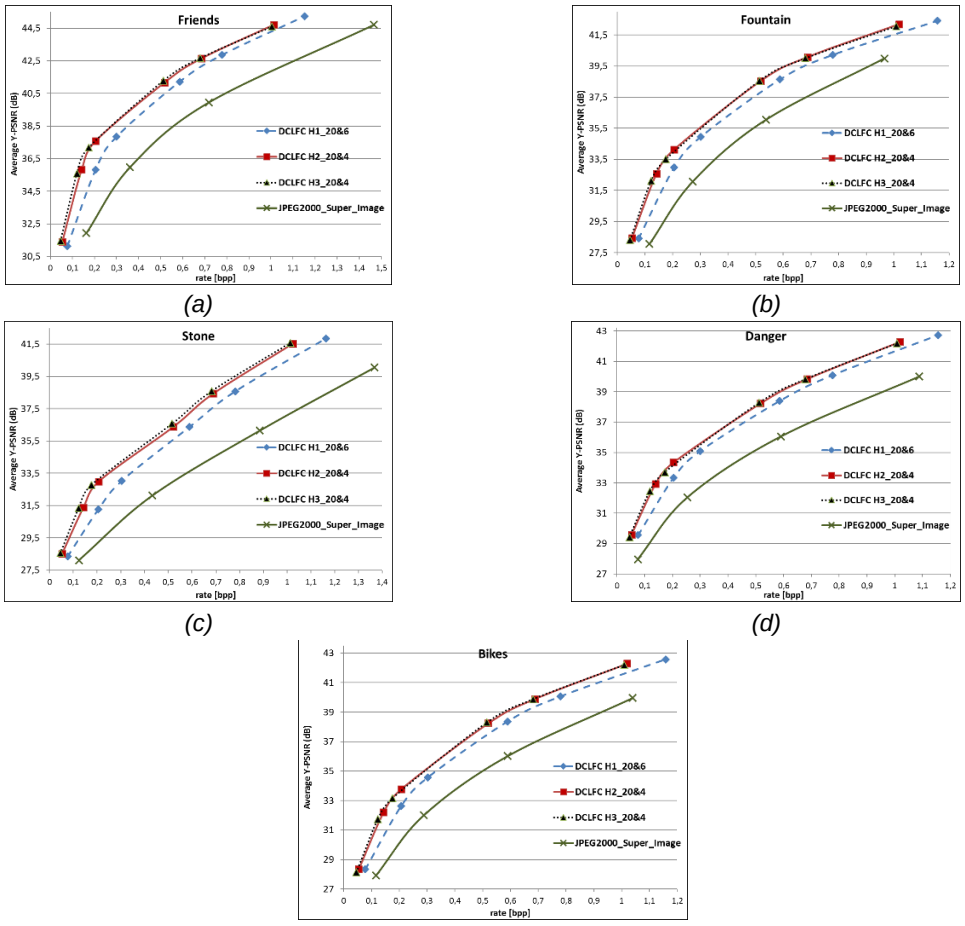
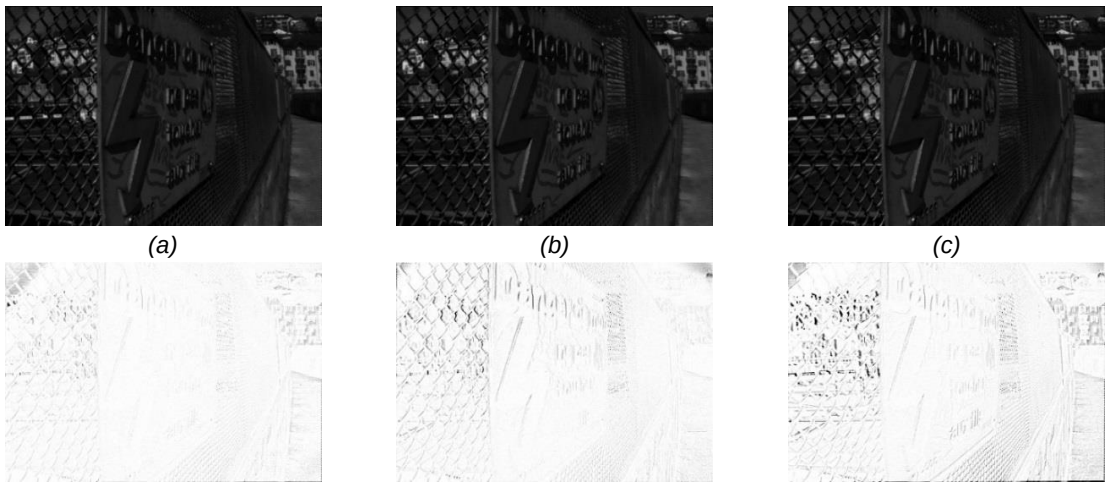


Figure 70: *DCLFC 1D horizontal RD performance results for: a) Friends; b) Fountain; c) Stone; d) Danger; e) Bikes.*

The results from Figure 70 allow taking the following conclusions:

i) **Number of decomposition levels impact** – Applying inter-view prediction, DCLFC H1, improves the compression performance compared to the situation where the SA images are encoded with JPEG 2000 (intra coding only) as a “super image”. The RD performance comparison is made in the situation where the SA images are coded as a “super image” because this scenario results in a better compression performance when compared to the situation where the SA images are coded individually. Increasing the number of decomposition levels of the inter-transform also allows increasing the RD performance although with a reducing gain for any additional level. Following the indications from the paper which inspired the DCLFC solution [83], the inter-transform scheme was only tested for up to 3 levels of decomposition as the compression performance is expected to decrease for higher values. These results and conclusions can be better understood with the help of the examples in Figure71 and Figure 72. By analyzing the high-frequency bands resulting from the inter-view transform, it is visible that the higher the number of decomposition layers, the smaller is the similarity between the low-frequency bands, which justifies the small RD performance increase for 3 decomposition levels shown in Figure 70. A high-frequency band resulting from 3 decomposition levels exhibits more perspective view differences than a high-frequency band resultant from 1 or 2 decomposition levels, thus supporting the conclusion that the low-frequency bands show reduced correlation with each decomposition level applied. Figure 71 shows examples of low-frequency and

high-frequency bands for the Danger light field for three cases in terms of number of decomposition levels.

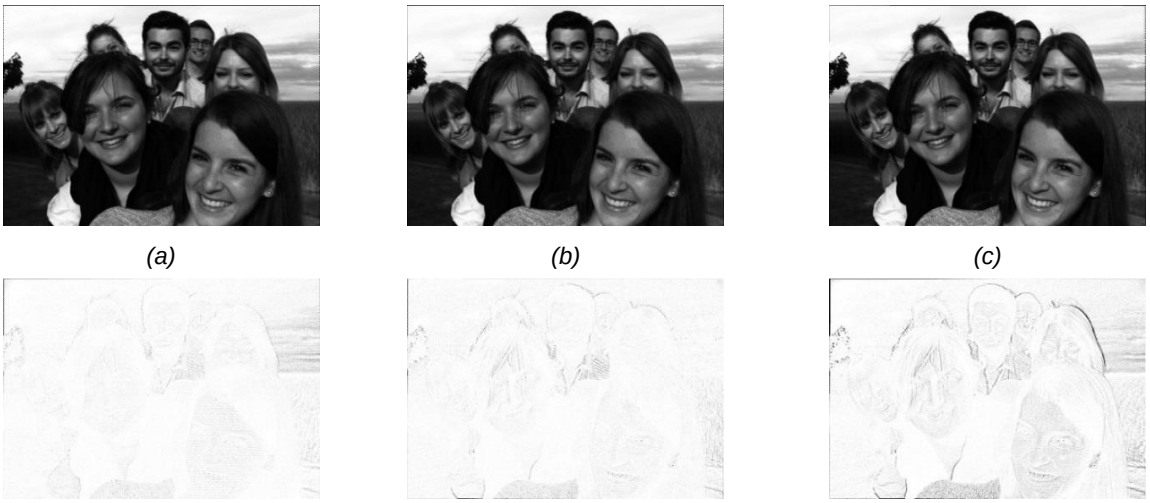


(a) *(b)* *(c)*

(d) *(e)* *(f)*

Figure 71: *top) Low-frequency bands from the Danger light field corresponding to inter-transform with: a) 1-level; b) 2-levels; c) 3-levels; bottom) High-frequency bands for the same situations.*

Note that, as the computed high-frequency bands are almost completely black, and thus its content very hard to perceive, it was necessary to multiply the high-frequency bands sample values by four (if greater than 255, it becomes 255) and then inverting the image (y=255-x) in order a clearer visual analysis of the high-frequency bands content is possible



(a) *(b)* *(c)*

(d) *(e)* *(f)*

Figure 72: *top) Low-frequency bands from the Friends light field corresponding to inter-transform with: a) 1-level; b) 2-levels; c) 3-levels; bottom) High-frequency bands for the same situations.*

In Figure 72, similar examples are shown for the Friends light field. The Friends light field is the one achieving the best RD performance with the proposed coding solution while the Danger light field corresponds to the opposite case. Comparing the two examples, it is notorious that the bands from Figure 71 exhibit more details in the level-3 high-frequency bands than those in Figure 72; this is mostly due to the detailed fence and the letters in the Danger light field. This allows to conclude that while the low-frequency bands exhibit high levels of similarity among each other, the application of the inter-transform will improve the RD performance. Otherwise, if the low-frequency bands correspond to views which are getting further away, the inter-transform scheme does not yield gains;

in this case, the high-frequency bands which are supposed to represent the details will include more and more details to represent the perspective differences for each view.

ii) **Overall RD performance –** Considering the behavior of the various RD curves along the rate and comparing with the JPEG 2000 results, it is possible to conclude: i) DCLFC H1 increases the RD performance on average 2 dB, and the gains remain rather constant along the different rates (except for Stone for which the high rates display higher RD gains); ii) DCLFC H2 increases even more the RD performance, but it is noticeable that the "gap" in RD performance from DCLFC H1 to JPEG 2000 is higher than between DCLFC H2 and DCLFC H1. It is also possible to conclude that the RD performance is enhanced for the lower rates, as it exhibits higher gain for the majority of the light fields; iii) lastly, DCLFC H3 shows RD performance improvements for some LFs, e.g. *Friends* and *Stone*, but the remaining 3 light fields do not see their performance improved with the increase on the number of decomposition levels. Like DCLFC H2, DCLFC H3 also displays larger RD performance gains for the lower rates; as the high-frequency bands are mostly 'black', it is possible to greatly reduce their size, when compared to the original SA images, and at the same time preserve the average reconstruction quality of the decoded light field.

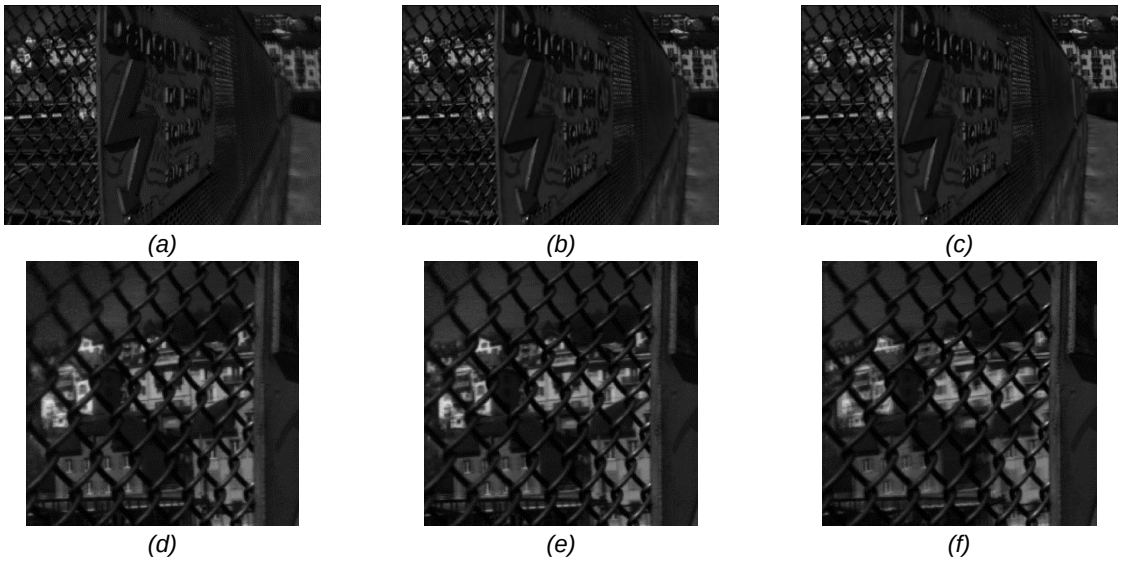
As a summary, Table 6 shows the BD-Rate and BD-PSNR for DCLFC H2 and DCLFC H3 in comparison to the JPEG 2000 standard. Regarding the RD performance, it is possible to conclude that *Friends* is the light field exhibiting higher gains, while *Danger* is the one with the worst RD performance gains. This is understandable as *Friends* exhibits a more homogenous background while *Danger* includes letters and much more details, thus reducing the proposed solution compression efficiency as there is less redundancy across the views.

Table 6: *Bjøntegaard delta results regarding JPEG 2000 with: left) DCLFC H2; right) DCLFC H3.*

Light Field	BD-PSNR[dB]	BD-Rate [%]
Friends	4.24	-61.44
Fountain	3.02	-46.75
Stone	3.48	-52.95
Danger	3.00	-49.69
Bikes	3.13	-47.56
Average	**3.38**	**-51.68**

Light Field	BD-PSNR[dB]	BD-Rate [%]
Friends	4.49	-64.24
Fountain	3.10	-48.75
Stone	3.70	-56.76
Danger	3.02	-50.75
Bikes	3.21	-49.36
Average	**3.51**	**-53.97**

Despite DCLFC H3 presenting a slightly improvement on the average BD-Rate and average BD-PSNR, the additional complexity and the quality decrease in the low-frequency bands makes DCLFC H2 a better choice for future comparisons. In Figure 73, it is possible to see the decrease in quality from the original SA image to the low-frequency band for DCLFC H3; for example, the buildings behind the fence appear to be blurred. These artifacts may result from the update step, when the low-frequency bands, taken as input to the inter-view transform, represent more distant views, which means they are an averaged version of those more distant views, as so some artifacts may be introduced. This effect is not persistent for all the low-frequency bands for DCLFC H3; for example, *Friends,* which is the light field with best RD performance, does not have these artifacts in the low-frequency bands for DCLFC H3.



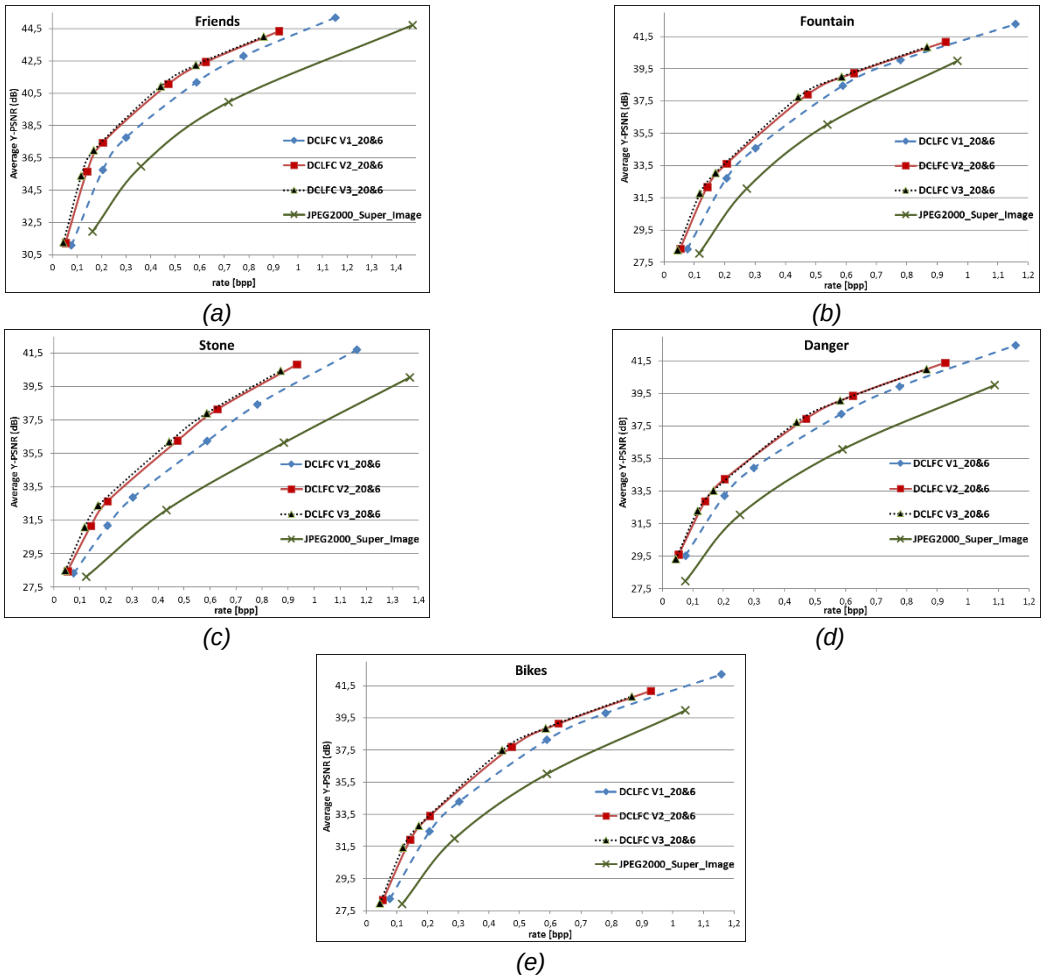
(a) *(b)* *(c)*

(d) *(e)* *(f)*

Figure 73: *a) Original SA image; b) Low-frequency band for DCLFC H2; c) Low-frequency band for DCLFC H3; d) Zoom of (a); e) Zoom of (b); f) Zoom of (c).*

4.3.1.2. DCLFC over the Vertical Dimension

Another possibility is to apply the DCLFC solution only to vertically neighboring SA images or low-frequency/high-frequency bands. At this stage, it is useful to understand if the RD performance behaves as for the horizontal dimension or if there are any relevant differences.



(a)

(b)

(c)

(d)

(e)

Figure 74: *DCLFC 1D vertical RD results for: (a) Friends; (b) Fountain; (c) Stone; (d) Danger; (e) Bikes.*

The RD performance results for the various levels of decomposition used in the vertical 1D are displayed in Figure 74 for the same 5 test light field images; four configurations are presented, namely: i) **DCLFC V1**, DCLFC encoding using only 1-level inter-transform over vertically neighboring SA images; ii) **DCLFC V2**, as DCLFC V1 but now using 2-level inter-transform; iii) **DCLFC V3**, as DCLFC V1 and DCLFC V2 but now using 3-level inter-transform, and finally a benchmark, iv) **JPEG 2000_Super_Image,** in the same conditions as in the horizontal case.

Following the same methodology used for DCLFC 1D horizontal, the conclusions regarding the DCLFC 1D vertical case are:

i) **Number of decomposition levels impact** – The same conclusions taken for the DCLFC 1D horizontal case remain true for the DCLFC 1D vertical case. By applying DCLFC V1, the RD performance is improved compared to JPEG 2000 and both DCLFC V2 and DCLFC V3 allow to further increase the RD results, although with the gains diminishing when the number of decomposition levels further increase. It is interesting to note that the DCLFC V3 shows here a better performance when compared to the DCLFC V2 approach, which may indicate that the low-frequency bands resulting from applying DCLFC V1 maintain more similarities between them, hence enabling an improvement in RD performance.

ii) **Overall RD performance** – Along the different rates, the RD performance for DCLFC V1 is similar to DCLFC H1. Most light fields have their compression efficiency improved for the lower rates zone, diminishing the gains when the rate increases. Finally, the light field with the largest gain is *Friends* and the worst case light field corresponds to *Fountain*; this naturally confirms that the light field content has a great impact on the RD performance and the horizontal and vertical redundancies are different within each light field. As for the horizontal approach, Table 7 confirms that, on average, DCLFC V3 provides the best compression performance, but again the for the same reasons explained earlier for the DCLFC 1D horizontal case, the extra complexity and lower low-frequency band quality makes preferable to select DCLFC V2 has the best configuration.

Table 7: *Bjøntegaard delta results regarding JPEG 2000 with: left) DCLFC V2; right) DCLFC V3.*

Light Field	BD-PSNR[dB]	BD-Rate [%]
Friends	4.18	-59.02
Fountain	2.67	-42.36
Stone	3.46	-52.78
Danger	2.95	-49.34
Bikes	2.89	-44.51
Average	**3.23**	**-49.60**

Light Field	BD-PSNR[dB]	BD-Rate [%]
Friends	4.58	-63.25
Fountain	2.95	-46.41
Stone	3.75	-56.94
Danger	3.03	-50.92
Bikes	3.05	-46.85
Average	**3.47**	**-52.88**

4.3.2. DCLFC Performance over Both Light Field Dimensions

This section presents the performance results and their analysis for the situation where the proposed DCLFC solution is first applied to horizontally neighboring SA images and after applied to vertically neighboring SA images; e.g. the DCLFC H2_V1 solution is implemented by applying first 2-levels of decomposition to horizontally neighboring SA images and after 1-level of decomposition to the resulting vertically neighboring low-frequency bands. This was the selected order because overall the application of the DCLFC solution first to horizontally neighboring SA images or low-frequency bands resulted in a better RD performance than the alternative vertical order. Considering the large number of DCLFC configurations that must be considered and to better organize their comparison, the RD results are presented considering a growing total number of levels of decomposition, independently of their

direction, i.e. the number of decomposition levels applied horizontally plus the decomposition levels applied vertically. Presenting the RD performance results organized by number of decomposition levels implies that each study concerns solutions which have the same number of view scalability layers and almost the same level of decorrelation is performed on the SA images or low-frequency bands.

4.3.2.1. 2-Levels DCLFC Performance

The first case studied concerns the use of 2-levels of decomposition, and so the RD results presented in Figure 75 compare the **DCLFC H2** solution with **DCLFC H1_V1** solution which both have 2 levels of decomposition. The DCLFC H1_V1 and DCLFC H2 solutions are here compared because they both have low-frequency and high-frequency bands resulting from a DCLFC with 2-levels of decomposition although DCLFC H1_V1 involves both dimensions of the light field and DCLFC H2 just the horizontal dimension. This should guarantee a fair comparison as the number of low-frequency and high-frequency bands is similar, thus allowing the same layers of view scalability and a rather similar compression performance. Moreover, it was already seen that the DCLFC H2 solution has larger BD-PSNR gains and BD-Rate savings than the DCLFC V2 solution.

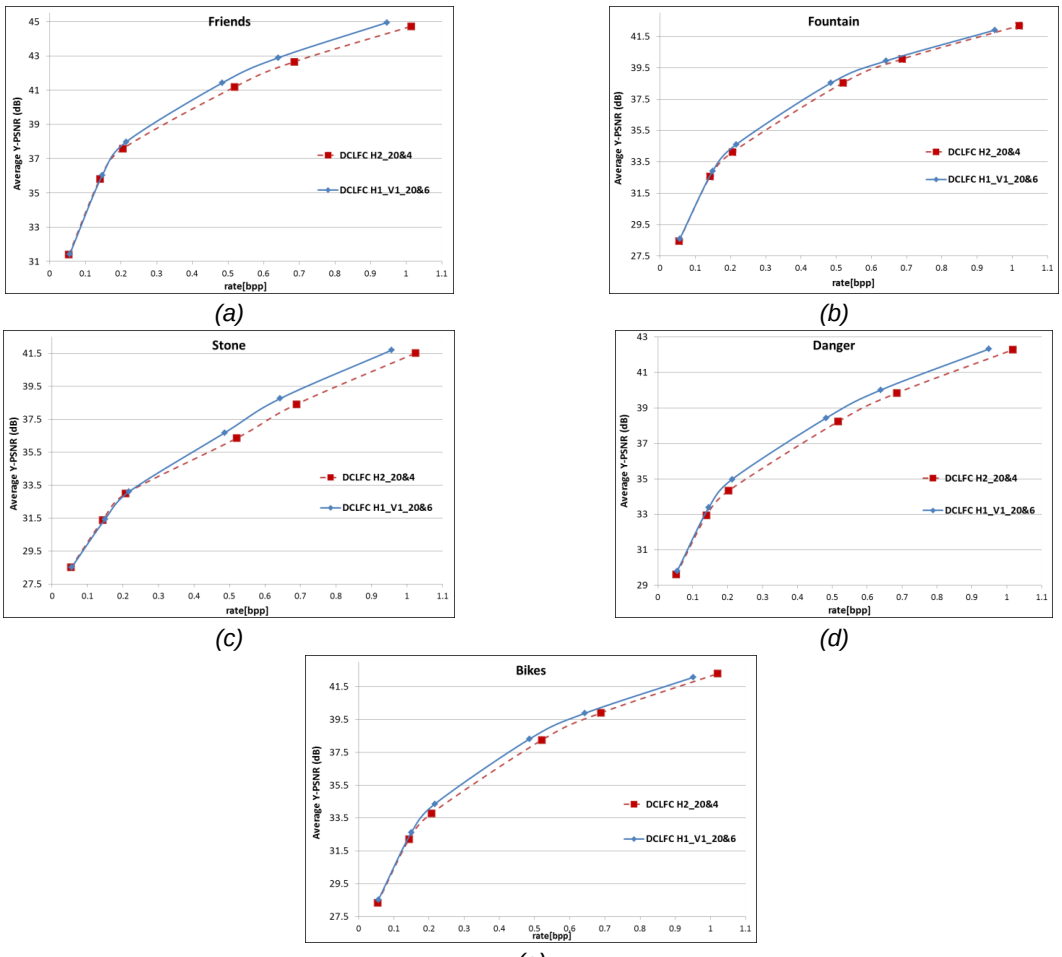


Figure 75: *DCLFC H1_V1 and DCLFC H2 RD performance for: a) Friends; b) Fountain; c) Stone; d) Danger; e) Bikes.*

The results from Figure 75 allow concluding that the DCLFC H1_V1 solution improves the RD performance compared to the DCLFC H2 solution. The behavior of the DCLFC H1_V1 solution against the DCLFC H2 solution is constant across all light fields, notably with no RD gains for the lower rates and increasing gains for increasing rates; naturally, the gains are not the same for all light fields as its

content impacts the RD performance. Curiously, both the *Bikes* and *Fountain* light fields show a decreasing RD performance for the highest rates in opposition to what happens for the remaining light fields. The RD performance is better especially for the higher rates, showing the same quality for lower rates, may be due to the better disparity compensation scheme with higher quality (2 light field dimensions). This can be seen in Figure 75, for the last three DCLFC H1_V1 RD points, which show less rate for a similar quality. The *Danger* light field is the one yielding higher BD-PNSR gains and BD-Rate savings, what is curious as it was usually the light field with lower RD performance improvements; this is anyway another indication that the DCLFC H1_V1 solution improves the quality of the disparity estimation. The Bjøntegaard delta results for the five light fields are shown in Table 8.

Table 8: *Bjøntegaard delta results for DCLFC H1_V1 regarding DCLFC H2.*

Light Field	BD-PSNR[dB]	BD-Rate [%]
Friends	0.18	-3.64
Fountain	0.20	-4.34
Stone	0.15	-5.61
Danger	0.41	-9.14
Bikes	0.30	-6.30
Average	**0.25**	**-5.81**

It is interesting to note that by just changing the dimension where the inter-transform is applied it is possible to achieve better RD performance; this may indicate that if too many decomposition levels are applied across one dimension, the correlation between the low-frequency bands tends to disappear faster than if the decomposition levels are applied alternately. In summary, it seems clear that is better to apply the DCLFC solution over two light field dimensions instead of just one, and so, in the next experiments, DCLFC solutions considering one single decomposition dimension will not be studied. Instead the comparisons will always be made with the best DCLFC solution until that moment.

4.3.2.2. 3-Levels DCLFC Performance

By increasing the number of decomposition levels to three, two different DCLFC solutions become relevant, namely **DCLFC H1_V2** and **DCLFC H2_V1**. While both exploit the correlation between neighboring SA images, horizontally and vertically, the first has a higher number of decomposition levels applied vertically contrarily to the second case. Figure 76 displays the RD performance results for both solutions and uses the previous **DCLFC H1_V1** solution as benchmark. With 3-levels of (joint) decomposition, both the BD-PSNR and BD-Rate gains are slightly improved compared to the previous 2-levels configurations; however, as for the DCLFC application over one single light field dimension, the RD performance improvements compared with 2-levels of decomposition are rather small.

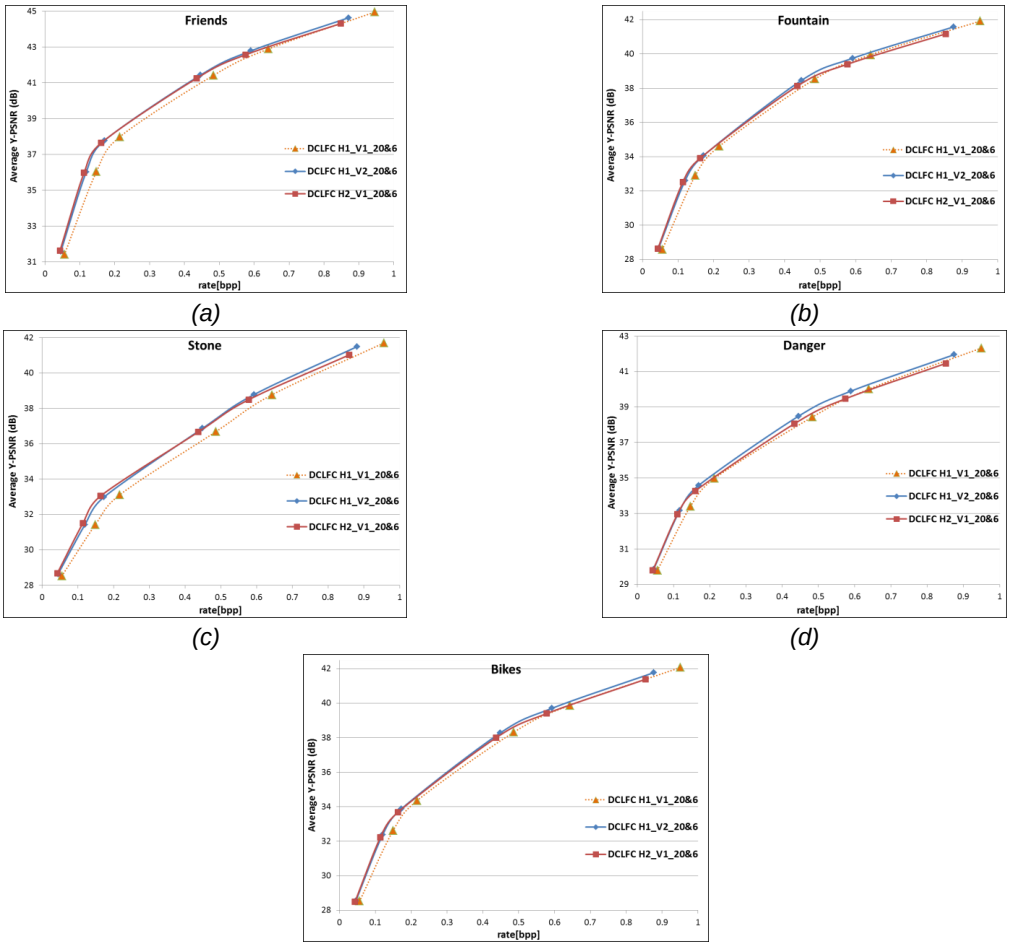


Figure 76: *DCLFC H1_V1, DCLFC H1_V2 and DCLFC H2_V1 RD performance for: a) Friends; b) Fountain; c) Stone; d) Danger; e) Bikes.*

The Bjøntegaard delta results in Table 9 allow concluding that the gains in BD-PSNR and BD-Rate remain similar for all light fields. *Stone*, which is one of the light fields with larger BD-PSNR gains for the DCLFC H2_V1 solution, in this case 0.86 dB, has only an increase of 0.75 dB for the DCLFC H1_V2 solution, thus showing that this light field has higher correlation between the horizontally neighboring SA images, again showing the obvious fact that the light field content impacts the RD performance. In summary, both DCLFC H2_V1 and DCLFC H1_V2 result, on average, in similar BD-PSNR gains and BD-Rate savings although DCLFC H2_V1 leads to a slightly better RD performance. Thus, for the next experiment, DCLFC H2_V1 will be the benchmark.

Table 9: *Bjøntegaard delta results using as reference DCLFC H1_V1: left) DCLFC H2_V1; right) DCLFC H1_V2.*

Light Field	BD-PSNR[dB]	BD-Rate [%]
Friends	0.83	-18.56
Fountain	0.47	-10.57
Stone	0.77	-16.73
Danger	1.98	-37.78
Bikes	0.49	-10.66
Average	**0.91**	**-18.86**

Light Field	BD-PSNR[dB]	BD-Rate [%]
Friends	0.75	-16.32
Fountain	0.47	-10.19
Stone	0.66	-14.19
Danger	2.13	-39.46
Bikes	0.50	-10.62
Average	**0.90**	**-18.16**

4.3.2.3. 4-Levels DCLFC Performance

With 4-levels of overall decomposition, it is relevant to study the configurations **DCLFC H1_V3**, **DCLFC H3_V1** and **DCLFC H2_V2**. Both the first and second solutions were implemented in the usual way, this means first the levels of horizontal decomposition are applied and after the levels of vertical decomposition are applied. However, for the DCLFC H2_V2 solution, for reasons to be explained later, each level of decomposition of the inter-transform was applied alternately between the horizontal and vertical dimensions. The RD performance results comparing these solutions with the previous best DCLFC H2_V1 solution are shown in Figure 77. It was previously mentioned that the DCLFC compression performance is expected to decrease for a number of decomposition levels higher than 3; however, such statement was made considering the DCLFC configurations applied to a single light field dimension. When the DCLFC solution is applied to both light field dimensions, the compression performance is expected to further increase up to the point where the compression performance finally starts decreasing.

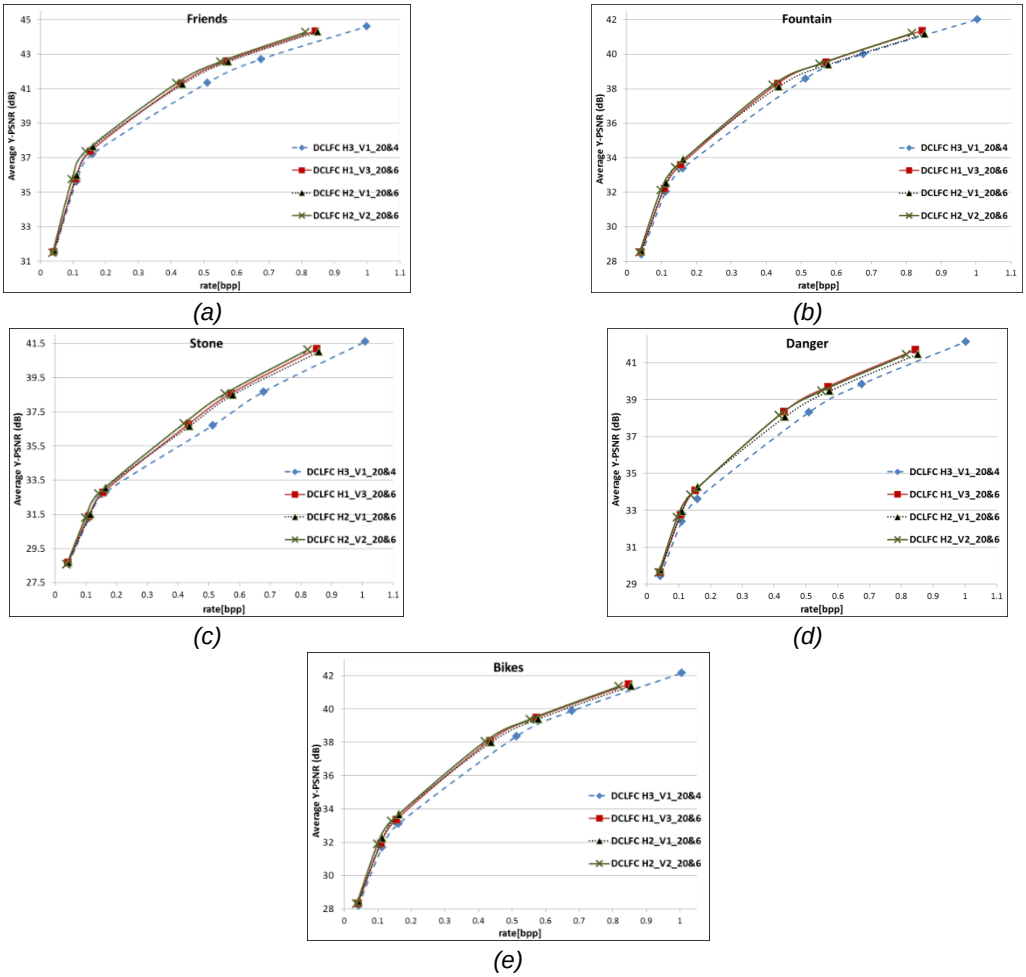


Figure 77: *DCLFC H3_V1, DCLFC H1_V3, DCLFC H2_V1 and DCLFC H2_V2 RD performance for: a) Friends; b) Fountain; c) Stone; d) Danger; e) Bikes.*

Analyzing the results in Figure 77, it is possible to conclude that overall the DCLFC H3_V1 solution reduces the RD performance when compared with the DCLFC H2_V1 solution; this may indicate that exploiting too much the correlation between horizontally neighboring SA images results in low-frequency bands representing increasingly different views, at some stage too different to bring performance gains; so the RD performance does not improve anymore. For the DCLFC H1_V3 solution, this effect does not

occur yet as this solution still shows a slightly better RD performance compared with the DCLFC H2_V1 solution; this leads to the conclusion that the low-frequency bands corresponding to the vertical inter-transform application result in more similar low-frequency bands. However, on average, both the DCLFC H3_V1 and DCLFC H1_V3 solutions failed to improve the DCLFC H2_V1 RD performance, as shown in Table 10. It is also clear that the best number of decomposition levels is reaching the saturation point; the RD gains achieved with each new DCLFC configuration are successively decreasing. For those reasons, it was decided to implement the DCLFC H2_V2 solution with an alternative approach, this time exploiting the correlation alternatively between the horizontal and vertical dimensions. This mean that the DCLFC H2_V2 solution is implemented following the order: i) DCLFC H1; ii) DCLFC H1_V1; iii) DCLFC H2_V1; iv) DCLFC H2_V2. Figure 77 and Table 10 show that this is the solution offering the best RD performance as it improves both the BD-PSNR gains and BD-Rate savings compared to DCLFC H2_V1 while providing one more view scalability layer.

Overall, the studied DCLFC configurations with 4-levels of decomposition presented very different RD performances compared with DCLFC H2_V1. While DCLFC H3_V1 shows a reduced RD performance, DCLFC H1_V3 shows almost no performance differences and DCLFC H2_V2 is the only configuration able to increase the DCLFC H2_V1 performance. It is a bit strange to have such large performance differences between the DCLFC H1_V3 and DCLFC H3_V1 RD performances as until now both the horizontal and vertical approaches resulted in rather similar RD performances, as long as the number of decomposition levels remained the same. Because of this unexpected behavior, all precautions were taken to ensure the correct implementation of the inter-transform scheme and JPEG 2000 compression parameters. For all light fields, there are larger RD gains for the higher rates, opposed to what used to happen for previous configurations, and the gains are rather uniform for the set of selected light fields. The light field with more extreme BD-PSNR gains and BD-Rate savings was *Friends* while *Danger* was the one with the lower BD-PNSR gains and BD-Rate savings.

Table 10: *Bjøntegaard delta results regarding DCLFC H2_V1 for: left) DCLFC H3_V1; mid) DCLFC H1_V3 ; right) DCLFC H2_V2.*

Light Field	BD-PSNR[dB]	BD-Rate [%]
Friends	-0.38	10.34
Fountain	-0.36	8.75
Stone	-0.37	11.06
Danger	-0.52	14.01
Bikes	-0.44	10.63
Average	**-0.41**	**10.95**

Light Field	BD-PSNR[dB]	BD-Rate [%]
Friends	-0.02	1.34
Fountain	0.01	-0.53
Stone	0.02	-1.06
Danger	0.10	-2.90
Bikes	-0.02	0.12
Average	**-0.01**	**0.18**

Light Field	BD-PSNR[dB]	BD-Rate [%]
Friends	0.32	-7.48
Fountain	0.22	-5.36
Stone	0.25	-6.73
Danger	0.18	-5.04
Bikes	0.23	-5.34
Average	**0.24**	**-5.99**

4.3.3. Final Benchmarking

Naturally, the most important results regard the direct comparison of the proposed solution with alternative solutions already available, both based on image coding standards and specifically developed for this type of imaging data. Figure 78 shows the RD performance results comparing the best configuration of the developed DCLFC solution, this mean **DCLFC H2_V2**, against the most relevant/popular standards available, **JPEG 2000**, **JPEG** and the three coding solutions proposed in the context of the JPEG Pleno Call for Proposals on Light Field Coding for which a brief description was provided in Section 4.1.

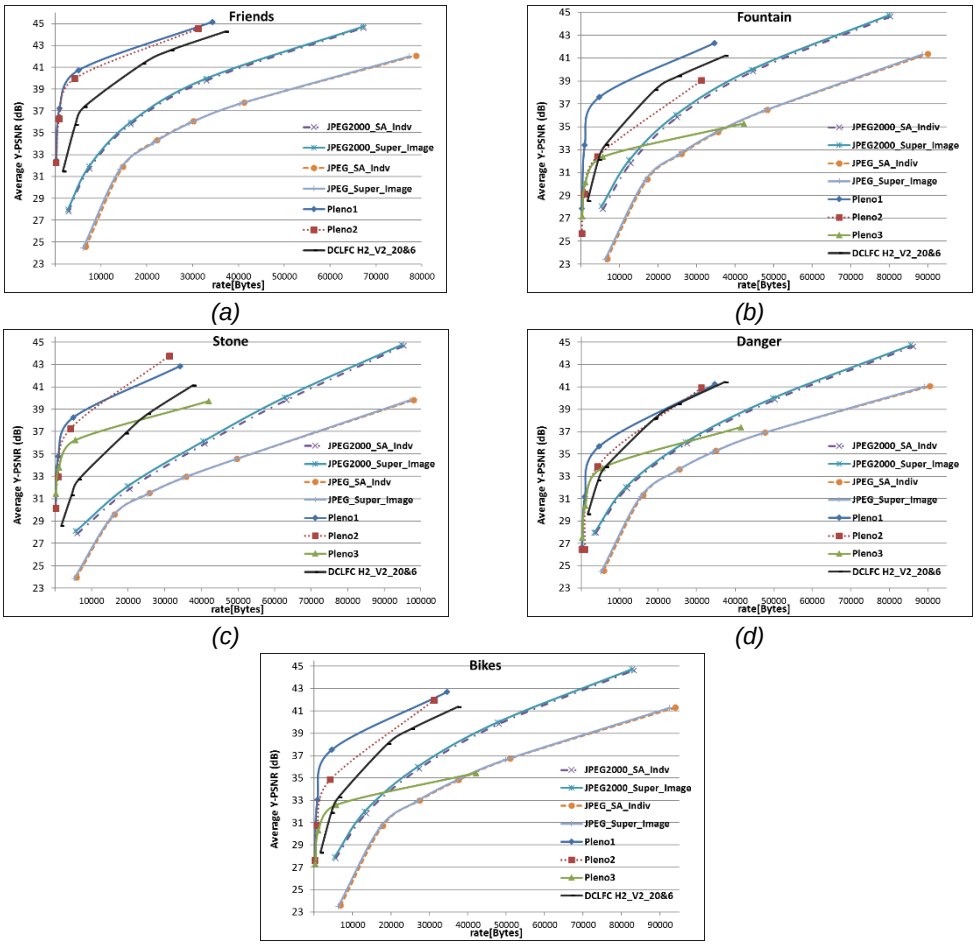


Figure 78: *Benchmarking RD results for: a) Friends; b) Fountain; c) Stone; d) Danger; e) Bikes.*

JPEG was chosen as it is still the most widely used image codec and JPEG 2000 because it is a standard and here an obvious candidate for comparison as the proposed solution is largely based on JPEG 2000. Both JPEG 2000 and the proposed DCLFC H2_V2 solution provide scalability and both are based on the wavelet transform. Moreover, the comparison with JPEG 2000 allows concluding how much influence the inter-view scheme has on the proposed codec RD performance. The conclusions regarding the two light field data organizations used for both JPEG and JPEG 2000 and the overall RD performance are:

A. Light Field Data Organization:

Concerning the two light field data organizations used both for JPEG and JPEG 2000 encoding, the conclusions were as expected. For JPEG, encoding the SA images individually or as “super image” provides basically the same RD performance. As for JPEG 2000, the performance increases slightly when using the “super image” as input; this is not unexpected as the 2D wavelet transform is performed along the entire image, thus exploiting a bit more of the correlation than the individually coded SAs.

B. Overall RD Performance:

The proposed DCLFC H2_V2 solution is able to outperform both the JPEG and JPEG 2000 standards, which is understandable as none of these coding solutions provides decorrelation capabilities between neighboring SA images; instead they rely on a fully Intra coding scheme, in opposition to the

DCLFC solution which uses both Intra and Inter decorrelation schemes. The RD gains remain rather constant along the rate and this behavior is observed for the five light fields tested. Table 11 shows the BD-PSNR gains and BD-Rate savings for the DCLFC H2_V2 solution regarding the benchmarks. The *Friends* light field proved to be the one with best RD performance while *Danger* proved to be the light field with the worst RD performance. Overall, on average, the DCLFC H2_V2 solution offers BD-PSNR gains of 4.39 dB and BD-Rate savings of 62.85% compared with JPEG 2000. Against JPEG, the gains are even higher which is understandable as JPEG is an older codec, not equipped with such powerful tools as JPEG 2000.

Table 11: *Bjøntegaard delta results using DCLFC H2_V2 as reference for: left) JPEG2000_Super_Image; right) JPEG_Super_Image.*

Light Field	BD-PSNR[dB]	BD-Rate[%]
Friends	5.61	-72.67
Fountain	3.91	-57.51
Stone	4.67	-65.65
Danger	3.92	-60.97
Bikes	3.84	-57.44
Average	**4.39**	**-62.85**

Light Field	BD-PSNR[dB]	BD-Rate[%]
Friends	9.30	-86.66
Fountain	7.62	-79.81
Stone	5.42	-70.56
Danger	6.64	-78.18
Bikes	7.43	-78.79
Average	**7.28**	**-78.80**

Following the comparison with the JPEG 2000 and JPEG standard codecs, a comparison with the JPEG Pleno proposals was performed. Two proposals bring significant BD-Rate gains regarding the proposed DCLFC H2_V2 solution, namely Pleno1 and Pleno2. Pleno2 is based on JPEG 2000 and can achieve large BD-PSNR gains and BD-Rate savings, as shown in Table 12; this solution also enables random access and offers the types of scalability naturally provided by the JPEG 2000 codec. However, the Pleno1 proposal based on HEVC is the coding solution achieving the best RD performance, also due to the superior performance of HEVC regarding JPEG 2000. Lastly, the Pleno3 proposal is the only one with similar RD performance when compared to the proposed DCLFC H2_V2 solution; the RD performance results of Pleno3 solution, regarding the *Friends* light field, are not considered as a bug decreased the solution performance. Note, that the bug was present in the Pleno3 proposal software.

Table 12: *Bjøntegaard delta results using DCLFC H2_V2 as reference for left) Pleno1; mid) Pleno2; right) Pleno3.*

Light Field	BD-PSNR[dB]	BD-Rate[%]
Friends	-3.94	81.47
Fountain	-4.32	79.01
Stone	-5.58	87.79
Danger	-2.23	52.63
Bikes	-4.57	78.77
Average	**-4.13**	**75.93**

Light Field	BD-PSNR[dB]	BD-Rate[%]
Friends	-3.60	77.55
Fountain	0.06	1.52
Stone	-5.58	83.28
Danger	-1.21	31.41
Bikes	-2.76	58.63
Average	**-2.62**	**50.48**

Light Field	BD-PSNR[dB]	BD-Rate[%]
Friends	-	-
Fountain	1.70	4.14
Stone	-2.99	73.48
Danger	0.51	5.95
Bikes	1.38	17.91
Average	**0.15**	**25,37**

To conclude this section, Figure 79 shows some low-frequency bands obtained with the DCLFC H2_V2 solution for the *Friends* and *Danger light* fields; they show that despite the 4-levels of decomposition applied, the low-frequency bands still look good.

Figure 79: *Examples of low-frequency bands for DCLFC H2_V2 for: left) Friends; right) Danger.*

4.3.4. Quality Scalable Stream Study

This chapter ends with a brief study on the scalability capabilities and its impact on the RD performance of the best achieved solution, this means the **DCLFC H2_V2** configuration. The added value of using a quality scalable stream is the possibility to easily decode from the same bitstream a number of different output qualities for the SA images, notably by just using a subset of the available bitstream. For the non-scalable case, the DCLFC solution was applied using just one quality layer, defined by the value of the CDQP applied; this means the JPEG 2000 is configured to provide a single scalability layer. The scalable stream case is achieved by controlling the JPEG 2000 codec asking for three quality layers; in this case, the CDQP value defines the first quality layer and the remaining two layers are defined by smaller CDQP values. While the non-scalable case allows decoding images with a single quality, the scalable case allows decoding images with three different qualities.

The results in Figure 80 show a slightly decrease in the RD performance when more scalability layers are requested, especially for the higher rates. The decrease in quality for the higher rates may be due to the difference in the compression parameter values used for each SA image or band; this implies that, for a given quality layer, some of the CDQP values may significantly change, thus degrading the light field quality reconstruction.

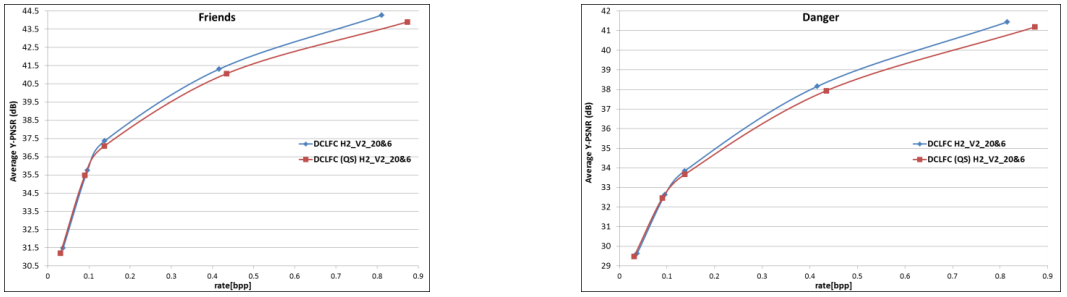


Figure 80: *RD results with or not quality scalability for: left) Friends; right) Danger.*

Lastly, Figure 81 shows an example of quality scalability; while Figure 81 left) shows the lowest layer decoded image, thus with the lowest quality, Figure 81 right) shows the last layer, using the totality of the available stream, thus yielding the best quality.

Figure 81: *Danger light field: left) SA image decoded with layer 1 stream; right) SA image decoded with layer 3 stream.*

Chapter 5

5. Conclusions and Future Work

The final chapter of this Thesis is divided in: ii) conclusions: the most relevant conclusions of this Thesis are highlighted; iii) future work: description of some work that may improve the proposed codec RD performance.

5.1. Conclusions

As shown in Chapter 4, it can be concluded that proposed solution improves the RD performance of the standard JPEG and JPEG 2000. Despite the light field content having a strong impact on the RD performance, the adopted solution improves the RD performance of every light field evaluated for all bitrates when compared with just Intra coding. From the RD performance assessment, it can be concluded:

1. As expected, the designed disparity compensated lifting based inter-view wavelet transform is able to improve the RD performance compared with only Intra coding.
2. The RD performance remained very similar when the proposed solution is applied to the horizontally or vertically to the 2D array of SA images.
3. Regarding the RD performance when quality scalability is provided, there is a slight decrease in the RD performance of the proposed solution when compared to the situation where only view scalability is provided. The result was expected, as offering quality scalability comes with a price due to the extra stream overhead.
4. The best codec configuration in terms of RD performance is to apply the proposed solution along a set of SA images horizontally and then along a set of low-frequency bands vertically, alternating between the two dimensions for further levels of decomposition. Also, the best codec configuration is to apply the proposed image codec with 4-levels of overall decomposition.
5. When there is more than one decomposition level, the inter-view transform can be applied to both low-frequency and high-frequency bands. However, it was verified experimentally, that the RD performance did not increase when compared to the case where the inter-view transform is only applied the low-frequency bands.
6. The homography parameters can be computed using the original SA images or low-frequency bands. The RD performance of the proposed solution was studied in the two cases and using the low-frequency bands was the best solution. Moreover, it is not possible to compute the homography parameters taking the high-frequency bands as input because the keypoint detection fails.

5.2. Future Work

Considering the objectives that the DCLFC solution aims to offer such as view scalability while still maintaining a high compression efficiency, the following research directions could be pursued:

- **Wavelet Transform**: as the solution seems to have reached saturation in terms of RD performance with 4-levels of overall decomposition to improve the RD performance while still requiring scalability,

a more complex wavelet transform as in [83] could be adopted. The biorthogonal Cohen-Daubechies-Feauveau 5/3 (CDF 5/3) wavelet provides a better RD performance compared to the Haar wavelet, in some cases and therefore it is expected to perform well for lenslet light field compression. The inter-view transform performance may increase because the CDF 5/3 improves the compression performance due to its bidirectional prediction and update steps, meaning that three neighboring SA images or low-frequency bands are used as input of the transform. Naturally, this comes at an increased computational cost.

- **Homography Parameters**: to improve the RD performance, several geometric (perspective) transformations can be used in different regions of the light field SA images and thus a better prediction can be obtained. This means that the residual that is coded (high-frequency band) will have a lower energy and therefore some bitrate savings can be achieved for the same target decoded quality. The estimation of how many perspective transformations and their respective parameters should be performed with a rate-distortion criterion. Naturally, this approach requires the transmission of some extra overhead (more parameters) but it may be advantageous since far away and close by regions (from the camera) in a light field are typically well predicted with different perspective transforms.

If spatial or quality scalability is not required, the HEVC Intra codec may also be used as base encoder for the proposed solution. HEVC is a more advanced codec, compared to JPEG 2000, and follows the block-based hybrid design of all major video coding standards. Each block of a picture is either coded in intra mode, i.e. without referring to other pictures of the video sequence, or it is temporally predicted, i.e. inter mode. In this case, the resulting bands of the proposed inter-view transform can be encoded with HEVC Intra and therefore, view scalability provided. Thus, the inter-view correlation is exploited in the same way as the presented in this Thesis and now the intra-view correlation is exploited using HEVC Intra instead of JPEG 2000. Another possibility is to use the scalable extension of HEVC, to provide the same scalability types as the proposed solution.

References

[1] THORLABS, "Microlens Arrays," [Online]. Available: https://www.thorlabs.de/newgrouppage9.cfm?objectgroup_id=2861. [Accessed 10 09 2017].

[2] L. Forum. [Online]. Available: http://lightfield-forum.com/tag/hacks/. [Accessed 2017 08 27].

[3] E. Carstens, "Lytro Illum Light Field Camera," [Online]. Available: http://www.dudeiwantthat.com/exclusives/lytro-illum-light-field-camera.asp. [Accessed 2017 08 27].

[4] "ExtremeTech," [Online]. Available: https://www.extremetech.com/extreme/191909-google-joins-the-vr-war-invests-in-light-field-cinematic-reality-company-magic-leap. [Accessed 2017 08 27].

[5] JBIG and JPEG, "JPEG Pleno Call for Proposals on Light Field Coding," Doc. number 73013, Changde, China, Oct. 2016.

[6] F. Pereira, E. A. Silva and G. Lafruit, "Plenoptic Imaging: Representation and Processing," R. Chellappa and S.Theodoris, editors, Academic Press Library in Signal Processing.

[7] E. M. Slayter and H. S. Slayter, "Wave interactions," in *Light and Electron Microscopy*, C. , United Kingdom, Cambridge University Press, 1992, pp. 25-33.

[8] Bergen, E. H. Adelson and J. T., "The Plenotpic Funtion and the Elements of Early Vision," MIT Press, Cambridge, Mass., 1991.

[9] G. Wetzstein, I. Ihrke, D. Lanman and W. Heidrich, "Computational Plenoptic Imaging," in *ACM SIGGRAPH* , New York, USA, 2012.

[10] S. J. Gortler et all, "The Lumigraph," in *Proc. ACM SIGGRAPH, pp. 43-54.*, New Orleans, USA, Aug. 1996.

[11] M. Levoy, "Light Fields and Computational Imaging," *Computer,* vol. 39, no. 8, Aug. 2006.

[12] "Illum," [Online]. Available: https://illum.lytro.com/illum. [Accessed 12 09 2016].

[13] F. Dai, J. Zhang and Y. Ma, "Lenslet Image Compression Scheme Based on Subaperture Images Streaming," in *IEEE International Conf. on Image Processing*, Québec City, Québec, Canada, Sep. 2015.

[14] C. Conti, L. D. Soares and P. Nunes, "3D Holoscopic Video Representation and Coding Technology," in *Novel 3D Media Technologies*, New York, Springer, 2014, p. ch. 5.

[15] "Seagul," [Online]. Available: http://www.tgeorgiev.net/795.jpg. [Accessed 17 09 2016].

[16] "Point Cloud," [Online]. Available: https://en.wikipedia.org/wiki/Point_cloud#/media/File:Geo-Referenced_Point_Cloud.JPG. [Accessed 23 08 2016].

[17] "Graphics, Geometry & Multimedia," [Online]. Available: https://www.graphics.rwth-aachen.de/person/37/. [Accessed 04 08 2016].

[18] "MVD format," [Online]. Available: https://cagnazzo.wp.mines-telecom.fr/?page_id=814&lang=en, accessed on 04/08/2016.. [Accessed 2016 08 04].

[19] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz and P. Hanrahan, "Light Field Photography with a Hand-Held Plenoptic Camera," Tech. Rep. CSTR 2005-02, Stanford University, Stanford, CA , USA, Apr. 2005.

[20] "Lytro," [Online]. Available: https://www.lytro.com/imaging. [Accessed 07 09 2016].

[21] "LIGHT FIELD CAMERA TECHNOLOGY," [Online]. Available: https://www.raytrix.de/technologie/. [Accessed 07 09 2016].

[22] T. Georgiev and A. Lumsdaine, "The Multi-Focus Plenoptic Camera," in *SPIE Electronic Imaging*, Burlingame, CA, USA, Jan. 2012.

[23] "Microlens-arrays," [Online]. Available: http://www.powerphotonic.com/products/micro-lens-arrays/microlens-arrays. [Accessed 08 09 2016].

[24] R. Ng, "Digital Light Field Photography," Ph.D. Thesis, Stanford University, Stanford, CA, USA, Jul. 2006.

[25] A. Lumsdaine and T. Georgiev., "Full Resolution Lightfield Rendering," Tech. Rep., Indiana University and Adobe Systems, Bloomington, IN, USA, Jan. 2008.

[26] "Digital Camera," [Online]. Available: https://en.wikipedia.org/wiki/Digital_camera. [Accessed 08 09 2016].

[27] Z. Yu, J. Yu, A. Lumsdaine and T. Georgiev, "An Analysis of Color Demosaicing in Plenoptic Cameras," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA, Jun. 2012.

[28] "Chromatic aberration," [Online]. Available: https://en.wikipedia.org/wiki/Chromatic_aberration. [Accessed 17 09 2016].

[29] "Lens aberrations," [Online]. Available: http://www.kshitij-iitjee.com/lens-aberrations. [Accessed 2016 09 08].

[30] D. G. Dansereau, O. Pizzaro and S. B. Williams, "Decoding, Calibration and Rectification for Lenslet-Based Plenoptic Cameras," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Portland, OR, USA, Jun. 2013.

[31] "Chromatic aberration lens diagram," [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/a/aa/Chromatic_aberration_lens_diagram.svg. [Accessed 08 09 2016].

[32] A. Lumsdaine and T. Georgiev, "The Focused Plenoptic Camera," in *IEEE Intl. Conf. on Computational Photography* , San Francisco, CA, USA, Apr. 2009.

[33] D. G. Dansereau, "Light Field Toolbox for Matlab," Feb. 2015. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4. [Accessed 04 12 2016].

[34] F. Murgia, S. Fernandez, D. Giusto and C. Perra, "Unfocused Plenoptic Camera Calibration," in *Telecommunications Forum Telfor*, Belgrade, Serbia, Nov. 2014.

[35] D. Cho, M. Lee, S. Kim and Y.-W. Tai, "Modeling the Calibration Pipeline of the Lytro Camera for High Quality Light-Field Image Reconstruction," in *IEEE International Conf. on Computer Vision*, Sydney, Australia, Dec. 2013.

[36] M. Rerabek, T. Bruylants, T. Ebrahimi, F. Pereira and P. Schelkens, "Call for Proposals ICME 2016 Grand Challenge: Light-Field Image Compression," Seattle, WA, USA, Jul. 2016.

[37] "LFDecodeLensletImageSimple," [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/48405-deprecated-light-field-toolbox-v0-3-v0-4-now-available/content/LFToolbox0.3/SupportFunctions/LFDecodeLensletImageSimple.m. [Accessed 27 09 2016].

[38] "Soft Light," [Online]. Available: https://en.wikipedia.org/wiki/Soft_light. [Accessed 21 09 2016].

[39] "Diffuser," [Online]. Available: https://en.wikipedia.org/wiki/Diffuser_(optics). [Accessed 21 09 2016].

[40] "LFBuildLensletGridModel," [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/48405-deprecated-light-field-toolbox-v0-3-v0-4-now-available/content/LFToolbox0.3/SupportFunctions/LFBuildLensletGridModel.m. [Accessed 27 09 2016].

[41] C. Heinze et al., "Automated Robust Metric Calibration Algorithm for Multifocus Plenoptic Cameras," *IEEE Transactions on Instrumentation and Measurement,* vol. 65, no. 5, pp. 1197 - 1205, May 2016.

[42] "Single Camera Calibration App," [Online]. Available: http://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html. [Accessed 21 09 2016].

[43] "Filtro de Bayer," [Online]. Available: http://wikipedia.qwika.com/en2pt/Bayer_filter. [Accessed 21 09 2016].

[44] I. Ihrke, J. Restrepo and L. Mignard-Debise, "Principles of Light Field Imaging: Briefly Revisiting 25 years of Research," *IEEE Signal Processing Magazine,* vol. 33, no. 5, pp. 59 - 69, Sept. 2016.

[45] M. Seifi, N. Sabater, V. Drazic and P. Perez, "Disparity-Guided Demosaicking of Light Field Images," in *IEEE International Conf. on Image Processing*, Paris, France, Oct. 2014.

[46] T. Bishop, Z. Zanetti and P. Favaro, "Light Field Superresolution," in *IEEE International Conf. on Computational Photography*, San Francisco, CA, USA, Apr. 2009.

[47] S. Shi, P. Gioia and G. Madec, "Efficient Compression Method for Integral Images Using Multi-view Video Coding," in *IEEE International Conf. on Image Processing*, Brussels, Belgium, Sep. 2011.

[48] "Raytrix," [Online]. Available: https://www.raytrix.de/. [Accessed 10 10 2016].

[49] "Lytro Illum," [Online]. Available: https://illum.lytro.com/illum. [Accessed 10 03 2016].

[50] "LYTRO FIRST GEN CAMERA," [Online]. Available: http://smithbw.com/2015/08/lytro-first-gen-camera/. [Accessed 10 31 2016].

[51] "Lytro Illum," [Online]. Available: https://illum.lytro.com/wedding. [Accessed 21 09 2016].

[52] "Raytrix Products," [Online]. Available: https://www.raytrix.de/produkte/#r42series. [Accessed 21 09 2016].

[53] "Lytro megaray," [Online]. Available: https://support.lytro.com/hc/en-us/articles/200863210-What-is-a-megaray-. [Accessed 10 03 2016].

[54] "Lytro Desktop App," [Online]. Available: https://illum.lytro.com/desktop#focusSpreadVid. [Accessed 18 10 2016].

[55] J. Lino, "2D Image Rendering for 3D Holoscopic Content using Disparity-Assisted Patch Blending," Ph.D. Thesis, Instituto Superior Técnico, Lisboa, Portugal, Oct. 2013.

[56] H. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. Tai and I. Kweon, "Accurate Depth Map Estimation from a Lenslet Light Field Camera," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, Jun. 2015.

[57] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet and F. Dufaux, "Full Parallax 3D Video Content Compression," in *Novel 3D Media Technologies*, New York City, NY, USA, Springer, Oct. 2014, pp. 49-70.

[58] T. Georgiev and A. Lumsdaine, "Focused Plenoptic Camera and Rendering," *Journal of Electronic Imaging,* vol. 19, no. 2, Apr. 2010.

[59] T. Georgiev, G. Chunev and A. Lumsdaine, "Superresolution with the Focused Plenoptic Camera," in *SPIE Computational Imaging IX*, San Francisco, CA, USA, Jan. 2011.

[60] "Raytrix 3D light field video," 10 02 2012. [Online]. Available: https://www.youtube.com/watch?v=msGZOjzreP8. [Accessed 18 10 2016].

[61] R. Ng, "Fourier Slice Photography," *ACM Transactions on Graphics,* vol. 24, no. 3, p. 735–744, Jul. 2005.

[62] G. Alves, F. Pereira and E. Silva, "Light Field Imaging Coding: Performance Assessment Methodology and Standards Benchmarking," in *IEEE International Conf. on Multimedia & Expo Workshops*, Seattle, WA, USA, Jul. 2016.

[63] R. Higa, R. Chavez, R. Leite, R. Arthur and Y. Iano, "Plenoptic Image Compression Comparison Between JPEG, JPEG2000 and SPITH," *Journal of Selected Areas in Telecommunications,* vol. 3, no. 6, Jun. 2013.

[64] T. Wiegand, G. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13, no. 7, pp. 560 - 576, Jul. 2003.

[65] J. Vanne, M. Viitanen, T. Hamalainen and A. Hallapuro, "Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 22, no. 12, pp. 1885 - 1898, Dec. 2012.

[66] G. Sullivan , J. Ohm , W. Han and T. Wiegand , "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 22, no. 12, pp. 1649 - 1668, Sep. 2012.

[67] A. Vieira, H. Duarte, C. Perra, L. Tavora and P. Assunção, "Data Formats for High Efficiency Coding of Lytro-Illum Light Fields," in *International Conf. on Image Processing Theory, Tools and Applications*, Orléans, France, Nov. 2015.

[68] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu and W. Zeng, "Pseudo-Sequence-Based Light Field Image Compression," in *IEEE International Conf. on Multimedia & Expo Workshops*, Seattle, WA, USA, Jul. 2016.

[69] A. Vetro, T. Wiegand and G. Sullivan, "Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC," *Proceedings of the IEEE,* vol. 99, no. 4, pp. 626 - 642, Jan. 2011.

[70] C. Conti, P. Nunes and L. Soares, "HEVC-Based Light Field Image Coding with Bi-Predicted Self-Similarity Compensation," in *IEEE International Conf. on Multimedia & Expo Workshops*, Seattle, WA, USA, Jul. 2016.

[71] R. Monteiro, L. Lucas, C. Conti, P. Nunes, N. Rodrigues, S. Faria, C. Pagliari, E. Silva and L. Soares, "Light Field HEVC-Based Image Coding Using Locally Linear Embedding And Self-Similarity Compensated Prediction," in *IEEE International Conf. on Multimedia & Expo Workshops* , Seattle, WA, USA, Jul. 2016.

[72] C. Conti, P. Nunes and L. Soares, "New HEVC Prediction Modes for 3D Holoscopic Video Coding," in *IEEE International Conf. on Image Processing*, Orlando, FL, USA, Oct. 2012.

[73] J. Ohm, G. Sullivan, H. Schwarz, T. Tan and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 22, no. 12, pp. 1669 - 1684, Dec. 2012.

[74] M. Flierl, T. Wiegand and B. Girod, "A Locally Optimal Design Algorithm for Block-Based Multi-Hypothesis Motion-Compensated Prediction," in *Data Compression Conf.*, Snowbird, UT, USA, Apr. 1998.

[75] L. Lucas, C. Conti, P. Nunes, L. Soares, N. Rodrigues, C. Pagliari, E. Silva and S. Faria, "Locally Linear Embedding-based Prediction for 3D Holoscopic Image Coding Using HEVC," in *European Signal Processing Conf.*, Lisboa, Portugal, Sep. 2014.

[76] C. Choudhury and S. Chaudhuri, "Disparity Based Compression Technique for Focused Plenoptic Images," in *Indian Conf. on Computer Vision Graphics and Image Processing*, Bangalore, KA, India, Dec. 2014.

[77] H. Zayed, S. Kishk and H. Ahmed, "3D Wavelets with SPIHT Coding for Integral Imaging Compression," *International Journal of Computer Science and Network Security,* vol. 12, no. 1, pp. 126 - 133, jan. 2012.

[78] A. Aggoun and M. Mazri, "Wavelet-based Compression Algorithm for Still Omnidirectional 3D Integral Images," *Signal, Image and Video Processing,* vol. 2, no. 2, p. 141–153, Jun. 2008.

[79] H. Kang, D. Shin and E. Kim, "Compression Scheme of Sub-images Using Karhunen-Loeve Transform in Three-dimensional Integral Imaging," *Optics Communications,* vol. 281, no. 14, p. 3640–3647, Jul. 2008.

[80] E. Elharar, A. Stern, O. Hadar and B. Javidi, "A Hybrid Compression Method for Integral Images Using Discrete Wavelet Transform and Discrete Cosine Transform," *Journal of Display Technology,* vol. 3, no. 3, pp. 321 - 325, Aug. 2007.

[81] S. Kishk, H. Ahmed and H. Helmy, "Integral Images Compression Using Discrete Wavelets and PCA," *International Journal of Signal Processing, Image Processing and Pattern Recognition,* vol. 4, no. 2, pp. 65 - 77, Jun. 2011.

[82] A. Aggoun, "Compression of 3D Integral Images Using 3D Wavelet Transform," *Journal of Display Technology,* vol. 7, no. 11, pp. 586 - 592, Sep. 2011.

[83] C. Chang, X. Zhu and P. Ramanathan, "Light Field Compression using Disparity-Compensated Lifting and Shape Adaptation," *IEEE Transactions on Image Processing,* vol. 15, no. 4, pp. 793 - 806, Mar. 2006.

[84] Wikipedia, "k-means clustering," [Online]. Available: https://en.wikipedia.org/wiki/K-means_clustering. [Accessed 20 03 2017].

[85] M. Levoy, "Light Fields and Computational Photography," http://graphics.stanford.edu, [Online]. Available: http://graphics.stanford.edu/projects/lightfield/. [Accessed 30 08 2017].

[86] T. Vijayan, "Performance Image Compression using Lifting based EEWITA," [Online]. Available: http://www.rroij.com/open-access/performance-image-compression-usinglifting-based-eewita.php?aid=44433. [Accessed 10 01 2017].

[87] P. Eisert, E. Steinbach and B. Girod, "Automatic Reconstruction of Stationary 3-D Objects from Multiple Uncalibrated Camera Views," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 10, no. 2, pp. 261-267, Mar. 2000.

[88] W. Chen, J. Bouget, M. Chu and R. Grzeszczuk, "Light Field Mapping: Efficient Representation and Hardware Rendering of Surface Light Fields," in *Proc. of SIGGRAPH Computer Graphics and Interactive Techniques*, San Antonio, TX, USA, Jul. 2002.

[89] C. Chang, X. Zhu, P. Ramanathan and B. Girod, "Shape Adaptation for Light Field Compression," in *Proc. of International Conf. on Image Processing*, Barcelona, Spain, Sep. 2003.

[90] M. Magnor and B. Girod, "Model-Based Coding of Multiviewpoint Imagery," in *Proc. of SPIE Visual Communications and Image Processing*, Perth, Australia, Jun. 2000.

[91] R. Singh and V. K. Srivastava, "JPEG2000: A Review and its Performance Comparison with JPEG," in *Power Control and Embedded Systems 2nd International Conf.*, Allahabad, Uttar Pradesh, India, Dec. 2012.

[92] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image Coding Using Wavelet Transform," *IEEE TRANSACTIONS ON IMAGE PROCESSING,* vol. 1, no. 2, pp. 205-220, Apr. 1992.

[93] W. Sweldens, "The Lifting Scheme: Construction Of Second Generation Wavelets," *SIAM J. Math. Anal.,* vol. 29, no. 2, pp. 511-546, 1997.

[94] D. Srikanth and M. Mittal, "Implementation of Haar Wavelet through Lifting For Denoising," *Int. Journal of Engineering Research and Applications,* vol. 3, no. 5, pp. 1311-1314, Oct. 2013.

[95] I. Daubechies and W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps," *Journal of Fourier Analysis and Applications,* vol. 4, no. 3, pp. 247 - 269, May 1998.

[96] M. Darbois, "DocJ2KCodec," OPENJPEG, [Online]. Available: https://github.com/uclouvain/openjpeg/wiki/DocJ2KCodec. [Accessed 15 08 2017].

[97] D. Taubman and M. Marcellin, "The 2D DWT," in *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Dordrecht, NE, Kluwer Academic Publishers, 2002, pp. 428 - 431.

[98] C. Christopoulos and A. Skodras, "The JPEG2000 Still Image Coding System: An Overview," *IEEE Transactions on Consumer Electronics,* vol. 46, no. 4, pp. 1103-1127, Nov.2000.

[99] M. Ghobadi, "Wavelet-Based Coding and its Application in JPEG000," SlidePlayer, [Online]. Available: http://slideplayer.com/slide/7978793/. [Accessed 15 08 2017].

[100] M. Marcellin , M. Lepley, A. Bilgin, T. Flohr, T. Chinen and J. Kasner, "An Overview of Quantization in JPEG 2000," *Signal Processing: Image Communication,* vol. 17, no. 1, p. 73–84, Jan. 2002.

[101] "Introduction to SIFT (Scale-Invariant Feature Transform)," OpenCV, [Online]. Available: http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html#keypoint-matching. [Accessed 27 01 2017].

[102] D. Lowe, "Distinctive Image Features from Scale-Invariant Key," *International Journal on Computer Vision,* vol. 60, no. 2, pp. 91-110, Jan. 2004.

[103] G. Zhao and Q. Song, "SIFT Image Stitching Technology Based on Music Score Scanning Recognition System," in *International Symposium on Computational Intelligence and Design* , Hangzhou, China, Dec. 2016.

[104] P. Monteiro, "Distributed Video Coding with Geometric Transforms," Ph.D. Thesis, Instituto Superior Técnico, Lisboa, Portugal, Mar. 2013.

[105] S. Zhao and C. Zhibo, "Lenslet Light Field Image Coding with Linear View Synthesis," ISO/IEC JTC 1/SC29/WG1 Coding of Still Pictures, EEIS Dept., University of science and technology of China, 2017.

[106] I. Tabus, P. Helin and P. Astola, "Lossy Compression of Lenslet Images From Plenoptic Cameras Combinning Sparse Predictive Coding and JPEG 2000," in *75th JPEG Meeting* , Sydney, AUS, Jul. 2017.

[107] Z. Alpaslan, S. Cen, W. Liu, R. Matsubara, H. El-Ghoroury and D. McNeil, "Ostendo Full Parallax Light Field Compression Codec for JPEG Pleno," Ostendo Technologies, Inc., Carlsbad, CA, Jun. 2017.

[108] M. S. P. Group, "Light-Field Image Dataset," [Online]. Available: http://mmspg.epfl.ch/EPFL-light-field-image-dataset. [Accessed 12 06 2017].

[109] JPEG, ISO/IEC JTC 1/SC29/WG1, "JPEG Call for Proposals on Light Field Coding," Doc. N74014, Geneva, CHE, Jan. 2017.

[110] "OpenCV," OpenCV, [Online]. Available: http://opencv.org/. [Accessed 11 08 2017].

[111] (UCL), Université de Louvain, "OpenJPEG," OpenJPEG, [Online]. Available: http://www.openjpeg.org/. [Accessed 11 08 2017].

[112] B. G., "Calculation of Average PSNR Differences Between RD-curves," VCEG Contribution VCEG-M33, Austin, Texas, Apr. 2011.

[113] A. Lumsdaine, L. Lin, J. Willcock and Y. Zhou, "Fourier Analysis of the Focused Plenoptic Camera," in *SPIE 8667 Multimedia Content and Mobile Devices*, Burlingame, CA, USA, Mar. 2013.

[114] "What is light field," [Online]. Available: http://blog.lytro.com/post/132599659620/what-is-light-field. [Accessed 09 09 2016].

[115] "Panasonic," [Online]. Available: http://av.jpn.support.panasonic.com/support/global/cs/dsc/knowhow/knowhow11.html. [Accessed 17 09 2016].

Appendix A

A. Lenslet Light Field Camera Architecture

Concerning the placement of the photosensor, there are two main architectures for a lenslet light field camera so-called ***focused*** and **unfocused**. The light field camera architecture considered in this Thesis is named unfocused [22] or plenoptic camera [36]. In this type of camera, the main (objective) lens focuses its image on the plane of the microlens array. In this case, each micro-image is defocused with respect to the image created by the main lens and the outside object; as the microlens array is focused at its optical infinity, as shown in Figure 82 left), therefore the resultant micro-images have low resolution [32]. The focused lenslet light field camera, often called plenoptic camera 2.0 [41], is called this way because the microlens array is placed in such a way that it focuses the image created by the main lens; this is the opposite of the unfocused approach, where the microlens focuses its optical infinity. As mentioned in [58] "in the conventional plenoptic camera, all of the directions for a given spatial sample are contained within a single micro-image and all of the spatial samples for a given direction are spread across micro-images". On the contrary, in the focused plenoptic camera, the different views for a given spatial sample are spread across micro-images. There are two possible configurations for the focused lenslet light field camera, as shown in Figure 82 mid) and right), the difference lying in the placement of the microlens array, which can be placed after or before the main lens focal plane [32].

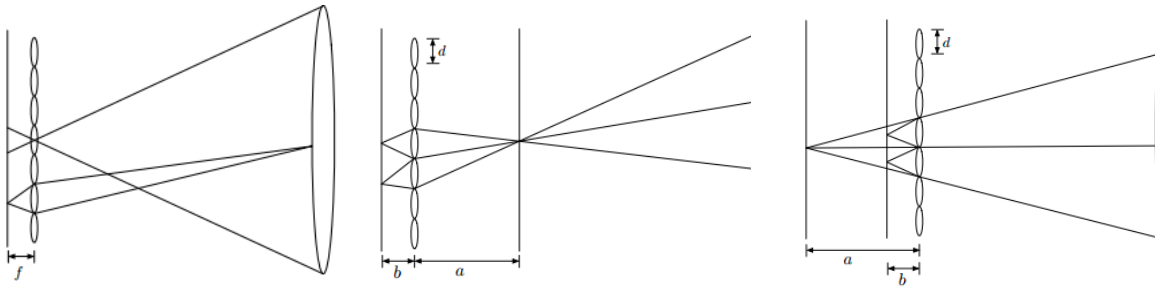


Figure 82: *left) Microlens focusing its optical infinity and main lens focusing on the microlens array [32]; mid) First approach, in where the microlens array is focused on a real image [32]; right) Second approach where the microlens array is focused on a virtual image [32].*

Despite having similar architectures, the fundamental difference between the two types of light field camera architectures comes from the microlens array placement. The *unfocused* lenslet light field camera architecture places the microlens array at the focus plane of the main lens (main lens image plane), as shown in Figure 83 left), at distance *f* from the photosensor, where *f* is the focal distance of each individual microlens. On the contrary, the focused approach places it behind or in front of the microlens array; this way, the microlens array forms a relay system with the main lens, meaning that the acquired micro-images are inverted and sharper. Figure 83 right) shows the most common focused lenslet light field camera architecture, where the photosensor is placed at a distance *b* behind the microlens array and focuses the main lens at distance *a* in front of it, where *a, b*, and *f* (the focal distance of each individual microlens.) satisfy the lens equation $\frac{1}{a}+\frac{1}{b}=\frac{1}{f}$ [113]. The focused approach leads to

a loss in angular resolution as a trade-off for an improvement in spatial resolution; for further details, consult [22].

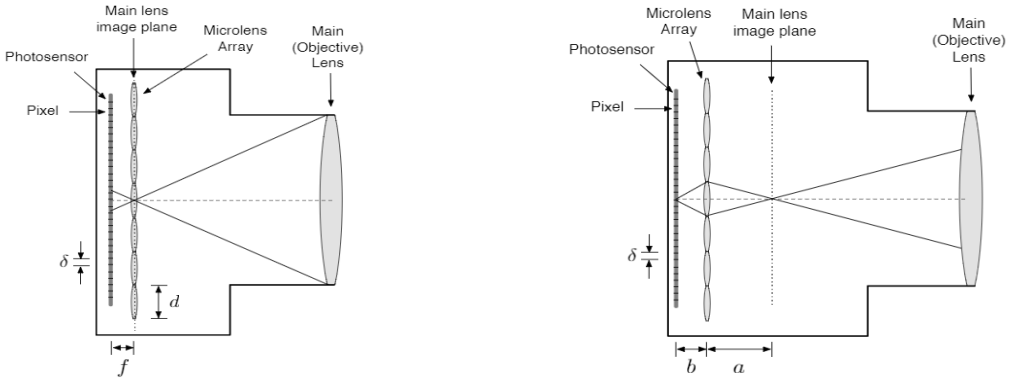


Figure 83: *left) Unfocused lenslet light field camera, where the microlens array is spaced by the distance f from the photosensor [113]; right) Focused approach, where the microlens array is placed at a distance b from the photosensor and a from the main lens image plane [113].*

The following procedures are intended for the **unfocused** lenslet light field camera's architecture. When a photo is taken with a lenslet light field camera, the light rays coming from the outside scene pass through the main lens and converges to the microlens array. There, the arriving light rays are refracted to then strike the photosensor, implying that each microlens records a small resolution image. Figure 84 represents the light path inside a lenslet light field camera illustrating the rays from the moment they enter the camera until they hit the photosensor. Each microlens is mapped to a specific number of pixels, with each pixel corresponding to a different ray of light and thus a different direction. Naturally, the baseline, i.e. the distance between the microlenses, is limited by the size of the microlens array, and the full photosensor resolution is shared by all the microlenses in the array.

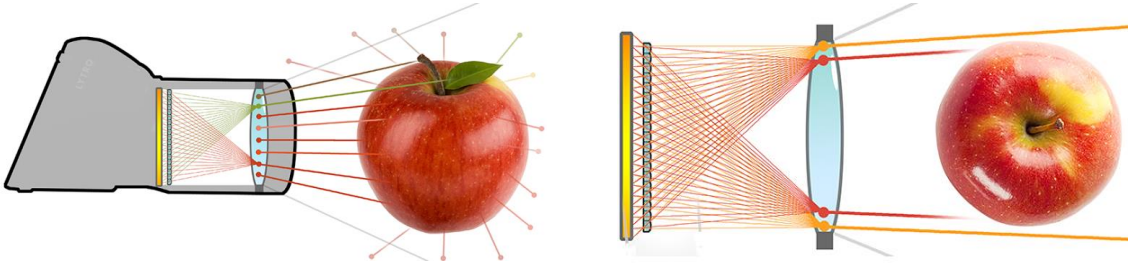


Figure 84: *left) Illustration of light rays mapping in a lenslet light field camera such as Lytro Illum [114]; right) Zoom in of the previous image detailing the light path inside the camera [114].*

Ideally, each microlens should cover the largest number of photosensor pixels to provide a micro-image with the highest angular resolution [19]. However, by doing this, the number of micro-images would be reduced for the same overall photosensor resolution, and so the spatial resolution for each angular direction. There is, thus, an important trade-off between spatial and angular information; a low angular resolution will lead to severe aliasing artifacts in refocusing and a low spatial resolution will result in rendered images with poor quality. The following steps describe how to optimize the microlenses' optics, for the so-called unfocused light field camera, to maximize the angular resolution without negatively impacting too much the spatial resolution [24]:

1. *Choosing the **f-number** of the main lens and microlenses*. Usually, the *f*-number is the aperture diameter divided by the focal length; however, in this case, the *f*-number for the main lens is the diameter of the main lens divided by the separation between the main lens plane and the microlenses

plane. Figure 85 left) illustrates the concepts of aperture diameter and focal length for a single lens. As the aperture diameter corresponds to the lens diameter where the light effectively passes through, it is possible to choose a different depth of field (which implies varying the amount of objects in focus) by regulating it. Figure 85 right) illustrates different *f*-numbers for the main lens, allowing to better understand how it is a function of the aperture diameter and focal length. When the *f*-numbers of the main lens and the microlens match, the micro-image is the largest possible, without overlapping with neighboring micro-images, as shown in Figure 86 left); for further details, consult [19]. In Figure 86 left), there is another example for similar conditions but now with a *f*-number discrepancy between the main lens and the microlenses, in this case *f*/4 for the microlenses and *f*/2,8 for the main lens, thus leading to overlapping micro-images and consequently wasting of resolution resources.

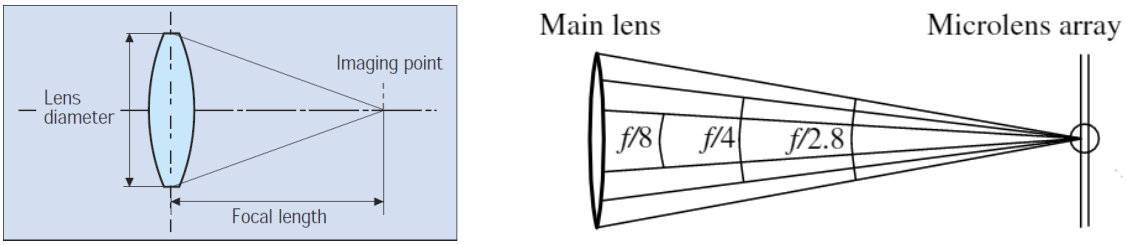


Figure 85: *left) Illustration for the concepts of focal length and lens diameter [115]; right) Illustration of different f-number configurations [19].*

2. *Optimizing the* ***location*** *of the microlens array and photosensor.* The microlens array is placed where the rays coming from the main lens converge, so it can separate them into the sensor behind it. The photosensor is placed on the focal plane of the microlens array, so that each microlens is focusing at its optical infinity (main lens principal plane) [27]. Figure 86 right) illustrates the main lens' and the microlens array planes' positioning for (a) the unfocused lenslet light field camera considered in this Thesis and (b) for the focused light field camera proposed by Lumsdaine et al. [22]. The separation between the microlens array and the photosensor should be accurate to $\Delta x_p . \left(\frac{f_m}{\Delta x_m}\right)$ where Δx_p is the sensor pixel length, f_m the microlens focal depth and, finally, Δx_m the length of the microlens [19]. If these requirements aren't fulfilled, the result is misfocus, thus blurring the micro-images.

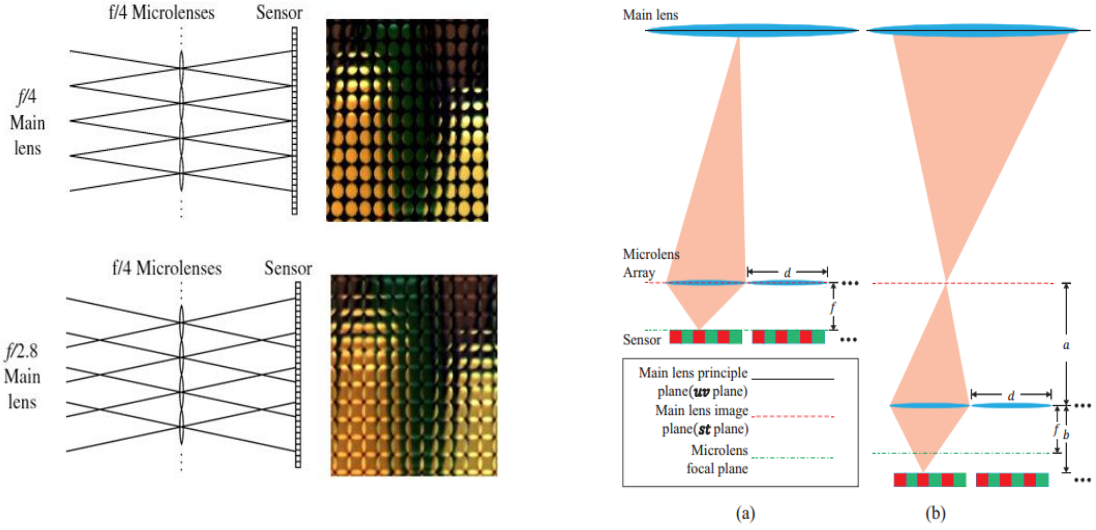


Figure 86: *left up) Representation of micro-images corresponding to equal f-numbers; left bottom) same situation for different f-numbers [27]); right) Illustration of the principal planes forming the optical system in both lenslet light field camera architectures, a) unfocused and b) focused [27]).*

Appendix B

B. DCLFC Applied to Low-frequency and High-frequency Bands

From Chapter 4, the main conclusion was that applying the DCLFC solution to some number of low-frequency bands increases the RD performance, so it seems interesting to analyze the RD performance impact of also applying the DCLFC solution to the high-frequency bands. While it is known that the low-frequency bands may benefit with the inter-transform decomposition, as there is still high correlation between the low-frequency bands, it is not clear if the same effect applies for the high-frequency bands. Figure 87 shows the RD performance for the **DCLFC H1_V1** and **DCLFC H2** configurations as these are the simplest DCLFC configurations where it is possible to study the impact of applying the inter-transform decomposition to the high-frequency bands. Moreover, the *Friends* and *Danger* light fields have been selected for this study as, on average, the first presented above the best and worst RD performances.

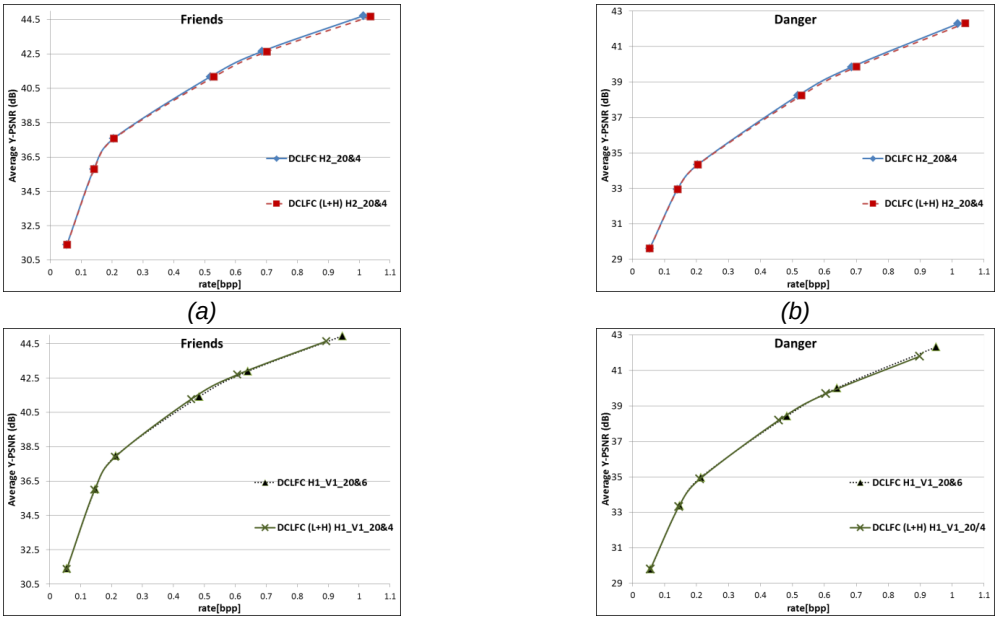


Figure 87: *DCLFC RD results with the inter-transform applied to the high-frequency bands for: a) Friends, DCLFC H2; b) Danger, DCLFC H2; c) Friends, DCLFC H1_V1; d) Danger, DCLFC H1_V1.*

As shown in Figure 87, the RD performance is not significantly changed when the DCLFC solution implements the inter-transform for both the low-frequency and high-frequency bands. The application of the inter-transform to the high-frequency bands implies more rate spent with homography parameters; moreover, the decomposition of the high-frequency bands decreases the reconstruction quality for the high rates, as seen in Figure 87 for the DCLFC H2 solution. A similar behavior is shown for the DCLFC H1_V1 solution, except for the *Friends* light field where there is a small improvement on the RD performance over DCLFC H1_V1, although just for rates between 0,4 and 0,6 bpp. In summary, there

are no significant advantages associated to the application of the DCLFC framework to the high-frequency bands, as so this feature is not further exploited.

Appendix C

C. Homography Parameters Related Performance

In this section, the objective is to assess the RD impact of using the SA images instead of the low-frequency bands to compute the homography parameters associated to the disparity estimation. While the first level of decomposition must take the SA images as the input for disparity estimation, for the higher levels of decomposition the low-frequency bands may replace the SA images as input for disparity estimation. An alternative approach to what was proposed in Chapter 4 consists in always using the SA images as input for disparity estimation; to measure the disparity between low-frequency bands, the SA images selection has to consider the corresponding low-frequency bands positions, as shown in Figure 88. The first case regards the bands configuration after a DCLFC solution with 1-level of decomposition is applied; in this case, the disparity estimation is performed taking as input two neighboring SA images, i.e. the SA images in positions 0 and 1 are replaced by the low-frequency band in position 0 and the high-frequency band in position 1, as shown in Figure 88 by the blue arrow. The remaining cases show the result of increasing the number of decomposition levels; when a DCLFC solution with 2-levels of decomposition is applied, the SA images selected correspond to the positions 0 and 2; finally, for 3-levels, the SA images selected correspond to the positions 0 and 4.

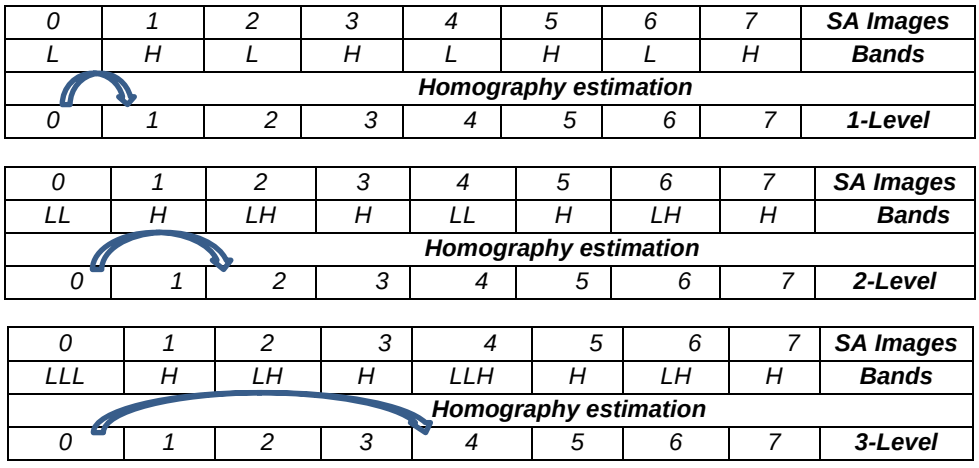


Figure 88: *SA images selection scheme for homography estimation.*

The RD results comparing the homography estimation using or SA images or low-frequency bands are displayed in Figure 89, notably for the light fields with the best and worse previous RD performances and for the simplest DCLFC solutions in which this impact may be studied, namely **H2**, **H1_V1** and **H3**.

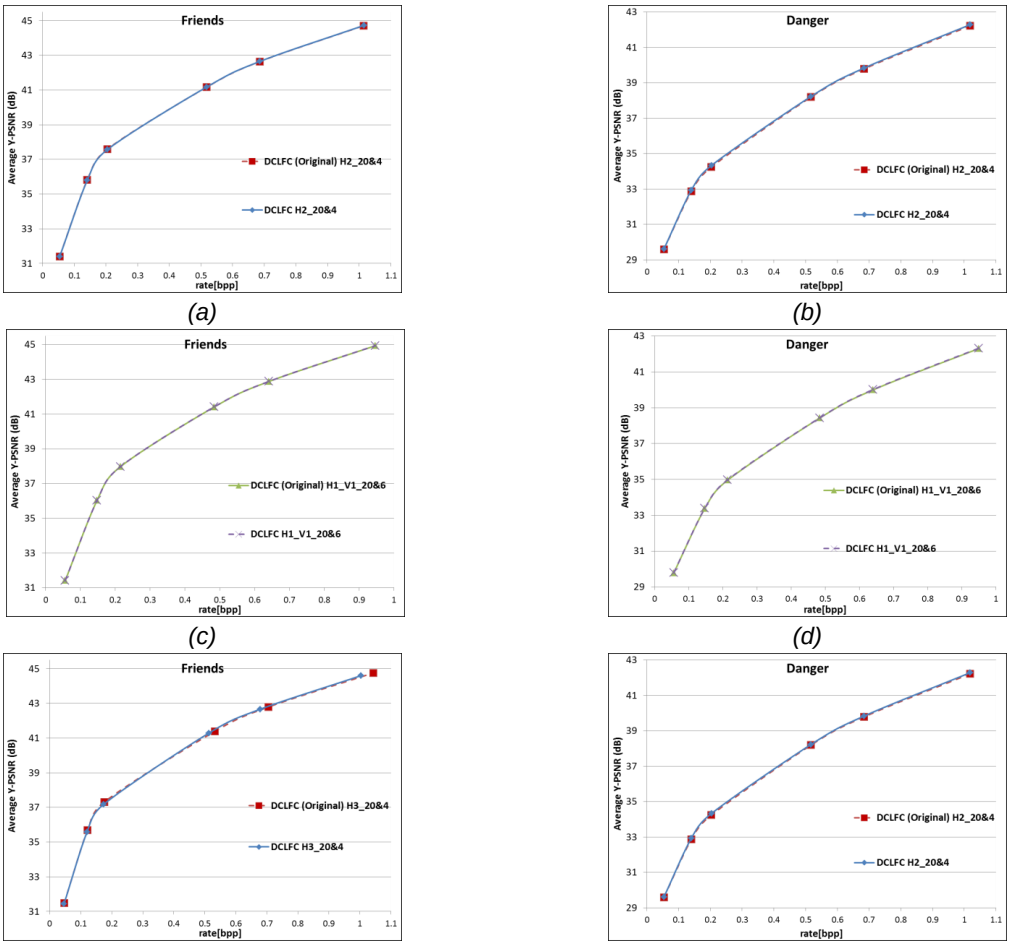


Figure 89: *RD results illustrating the impact of computing the homography parameters using original SA images instead of low-frequency bands for: a) Friends, DCLFC H2; b) Danger, DCLFC H2; c) Friends, DCLFC H1_V1; d) Danger, DCLFC H1_V1; e) Friends, DCLFC H3; f) Danger, DCLFC H3.*

The main conclusion is that computing the homography with the original SA images does not improve the previously proposed solution which computed the homography with the low-frequency bands. For the DCLFC H2 and DCLFC H1_V1 solutions, almost no RD performance difference is observed between the curves associated to each situation; for the DCLFC H3 solution, a small decrease in the RD performance is emerging. Observing the pattern used for the selection of the SA images to compute the homography parameters, it can be pointed out that for a number of decomposition levels higher than one, the disparity between bands does not correspond necessarily to the disparity between the original SA images, as the low-frequency bands are an averaged representation of the inter-transform input; this means that, for a decomposition with more than 3 levels, the low-frequency bands may represent views which do not exactly correspond to the SA images they are 'replacing'.