Graph based transforms for compression of new imaging modalities

Mira Rizkallah

▶ **To cite this version:**

Mira Rizkallah. Graph based transforms for compression of new imaging modalities. Image Processing [eess.IV]. Université de Rennes, 2019. English. NNT : 2019REN1S021 . tel-02285386

HAL Id: tel-02285386
https://theses.hal.science/tel-02285386

Submitted on 12 Sep 2019

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

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

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITE DE RENNES 1
COMUE **U**NIVERSITE **B**RETAGNE **L**OIRE

Ecole Doctorale N°601
Mathématique et Sciences et Technologies de l'Information et de la Communication
Spécialité : Signal, Image et Vision

Par

Mira RIZKALLAH

Graph-based transforms for compression of new imaging modalities

Thèse présentée et soutenue à RENNES , le 26 Avril 2019
Unité de recherche : SIROCCO, Inria Rennes Bretagne Atlantique

Rapporteurs avant soutenance :

Gene Cheung	Professor, EECS Department, York University, Canada
Markus Flierl	Professor, KTH University, Sweden

Composition du jury :

Président :	Luce Morin	Professor at INSA Rennes
Examinateurs :	Laura Toni	Assistant Professor at UCL, London, United Kingdom
Dir. de thèse :	Christine Guillemot	Research Director, Inria Rennes Bretagne Atlantique
Co-dir. de thèse :	Thomas Maugey	Researcher, Inria Rennes Bretagne Atlantique

Acknowledgement

I owe my deepest gratitude to all the people in Inria Rennes Bretagne Atlantique, who have made my three years of Phd a great experience.

Christine, know that your words of wisdom and advice have always been received with highest degrees of respect. Your support, confidence and commitment to my project were instrumental to its successful completion. I extend my thanks and gratitude to Thomas who has shown me that there is no limit to aspiration. It was a great honor and a privilege to work with you and I'll always be looking up to you when I think of dedication to science. I genuinely appreciate your honesty, persistence, and fruitful scientific and human discussions which have been invaluable to both my thesis and future professional career.

My sincere gratitude also goes to my supervisors during my stay in Lausanne Switzerland, Francesca de Simone and Pascal Frossard who have played a great role in this thesis and enormously contributed to its success. I am greatly thankful for your assistance, advice and support and I am quite happy that I had the opportunity to work with someone with amazing scientific and human qualities like yours.

I would also like to express my gratitude towards all the jury members (Luce Morin (jury president), Laura Toni, Markus Flierl and Gene Cheung) for accepting to be a part of this long process. In your own ways, each one of you have added to this work and helped in shaping me both as a student and a researcher.

I would particularly like to thank Dr. Charles Yaacoub, who is undeniably the first reason of what I am today. I will never find enough words to express my gratitude to such a great teacher. Thank you to all my teachers through the years for being such an exceptional inspiration for me.

For all future Phd students: "don't pick a thesis subject, pick a team, as your team is the biggest factor in your Phd success. A team where you don't feel like home won't give you opportunities to grow. " I couldn't have picked more wisely. My deepest heartfelt appreciation goes to all my colleagues in SIROCCO. You are my role models and you have inspired me in so many ways both at the personal and scientific levels and for that I'll be forever grateful.

Pierre, my loving boyfriend, I am so lucky to have you in my life. Without you, my last year of thesis would have been unbearable. Thank you for your presence and encouragement during my stressful moments. Thank you for your understanding and unconditional love.

A very special thought goes to my friends in France and Lebanon for being there for me. Also, I am extremely grateful for Pierre's family. I have learned a lot from your perseverance and I feel extremely lucky to know you. I have a special place in my heart for all of you, thank you for being my big family.

Last but not least, I want to dedicate this thesis to my father (my hero), my loving mother (my everything), my brother (my prince) and all my family who have always supported me. I am so lucky to have you. Your faith in me, your unconditional love and my eagerness to make you proud was the driving force behind my success.

This project has been, without any doubt, the largest test of my own commitment, spanning about

three years away from my family and loved ones. But I have not achieved this alone. Along the way, I have received so much support from so many people to count. I definitely wouldn't have done it if some of you weren't in my life. Thank you for all the gracious sacrifices you have made for me while I realized this goal. Though you may not see your names here, know that your various contributions have not gone unnoticed or unappreciated.

Table of Contents

Résumé en Francais

Contexte

Au cours des dernières années, il existe un intérêt particulier à donner une impression plus aigue de profondeur, et de géométrie au contenu visuel que nous cherchons à capturer et à diffuser, que ce soit pour des applications immersives ou pour la photographie. Celà a du sens car après tout, nous avons deux yeux avec lesquels nous pouvons percevoir le monde qui nous entoure. Nous pouvons bouger et donc changer de perspectives. Nous pouvons également nous concentrer sur un objet particulier de la scène. En tant que tels, la disparité, le changement de perspective et la mise au point sont des capacités capitales de notre système visuel. Il s'est avéré qu'avec une photographie conventionnelle, nous ne sommes pas capables d'imiter notre système visuel. Ce que nous capturons avec une seule caméra conventionnelle nous en dit assez peu sur la géométrie de la scène. En particulier, les cameras traditionnelles n'enregistrent pas la quantité de lumière circulant le long des rayons qui contribuent à l'image. Ils ne nous disent que la somme des rayons lumineux frappant chaque point de l'image. En quête des "informations géométriques manquantes", de nouvelles modalités d'imagerie ont récemment été proposées.

Un exemple en est le *champs de lumière*. Comme son nom l'indique, un *champ de lumière* [62] [76] [61] représente l'ensemble des rayons de lumière émis par la scène selon différentes orientations. C'est une description formelle des intensités des rayons qui se déplacent de et vers chaque point de l'espace. D'un point de vue plus général, il s'agit d'un terme plus large désignant la capture synchrone d'une scène sous différents points de vue. Dans de nombreux cas, il peut s'agir d'un ensemble de vues décrivant la même région d'intérêt. Ce contenu visuel a récemment été capturé par de nouvelles caméras plénoptiques [40] [75] et a apporté d'énormes contributions dans de nombreux domaines, notamment l'imagerie médicale, les systèmes de sécurité et autres [80].

Les caméras plenoptiques basées sur des matrices planaires de capteurs ont cependant une limitation importante: elles ne capturent que les rayons lumineux dans une partie limitée de l'espace directionnel; Elles ont un champ de vision limité. Celà a conduit au développement de caméras omnidirectionnelles parallèlement à l'étude des caméras à champ lumineux. Les *images omnidirectionnelles* ont commencé à susciter un intérêt vif dans les communautés de traitement d'images et de vision par ordinateur en raison de leur large champ de vision. Ceci est une propriété intéressante qui apporte beaucoup à de nombreuses applications telles que la navigation à 360°, les systèmes de surveillance et la modélisation 3D des environnements [39]. Ils permettent également à un très grand nombre d'utilisateurs de regarder vidéos 360 en ligne, sur leur smartphone, leur tablette ou leurs lunettes de réalité virtuelle.

Fournir l'information géometrique manquante, que ce soit dans un *champ de lumière* ou dans un contenu *omnidirectionnel*, s'est fait au détriment de la collecte de grands volumes de données à très haute dimension. Ainsi, le développement de ces nouvelles modalités d'image que nous souhaitons stocker ou délivrer a donné un nouvel élan à la recherche sur les schémas de compression [117] [18] [6] . Avec

la forte demande d'images de haute résolution et de haute qualité, le développement de schémas de compression à la fois efficaces et non complexes est crucial.

Motivation et objectifs

Dans un schéma de codage classique de tout type de contenu, nous visons à réduire le nombre de bits nécessaires pour représenter des données avec une qualité fixe. Une étape importante dans le développement d'un schéma de compression est l'étape de la transformée, où le signal initial à coder est projeté dans un autre domaine et les redondances sont réduites. La parcimonie du signal de sortie et la compaction de l'énergie sont des propriétés très importantes recherchées la plupart du temps. Afin de spécifier une transformée, il convient de délimiter les supports sous-jacents, les signaux à transformer et les bases de la transformation elles-mêmes. La transition entre la définition d'une transformée dans le domaine 1D vers le domaine des images 2D s'est naturellement déroulée, car la géométrie sous-jacente est toujours régulière et les supports peuvent être étendus intuitivement d'un espace régulier à un autre plus étendu.

Après tout, on peut facilement supposer que les pixels voisins qui correspondent à un même objet sont généralement très dépendants. Nous avons toujours eu tendance à définir des supports en regroupant des pixels corrélés avant la transformée. Les blocs de differentes tailles dans un schéma de codage 2D peut être considérée comme un support prenant en compte la géométrie d'une image 2D [44]. Plus récemment, des supports ayant des formes variées, tels que les super-pixels [1], ont été proposés, offrant un moyen plus souple qui s'adapte plus facilement au contenu de la scène.

Afin de réduire la redondance d'un signal se trouvant sur de tels supports dans une image classique, une méthode traditionnelle consiste à projeter le signal sur des fonctions de base assurant à la fois la décorrélation et la compaction de l'énergie du signal de sortie. Par exemple, le DCT classique en 2D [107] et la transformée en ondelettes [95] ont été largement appliqués sur des blocs réguliers. La DCT à adaptation de forme (SA-DCT) est également apparu comme une transformée prometteuse pour les super-pixels [93].

Aller au-delà d'une image 2D classique reposant sur des grilles uniformes, jusqu'à un volume 4D couplé à une géométrie complexe dans le cas des *Champs de Lumière* et à un domaine structurel non uniforme avec le contenu *omnidirectionnel* a soulevé de nombreuses questions. Comment généraliser les outils traditionnels de traitement et de compression des images 2D à de nouveaux domaines non nécessairement échantillonnés de manière uniforme sous forme de grille 2D? Plus précisément, avec les informations géométriques déjà acquises, comment pouvons-nous définir les supports de transformées de manière à exploiter à la fois le contenu et la structure sous-jacente? Une fois les supports définis avec soin, comment concevoir une transformation basée sur ces supports à la fois efficace et non complexe?

Que ce soit dans un *champs de lumière* ou une *image omnidirectionnelle*, les données capturées reposent sur des structures irrégulières. Dans le premier cas, l'irrégularité provient du fait que la géométrie de la scène est complexe. Dans le dernier cas, les pixels sont le résultat d'un échantillonnage de nature non uniforme et correspond à un échantillonnage équi-angulaire sur une sphère. Les outils de traitement de signal classiques se révèlent inappropriés pour des domaines aussi complexes et irréguliers, car ils sous-évaluent généralement la structure inhérente. D'où la nécessité d'une nouvelle représentation plus

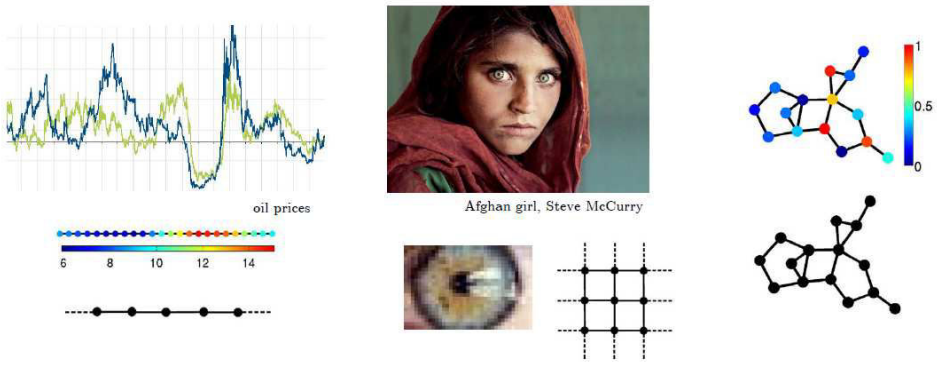


Figure 1: Le graphe capture la structure géométrique sous-jacente des données.

flexible. Dans cette thèse, nous nous appuyons sur la notion de graphes (Fig. 1). La communauté de traitement des signaux sur les graphes s'est récemment impliquée dans la conception de nouveaux outils et algorithmes capables de gérer les défis posés par la nature irrégulière des supports de graphes et de traiter efficacement les signaux définis sur les nœuds des graphes [92].

Si nous limitons notre attention au cas *champs de lumière*, nous pouvons imaginer un graphe *énorme* représentant les corrélations existantes en traçant des arêtes entre les différents pixels dans, et entre les différentes vues. Ce graphe peut être considéré comme un support pour définir une transformée à base de graphes. Néanmoins, sa grande dimensionnalité soulève des problèmes de complexité frappants. Dans cette thèse, nous visons à réduire cette complexité. Nous sommes donc intéressés à trouver des supports plus locaux en nous basant sur le concept de super-rayons inhérent à la géométrie et au contenu du *champs de lumière*. Le concept de super-rayons introduit dans [46] en tant qu'extension du concept de super-pixels au domaine 4D d'un *champs de lumière*. Un example de résultat de super-rayons est illustré dans la Fig. 2.

Après avoir effectué le partitionnement, les super-rayons constituent le support des transformées locales basées graphes. Maintenant que nous avons les supports dépendents de la géométrie et la couleur, cela nous amène au premier objectif que nous abordons dans cette thèse: Comment concevoir des transformées basées graphes, locales non complexes et efficaces tout en tenant compte de la géométrie?

Cependant, la localité des supports de super-rayons pose un autre problème: elle ne nous permet pas de capturer les dépendances spatiales à long terme du signal de couleur, contrairement aux schémas prédictifs efficaces utilisés dans les codeurs classiques (par exemple, HEVC). Plus précisément, dans le cas des transformées locales basées graphes, à l'intérieur de chaque super-rayon, une grand partie de l'énergie est concentrée dans un nombre limité de coefficients. Néanmoins, en les codant directement, les corrélations entre différents super-rayons ne sont pas exploitées. Ceci nous oriente vers la deuxième contribution pour les *champs de lumière*: la conception d'algorithmes d'échantillonnage et de prédiction qui permettront la meilleure compression des coefficients qui détiennent les énergies les plus élevées. Ceci en exploitant la corrélation spatiale au-delà des limites du support de transformée locale basée graphe .

De la même manière que le graphe *énorme* de *Champs de lumière*, un gros graphe peut également

Figure 2: Un example de super-rayons obtenus en utilisant l'agorithme proposé dans [46]. Pour une bonne visualisation, le resultat est montré pour seulement une vue du champs de lumiere *StillLife*. A gauche, la vue originale en couleur. A droite, les super-rayons obtenus.

être conçu pour représenter la structure sphérique sous-jacente du contenu de *omni-directionnel*. Cela tient principalement au fait que les schémas de compression existants pour le contenu omnidirectionnel ne s'adaptent pas à la géométrie sphérique sous-jacente et sont donc sous-optimaux. Confrontés aux mêmes problèmes de complexité, nous abordons les objectifs suivants: comment définir des supports locaux pour limiter la complexité des transformées basées graphes? Comment diviser efficacement le graphe en supports plus restreints, tout en tenant compte de la géométrie sphérique? Comment optimiser le lissage des signaux sur les sous-graphes tout en conservant un léger sur-coût pour coder la description de la partition?

Dans les représentations à base de graphes évoquées précédemment, le graphe a été utilisé pour caractériser soit les pixels corrélés, soit ceux dépourvus de corrélation, sans aucune notion du niveau de corrélation. Si nous souhaitons donner des poids aux connections et par ça modéliser les similitudes des signaux, nous pouvons définir un modèle du signal. Parfois, un bon modèle peut améliorer l'efficacité du codage. Mais, dans quelle mesure pouvons-nous utiliser ce modèle, au lieu de nous fier uniquement à une topologie? Cela a motivé notre étude théorique finale où nous cherchons à trouver une limite théorique sur l'incertitude qu'une transformée basée sur un modèle peut gérer.

Résumé des Contributions

Cette thèse est structurée en deux parties principales.

Dans la partie I, le chapitre 1 fournit une compréhension globale des transformées basées graphes. Plus précisément, nous introduisons la définition formelle d'un graphe et d'un signal sur un graphe avec la taxonomie et les notations pertinentes utilisées dans la suite de ce travail de thèse. Nous nous concentrons sur l'application des transformées basées graphe dans le domaine de la compression d'image. Ensuite, au chapitre 2, nous donnons un aperçu des deux modalités d'image abordées dans cet ouvrage:

les *champs de lumière* et les *images omni-directionnelles*. Plus précisément, nous présentons la définition de chacune d'elles, leurs différentes représentations et les nouvelles possibilités qu'elles offrent dans les differents domaines.

Dans la partie II, nous présentons nos différentes contributions présentées ci-dessous. En résumé, cette partie est orientée selon deux axes principaux correspondant aux deux modalités d'image deja mentionnées. Plus en détail, cette partie est organisée comme suit:

- **Chapitre 3:** Ce chapitre explore deux manières de résoudre les deux problèmes principaux de la conception de la transformées basées graphe pour la compression des champ de lumière: (a) Trouver le graphe optimal sur lequel le signal à transformer est régulier et (b) Trouver le meilleur compromis entre complexité et précision de la représentation. Nous présentons une première solution qui consiste à utiliser des supports de graphes qui ne tiennent pas compte de la géométrie. Ils seront fixés pour toutes les images de sous-ouverture du champ de lumière, associés à un mécanisme de prédiction puissant basé sur les réseaux de neurones convolutionnels (CNN). Une autre solution est également proposée en considérant les super-rayons quasi-idéaux qui sont plutot conscients de la géométrie. Ceux-ci sont construits avec prudence et associés à des transformées basées graphes séparables optimisées, afin de préserver les corrélations angulaires. Les résultats expérimentaux montrent l'intérêt des approches en termes de compaction de l'énergie. Des schémas de codage sont également décrits pour évaluer les performances débit-distorsion des transformées proposées et sont comparés aux codeurs classiques notamment HEVC et JPEG Pleno VM 1.1.

- **Chapitre 4:** Dans le chapitre précédent, l'efficacite des transformées locales basées graphes était montrée en terme de compaction d'énergie. Néanmoins, la localité des supports nous permet pas d'exploiter pleinement les dépendances à long terme du signal. Dans ce chapitre, nous décrivons une solution de prédiction basée sur des graphes . Celle-ci permet de tirer parti des mécanismes de prédiction intra ainsi que des bonnes propriétés de compaction d'énergie des transformees proposées dans le chapitre précedent. Nous nous appuyons sur des transformées spatio-angulaires non séparables et séparables et nous déduisons des coefficients spatio-angulaires à basse fréquence à partir d'une seule image de référence comprimée et des coefficients de hautes fréquences. L'image de référence est composée d'échantillons choisis avec soin dans le champ de lumière. Les approches se révèlent très efficaces dans un contexte de compression de haute qualité des champs de lumière quasi sans perte.

- **Chapitre 5:** Dans ce chapitre, nous nous intéressons au codage du contenu omnidirectionnel. Récemment, afin d'être comprimés à l'aide de codeurs existants, ces signaux sont projetés sur le domaine planaire 2D. Une représentation plane couramment utilisée est la représentation équirectangulaire, qui correspond à un motif d'échantillonnage non uniforme sur la surface sphérique. Cette particularité n'est pas explorée dans les schémas de compression d'images classiques, qui traitent le signal d'entrée comme une image en perspective sur une grille regulière. Dans ce travail, nous construisons un codeur basé sur les graphes, adapté à la surface sphérique. Nous construisons un graphe directement sur la sphère. Ensuite, pour obtenir des transformées basées graphes

non complexes, nous proposons un algorithme de partitionnement de graphes optimisé en terme de debit-distorsion. Ceci nous permet d'obtenir un compromis efficace entre la distorsion des signaux reconstitués, la régularité du signal sur chaque sous-graphe et le coût de codage de la partition. Les résultats expérimentaux démontrent que notre méthode dépasse le codage traditionnel JPEG des images équi-rectangulaires.

- **Chapitre 6:** Les chapitres précédents sont principalement liés à des applications pratiques et ont été construits sur l'hypothèse qu'un pixel est corrélé ou non avec un autre. Afin de donner plus de degrés de liberté aux supports de graphe que nous voudrions utiliser, nous pouvons supposer un modèle de signal représenté par des poids. Dans ce contexte, dans ce chapitre, nous abordons un problème plus théorique de compression concernant les transformées basées graphes: l'incertitude d'un modèle et son impact sur les transformées. Plus précisément, nous étudions dans quel cas le fait de s'appuyer sur un modèle incertain pour définir une transformée (avec plus de degrés de liberté pour les poids) apporte une amélioration aux transformées qui se basent uniquement sur la topologie (où seuls les poids binaires sont utilisés). Nous développons les différentes équations qui mènent à une discussion intéressante sur l'effet de l'incertitude des poids sur l'efficacité de la compression. Nous validons ensuite notre étude théorique sur des données synthétiques.

Enfin, nous concluons ce manuscrit en discutant des contributions et des perspectives futures de ce travail. La liste des publications de l'auteur de cette thèse est disponible dans la section publications de l'auteur. En annexes, figurent les differentes contributions de l'auteur qui ne sont pas incluses dans ce manuscrit.

Les travaux présentés dans ce manuscrit ont été soutenus par le Ministère de l'Enseignement supérieur et de la Recherche. Ils ont également été financés en partie par le programme de recherche et d'innovation H2020 de l'UE dans le cadre de la convention de subvention n°694122 (*ERC Advanced Grant CLIM*).

Introduction

Context

In the recent years, there exists a special arousal to grant a greater sense of geometry to the visual content we seek to capture and deliver, may it be for immersive applications or photography. It makes sense because after all, we have two eyes with which we are able to perceive the world around us. We can move and therefore shift perspectives. We can also focus on a particular object in the scene. As such, disparity, motion and focus are chief geometrical cues our visual system is capable of doing. It turned out that with a conventional photograph, we are not capable of mimicking our visual system. What we capture with a single camera tells us rather little about the geometry of the scene. In particular, they do not record the amount of light traveling along individual rays that contribute to the image. They tell us only the sum total of light rays striking each point in the image. Going after the "missing geometrical information", new imaging modalities have been recently proposed.

An example of such is the so called *Light Field*. In essence, as its name suggests, a Light Field [62] [76] [61] is the embodiment of the concept of representing light as a vector field. It is a formal description of the intensities of rays, flowing from and into every point in space. From a higher perspective, it is a broad term referring to the synchronous capture of a scene from different viewpoints. In many cases, it can be portrayed as a collection of views describing the same region of interest. Thus, the captured data for a static light field contains redundant information in both the spatial and angular dimensions. Moreover, this visual content has been recently captured by novel plenoptic cameras [40] [75] and has offered enormous contributions in a lot of fields including medical imaging, security systems and others [80].

Light field cameras based on planar arrays of sensors have, however, a severe limitation: they capture only light rays in a limited portion of the directional space, i.e., they have a limited field of view. This led to the development of omni-directional cameras running in parallel to the study of light field cameras. *Omni-directional images* began to spark a tremendous interest in the image processing and computer vision communities due to their large field of-view. A large field of view is an interesting property that brings a lot to many applications such as 360 navigation , surveillance systems, and 3D modeling of environments [39]. It also enabled users to watch 360 videos online, on their smartphone, tablet or Virtual Reality glasses [1].

Providing the missing information, whether in a captured light field or an omnidirectional content, has come at the expense of the collection of large volumes of high dimensional data. Thus, the outgrowth of those novel imaging modalities that we wish to store or deliver has brought a new impulse to research in compression techniques [117] [18] [6] . With the high demand of high resolution images and high

[1] "Google cardboard," https://vr.google.com/cardboard/
"Google daydream," https://vr.google.com/daydream/
"Google odyssey," https://gopro.com/odyssey
"Facebook surround 360," https://facebook360.fb.com/ facebook-surround-360/

qualities, the development of compression schemes that are at the same time efficient and not complex, is crucial.

Motivation and Goals

In a classical coding scheme of any kind of content, we aim at reducing the number of bits needed to represent data for a fixed quality. One important step in the development of the compression scheme is the transform stage, where the initial signal to be coded is transformed and redundancies are reduced. The sparsity of the output signal is a very important property that is seeked most of the times. In order to specify a transform, one should delineate the underlying supports, the signals to be transformed and the transform basis functions themselves. The transition between the definition of a transform in the 1D domain to 2D domain of images was smooth since the underlying geometry is always regular and the supports can be intuitively extended to the 2D regular space.

After all, we can easily assume that neighboring pixels that correspond to a same object are usually highly dependent. We have always tended to define supports by grouping correlated pixels together prior to the transform. The size varying blocks in a 2D coding scheme can be though as supports taking into account the geometry of a 2D image [44]. More recently, the shape-varying supports such as super-pixels [1] have been proposed providing a more flexible way that more easily adapt to the content of the scene.

In order to reduce the redundancy of a signal lying on such supports in an 2D classical image, a traditional way of doing is to project the signal onto some basis functions that provide at the same time the decorrelation and the energy compaction of the output signal. For example, the classical 2D DCT [107] and wavelet transform [95] have been widely applied on regular blocks. Shape adaptive DCT SA-DCT has also appeared as a promising transform for super-pixels [93].

Moving beyond a traditional 2D image lying on uniform grids, to a 4D volume coupled with complex geometry in *Light Fields* and a non uniform structural domain with the *omnidirectional* content has led to numerous questions. How can we generalize the traditional 2D image processing and compression tools to new domains that are not necessarily uniformly sampled as a 2D grid? More precisely, with the geometrical information already acquired, how can we define the transform supports in a way that we exploit both the content and the underlying structure? Once the supports are carefully defined, how to design a transform based on those supports that is at the same time efficient and non complex?

Whether it was in a *Light Field* or an *Omni-directional image*, the captured data lies on irregular structures. In the former, the irregularity comes from the fact that the scene geometry is complex. While in the latter, the sampling is already non-uniform and corresponds to an equi-angular sampling on a sphere. Classical signal processing tools are revealed to be inappropriate for such complex and irregular domains since it usually undervalue the inherent structure. From here arises the need for a new representation that is more flexible. In this thesis, we rely on the notion of graphs. (See Fig. 3) The Graph Signal Processing community has been lately devoted to design new tools and algorithms that can proficiently handle the challenges arising from the irregular nature of graph supports and efficiently process the signals living on the vertices of graphs [92].

If we restrict our attention to the *Light Fields* case, we can imagine a *huge* graph to represent the ex-

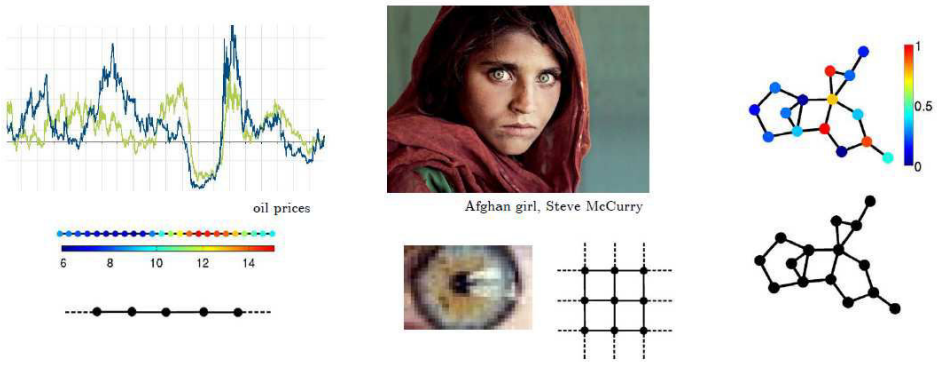


Figure 3: The graph describes the underlying geometric data structure.

isting correlations by drawing edges between different pixels inside and between the different views. Such graph can be considered as a support to define a graph transform. Nevertheless, striking complexity issues arise from its high dimensionality. In this thesis, we aim at reducing this complexity. We are thus interested in finding more local supports relying on the concept of super-rays that are inherent from the geometry and the content of the *Light Field*. The concept of super-ray introduced in [46] as an extension to 4D *Light Field* of the concept of super-pixels.The result of the partitioning in one light field view is shown in Fig. 4.

After performing the partitioning, super-rays constitute the supports of local graph transforms. Now that we have the supports denoting the geometry and the color information, this leads us to the first problem we address in this thesis: How to design efficient non complex local geometry-aware graph transforms for *Light Fields*?

However, another problem arises with the locality of the super-ray supports: it does not allow us to capture long term spatial dependencies of the color signal, unlike efficient predictive schemes used in state of the art coders (e.g. HEVC). More precisely, in the case of the local graph transforms, inside each super-ray, most of the energy is concentrated in a small number of coefficients. Nevertheless, the correlations between different super-rays are not exploited. This motivates the second contribution for *Light Fields*: the design of a prediction and sampling scheme that would allow the best compression of the coefficients with the highest energy, by exploiting spatial correlation beyond the limits of the local graph transform support.

Similarly to the *huge* graph for *Light fields*, a big graph can also be used to represent the underlying spherical structure of the *omni-directional* content. The motivation behind this comes from the fact that the existing compression schemes for omnidirectional content do not adapt to its underlying spherical geometry and are therefore sub-optimal. Faced with the same complexity issues, we tackle the problem of: how to define local supports and graph transforms ? How to efficiently partition the graph into smaller supports, taking into account the geometrical information? How to optimize the smoothness of the signals on the sub-graphs while keeping a small overhead to code the description of the partition?

In the previously evoked graph based representations, the graph has been used to characterize either pixels that are correlated or those with no correlation at all without any notion of the level of correlation.

Figure 4: An example of super-rays obtained using the algorithm in [46]. For visualization purpose, the result is shown for only one view of the Light Field *StillLife*. On the left, the original view in the RGB space. On the right, the resulting super-rays.

If we wish to give more degree of freedom to the weights and by that model the vertices similarities, then we can define a *signal model*. Sometimes a good model can improve the coding efficiency. However, to what extend are we able to use this model, instead of only relying on a topology? This has motivated our final theoretical study where we seek to find a theoretical limit on the uncertainty that a transform based on a model can handle.

Thesis Roadmap

This dissertation is structured in two main parts.

In Part I, Chapter 1 provides a global understanding of the transforms defined for signals on graphs. Specifically, we introduce the formal definition of a graph and a graph signal along with the relevant taxonomy and notations used in the rest of the thesis work. We focus on the application of the graph transforms in image compression. Then, in Chapter 2, we give a broad background on the two imaging modalities tackled in this work: The *Light Fields* and *Omni-directional images*. Specifically, we present the formal definition of each, the different representations they have and what new possibilities they provide.

In Part II, we present our different contributions presented above. In a nutshell, this part is oriented along two main axes corresponding to the two imaging modalities. More in detail, this part is organized as follows:

- **Chapter 3 :** This chapter explores two ways of solving the two main issues concerning the graph transform conception for light field compression: (a) The careful graph support design in a way that the signal to be transformed is smooth on the graph and (b) Finding the best trade-off between complexity and representation accuracy. We present a first solution that consists of using geometry-blind graph supports which are fixed for all light field sub-aperture images cou-

pled with a powerful prediction mechanism based on Convolutional Neural Networks (CNN). Another solution is also proposed considering quasi-ideal geometry-aware super-rays which are cautiously built coupled with an optimized separable graph transform to preserve angular correlations. Experimental results show the benefit of the approachs in terms of energy compaction. Coding schemes are also described to assess the rate-distortion performances of the proposed transforms and are compared to state of the art encoders namely HEVC and JPEG Pleno VM 1.1

- **Chapter 4 :** In the previous chapter, the local graph-based transforms have been shown to be powerful tools in terms of energy compaction. Nevertheless, the locality of the supports may not allow us to fully exploit long term dependencies in the signal. In this Chapter, we describe a graph-based prediction solution that allows taking advantage of intra prediction mechanisms as well as of the good energy compaction properties of the graph transforms proposed in chapter 3. We rely on both non-separable and separable spatio-angular transforms and derives low frequency spatio-angular coefficients from one single compressed reference image and from the high angular frequency coefficients. The reference image is made of samples that are carefully chosen from the light field. The approaches is shown to be very efficient in a context of high quality quasi-lossless compression of light fields.

- **Chapter 5 :** In this chapter, we are interested in coding the omnidirectional content. Recently, in order to be compressed using existing encoders, these signals are mapped to planar domain. A commonly used planar representation is the equi-rectangular one, which corresponds to a non uniform sampling pattern on the spherical surface. This particularity is not explored in traditional image compression schemes, which treat the input signal as a classical perspective image. In this work, we build a graph-based coder adapted to the spherical surface. We build a graph directly on the sphere. Then, to have computationally feasible graph transforms, we propose a rate distortion optimized graph partitioning algorithm to achieve an effective trade-off between the distortion of the reconstructed signals, the smoothness of the signal on each subgraph, and the cost of coding the graph partitioning description. Experimental results demonstrate that our method outperforms JPEG coding of planar equi-rectangular images.

- **Chapter 6 :** The previous chapters are mostly related to practical applications, and were built on the assumptions that a pixel is either correlated or not with another. In order to give more degrees of freedom to the graph supports we would want to use, we can assume a signal model represented by edge weights. In this chapter, we tackle a more theoretical problem in transform based compression on graphs: the uncertainty of a graph model and its impact on the transforms. More precisely, we study when does relying on an uncertain model to define a transform (with more degree of freedom for weights) brings improvement to only topology based transforms (where only binary weights are used)? We develop the different equations that leads us to an interesting discussion about the effect of uncertainty of the edge weights on the compression efficiency. We then validate our theoretical study on synthetic data.

Finally, we conclude this manuscript by discussing contributions and future perspectives of this work. The list of publications produced by the author of this thesis can be found in the the author publications section. The authors contributions that are not discussed in this manuscript are joined in Appendix A and B.

The work presented in this manuscript has been supported by the French Ministry of Higher Education and Research. It has also been supported in part by the EU H2020 Research and Innovation Programme under grant agreement No 694122 (ERC advanced grant CLIM).

PART I

State of the Art

Chapter 1

Transforms for signals on graphs and applications in image compression

Traditionally, Digital Signal Processing (DSP) deals typically with signals residing in continuous domains, which may be then sampled to get a particular digital representation to be processed afterwards. Usually, those signals are acquired and sampled uniformly representing some evolution in time of a variable or some luminance distribution on a regular lattice (two dimensional images). Thus, it is expected that the main part of the signal processing research targets uniform grids. Yet, in applications such as social, energy, transportation, sensor and neural networks, high dimensional data resides on the vertices of weighted graphs. Also, weighted graphs are commonly used to represent similarities in statistical learning problems addressed in computer vision. Even images and three dimensional scenes can be considered as graphs where pixels or regions are nodes connected based on their similarity, dependencies and distance in the 3D scene. Classical signal processing tools are revealed to be inappropriate for such irregular structures. Therefore, a lot of research effort has been lately devoted to design new tools and algorithms that can proficiently handle the challenges arising from the irregular nature of graph support and efficiently process the signals living on the vertices of graphs.

In this chapter, we review the main principles of Graph Signal Processing. The graphs and the signals on irregular domains are first recalled. We then present the graph signal spectral representations and the generalized operators for signals on graphs. With the aid of the aforementioned notions, we review the main strategies used to define the graphs and design the transforms for graph signals.

1.1 Graphs and signals defined on graphs

In this first section, we briefly recall the basic definitions for graphs and signals residing on graph nodes (or vertices). We commonly consider a weighted undirected graph (see Figure 1.1) $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{W}, \mathbf{V}\}$ where $\mathcal{V}$ and $\mathcal{E}$ represent the set of vertices(or nodes) and edges. The Adjacency (or connectivity) matrix $\mathbf{A}$ of $\mathcal{G}$ is a $n \times n$ symmetric matrix such as $\mathbf{A}(i,j) = 1$ if there is an edge connecting the vertices i and j, and $\mathbf{A}(i,j) = 0$ otherwise, for $i, j = 1...n$. $\mathbf{W}$ is a weight matrix, positive and symmetric in our case of undirected graphs typically denoting similarities between connected nodes. In such matrix, if the nodes m and n are connected by an edge, $\mathbf{W}(m,n)$ equal to $\mathbf{W}(n,m)$, represents the weight of this edge. Otherwise $\mathbf{W}(m,n)$ is equal to zero. The self-loops matrix $\mathbf{V}$ of $\mathcal{G}$ is an $n \times n$ diagonal matrix with entries $\mathbf{V}(i,i)$ for $i = 1...n$, that are non-zero only if there is a self-loop connecting the node i to itself, and $\mathbf{V}(i,j) = 0$ for $i \neq j$. The degree of a vertex n is the sum of weights of all the incident edges that can be computed by summing the elements of the n^{th} row of the weight matrix $\mathbf{W}$ or the adjacency $\mathbf{A}$

if the graph is unweighted. The degree matrix $\mathbf{D}$ is a diagonal matrix with the i^{th} diagonal value is the degree of a vertex i.

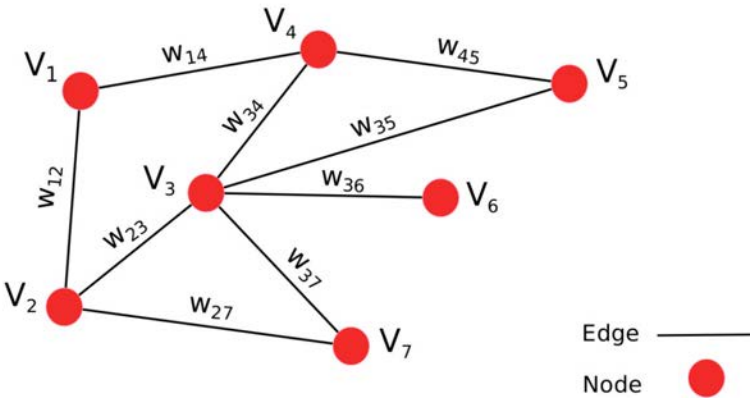


Figure 1.1: An example of a arbitrary graph consisting of 7 nodes $V_1, V_2, \ldots V_7$ that are connected by undirected weighted edges

Assuming that the graph $\mathcal{G}$ consists of N nodes and is connected, i.e. every node can reach another node in this graph following some paths, we denote by the j-hop neighborhood of a vertex n expressed in Eq.(1.1), the set of vertices which are at most j hops away from node n. The distance d_h here represents the minimum distance in hops between two nodes. An example of neighborhood of a vertex is depicted in Figure 1.2.

$$N_j(n) = \{v \in \mathcal{V}, d_h(v, n) \leq j\}, \tag{1.1}$$

If we do not take the weights into account, and we define the degree of a vertex as the number of connections it has, we can define the unweighted combinatorial graph laplacian as:

$$\mathbf{L}_u = \mathbf{D} - \mathbf{A}, \tag{1.2}$$

More generally, the generalized graph Laplacian embodies the intrinsic structure of the graph and is primarily useful for its spectral interpretations. It is defined as:

$$\mathbf{L}_g = \mathbf{D} - \mathbf{W} + \mathbf{V}, \tag{1.3}$$

Where $\mathbf{D}$ is a diagonal degree matrix with the n^{th} diagonal entry d_n is equal to the degree of the vertex n, i.e. the sum of the weights of all incoming edges at the node n. If we do not take the self loops into account (assume $\mathbf{V} = 0$), the resulting laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the so-called combinatorial laplacian.

For the sake of simplicity, we restrict our attention in the following to the latter combinatorial graph laplacian $\mathbf{L}$. It is a real symmetric positive semi-definite matrix that has a complete set of real orthonormal eigenvectors $\mathbf{U} = \{\mathbf{u}_0, \mathbf{u}_1, \ldots \mathbf{u}_{N-1}\}$ corresponding to non-negative eigenvalues $\sigma(\mathbf{L}) = \{\lambda_0, \lambda_1, \ldots \lambda_{N-1}\}$ satisfying $\mathbf{LU} = \lambda\mathbf{U}$. Note that there is not necessarily a unique set of graph laplacian eigenvectors as it can be seen later on in this chapter. Also, more generally, zero appears as an eigenvalue

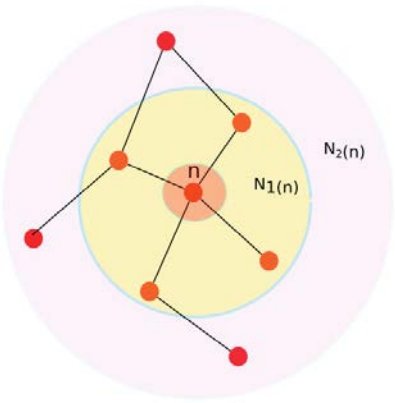


Figure 1.2: An example of node neighborhood: $N_1(n)$ and $N_2(n)$ represent the 1-hop and 2-hop neighborhood of the vertex respectively.

with multiplicity equal to the number of connected components of the graph and the largest eigenvalue depends on the largest degree of the graph. Hence, since we restrict our attention to connected graphs, the sorted spectrum of eigenvalues is denoted by (1.4) :

$$0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \cdots \leq \lambda_{max}, \tag{1.4}$$

The first eigenvalue (i.e. λ_0) is always zero and the corresponding eigenvector $\mathbf{u}_0$ is constant, which is a useful property in extending intuitions about the DC components of the signals related to classical signal processing theory.

Another popular option for analyzing the connected graph structure is to normalize each weight by $\frac{1}{\sqrt{d_i d_j}}$, leading to the normalized graph laplacian of the form (1.5):

$$\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}, \tag{1.5}$$

The first eigenvalue of the normalized laplacian is also equal to zero however in that case its associated eigenvector is not constant. A nice property of this normalized version is that its eigenvalues are always enclosed in the interval $[0, 2]$ which makes evaluation and comparison of graph signals' spectral representations simpler, particularly if we deal with graphs having a large difference in the total number of connected nodes.

A third popular matrix often used in dimensionality reduction techniques is the random walk matrix computed as $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$. Each entry in this matrix is a probability P_{ij} of going from a vertex i in the graph to another vertex j in one step of a Markov Random walk on the graph. It should be noted here such matrix is used when the nodes by themselves symbolize possible signal states or values.

A graph signal $\mathbf{f} : \mathcal{V} \to \mathbb{R}$ is a real-valued function, where each vertex n is assigned a real value $\mathbf{f}(n)$. Thus, such function may be represented as a vector $\mathbf{f} \in \mathbb{R}^n$ with the n^{th} element corresponding to the value of the signal at the vertex n in $\mathcal{V}$. An example of a signal defined on a graph is depicted in Figure 1.3.

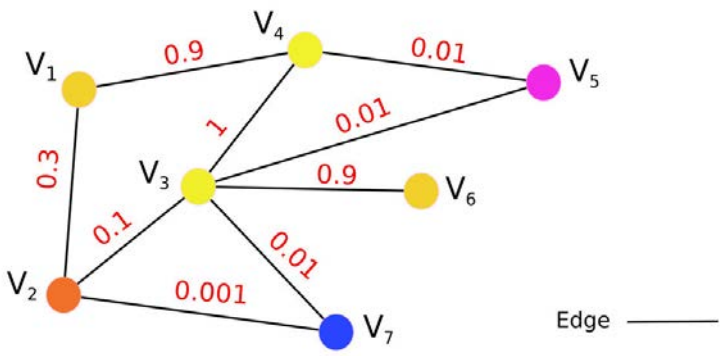


Figure 1.3: An example of a signal defined on a graph. The color of the nodes denotes the different values of the signal on the graph vertices. The weights are written in red on each edge.

The signal in Figure 1.3 is a special case of a smooth signal where there is no brutal signal variation between any two strongly connected nodes in the graph. The smoothness of a signal is thus potently dependent on the structure of the graph on which it is defined. More precisely, a signal is considered to be smooth with respect to a graph $\mathcal{G}$ when it exhibits small variations between strongly connected vertices. Typically, the global smoothness of a graph signal $\mathbf{f}$ is expressed though the discrete p-Dirichelet norm of $\mathbf{f}$ [92] (Eq. 1.6):

$$S_p(\mathbf{f}) = \frac{1}{p}\sum_{v\in\mathcal{V}} \|\nabla_v \mathbf{f}\|_2^p = \frac{1}{p}\sum_{v\in\mathcal{V}} \Big[\sum_{u\in\mathcal{N}_v} \mathbf{W}(v,u)[\mathbf{f}(v)-\mathbf{f}(u)]^2\Big]^{\frac{p}{2}}, \tag{1.6}$$

Where $\mathcal{N}_v$ denotes the one-hop neighborhood of the vertex v, and $\mathbf{W}(v,u)$ is the edge weight between the nodes v and u. When $p = 1$, Eq. (1.6) defines the total variation of the signal $\mathbf{f}$ on the graph. A widely-used and well-known Laplacian based form of smoothness is derived by fixing $p = 2$ as in Eq. (1.7).

$$S(\mathbf{f}) = \sum_{u,v\in\mathcal{E}} \mathbf{W}(v,u)[\mathbf{f}(v)-\mathbf{f}(u)]^2 = \mathbf{f}^T\mathbf{L}\mathbf{f}, \tag{1.7}$$

Eq. (1.7) implies that a signal is smooth , i.e. $S(\mathbf{f})$ is small only if the signal has similar values on neighboring vertices connected by an edge with high weights. This notion of smoothness has been widely used in semi-supervised learning literature, where the goal was to recover missing signal values under smoothness priors exploiting the assumption that the signal's values vary slowly between nodes connected with strong edges.[7] The smoothness is one of many signal processing tasks that one can generalize to graph signals.

To our knowledge, with the emergence of the graph representations and the graph signal processing research field, different frameworks have been developed and adopted to efficiently process signals residing on graph nodes taking into account the underlying graph support. David Shuman et al [92] were among the first in this academic movement, leading the way as they published an influential paper

stipulating the graph signal processing (GSP) framework which is based on spectral graph theory [16] and relies on an analogy built between the traditional Fourier transform and the spectral decomposition of the sets of eigenvalues/eigenvectors of the graph laplacian matrix. Aliaksei Sandryhaila and Jose M.F. Moura [87][86] developed a different framework relying on the algebraic signal processing theory [81] where the adjacency matrix was used to define a graph shift operator. The former framework has been principally defined for weighted undirected graphs with real non-negative edge weights, whereas the key advantage of the latter is that it can be generalized to account for directed graphs with negative and complex weights. After adopting a particular framework, researchers have been interested in generalizing the signal transforms from the classical euclidean domain to the graph settings.

In the traditional digital signal processing, the aptitude of defining and applying global or localized transforms on signals such as Fourier transform, wavelets, curvelets and windowed Fourier transforms to sparsely represent high dimensional data lying on regular spaces has led to significant improvement in the aforementioned compression tasks. Interestingly, a signal residing on graph nodes can be viewed as a vector in $\mathbb{R}^n$. However, a major obstacle to the applications of classical transforms on graph signals is that treating the graph signals in the same way as handling a discrete-time signal completely ignores significant dependencies and interactions arising from the underlying irregular structure and the connectivity in the graph domain. Relying on the spectral graph theory, some of the research effort [92] [16] has been recently dedicated to find analogies between the traditional signal processing and the graph signal processing.

1.2 Spectral representation and generalized operators for signals on graphs

1.2.1 Graph spectral representation

In classical Fourier analysis, the eigenvalues of the 1-D Laplace operator were revealed to carry an exact notion of frequency: eigenvectors associated to low eigenvalues are slowly oscillating complex exponential exhibiting low variations (i.e. low frequency) whereas for larger eigenvalues, the associated Eigenfunctions oscillate more rapidly (i.e. high frequency). Analogously, in the graph setting, the eigenvectors associated to small eigenvalues of the Laplacian matrix are signals that vary slowly across the graph edges as opposed to those associated to large eigenvalues where they take values changing more rapidly. In other words, in the former case, the values of the low frequency eigenvectors are expected to be similar on vertices connected by an edge with a high weight. In the latter, they are more likely to have dissimilar values at those locations. The set of eigenvectors of the graph Laplacian matrix are thus considered as a Fourier basis for signals defined on the graph vertices.
Formally, for a function $\mathbf{f}$ defined on the vertices of a graph $\mathcal{G}$, the graph Fourier transform $\hat{\mathbf{f}}(\lambda_l)$ at frequency λ_l is hence defined as the inner product with the associated eigenvector $\mathbf{u}_l$ (Eq. (1.8)):

$$\hat{\mathbf{f}}(\lambda_l) =< \mathbf{f}, \mathbf{u}_l >= \sum_{n=1}^{N} \mathbf{f}(n)\mathbf{u}_l^*(n), \tag{1.8}$$

where the inner product is linear with respect to the first argument and conjugate-linear with respect to the second argument of the previous equation, and $\mathbf{u}_l^*(n)$ is the conjugate value of the eigenvector $\mathbf{u}_l$ at the node n. The inverse Graph Fourier Transform $\mathbf{f}(n)$ at node n is given by Eq. (1.9):

$$\mathbf{f}(n) = \sum_{l=0}^{N-1} \hat{\mathbf{f}}(n)\mathbf{u}_l(n) \tag{1.9}$$

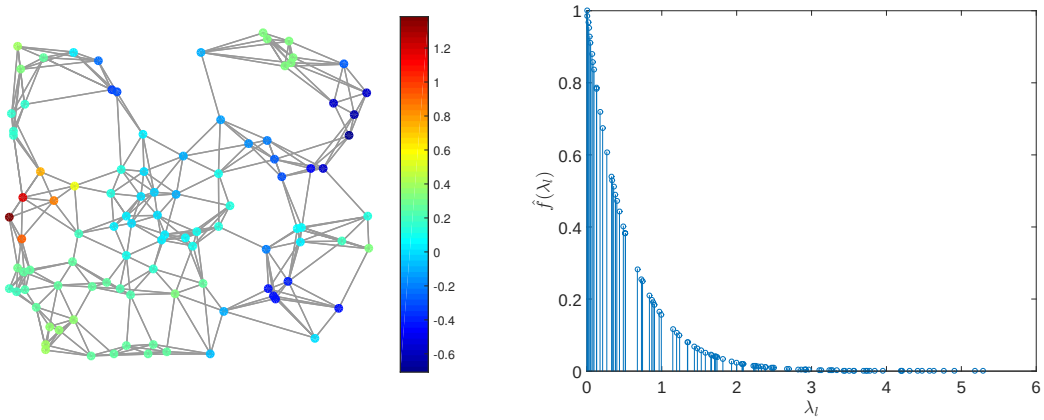


Figure 1.4: Equivalent representations of a graph signal in the vertex and graph spectral domains. On the left, a signal f that resides on the vertices of a sensor network with Gaussian edge weights. The signal values are represented by the colors of the vertices. On the right, the same signal in the graph spectral domain. In this case, the signal is a heat kernel, which is actually defined directly in the graph spectral domain as $\hat{\mathbf{f}}(\lambda_l) = e^{\frac{-10}{\lambda_{max}}\lambda_l}$. The plotted signal in is then determined by taking an inverse graph Fourier transform in Eq [1.9] of $\hat{\mathbf{f}}$.

An example of the two equivalent representations of a graph signal is depicted in figure (1.4) where a sensor network consisting of 100 vertices is drawn with random edge weights. A heat kernel $\hat{\mathbf{f}}$ has been designed in the graph spectral domain as shown on the right of Figure (1.4). The plotted signal in Figure (1.4) is then determined by taking an inverse graph Fourier transform in Equation [1.9] of $\hat{\mathbf{f}}$. The graph Fourier basis functions can be chosen as the eigenvectors of either the combinatorial or the normalized graph Laplacian. In both cases, the spectrums hold a frequency-like analysis. Moreover, as for the classical Fourier analysis, the spectral representation proficiently provides central information about the graph signal. More precisely, the smoothness of the signal as measured in equation 1.7 can also be written as:

$$S(\mathbf{f}) = \mathbf{f}^T\mathbf{L}\mathbf{f} = \sum_{l=0}^{N-1} \lambda_l \hat{\mathbf{f}}^2(\lambda_l), \tag{1.10}$$

The signal is smoother when the most of the corresponding graph Fourier coefficients are concentrated in the low eigenvalues. This is the case heat kernels for instance. This property is useful for data compression and graph regularization techniques as such signals can be closely estimated by a sparse set of coefficients [92].

1.2.2 Generalized operators

Besides its frequency interpretation and its use for the graph spectral representation, the Graph Fourier Transform has been also valuable for defining generalized operators for graph signals such as filtering, convolution, translation and scaling ... [92] which are some of the ingredients used to develop localized and multiscale transforms on graph signals discussed in a later section.

In essence, in the classical signal processing framework, frequency filtering consists of representing a signal as a linear combination of complex exponentials then attenuating or amplifying contributions of some of those signal components. The Fourier coefficients of the signal are thus multiplied by a so-called transfer function. An inverse Fourier transform of the resulting filtered coefficients corresponds to the convolution in the time domain. Intuitively, and since a Graph Fourier Transform is well-defined (in Eq (1.8)), the outcome $\hat{\mathbf{f}}_{out}$ of the filtering of a signal $\mathbf{f}$ on a graph $\mathcal{G}$ with a graph filter with transfer function $\mathbf{h}$ is defined in the graph spectral domain as the multiplication of the Graph Fourier coefficients $\hat{\mathbf{f}}(\lambda)$ with the transfer function $\hat{\mathbf{h}}(\lambda)$ as follows:

$$\hat{\mathbf{f}}_{out}(\lambda) = \hat{\mathbf{f}}(\lambda)\hat{\mathbf{h}}(\lambda) \qquad \forall \lambda \in \sigma(\mathbf{L}) \tag{1.11}$$

In the above mentioned formulation, $\sigma(\mathbf{L})$ denotes the spectrum of the graph $\mathcal{G}$. Equivalently, the filtered signal in the vertex domain is consequently computed by taking the inverse Fourier transform of the result of Eq. (1.11), such that:

$$\mathbf{f}_{out}(n) = \sum_{l=0}^{N-1} \hat{\mathbf{f}}(\lambda)\hat{\mathbf{h}}(\lambda)\mathbf{u}_l(\lambda) \tag{1.12}$$

In matrix notations, we can also write the former Equation as

$$\mathbf{f}_{out} = \hat{\mathbf{h}}(\mathbf{L})\mathbf{f} \tag{1.13}$$

Where

$$\hat{\mathbf{h}}(\mathbf{L}) = \mathbf{U} \begin{pmatrix} \hat{\mathbf{h}}(\lambda_0) & 0 & \cdots & 0 \\ 0 & \hat{\mathbf{h}}(\lambda_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\mathbf{h}}(\lambda_{N-1}) \end{pmatrix} \mathbf{U}^T. \tag{1.14}$$

When the graph filter $\mathbf{h}$ is a polynomial of order K with coefficients $\{\alpha_k, \quad k = 0, \dots K\}$ such that:

$$\mathbf{u}(\lambda_l) = \sum_{k=0}^{K} \alpha_k \lambda_l^k \tag{1.15}$$

The frequency filtering's outcome of an input signal $\mathbf{f}(n)$ at a vertex n can be inferred as a linear combination of the signal values at vertices restricted within K-hop neighborhood of the node n. [92]

This property is suitable for designing signals that are well localized in the vertex domain and was extensively exploited by many researchers in the graph signal processing field more particularly for learning dictionaries highlighted in a later section [116].

Furthermore, enforcing the well-known property that convolution in the vertex domain is equivalent to a multiplication in the frequency domain, a generalized convolution is defined by taking the inverse Fourier transform of a multiplication of two signals in the graph spectral domain. Thus, given two signals $\mathbf{f}$ and $\mathbf{g}$ residing on vertices of the same graph structure, the result of their convolution on a vertex n is computed by:

$$(\mathbf{f} * \mathbf{g})(n) = \sum_{l=0}^{N-1} \hat{\mathbf{f}}(\lambda_l)\hat{\mathbf{g}}(\lambda_l)\mathbf{u}_l(n) \tag{1.16}$$

The definition of convolution can't be directly generalized in the vertex domain because of the unavoidable fact that weighted graphs are irregular structures which lack a shift-invariant notion of translation in the vertex domain. The translation in the frequency domain can still be generalized though. More precisely, the translation $T_v\mathbf{f}(n)$ of a signal $\mathbf{f}$ to a node v can be defined as a convolution (following from Eq. 1.16) with a Kronecker function centered at vertex v (δ_v) as in Eq. (1.17):

$$T_v\mathbf{f}(n) = \sqrt{N}(\mathbf{f} * \delta_v(n)) = \sqrt{N}\sum_{l=0}^{N-1} \hat{\mathbf{f}}(\lambda_l)\mathbf{u}_l^*(n)\mathbf{u}_l \tag{1.17}$$

$\sqrt{N}$ is a normalizing constant which ensures that the translation operator preserves the mean of the signal. The Kronecker function δ_v is an N-dimensional signal that is equal to one at node v and zero everywhere else. An example of the translation of a signal to different locations on the graph is illustrated in Figure (1.5). Looking more closely at the properties of the translated signals on such irregular topologies, it is clear that the signal does not maintain its original values. The translation operator is actually a kernelized operator that acts on the kernel $\hat{\mathbf{f}}(.)$ defined directly in the graph spectral domain. In these examples, we can understand the subtle difference between the standard translation on a regular structure and the generalized translation on graphs.

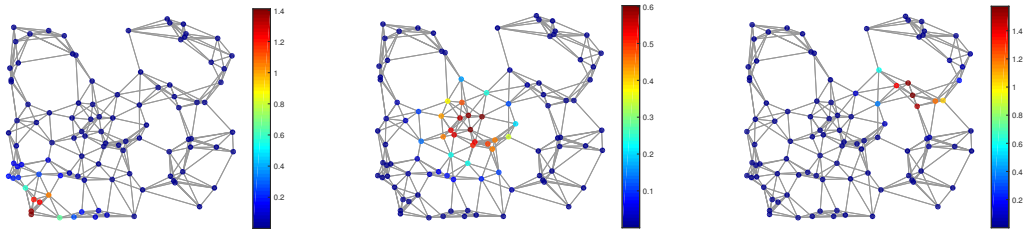


Figure 1.5: The translated signals (a) $T_{20}\mathbf{f}$ (b) $T_{53}\mathbf{f}$ (c)$T_{78}\mathbf{f}$, where $\mathbf{f}$ is the heat kernel shown in figure (1.4)

Moving to another essential operator mainly for wavelets, scaling or dilation can be generalized to the graph setting by performing the scaling to the Graph Fourier domain. Assuming a kernel $\hat{\mathbf{f}}(\lambda) : \mathbb{R}^+ \rightarrow \mathbb{R}$, we can define the dilation of a signal$\mathbf{f}$ by a factor s in the Graph Fourier domain by:

$$\widehat{D_s\mathbf{f}}(\lambda) = \hat{\mathbf{f}}(s\lambda) \tag{1.18}$$

A key thing to note here that the scaling requires the kernel $\hat{\mathbf{f}}$ to be defined on the entire real-line, not

only on $\sigma(\mathbf{L})$ [43].

The generalized operators detailed above are mainly defined in the spectral domain. Other works such as [77] have defined translation and convolution in the spatial domain, since the translation in the spectral domain is not really satisfying and does not preserve neighborhood information. However, in our framework, we choose to work in the continuation of the spectral-based methods.

Having discussed the major operators on graphs, we will discuss in the following section the transforms that have been proposed relying on them.

1.3 Graph based transforms

Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{W}, \mathbf{V}\}$ where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ the set of edges drawn between the nodes. $\mathbf{A}$, $\mathbf{W}$ and $\mathbf{V}$ are the Adjacency, weights and self-loops matrices respectively. Also, consider the signal $\mathbf{x}$ is living on the vertices set $\mathcal{V}$. A large number of Graph-based transforms have been proposed in the graph signal processing literature. They can be split into two categories: Non-localized and Localized.

1.3.1 Non-localized transforms: Fourier-like

Formally, for the signal $\mathbf{x}$ defined on the vertices of the Graph $\mathcal{G}$, the graph Fourier transform $\hat{\mathbf{x}}(\lambda_l)$ at frequency λ_l is defined as the inner product with the associated eigenvector $\mathbf{u}_l$ (see Equation 1.8 in section 1.1). ifferent types of Laplacian matrices have been defined. We can thus characterize various Graph Fourier Transforms depending on the matrix involved in the Transform.

(1) **Unweighted Graph Fourier Transform (uGFT)** corresponds to projecting the signal onto the eigenspace of the unweighted combinatorial graph Laplacian of the graph $\mathcal{G}$.

(2) **Weighted Graph Fourier Transform (wGFT)** corresponds to projecting the signal onto the eigenspace of the weighted combinatorial graph Laplacian of the graph $\mathcal{G}$.

(3) **Unweighted Normalized Graph Fourier Transform (uNGFT)** corresponds to projecting the signal onto the eigenspace of the normalized unweighted graph Laplacian of the graph $\mathcal{G}$.

(4) **Weighted Normalized Graph Fourier Transform (wNGFT)** corresponds to projecting the signal onto the eigenspace of the normalized weighted graph Laplacian matrix of the graph $\mathcal{G}$.

(5) **Generalized Graph Fourier Transform (gGFT)** corresponds to projecting the signal onto the eigenspace of the generalized graph Laplacian of the graph $\mathcal{G}$.

The Graph Fourier Transform is global as most of the eigenvectors of the graph Laplacian matrices are not localized in the vertex domain of the graph. Yet, many existing applications such as the analysis of graph signals, source localization and detection necessitate the localization of the transforms.

1.3.2 Localized transforms: Wavelet-like

To provide localization in the vertex and spectral domains, wavelet-like transforms for signals on graphs [21, 73, 71, 70, 43, 108] have gained considerable attention mostly due to their capability of providing multiresolution representations obtained with the definition of graph translation and scaling. Moreover, these signal representations can be used to develop local analysis tools, so that a graph signal can be treated "locally" around a vertex using data residing in its small neighborhood. Existing designs of wavelet-like filterbanks on the graph can be divided into two types, namely, spatial and spectral designs.

The vertex domain designs of graph wavelets and transforms are founded on spatial features of the graph support such as k-hop neighborhoods and k-hop connectivity between nodes. Wang and Ramchandran [108] designed spatially localized transforms for sensor network graphs with binary edges (having weights equal to 0 or 1). They proposed computing a weighted average or a weighted difference in a k-hop neighborhood around each node in the graph. Their approach intuitively defines a two-channel wavelet filterbank with approximation and detail filters, however suffers from two main glitches: a non-zero DC response of the filters and an oversampled output. In [21], the proposed graph wavelets are effectively localized in space with respect to a range of positions and scaling indices. Specifically designed in the vertex domain to analyze network traffic, the wavelet's construction relies on the geodesic distance or the shortest path between nodes on the graph. Yet, the construction algorithm is restricted to unweighted graphs, and the transform is generally not invertible.

Additionally, graph wavelet lifting schemes have been proposed in [89] [72] and offer a natural way of constructing local two-channel critically sampled filterbanks on graph-signals. In such approaches, the vertices are divided into two sets, the odd and the even nodes using graph coloring techniques [50]. The odd nodes compute their "prediction" coefficients using their own data and data from their even neighbors, then even nodes compute their "update" coefficients from their own data and the prediction coefficients of their odd neighboring nodes. Although the transform can be applied on any arbitrary graph, the scheme by itself dictates that any two nodes having the same parity can't use each other's data even if they are connected by an edge. This remains as the underlying reason why those schemes do not provide optimal signal decorrelations.

The graph spectral domain designs of graph wavelets are based on the spectral features encoded in the eigenvalues and eigenvectors of the graph Laplacian. The main goal behind those designs is providing localized bases in both the vertex and the graph frequency domains. Coifman and Maggioni [17] introduced the "diffusion wavelet" which interacts with the underlying graph structure through the repeated applications of the powers of a diffusion operator (such as Laplacian) to capture different resolutions. The localized basis functions at each resolution are then downsampled and orthogonalized appropriately. Another approach proposed to define wavelets relies on the precise analogy with the time domain wavelets that is translating and dilating band-pass filters defined in the graph spectral domain [43]. The graph wavelet filterbanks defined in [73, 71, 70] are fundamentally inspired from the classical multiresolution analysis based on filter banks in the Euclidean domain. Under some conditions, the defined filter-banks are critically sampled and can either be orthogonal with a perfect reconstruction [73] or bi-orthogonal and localized with a compact support at the expense of a small reconstruction error [71]. We refer readers to the above cited papers and references therein for more thorough analysis and details.

The aforementioned transforms draw on pre-defined structures merely derived from the graph and some of them can be efficiently implemented. However, they are commonly not well-adapted to the signals in hand. Addressing this issue, very few researchers dedicated their work to offer an extra adaptivity of the transforms. More precisely, the authors in [9] used a large number of bases, each one adapted to specific space and frequency localization properties, to construct diffusion wavelet packets. The choice of the bases depends on the task in hand. Another approach consisted of the use of deep learning to design lifting schemes that resembles a deep auto-encoder network [85]. In both works, nevertheless, the training signals living on the graph vertices are not taken into account. In addition, the definition of the trees in [82] denotes the geometry and the structure of the input data and the adaptivity is obtained after reordering and permutations derived from the tree which inevitably makes the performance of the scheme dependent on the tree construction and reordering involved.

Later on, there have been a growing interest in learning structured dictionaries from signals on graphs in order to sparsely represent the graph signals at a low computational cost. Zhang et al [116] were the first to introduce structured dictionaries for signals living on arbitrary graphs. Making use of the graph Laplacian operator, the algorithm is built on a sparse approximation step followed by an updating step which iteratively leads to a structured dictionary. The learned dictionaries were able to capture the spectral features of the different considered signals, in return providing sparser representations.

Afterwards, Thanou et al [101] have proposed a parametric family of structured dictionaries formulated as unions of polynomial matrix functions of the graph Laplacian, to sparsely represent signals on a given weighted graph, and an algorithm to efficiently learn the parameters of a dictionary belonging to this family from a set of training signals on the graph. When translated to a specific vertex, the learned polynomial kernels in the graph spectral domain correspond to localized patterns in the graph vertex domain. The translation on different locations in the graph leads to an efficient and sparse signal representation with a greater performance than non-adapted transforms such as spectral graph wavelets, and comparable to state of the art algorithms as K-SVD.

Now that we have surveyed the graph based transforms already proposed in the literature, we focus in the following on their use in image compression schemes. In the following section, we only give a brief state of the art relevant to the work in this thesis. A more complete overview about graph based transforms designed for image compression can be found in [14].

1.4 Graph based transforms in image compression

In essence, image compression consists of encoding an image $\mathbf{I}$ onto some code-word c, that is carefully chosen in a way that the distortion on the reconstructed image $\tilde{\mathbf{I}}$ is minimized under a total bitrate constraint. This minimization can be written as:

$$\min_{\mathbf{c}} D(\mathbf{I}, \tilde{\mathbf{I}}) + \lambda R(\mathbf{c}(\mathbf{I})) \tag{1.19}$$

where $R(\mathbf{c}(\mathbf{I}))$ is the average code-word length. In most of the lossy compression schemes, a first step

consists of transforming the signal and projecting it in another domain. At the output of this stage, we have a novel image representation made of coefficients $\hat{\mathbf{I}}$ that are approximately uncorrelated. Also, we aim at having most of the energy compacted in fewer coefficients and thus the sparsity of the output is a researched asset. We refer to those two properties as the decorrelation efficiency and energy compaction performance. Those are critical to achieve an acceptable compression performance. The coefficients $\hat{\mathbf{I}}$ are subsequently quantized, and the quantization indices are coded with with some lossless compression algorithms such as Huffman or arithmetic coding. Note that if the transform (*i.e.* the projection matrix) to be used is not known in advance in both encoder and decoder, then the total rate consists of two different terms. The former consists of the bits needed to code the quantization indices. The latter is made of auxiliary information for the decoder to be able to reproduce the suitable inverse transform and retrieve the original image signal. Both terms are maybe dependent on $\mathbf{I}$, making the design of adaptive transforms a challenging problem.

One of the first proposed transforms is the *Karhunen-Loeve transform* (KLT). It is based on the eigen-decomposition of the estimated covariance matrix of the input process. Projecting the signal into the eigenvectors of its covariance matrix has been shown to be optimal under mean square error metric and fixed-rate coding [42]. The *Discrete Cosine Transform* (DCT) [96] is almost equivalent to the KLT for a first order auto regressive process [49]. While many common transforms used in image compression schemes (for example JPEG and JPEG 2000 and HEVC), such as the DCT and wavelet transforms [26], make use of a fixed set of basis vectors that do not need to be communicated to the decoder side, the KLT is a signal-adaptive transform and the necessity of sending additional information is stands still as a challenging problem. Also, the KLT has no structure and lacks any fast implementation. Those are the main disadvantages that have limited the use of the KLT in image compression.

Nevertheless, all the models defined before based on stationary Gaussian assumptions fail to capture the complex and non stationary behavior typically occurring in digital images. In this context, the *Graph Fourier Transform* has been proposed and can be seen as an adaptive DCT taking into account both the image structure and the image signal in hand. Like the KLT, it is based on an eigendecomposition of a kernel matrix (Laplacian matrix). As has been seen in 1.1, the graph topology $\mathbf{A}$ and set of weights $\mathbf{W}$ fully define the graph Laplacian matrices, from which the Graph transforms are computed. Hence, obtaining a "good" Graph Fourier Transform depends on the selection of the topology and weights yielding the best compression performance in an RD sense as in 1.19.

When the data to be handled is already structured, the knowledge of meaningful graph topology $\mathbf{A}$, weights $\mathbf{W}$ and self-loops $\mathbf{V}$ (if they exist), plays a crucial role in the success of graph-based representations and transforms for compression applications. More precisely, if a function is assumed to be smooth or piecewise smooth on a graph, it can be described by a very small number of coefficients in a well-chosen basis which is usually related to the graph topology ($\mathbf{A}$) or model ($\mathbf{W}$). For certain types of signals such as 1D and 2D images, one can often construct the graph in an intuitive manner: Each pixel of the image is represented by a node in the graph. Edges are drawn between neighboring nodes intuitively based on the knowledge of the structural information. A signal lying on a structure with Adjacency matrix $\mathbf{A}$ can be transformed using the eigenvectors of the unweighted combinatorial graph Laplacian $\mathbf{L}_u$ or the normalized unweighted graph Laplacian $\mathcal{L}_u$. Intuitive models are convenient to analyze since usually such graphs are not highly connected thus the precision matrix is sparse. The very

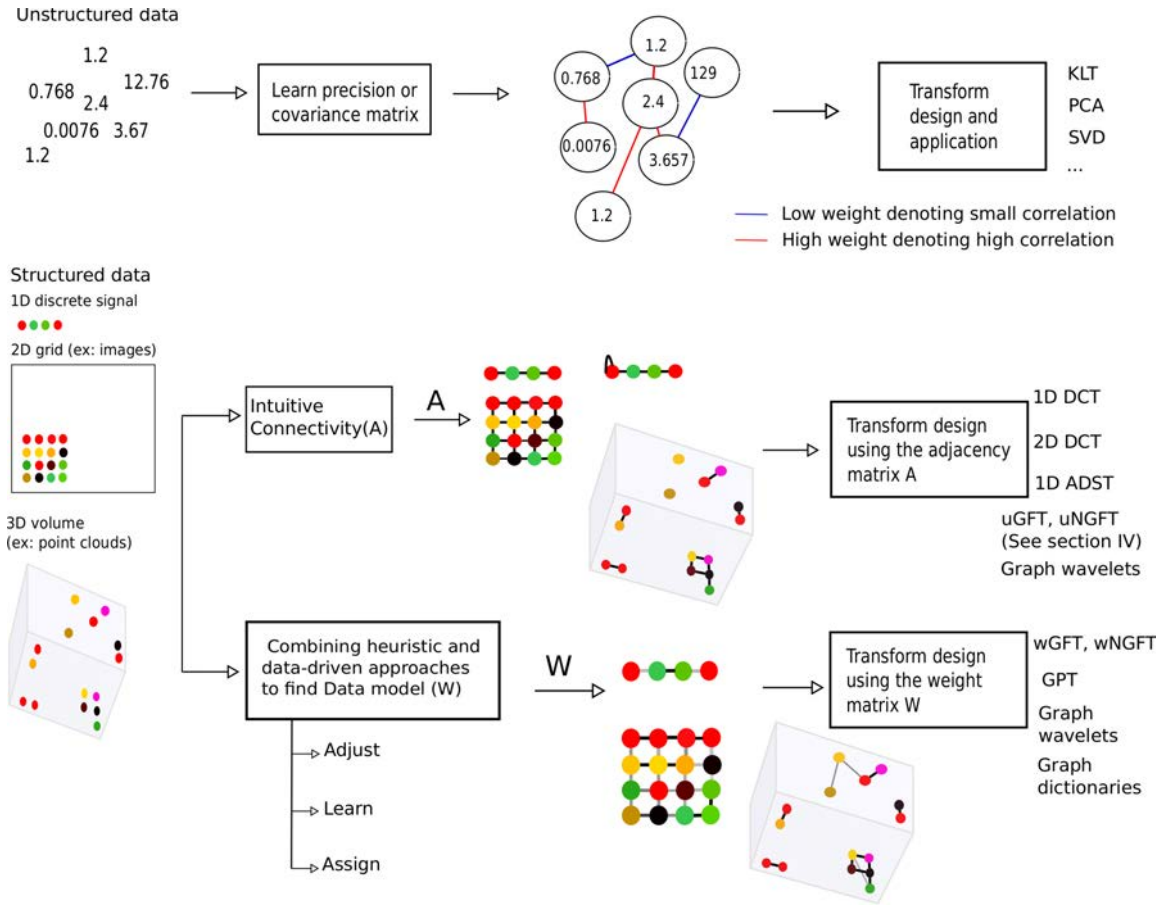


Figure 1.6: Different strategies to design a decorrelating transform

well known transform DCT falls in this category. For example, behind the 2D-DCT, lies an assumption of a 4-neighbors connected graph where all weights are unity [114]. The 1D-ADST is a Graph Based Transform derived from the generalized graph Laplacian $\mathbf{L_g}$ of a line graph whose weights are all equal to w_u and having a single self-loop on the first sample with the same weight w_u [32]. Likewise, relying on the structure of the graph and its spectral properties, structured graph dictionaries [116] can be learned and Graph wavelet filter banks [73] [71] can be used to transform the signal.

Ideally, the heuristic intuition-based and data-driven schemes should be combined together for better performance especially when the structural information is not entirely reliable and meaningful. Three major design approaches exist for that case:

(i) **Adjusting** the already known topology and intuitive model matrices depending on signal characteristics.

(ii) **Learning** new model from the data taking into account the underlying structure.

(iii) **Assigning** model weights depending on the inherent graph topology and neighborhood information.

The weight w_{ij} on each edge of the graph is conventionally defined as a function of the difference in pixel values $\mathbf{I}_i$ and $\mathbf{I}_j$ connected by that edge. Real-valued graph weights are however too expensive in terms of signaling rate. To overcome this issue without loosing adaptivity, the weights are constrained to be in the set $\{1, 0\}$ in [90] [58], [35]. This implies that the weights are restricted to describe strong or zero correlations; the weights can be defined as a result of an edge detection algorithms [90], using some greedy optimization algorithms [58]. In [35], the authors proposed to segment an image into uniform regions that adhere well to object boundaries (the so-called superpixels) and apply a unweighted Graph Fourier Transform within each superpixel. This method can be seen as removing unreliable links in the whole graph representing the image therefore avoiding the filtering around the edges and reducing the overhead of representing the graph structure within each superpixel as well. In [33], the difference $|\mathbf{I}_i - \mathbf{I}_j|$ is quantized to two values using a pdf-optimized uniform quantizer, yielding a graph that is always connected by construction; although weight binarization leads to suboptimal compression efficiency, it is shown that a suitably designed quantizer makes the performance loss very small. Offering more adaptivity, in [48], two sets of weight values are used, i.e. $w_{ij} = \{1; 0\}$ for image blocks characterized by strong or zero correlation, and $w_{ij} = \{1; c\}$ for blocks exhibiting strong or weak correlation. The constant c is optimized using a model suitable for piecewise smooth signals, and very good results are obtained in the compression of depth map images. However, the overhead incurred by the graph for natural images makes it harder to obtain significant gains. This problem has been addressed in [33], where edge prediction followed by coding is used to reduce the overhead, leading to performance gains between 1 and 3 dB in peak signal-to noise ratio (PSNR) over the DCT. More sophisticated graph coding techniques may also contribute in reducing the overhead, e.g. one might think of applying contour coding techniques as in [118], [23], [104] to reduce the cost of representing the graph. Moreover, in [104] directional graph weight prediction modes are proposed, which avoid transmitting any overhead information to the decoder.

For natural image compression, in [32], two graph topology and model adaptations were proposed for inter-predicted residual blocks taking into account the statistical properties of the residuals. The first method is edge-adaptive where graph link weights are reduced depending on the directionality of the edges detected in the image. A similar work [84] explores the structure tensor's properties to design graph Adjacency templates before computing the optimal edge weights that successfully describe the inter-pixel correlations. One of the main drawbacks of such methods lies in the cost required to represent and encode the graph, which may outweigh the coding gain provided by the edge-adapted transform. To reduce the signaling overhead, the coder can choose one out of some fixed graph templates to describe the signal by minimizing its total variation with respect to the Graph [32]. An alternative strategy consists of maximizing the Gaussian likelihood subject to a constraint set defined by a (matrix type - graph template) pair as in [78]. However, with the previously evoked set of constraints on the graphs, the resulting transform may not retain the advantages of the edge-aware operator.

As for the problem of estimating and learning models from the data taking into account the known structure, recently, there has been a growing interest in learning Laplacian matrices (with non negative weights) that can successfully describe the graph signals due to their spectral properties and intuitive interpretations. Dong et al [29] estimate the graph Laplacians under signal smoothness priors with respect to the graph. To this end, a factor analysis model for the graph signals is adopted and a Gaussian

prior on the latent variables is imposed to control these signals. In their algorithm, the learned Laplacian is combinatorial positive semi-definite. In a more recent work [31], more generalized graph estimation problems are formulated and three different Laplacian matrix types are estimated: The generalized Laplacian, the diagonally dominant Laplacian, and the combinatorial one. Unlike the previously cited work, the estimation does not rely on approximations or smoothness priors, however on a strong minimization of the following objective function:

$$\underbrace{Tr(\mathbf{\Theta S}) - logdet(\mathbf{\Theta})}_{\mathcal{D}(\mathbf{\Theta},\mathbf{S})} + \underbrace{\frac{\alpha}{2}||\mathbf{\Theta}||_{1,off}}_{\mathcal{R}(\mathbf{\Theta},\alpha)} \tag{1.20}$$

where $\mathbf{\Theta}$ is the target variable matrix (i.e., a specific type of graph Laplacian), $\mathcal{R}(\mathbf{\Theta}, \alpha)$ refers to the sparsity promoting regularization term with parameter α and $\mathcal{D}(\mathbf{\Theta}, \mathbf{S})$ is the data fidelity term, whose minimization is equivalent to the maximum likelihood estimate of precision matrices under the assumption of an attractive Gaussian Markov Random Field (GMRF). Prior knowledge about the graph connectivity (data structure) is built into the choice of the added structural constraints to the main minimization problem.

Naturally, learning problems are complex and require a lot of computational efficiency especially for large data and graphs. To alleviate this problem, for nodes that lie on a vertex set $\mathcal{V}$ embedded in an euclidean domain (2D images, 3D scenes, voxelized point clouds), it is common practice for the neighborhood structure (the weights matrix) of the underlying graph to be inherited from the neighborhood structure of the containing domain since the structure of the data is more meaningful and can successfully model the inter-pixels correlations. More precisely, for signals living on such nodes, the signal model precision is usually defined using the neighborhood information provided by the structure graph. For instance, in [15], the authors examined two weights models: the auto-regressive model and the inverse-distance model. The former fits the best weights to the signal in hand under some auto-regressive assumptions. The latter, also applied in [115], exploits the neighborhood information by assigning weights that fall off inversely with distance up to a threshold. A similar strategy was adopted in [69] where the weights of the links are computed as an exponential term of the euclidean distance square of the inter-pixel distance.

As an alternative to the weight functions, and under the assumption of a Gaussian process, covariance functions can be used to characterize the signal. For example, one can estimate the covariance function by assuming that it depends only on the distance between the nodes. Authors in [15] refer to this as the Non-Parametric model *NP Model*. Another option is to assume that the graph signal values are samples of an Ornstein-Uhlenbek process and thus define the covariance as a function of the distance with only one parameter which models its decay.

Once the graph Adjacency, Weights or Covariance functions are well defined, the signal is then transformed using the graph transforms as outlined in section 1.3 or traditional Karhunen-Loeve transforms. In the rest of the thesis, we will restrict our attention to the special case of structured data where the signals are sampled on nodes embedded in a structured domain. Light fields and omni-directional images are two examples of structured data and will be extensively studied in the following chapters.

Chapter 2

Background on light fields and omni-directional imaging

2.1 Light Field imaging

From a purely geometric point of view, light is composed of rays: directed lines in 3D space that carry a certain intensity value, or radiance. Conventional photographs do not record most of the information about the rays flowing in the camera at the moment of capture. For example, if we think about the light striking a sensor and consequently contributing in one pixel, the photograph tells us nothing about the distribution of the rays and their individual contribution in the final pixel. It turns out this geometrical information is the crucial piece of the omitted information that leads to different problems in conventional photography. This has led to the development of the light fields concept: a representation of a 3D scene as a collection of light rays coming through every point in space and flowing in all possible directions.

In this section, we introduce the basic notion of light fields, which represents the total geometrical distribution of the rays entering the camera at the moment of capture. For the sake of completeness, we then briefly overview the acquisition techniques and the most prominent applications. We then focus on the major problem related to the core of the thesis which is the compression of the large amount data that a light field represents.

2.1.1 Formal definition

The Plenoptic function

Thinking about the geometrical distribution and representation of the light travelling in the world has an extensive investigation history. Adelson and Bergen [2] P were among the first in this academic movement, leading the way as they published an influential paper stipulating the representation of a light ray $\mathcal{R}$ as a 7 dimensional (7D) function known as the *Plenoptic function*:

$$\mathcal{R} = \Pi(x, y, z, \theta, \phi, \lambda, t) \tag{2.1}$$

The first three dimensions outlining the ray position, θ and ϕ define its direction, for all wavelengths λ and at every time t. The sampling of the temporal dimension is usually dependent on the capturing device's frame rate and the wavelength is decomposed into 3 channels, namely the Red-Green-Blue (RGB) components. Without any consideration about the range and sampling of the function, intuitively,

the plenoptic function can be used to represent all possible observable scenes. For instance, to generate a conventional photo, a pinhole camera samples Π at a range of (θ, ϕ) for a fixed (X, Y, Z).

In practice, this function can be simplified by omitting dimensions. McMillan and Bishop [68] reduced the 7D plenoptic function to 5D by removing the wavelength and time parameters (See Fig. 2.1). They recorded cylindrical views (2D) of a static scene at multiple 3D camera positions in order to sample the 5D function.

The Lumigraph

Nevertheless, it is very difficult to capture the entire or even a bounded sampling of the plenoptic function. And even if we can do so, this will lead to highly redundant information. Indeed, if we take the reasonable assumption that the air around an object does not absorb or deflect light (i.e has a transmittance of 1 and a constant refractive index), all the ray intensities remain almost constant along their path. This observation led Levoy and Hanrahan [62] to introduce to computer graphics the notion of 4D light field so called *Lumigraph*, hence eliminating 3 dimensions assuming that all rays are flowing in free-space for the same wavelength at a fixed time. Although the 4D collection of rays can be parameterized in different ways, the simplest way is to use two parallel planes (The so called Two planes parameterization). More precisely, each ray is described by its intersection with two distinct and parallel planes that we denote (x, y) and (u, v) (as illustrated on Fig. 2.1). The plenoptic function is thus reduced to the 4D light field function :

$$\mathcal{R} = L(u, v, x, y) \tag{2.2}$$

Similarly to the plenoptic function, if the light field is colored then we can add a color dimension, and a dynamic light field (a video) adds the temporal dimension.

By convention, (u, v) are the *angular dimensions*, as they define the angle of the ray relative to the second intersection (x, y), the *spatial dimensions*.

2.1.2 Light field rendering

Lately, computational imaging has gained a considerable attention promising a wind range of applications in security and medicine. Researchers in computer graphics, and computer vision, have explored the acquisitions of light fields and built devices to capture them in order to enable many post-capture functionalities when coupled with some computational approaches. light field's gathering requires taking an important amount of photographs of the same entity from different perspectives under unchanged light conditions. This notion is so-called the *light lield rendering*.

Levoy and Hanrahan [62] have proposed moving a single-camera across the scene with a changing time of capture. An alternative technique suggested by Wilburn et al [110] was the use of an array of cameras which revealed the "see-through" effect enabled by the light field post-capture processing, precisely, the digital refocusing. If the cameras are arranged along a 1D path, then displaying the views successively would give the impression of orbiting around the scene at a constant time. If arranged in 2D arrays, then the full light field is captured. Doing so, new focused images can be generated computationally.

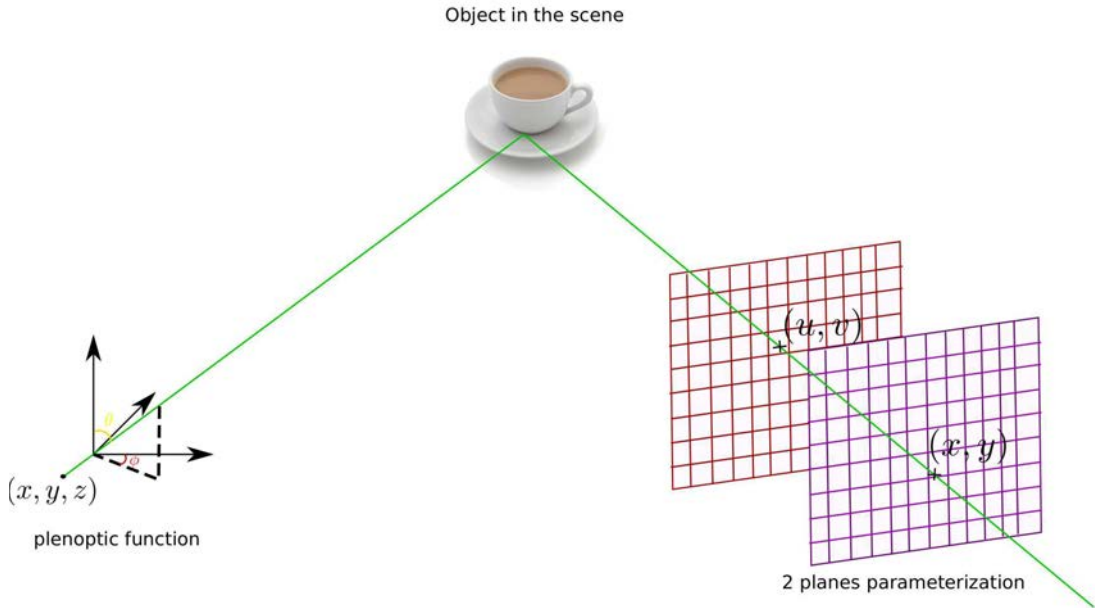


Figure 2.1: 5D Plenoptic function vs the two-plane parameterization: On the left, the ray is sampled with the plenoptic function by 5 coordinates, (θ, ϕ) being the direction and (x, y, z) the 3D position of the generating point. The ray on the right is sampled with the two-plane parameterization. A ray is described by 4 coordinates corresponding to its intersection points with two parallel planes noted (u, v) and (x, y).

In 2006, Ren Ng [76] proposed the complete architecture of a plenoptic camera 1.0 i.e. light field camera, by introducing a microlens array positioned above the standard photosensors. In such camera, sensors under each microlens record light striking the microlens from different positions spread over the main lens aperture. This constitutes the light field, whose (X, Y) or spatial resolution, depends on the number of microlenses and whose (U, V) or angular resolution depends on the number of pixels behind each microlens. Because the rays are multiplexed by direction into a single sensor, plenoptic cameras trade spatial resolution for angular resolution. And because the distance between the microlenses is very small, their angular sampling is dense. Conversely, because there is a limit to the number of microlenses we can use, the generated images are fairly low resolution: the spatial sampling is sparse. Therefore with a fixed sensor resolution, collecting more directional information necessitates scarifying in final output spatial resolution.

An enhanced plenoptic camera 2.0 has been proposed to overcome this issue and was proposed in a more recent paper [40]. The main difference between this camera and the former one is that instead of placing micro lenses at the focal distance from the sensor plane, the array is positioned at a distance a from the main lens aperture and at another distance b from the sensor plane. This difference results in a rise of the capability of the plenoptic camera 2.0 since the former had an output of simply one pixel per microlens whereas this one provides many pixels yielding improved rendering results, and greater spatial resolution.

Additionally, if we move further down the scale of the sights we might want to perceive, and as suggested by Levoy[61], adding a microlens array to a traditional microscope will contribute with no

doubt in a variety of additional features that can be exploited in future investigations in the medicine and science fields.

There are a few plenoptic cameras available in the market. For instance, the *Lytro 1* and *Lytro Illum* are two affordable cameras focused to the consumer level, while the company *Raytrix* focuses on making high-end focused plenoptic cameras for the industry. Some high-end camera (e.g. the *Canon 5D Mark IV*) also have a microlens covering two pixels and could arguably be considered as a light field camera, however in practice they are used for auto-focusing.

2.1.3 Light field representations

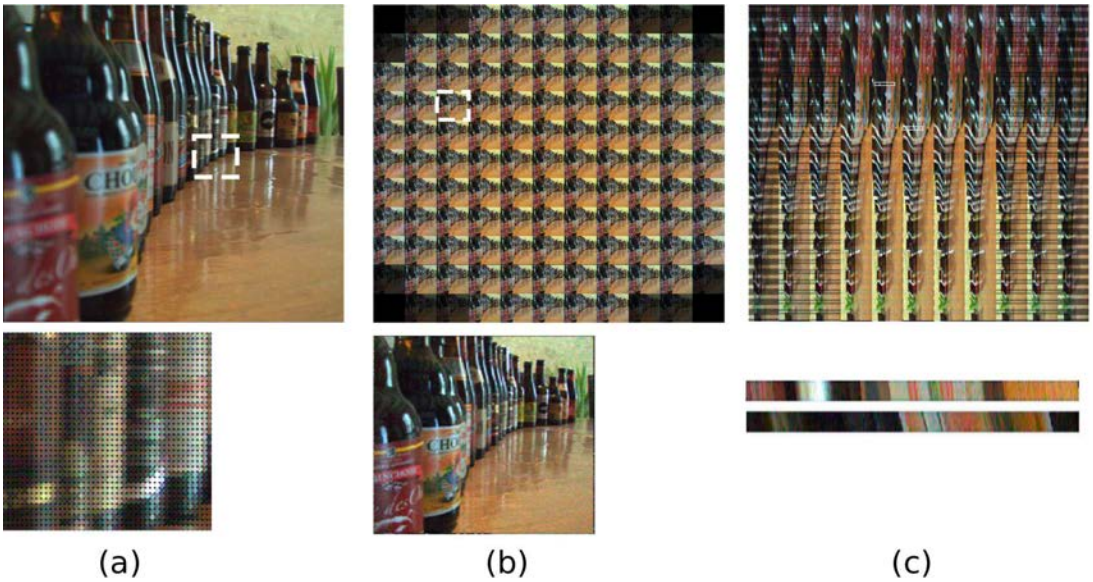


Figure 2.2: Example of light field captured with a plenoptic camera. (a) The raw photograph is an (x, y) grid of images, where each one reveals the amount of light striking that microlense from different (u, v) positions on the main lens aperture. (b) Reorganizing the pixels by angle into different images gives a matrix of views called sub-aperture images however with a small baseline. (c) An epipolar plane image shows, for a fixed pair of vertical (or horizontal) angular and spatial coordinates, the variations in intensities depending on the horizontal (or vertical) angular and spatial coordinates. This last representation is mostly used for depth estimation.

When sampling the light travelling along the rays inflowing the plenoptic camera at the moment of capture, the light field offers rich material about the scene. One way to think about it is that it provides information about the light striking each microlense. This is the *raw light field photograph* which is usually hard to visualize.

A more popular representation is the *sub-aperture* representation which consists in gathering pixels with the same relative position in a microlens image and placing them in a new image depending of the microlens image it came from. Doing so, we obtain an array of images similar to a camera array view matrix however with much smaller baseline. It is easy to see that the view resolution depends on the number of microlenses while the number of subaperture images depends on the number of pixels

underneath each microlens.

Another way of slicing such light field could be noted $L(u, x)$ and $L(v, y)$. First proposed to analyze a spatio-temporal volume , Epipolar Plane Images (EPIs) intuitively offer a way to visualize pixel intensities as a function of its horizontal (or vertical) spatial and angular coordinates. The obtained image, as in 2.2(c), is of particular interest as it allows direct study of intensity variations and thus gives interesting information about the geometry and the disparity of objects in the scene.

2.1.4 Applications

We can generally classify light field applications in two main classes. First, we have the depth estimation methods, that seek to expressly recover the scene geometry from the captured light field. Then image rendering techniques that aim at producing conventional images from the extracted geometrical information. Because the application fields are not related to the core of the thesis, we only give a non-exhaustive overview, rather than a comprehensive list of applications. We also focus on lenslet-based light field images which will be considered in the rest of the thesis.

Depth estimation

Depth estimation from a pair of stereo cameras is one of the most studied problem in the computer vision literature [88]. Notwithstanding that the classical correlation-based methods can be extended to several views instead of two [3][22][113], there are various different ways of leveraging the views redundancy that are exclusive to light fields.

Having very narrow baselines, lenslet-based light field images could not be efficiently used in stereo matching techniques as they usually involve interpolation with blurriness due to sub-pixel shifts in the spatial domain. Researchers have been devoted to find other constraints and cues for estimating the depth. One way was to compute cross-correlations between microlens images to estimate the disparity map [41]. Also, the epipolar images (EPI) as mentioned in the previous section have been shown to be very useful for depth estimation. The slope of the line formed by the pixels corresponding to the same scene point in an EPI is proportional to its depth [8]. Structure tensors can then be used on the EPIs to measure the slope and infer the corresponding disparity [109]. The resulting depth values can be regularized [109] or given sufficient sampling, simply filtered [57] to take into account occlusion and produce the final depth map.

Likewise, estimating depth from defocus and correspondence has been studied expansively. Assessing depth from focus (defocus) has been achieved using multiple image exposures or complicated equipment to be able to record the data at the same lighting conditions at the same time. Using defocus cues, we should analyze the optimal contrast within a patch. In such measurements, obstructions may easily affect the outcome therefore it is necessary to use patch based measurements to improve the stability. Also, some out of focus regions such as high frequency regions and bright light can yield to higher contrast yielding many ambiguities in defocus measurements.

Alleviating some of the defocus ambiguities and problems, correspondence cues between multiple views have also been extensively considered to estimate depth maps. However, the only use of this cue did not bring much promising outcomes from traditional light fields facing many errors due to large

stereo displacements with a limited search space. Matching uncertainties and inaccurate results also occur in repetitive patterns and noisy areas of the image.

In the new plenoptic cameras, the array of microlens inserted between the main lens and the photosensors provided sufficient information that one can refocus after light field acquisition, and effectively shift viewpoints within the main lens aperture. Using those two assets, Tao et al. [100] successfully combined both defocus and correspondence cues, relying on a confidence measurement of each cue to finally estimate the depth map.

Image rendering

Light field imaging belongs to the image-based rendering field, where we aim at generating new conventional photographs using the geometrical and texture information captured in the light field. One of the first prominent applications was the generation of new point of views from the 4 dimensional volume in [62].

Digital refocusing or the *Synthetic aperture* from a light field has also been an active research field. With the information we have in a light field, we are capable of retrieving small details in the scene with a high fidelity. The underlying reason of this fidelity is that the refocusing algorithms are based on ray tracing techniques simulating a virtual camera and a sensor that sums the rays with a great precision. More precisely, it is possible to computationally replicate the ray angular integration happening inside of the body of a conventional camera [63]. In the case of the 4D light field, this can be simply done with a *shift and add* procedure of all or a subset of the sub-aperture images.

2.1.5 Light fields compression

In the context of computer graphics techniques, the input for previously detailed rendering and post-capture manipulations comprises geometrical information of the scene along with lighting attributes (texture information). Despite the significant progress in light field acquisition and sampling, it is still challenging to render new photographs in real time because of the computational burden and the high data rate constraints. Additionally, the spatial sampling during the acquisition stage should be fine enough to permit acceptable quality of rendered photographs with a minimal amount of distortion, thus inferring a tremendously large amount of captured image data.

To overcome the former problems, compression and coding techniques are essential for transmission as well as fitting all the information into the local memory during post-capture manipulation, while random access to any light field fragment is also crucial to achieve interactive rendering rates.

Existing light fields compression solutions can be broadly classified into two categories: approaches directly compressing the lenslet images or approaches coding the views extracted from the raw data. The authors in [**chao2017**] propose a coding scheme for light field image compression based on graph-based lifting transform. The scheme is able to encode the original raw data without introducing redundancies from demosaicking and calibration.

Other Methods proposed for compressing the lenslet images mostly extend HEVC intra coding modes by adding new prediction modes to exploit similarity between lenslet images (*e.g.* [18], [19],

[20], [64]). The authors in [99] propose a lenslet-based compression scheme that uses depth, disparity and sparse prediction followed by JPEG-2000 residue coding.

A second category of methods consists in encoding the set of views which can be extracted from the lenslet images after de-vignetting, demosaicing and alignment of the micro-lens array on the sensor, following e.g. the raw data decoding pipeline in [25]. Several methods code the views as pseudo video sequences using HEVC [66], [83], or the latest JEM coder [51], or extend HEVC to multi-view coding [4]. Low rank models as well as local Gaussian mixture models in the 4D rays space are proposed in [55] and [105] respectively. View synthesis based predictive coding has also been investigated in [117] where the authors use a linear approximation computed with Matching Pursuit for disparity based view prediction. The authors in [53] and [97] use instead a the convolutional neural network (CNN) architecture proposed in [56] for view synthesis and prediction. The prediction residue is then coded using HEVC [53], or using local residue transforms (SA-DCT) and coding [97]. Most of the compression schemes have been designed based on the impact on the light field quality. In our work [83], we have studied how the compression of light fields may impact the post-capture functionalities namely the *refocusing* and *extended depth of field*.

2.2 Omni-directional imaging

In this section, we provide a broad overview on *omni-directional* images, that is substantial for the understanding of the rest of the thesis.

2.2.1 Formal definition

In traditional photography, an *omni-directional* camera is a camera that has a large field of view covering approximately the entire sphere or at least a full circle in the horizontal plane. Since it covers all the directional space, it is called *omni-directional* meaning "all-directional". It is also referred to as 360 camera since it covers the 360 degrees of the sphere.

The omni-directional image is thus an image captured by the 360 camera that represents the light activity arriving at a point (the image center) from every direction (360 degrees field of view). In practice, a normal camera can capture at most light falling onto the focal point through a hemisphere. Thus, an omni-directional image is most of the time, the result of several stitching algorithms of images captured by different types of cameras such as catadioptric or fish-eye etc.

2.2.2 Omni-directional image representations

There exist different ways to represent the omni-directional content. The most popular one is the equi-rectangular (or "equi-angular") representation where each point of the sphere (at specific angles θ and ϕ on the sphere) is projected into a pixel in a rectangular image. It has been studied extensively since it can be useful for image processing applications. However, it does suffer from stitching artifacts and radial distortion.

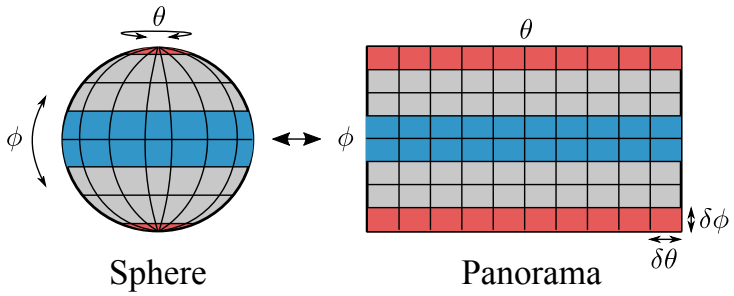


Figure 2.3: Equirectangular representation [67] of a omnidirectional content. Each sample on the sphere is projected on a traditionnal 2D grid to form an equi-rectangular image.

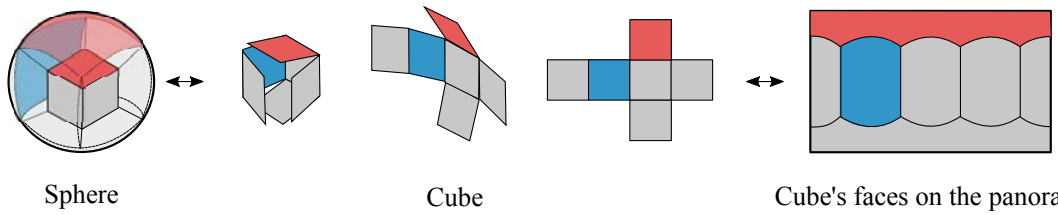


Figure 2.4: Cube maps representation of a omnidirectional content. The environment is projected onto the sides of a cube and unfolded into six regions of a single texture or stored as independent textures.

Another way to represent the content is to use the cube map projection technique. In essence, cube mapping represents a method of environment mapping that uses the six faces of a cube as shown in Fig. 2.4.

An alternative representation is a based on a pyramid. This representation has been adopted as a multi-resolution approach, to provide a view angle at a high resolution (the blue part on Fig. 2.5) and the rest of the sphere is covered however at a lower resolution. This is mainly useful for streaming purpose.

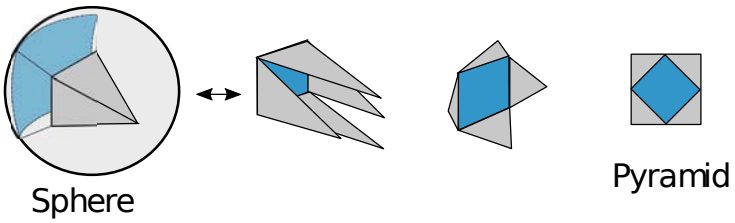


Figure 2.5: Pyramidal representation of a omnidirectional content

While all the aforementioned representations suffer from projection errors and glitches, the uniform sampling on the sphere provides a more flexible approach where we don't deal with a 2D image anymore, the samples lie directly on the sphere (see Fig 2.6).

2.2.3 Coding of omni-directional visual content

Several approaches to code omni-directional visual content have been proposed in the past few years. They can be grouped into three main categories:

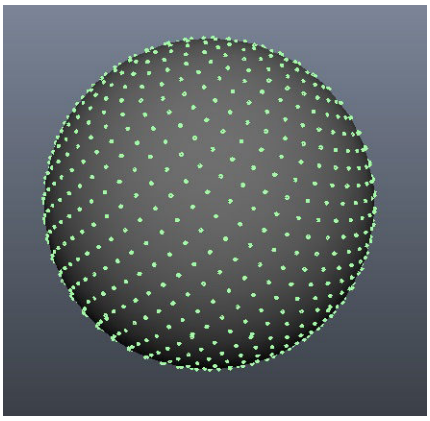


Figure 2.6: Uniform sampling on the sphere for the representation of omnidirectional content

1. Coding based on adaptive and partial content delivery.
2. Coding while exploiting or adapting to the 2D spherical surface geometry of the content.
3. Coding after a geometric representation or projection based method.

One of the very first solutions to code an omnidirectional image was developed by Apple Inc. The so-called Quicktime VR [11] which refers to both a file format and visualization software. It allows for creation and display of panoramic images. More specifically, it stores 360 degree cylindrical panoramic images divided in tiles. For display, only the tiles visible in the current viewport are decoded. The idea of using the Region of Interest (RoI) to code omnidirectional visual content recently appeared in [5]. The authors propose to deliver only the omnidirectional content's part which is being viewed. Each frame, after an equirectangular projection, is divided into regions which are then coded separately with different quality according to an adaptive model. The tiles corresponding to the portion of the frame being viewed are encoded with the highest possible quality. The quality of other regions is determined considering their probability of being viewed next. A major problem in such approaches is that they do not explore the spherical geometry which makes this type of coding sub-optimal.

To deal with this problem, other works [102] investigate coding strategies that take into account a specific 2D spherical surface. Assuming that a raw image can be mapped into a sphere after stitching, the authors proposed a generic compression method based on the decomposition over a dictionary of geometric atoms. A redundant dictionary is built over two generating functions (denoting both low and high frequencies) extended with scaling and affine transformations on a 2D sphere. The matching pursuit algorithm is then performed to choose atoms from the dictionary, followed by sorting the atoms along the decreasing magnitude of their coefficients, and then applying adaptive quantization. The proposed codec outperforms JPEG 2000 at low bitrates.

Representing omni-directional visual content using geometric projections, which produce less amount of data, is another strategy. An example of such approaches can be found in [38] where authors propose a rhombic dodecahedron (RD) mapping model. This convex polyhedron was selected considering the limitation that faces should be of quad-based nature, allowing the construction of unfolded rectangular images. The model provides almost uniform pixel distribution without significant oversampling or

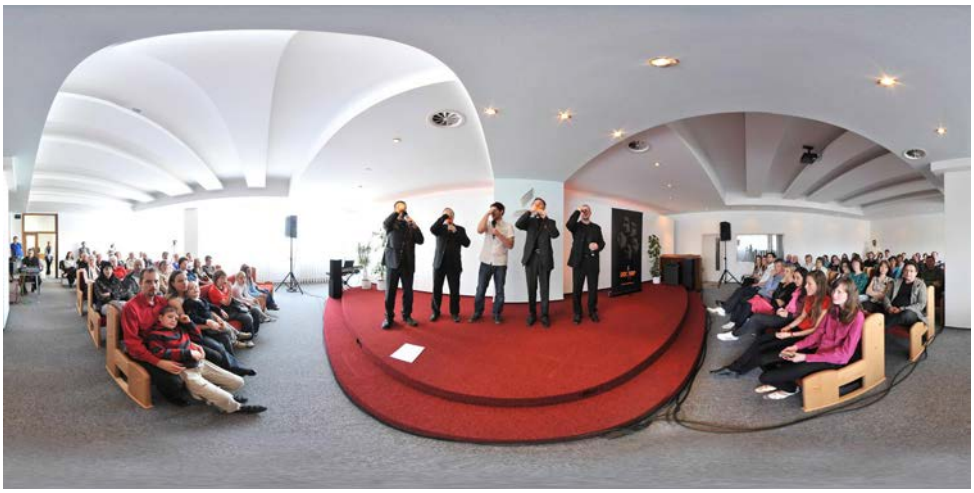


Figure 2.7: Equirectangular planar representation of the omnidirectional image *Theater* from the SUN360 dataset

undersampling. This grants the possibility of applying traditional transform coding more efficiently when compared to alternatives, such as cubic mapping. Nevertheless, this method has not been widely adopted because of its complexity.

2.2.4 Metrics to assess the compression performance

Metrics are required to compare the coding performances between different proposed coders. The most used one is the Peak Signal to Noise Ratio (PSNR) to measure the quality of a compressed image at a given bit-rate. However, in the omni-directional image case, it is computed in different domains to accurately reflect the visual quality of the spherical content.

PSNR in the equirectangular domain

The traditional PSNR is computed in the equirectangular (see example from the SUN360 dataset [1] in Fig. 2.7) domain where it measures the distortion of a retrieved signal compared its original version.

It is computed pixel-wise between the two original and decoded equi-rectangular images:

$$\mathrm{PSNR} = 10 \cdot \log 10 \left(\frac{\mathrm{d}^2}{\mathrm{MSE}} \right) \tag{2.3}$$

with d the maximum possible value for a pixel (e.g. 255 for a 8 bits image), and the Mean Squared Error (MSE) computed for two single channel images $\mathbf{I}$ and $\mathbf{J}$ of size $m \times n$ as:

$$\mathrm{MSE} = \frac{1}{\mathrm{m} \times \mathrm{n}} \sum_{\mathrm{i}=0}^{\mathrm{m}-1} \sum_{\mathrm{j}=0}^{\mathrm{n}-1} \left(\mathbf{I}(\mathrm{i},\mathrm{j}) - \mathbf{J}(\mathrm{i},\mathrm{j}) \right)^2 \tag{2.4}$$

The main drawback of this evaluation is that it does not take into account the spherical geometry and

[1] "Sun360 dataset." Available online at : `http://people.csail.mit.edu/jxiao/SUN360/main.html`

Figure 2.8: Cube-maps planar representation of the *Theater* omnidirectional image in the SUN360 dataset.

the redundancy around the poles, neither the frequency of accessing different viewports by the users.

PSNR in the spherical domain

In order to alleviate this issue, the authors in [112] propose a method to compare the original and coded omnidirectional content when the head motion data is not known beforehand. They propose a sphere based PSNR computation, denoted as S-PSNR, to approximate the average quality over all possible viewing directions.

If we perform a uniform sampling on the sphere, the error over this entire set of points on the sphere is averaged to compute S-PSNR of different coded representations with respect to the ground truth uniformly sampled sphere. The MSE is thus computed between two sets of points $\mathbf{p}$ and $\mathbf{q}$ made of N samples each:

$$\mathrm{MSE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=0}^{\mathrm{N}} \left(\mathbf{p}(\mathrm{i}) - \mathbf{q}(\mathrm{i})\right)^2 \tag{2.5}$$

Then, they propose to take into account the fact that not all viewports are equally likely, e.g., users are more likely to view areas around the equator than the poles. Using head motion data over a set of users, they estimate relative frequencies of accessing different points on the sphere. The resulting frequencies are then used to compute a weighted S-PSNR with a weighted summation of errors instead of the traditional MSE.

PSNR in the cubemaps domain

Another strategy is to compute the PSNR of the cubemap projections of the omni-directional content. In essence, cube mapping represents a method of environment mapping that uses the six faces of a cube. An example of such image is shown in Fig. 2.8.

The scene is projected onto the sides of a cube and stored as six square textures as we can see in Figure 2.8, or unfolded into six regions of a single texture. Once the six images are rendered, we can compute the PSNR between the resulting images after decoding and projection, and the original ground truth cubemap. This is another way to efficiently reflect the quality of viewpoints asked by a user. In our experiments, the PSNR computed is the mean PSNR over the six faces of the cube.

PART II

Contributions

Chapter 3

Local graph based transforms for light field compact representation

3.1 Introduction

In this chapter, we address the problem of designing graph based transforms for light fields compact representation. Light fields record illumination of light rays emitted by a scene in different orientations. The captured data for a static light field is represented by a 4D function $LF(u, v, x, y)$. It can be seen as a collection of images of the same scene taken from different points of view. It contains redundant information in both the spatial (x and y) and angular dimensions (u and v).

The existing spatio-angular correlations should ideally be represented by a *huge* non separable weighted graph connecting pixels within and across views of the entire light field. The basis functions of a graph Fourier transform [92] could then be used to decorrelate the color signal residing on the graph vertices. However, such graph would have a very high number of vertices, each vertex corresponding to a light ray. This makes the diagonalization of the Laplacian matrix unfeasible, hence, the computation of the graph Fourier transform intractable.

To lower the dimensionality of the problem, we can partition the global graph into smaller ones that are coherent and correlated inside and across the views. This can be viewed as cutting relatively unreliable edges from the *global* graph. We therefore group similar pixels within and across views based on the concept of super-rays defining the supports of the set of *local graph transforms*. The concept of super-ray has been introduced in [45] as an extension of super-pixel's concept to light fields .

Despite the local support of limited size defined by the super-rays, the local Laplacian matrix remains of high dimension and its diagonalization to retrieve the transform eigenvectors is computationally expensive. An intuitive way to solve this problem is to perform the transform in a separable manner:

- A first spatial transform applied per super-pixel inside each view to capture spatial correlations. Then,
- An angular transform between corresponding super-pixels across the views to capture angular dependencies.

We have however observed that if the shape of the super-ray undergoes a slight change between views, the basis functions computed from the graph Laplacian have very different forms from one super-pixel to the corresponding ones in the other views (refer to Figure 3.1), resulting in a decreased correlation between spatial transform coefficients.

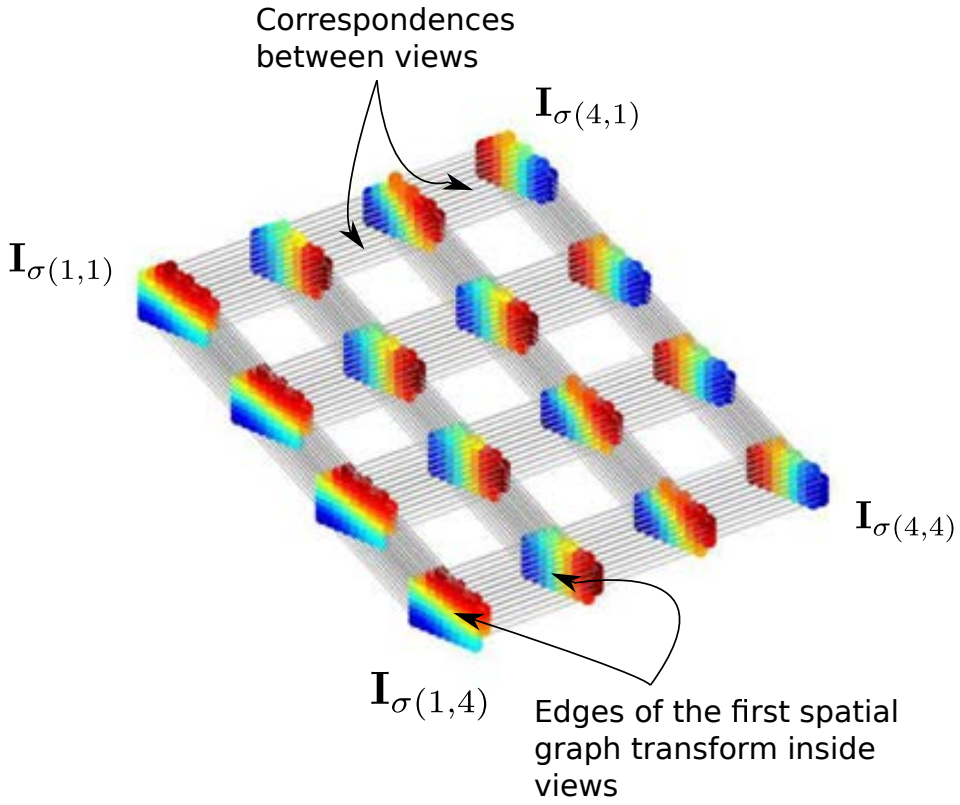


Figure 3.1: Second eigenvector of shape-varying super-pixels belonging to the same super-ray.

This chapter explores two ways of solving the two main issues concerning the graph transform conception for light field compression mentioned above: (a) the careful graph support design in a way that the signal to be transformed is smooth on the graph (low total variation $\mathbf{x}^\top \mathbf{L}\mathbf{x}$) and (b) finding the best trade-off between complexity and representation accuracy. The first solution consists of using geometry-blind graph supports which are fixed for all light field sub-aperture images coupled with a powerful prediction mechanism based on Convolutional Neural Networks (CNN). The second solution considers quasi-ideal geometry-aware super-rays which are cautiously built coupled with an optimized separable graph transform to preserve angular correlations.

In summary our main contributions are as follows:

- We first examine the case where the structure of the local graphs is derived from a coherent super-pixel over-segmentation of the different views to cope with the feasibility of separable graph transforms. In order to decrease the energy of the signals to be transformed, a powerful prediction mechanism based on view synthesis is used as a first step to exploit inter-view correlation. The local Graph Transform is computed and applied in a separable manner with a first spatial unweighted transform followed by an inter-view Graph Transform. For the latter, both unweighted and weighted versions have been considered. A dedicated predictive coding scheme is then described to assess the rate-distortion performance on a set of real light fields.

- We then consider the design of local Graph Transforms based on shape-varying super-rays that are adapted to scene geometry. To define the supports of the geometry-aware local transforms, we pro-

pose (section 3.3.1) a new algorithm to segment the light field into super-rays. The method takes as input only the top-left color image and a sparse set of disparities. The resulting segmentation defines the supports of local graph transforms. We then introduce (section 3.3.3) a novel method to optimize the spatial transforms in such a way that the basis functions are coherent across the views, given the scene geometry. We analyze the properties in terms of energy compaction of the proposed super-rays based graph transforms. A complete color coding scheme (section 3.3.4) is also described to assess the rate-distortion performance of these novel transforms on a set of real light fields.

3.1.1 General light field notation

Suppose we are dealing with a light field consisting of $N = U \times V$ views. In this chapter and the one that follows, we will alternate between the two different notations: $\mathbf{I}_{\sigma(u,v)}$ or $\mathbf{I}_v$ to represent a light field view in a way that we best serve the clarity of our presentation. If we need to show a 2D position, we refer to a image view at angular positions (u, v) in the light field as $\mathbf{I}_{\sigma(u,v)}$ where $\sigma(u, v)$ is a mapping from the $\mathbb{N} \times \mathbb{N}$ space to the range $[1, N]$ in $\mathbb{N}$. More precisely, for each u and v, $\sigma(u, v) = (u - 1)U + v$ where U is the number of views in the column wise angular dimension. Otherwise, if we read through the light field views with a raster scan, we just use $\mathbf{I}_v$ for the view v.

3.2 Separable graph transforms on fixed graph supports

3.2.1 Fixed graph supports: super-pixels

Figure 3.2: The original view $x_{\sigma(4,4)}$ of "*Cars*" dataset in [56] (left) and the corresponding super-pixel segmentation (right).

To exploit the local redundancies in images and video, various pixel grouping strategies are used in image/video compression, *e.g.*, fixed square patches, blocks with adaptive size. Compared with the traditional block based grouping, super-pixels aim at gathering similar pixels into more meaningful regions or objects which however requires the design of shape-adaptive decorrelating transforms.

We consider here the design of graph based transforms adapted to the local signal characteristics. In order to define these local transforms, super-pixels are computed on a reference view using the SLIC algorithm [1] which groups pixels having similar color values and that are close spatially, as shown in Fig.3.2. The segmentation in the *central* view is propagated to other views without changing the position and size of the segmentation mask to cope with the feasibility of the separable graph transform.

Blindly propagating the segmentation map to the whole set of views of the light field does undeniably violate our main hypothesis that the signals residing on the local graphs are smooth. This will for sure drop the energy compaction capacity of our graph based transforms. Yet, to overcome this issue, we can exploit effective prediction techniques. Local graph transforms are applied on the residual signals and the energy will be compacted in fewer coefficients prior to coding.

3.2.2 Graph signal: residuals after CNN based prediction

Machine learning methods have been recently considered for view synthesis. In [56], the authors only use the four corner sub-aperture views to synthesize the whole light field with high quality by two convolutional neural networks (CNN). One of the CNNs is trained to model the disparity in the given light field, while the other one is used to estimate the color of the synthesized views.

To obtain a pleasing prediction of the color signal and decrease the energy to be coded prior to transform, we use this architecture to predict the light field views from four corner views, as shown by the yellow parts in Fig.3.8.

Once a prediction $\tilde{\mathbf{x}}$ of the light field signal $\mathbf{x}$ is obtained, we can compute the residuals $\mathbf{r} = \mathbf{x} - \tilde{\mathbf{x}}$ that will be the graph signals during the following transform stage.

3.2.3 Separable graph transforms: spatio-angular

Thanks to the superpixel ability to adhere to image borders, the sub-aperture images are subdivided into uniform regions where the residual signal is supposed to be smooth. We can think of the local graph support as a super-ray that groups uniform super-pixels from different views that are supposed to be correlated. In order to capture the spatial and angular correlations within each super-ray and to avoid the complexity of the non-separable version, we use a separable Graph Transform comprising a local super-pixel based spatial GT followed by a local angular GT. The graphs used to compute the local separable transforms are depicted in Figure 3.3.

First spatial graph transform

If we consider only one residual view v of the light field and a segmentation map S, the k^{th} super-ray $SR_{k,v}$ can be represented by a signal $\mathbf{r}_{k,v} \in \mathbb{R}^{N_{k,v}}$ defined on an local spatial graph with only connections in the spatial domain (*i.e.* between the neighboring pixels in a super-pixel, and not across the views in a super-ray). As proposed in [92], a graph transform is defined based on the Laplacian matrix $\mathbf{L}_{k,v} = \mathbf{D}_{k,v} - \mathbf{A}_{k,v}$, where $\mathbf{D}_{k,v}$ is a diagonal degree matrix whose i^{th} diagonal element is equal to the sum of the weights of all edges incident to node i. The spatial graph transform basis is given by the eigenvectors $\mathbf{U}_{k,v}$ of the Laplacian $\mathbf{L}_{k,v}$. For the signal $\mathbf{r}_{k,v}$ settled on the vertices of the graph, the

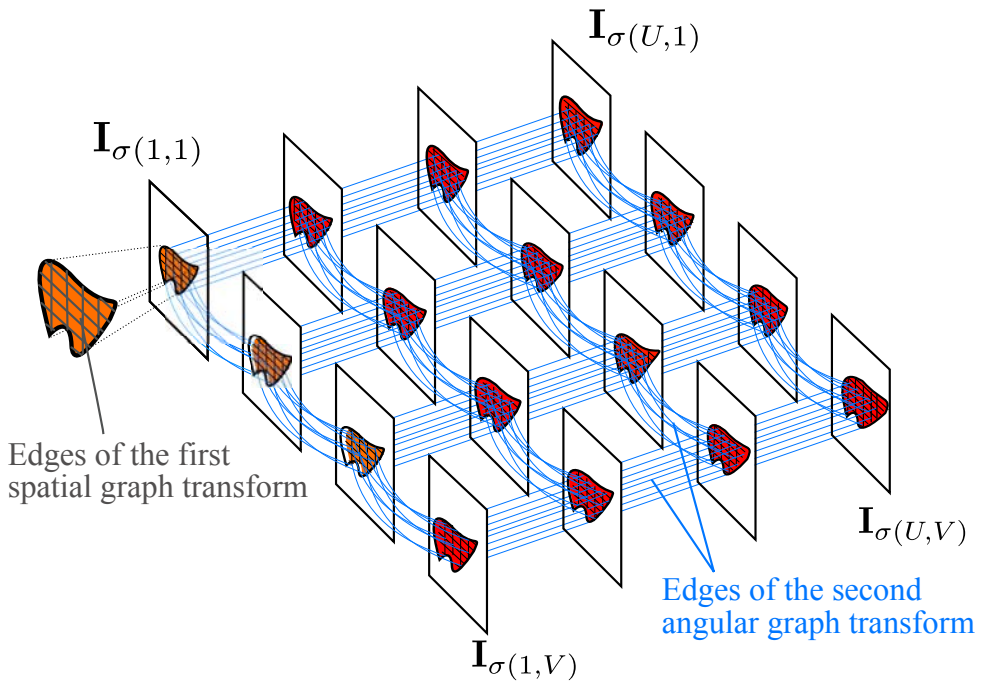


Figure 3.3: Illustration of the two graphs used to compute the two local separable graph transforms.

transformed coefficients vector $\hat{\mathbf{r}}_{k,v}$ are outlined as:

$$\hat{\mathbf{r}}_{k,v} = \mathbf{U}_{k,v}^{\top} \mathbf{r}_{k,v} \tag{3.1}$$

The inverse spatial graph Fourier transform is then given by

$$\mathbf{r}_{k,v} = \mathbf{U}_{k,v} \hat{\mathbf{r}}_{k,v} \tag{3.2}$$

Fig. 3.4 shows the luminance values of a cropped region of the residues for a subset of views of the Flower 1 dataset. Although the disparity is not taken into account, the signals in super-pixels which are co-located across the views are correlated for light fields with narrow baselines.

Second angular graph transform

The purpose of this transform is to tract the similarity between the transformed coefficients of each band b, $\hat{\mathbf{r}}_{k,v}(b)$ across the views which can be observed as in Figure 3.5. In a general case, for a given super-ray, we do not necessarily have the same number of pixels in all the views, hence the number of coefficients resulting from the spatial transforms is not identical in all the views. Therefore, for each band b, we build a different graph between the views where the band b exists. In the specific case of a fixed segmentation used here, for a given band b (coefficients corresponding to the b^{th} eigenvectors of the spatial transforms), we always construct a 2D grid of $M \times N$ vertices corresponding to all light field views to be coded.

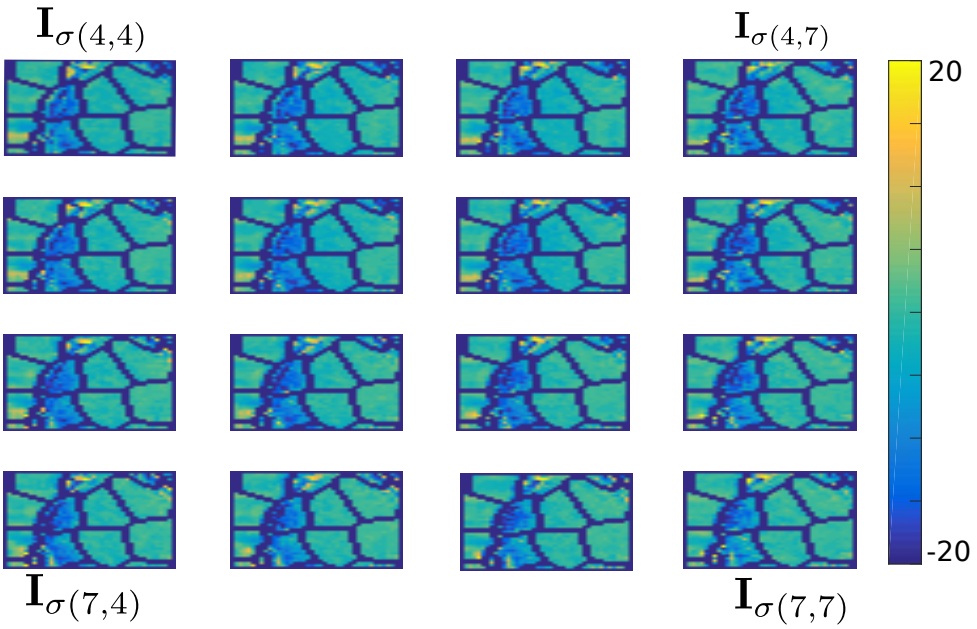


Figure 3.4: Illustration of coherent residual signals in superpixels for a subset of views of "*Flower 1*" (luminance).

Unweighted Angular Graph Transform For each super-ray k, and for each band b, the Adjacency $\mathbf{A}_k^b$ and degree $\mathbf{D}_k^b$ matrices are used to compute the inter-view Laplacian as $\mathbf{L}_k^b = \mathbf{D}_k^b - \mathbf{A}_k^b$. The eigenvectors $\mathbf{U}_k^b$ of $\mathbf{L}_k^b$ are then used to characterize the angular graph transform basis. Moreover, the spatial-band vector is defined as $\hat{\mathbf{r}}_k^b = [\hat{\mathbf{r}}_{k,v}(b)]_{v\in\{1,2,\ldots,V\},\ \text{s.t. } b<|\mathbf{r}_{k,v}|}$, where V is the number of views. The angular transform coefficients are obtained by calculating:

$$\hat{\hat{\mathbf{r}}}_k^b = {\mathbf{U}_k^b}^\top \hat{\mathbf{r}}_k^b. \tag{3.3}$$

The inverse angular Graph Transform is then given by

$$\hat{\mathbf{r}}_k^b = \mathbf{U}_k^b \hat{\hat{\mathbf{r}}}_k^b \tag{3.4}$$

A major assumption lying behind the use of the unweighted version of a Laplacian is a constant pair-wise relationship between neighboring nodes which may not accurately reflect the statistical precisions in our case especially for high frequencies where different patterns of correlations can be observed (refer to Figure 3.6).

Weighted Angular Graph Transform Instead of applying the same graph transform to all the bands, we divide them into 64 groups, ranging from low to high frequencies. For each group, we compute the sample covariance matrix from a set of training superpixels spatial coefficients. We show an example of covariance matrices obtained for the first 16 groups in Figure 3.6. It is evident that we can note different models of correlations when we move from low to high frequencies.

We solve the minimization problem defined in [30] to compute 64 different generalized Laplacian

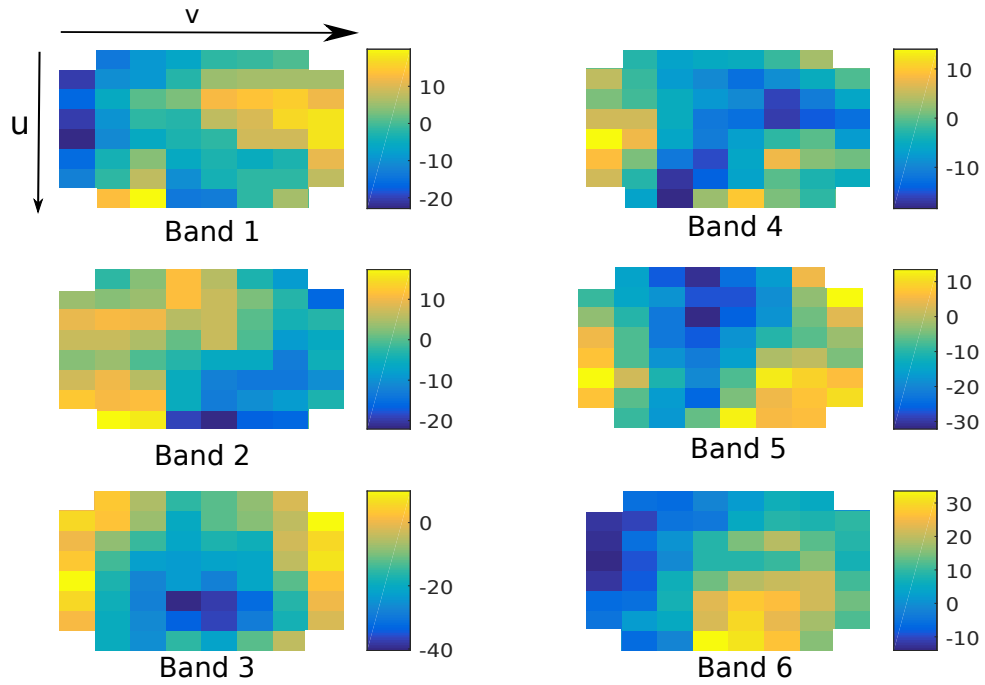


Figure 3.5: An example of the transformed coefficients of the 9 first bands b, $\hat{\mathbf{r}}_{k,v}(b)$ across the views for a super-ray.

matrices [1], that can be either computed separately for each dataset and sent as additional information or learned for a set of training datasets and stored in the decoder side. Due to the high computational cost of the first option, we will learn a fixed set of 64 Laplacian matrices to be exploited for all datasets. Let $\mathbf{U}^h$ be the matrix whose columns contain the weighted Graph Transform basis for a specific group h i.e., the eigenvectors of the corresponding weighted Laplacian. The band signals belonging to this group are thus projected onto this basis.

Energy compaction gain with angular transforms

We evaluate the energy compaction of the transformed coefficients for the three transforms (only spatial GT, spatial + unweighted angular GT, spatial + weighted angular GT) to show the utility of exploring inter-view correlation.

Energy compaction is measured by ordering all coefficients (for the luminance component) according to their decreasing variances. The total energy in the transform coefficients is the same as that in the Light Field residual signal, due to the orthogonality of the transforms. Fig. 3.7 shows the fraction of the total energy captured by α % of transform coefficients as a function of α for the residuals of a dataset used in our experiments, namely "*Flower 1*". Higher energy compaction is observed with the second angular transform compared with only applying the spatial transform, with a slight improvement for the wGT. This shows the utility of exploring the inter-view correlations between residues in different views and

[1] We adapt the code provided with the paper [30] to solve problem 1 as defined in their paper for our estimated covariance matrices.

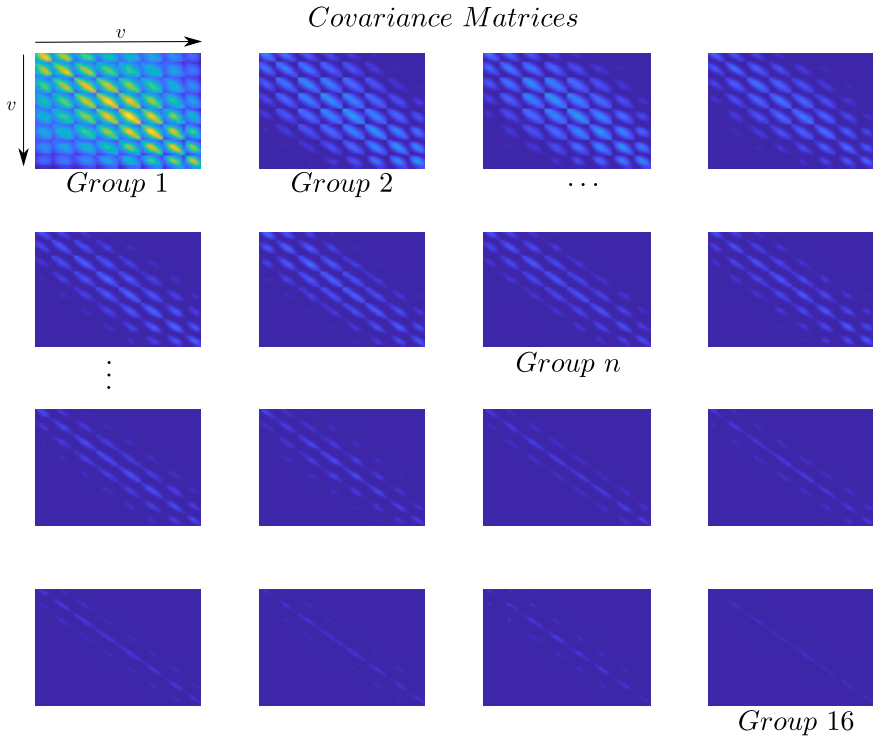


Figure 3.6: Covariance matrices of the transformed coefficients of 16 group of bands $\{\hat{\mathbf{r}}_k^b\}$ across the views. Going through line by line, from left to right, is equivalent to moving from low to high frequencies groups

adapting the graph weights for that purpose compared to only performing local spatial transforms.

After performing the segmentation and two transforms, the energy of the residual signal is indeed expected to be small and mostly concentrated in a small number of coefficients. In the following section, we aim at exploiting this energy compaction property to code the residual signals and improve the quality of the prediction at the decoder side.

3.2.4 Light field predictive coding scheme

Figure 3.8 depicts the proposed predictive coding scheme. Let $\mathbf{LF} = \{\mathbf{I}_{\sigma(u,v)}\}$ denote a light field, where $u = 1, \ldots, U$ and $v = 1, \ldots, V$ are the view indices.

Four views at the corners $\mathbf{LF}^{\text{cor}} = \{\mathbf{I}_{\sigma(1,1)}, \mathbf{I}_{\sigma(1,V)}, \mathbf{I}_{\sigma(U,1)}, \mathbf{I}_{\sigma(U,V)}\}$ are encoded using HEVC-Inter and used to synthesize the whole light field with the CNN based synthesis method [56], as shown in Fig.3.8 (red arrows). To improve the quality of the synthesized light field, the residuals between the synthesized and original views are encoded by graph transforms, (see Fig.3.8, blue arrows). The residuals of all the views but the 4 corner views $\mathbf{LF} \backslash \mathbf{LF}^{\text{cor}}$ are considered here. These residual signals are grouped into super-pixels using the SLIC algorithm [1] as explained in section 3.2.1, then graph transforms are

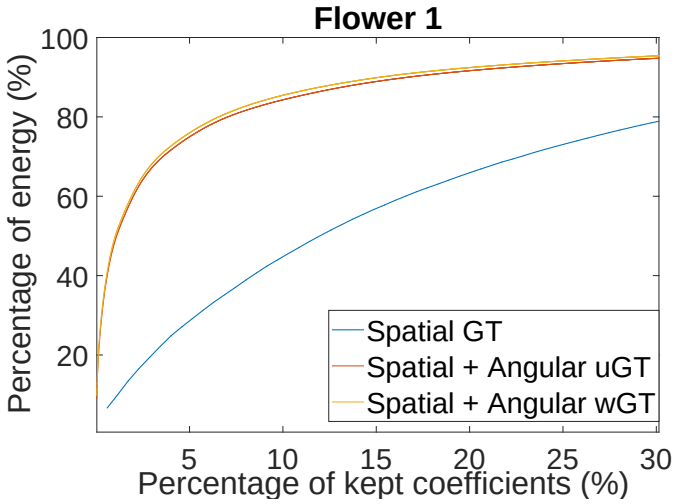


Figure 3.7: Energy Compaction of the transformed residues $\mathbf{r}$ for "*Flower 1*".

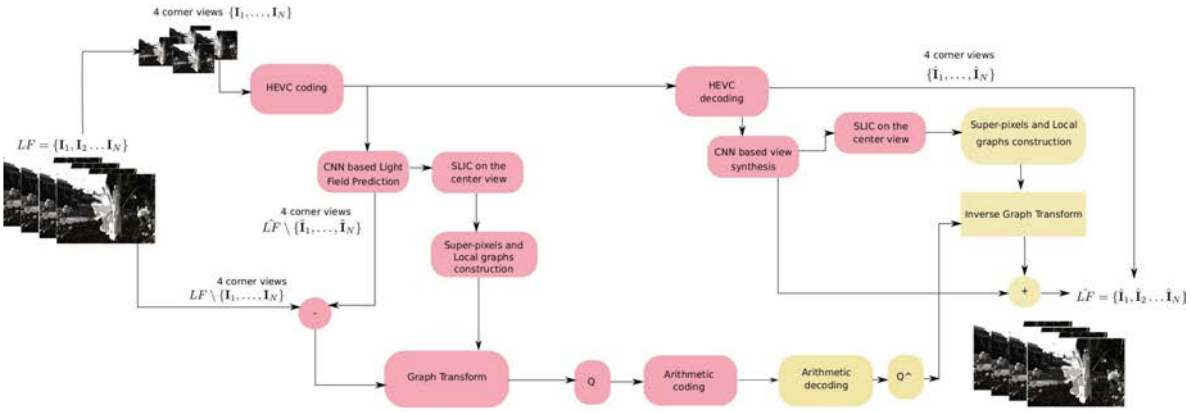


Figure 3.8: Overview of proposed light field predictive coding scheme.

applied on each super-pixel as in section 3.2.3. At the end of the transform stage, coefficients are grouped into a three-dimensional array $\mathbf{R}$ where $\mathbf{R}(i_{SR}, i_{bd}, v)$ is the v^{th} transformed coefficient of the band i_{bd} for the super-ray i_{SR}. Using the observations on all the super-rays in some training datasets, we can find the best ordering for quantization. We first sort the variances of coefficients with enough observations in decreasing order. We then split them into 64 classes assigning to each class a quantization index in the range 1 to 64. All the remaining coefficients with less observations will be considered in the last group. We use the zigzag ordering of the *JPEG* quantization matrix to assign the quantization step size for each. The quantized coefficients are further coded using an arithmetic coder.

Note that the super-pixels are computed on the synthesized view, since those are available at both the encoder and decoder. We can thus, at the decoder side, recover the super-pixel segmentation from the four corner views $\hat{\mathbf{LF}}^{\text{cor}}$, then construct the spatial and angular unweighted graphs. Also, for the weighted angular graph transform, the Laplacians are learned on a training set of light fields and then fixed. After applying the inverse separable graph transform, the decoded residuals are added to the

synthesized light field to obtain the final decompressed light field.

3.3 Geometry-aware graph transforms for color coding of light fields

The compression efficiency of any coder based on block partitioning and transform coding does undeniably depend on the way the partitioning is done, and on how the resulting segmentation adheres to object boundaries. In the previously presented coder, fixing the graph supports for the transforms may result in a decrease in the energy compaction due to high frequencies captured on the object boundaries. Here, we instead rely on a segmentation of the entire 4D light field into geometry-aware super-rays.

3.3.1 Geometry-aware graph supports: super-rays

The concept of super-ray has been introduced in [46] as an extension of super-pixels [1] to group light rays coming from the same 3D object, *i.e.* to group pixels having similar color values and being close spatially in the 3D space. The method performs a k-means clustering of all light rays based on color and distance in the 3D space. To deal with dis-occlusions, a slightly modified formulation is proposed in [98] where the dense depth information is also used in the clustering. When the depth information is not fully reliable, this method results in inconsistent super-rays across views, *i.e.* shape-varying super-rays. In addition, the signaling cost of such a global light field segmentation is high.

In order to make the super-rays more consistent across the views, we propose a modified version where we compute super-pixels in the top-left view. Then, using the disparity map, we project the segmentation labels to all the other views. Namely, having a segmentation map in the top left view and the corresponding disparity map, we compute the median disparity per super-pixel, and use it to project the segmentation mask to the other views. More precisely, the algorithm proceeds row by row. In the first row of views, we perform horizontal projections from the top-left $\mathbf{x}_{\sigma(1,1)}$ to the $V-1$ views next to it. For each other row of views, a vertical projection is first carried out from the top view $\mathbf{x}_{\sigma(1,1)}$ to recover the segmentation on view $\mathbf{x}_{\sigma(u,1)}$, then $V-1$ horizontal projections from $\mathbf{x}_{\sigma(u,1)}$ to the $V-1$ other views are performed, as shown in Figure 3.9.

An example of segmentation S is shown in Figure 3.9, where the background consists of two yellow superpixels, and two foreground objects are labeled with red and pink. The disparity of the two objects is equal to 1, while the background is almost fixed with a disparity equal to 0. At the end of each projection, some shapes are projected in all the views without interfering with others. Those typically represent flat regions inside objects (for example, the object labeled in pink). While others, mainly consisting of occluded and occluding segments end up superposed in some views, for example, the red object occluding pixels from the yellow background. In this case, the occluded pixels are assigned the label (e.g. red) of the neighboring super-ray corresponding to the foreground objects (*i.e.* having the higher disparity). As for appearing pixels, for example, between the yellow background and pink object, they will be clustered with the background super-rays (*i.e.* having the lower disparity e.g. yellow). The super-rays that end up with different shapes in the views are marked with a dashed contour.

We assess how the proposed super-ray construction method deals with occluded and dis-occluded parts, and to which extent the super-rays are consistent despite uncertainty on the disparity informa-

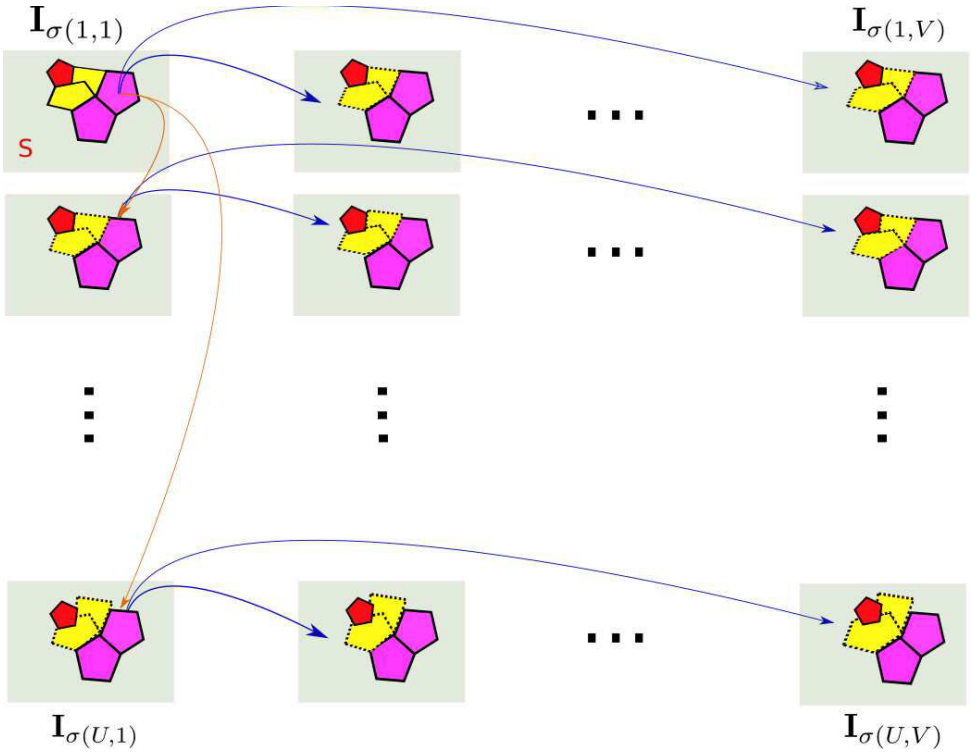


Figure 3.9: Image showing the super-ray construction. The algorithm proceeds row by row. In the first row, only horizontal projections (blue arrows) are performed. In every other row, first a vertical projection (red arrow) than $V-1$ horizontal projections (blue arrows) are performed.

tion. Figure 3.10 shows examples of super-rays obtained with different sets of synthetic and real light fields captured by a Lytro Ilum camera ("*MonasRoom*", "*Butterfly*" from the HCI old Light Field Dataset, "*Flower 2*", "*Rock*" used in [56], and "*FountainVincent*", "*StonePillarInside*" used in [106]). In the first three columns, we have the original top left corner view, its corresponding disparity map and super pixel segmentation using the SLIC algorithm [1] respectively. In the fourth column, we show horizontal and vertical epipolar segments taken both from the 4D light field color information and our final segmentation in specific regions of the image (the red blocks). We can see that we are following well the object borders, especially when the disparity map is reliable. Also, we have always attained a high percentage of coherent super-rays across views (higher than 40% as measured with Cons(%) in the fifth column). More precisely, Cons(%) gives the percentage of coherent super-rays: a super-ray is coherent when it is made of super pixels having the same shape in all the views, with or without a displacement.

At the end of this segmentation stage, we end up with a segmentation map with consistent super rays in flat objects and shape-varying super-rays mainly on the borders.

3.3.2 Graphs and graph signals

In order to jointly capture spatial and angular correlations between pixels in the light field, we first consider a local non separable graph per super-ray. While in section 3.2.3, we were dealing with each

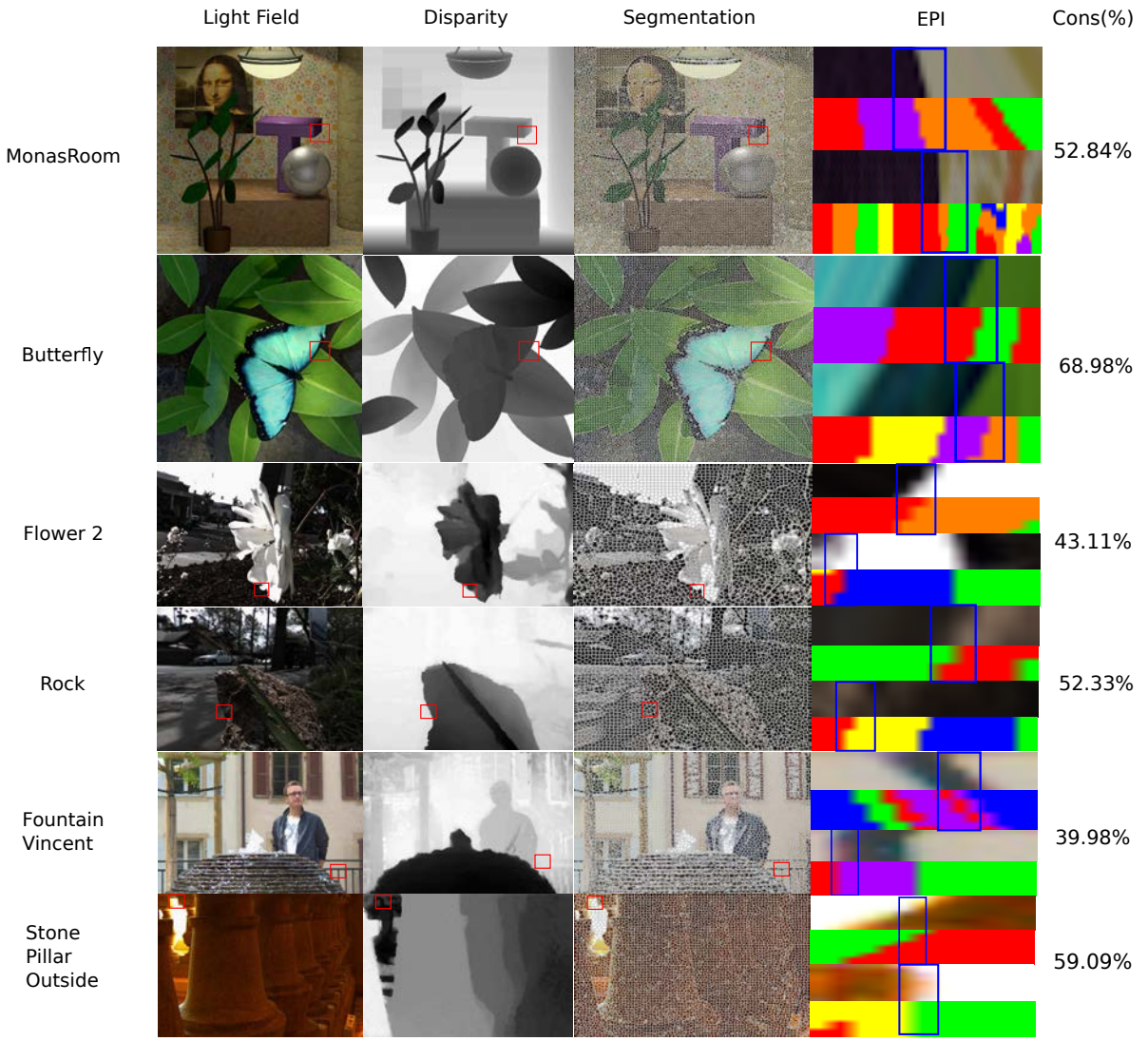


Figure 3.10: Consistent Super-rays performance:In the first three columns, we have the original top left corner view, its corresponding disparity map and super pixel segmentation using the SLIC algorithm [1] respectively. In the fourth column, we show horizontal and vertical epipolar segments taken both from the 4D light field color and our final labeling in specific regions of the image(the red blocks). We use the prism color map in Matlab for the segmentation, just for illustration purposes.

super-ray inside a view at a time, here, we take the whole super-ray in all the light field views. More precisely, if we consider the luminance values in the whole light field and a segmentation map S, the k^{th} super-ray SR_k can be represented by a signal $\mathbf{x}_k \in \mathbb{R}^{N_k}$ defined on an undirected connected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ which consists of a finite set $\mathcal{V}$ of vertices corresponding to the pixels at positions $\{u_l, v_l, x_l, y_l\}, l = 1 \dots N$ such that $S(u_l, v_l, x_l, y_l) = k$. A set $\mathcal{E}$ of edges connect each pixel and its 4-nearest neighbors in the spatial domain (i.e. in each view), and to its corresponding pixels, found by disparity based projection, in the 4 nearest neighboring views. The disparity used for projection is fixed for all pixels in super-ray and is actually equal to the median disparity in the corresponding super-pixel of the top-left view. An example of graph built inside a super-ray is shown in Figure 3.11.

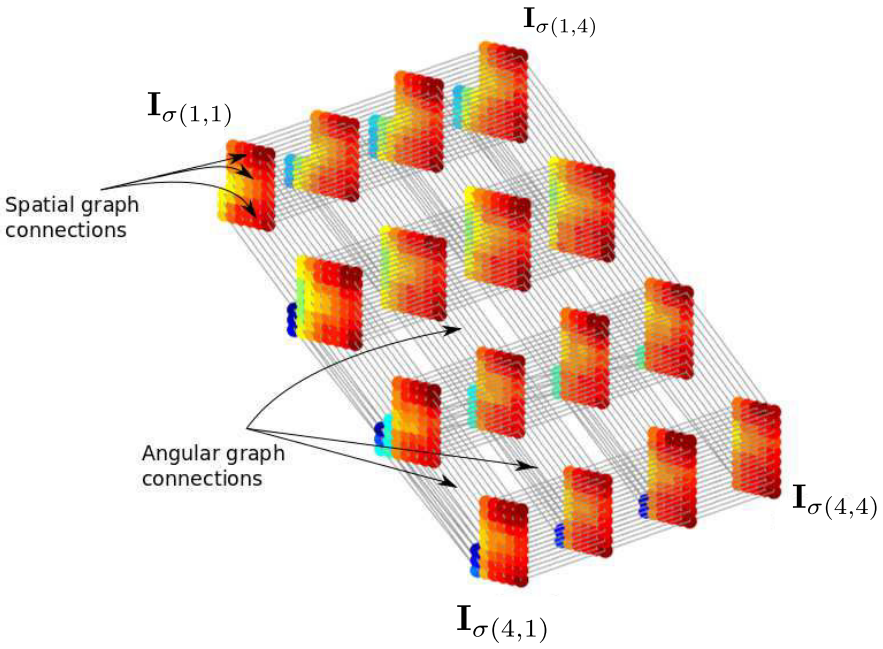


Figure 3.11: Example of local non-separable graph within a super-ray. We can see the connections within super-pixels in each view, as well as connections between pixels belonging to different views. The color assigned to the vertices is the luminance value of the pixels. For visualization purposes, we show the luminance in false colors.

3.3.3 Geometry-aware graph transforms

In this section, we focus on the design of suitable transforms for the light field luminance values residing on the local graphs defined above.

Non separable graph transform

The optimal local decorrelating transform within a super-ray is unquestionably the one who follows both spatial and angular correlation structures of the light field, precisely the one that relies on the Non Separable graph built inside each super-ray.

Let us consider the k^{th} super-ray SR_k and its corresponding local graph $\mathcal{G}_k$. We start by defining its adjacency matrix $\mathbf{A}_k$ with entries $\mathbf{A}_k(m,n) = 1$, if there is an edge $e = (m,n)$ between two vertices m and n, and $\mathbf{A}_k(m,n) = 0$ otherwise. The adjacency matrix is used to compute the Laplacian matrix $\mathbf{L}_k = \mathbf{D}_k - \mathbf{A}_k$, where $\mathbf{D}_k$ is a diagonal degree matrix whose i^{th} diagonal element $\mathbf{D}_k(i,i)$ is equal to the sum of the weights of all edges incident to node i. The resulting Laplacian matrix $\mathbf{L}_k$ is symmetric positive semi-definitive and therefore can be diagonalized as:

$$\mathbf{L}_k = \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^\top \tag{3.5}$$

where $\mathbf{U}_k$ is the matrix whose columns are the eigenvectors of the graph Laplacian and $\mathbf{\Lambda}_k$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues. The Laplacian eigenbases $\mathbf{U}_k$ are analogous to the Fourier bases in the Euclidean domain and allow representing the signals residing on the graph as a linear combination of eigenfunctions akin to Fourier Analysis. This is known as the Graph Fourier transform. For the signal $\mathbf{x}_k$ defined on the vertices of the local graph, the transformed coefficients vector $\hat{\mathbf{x}}_k$ is defined in [92] as:

$$\hat{\mathbf{x}}_k = \mathbf{U}_k^\top \mathbf{x}_k \tag{3.6}$$

The inverse graph Fourier transform is then given by

$$\mathbf{x}_k = \mathbf{U}_k \hat{\mathbf{x}}_k \tag{3.7}$$

Although this would be the ideal local decorrelating transform for the signal, the Laplacian of such graph, despite the locality, remains of high dimension (almost 6000 nodes per super-ray) leading to a high transform computational cost. To limit the computational cost, we then consider separable local transforms.

Coherent separable graph transform

The separable graph transform is defined by a first spatial transform followed by a second angular transform as detailed in section 3.2.3.

The spatial graphs in the different super-pixels forming one super-ray may not have the same shape particularly on the object boundaries. Furthermore, we have observed that for a specific super-ray, when the spatial graph topology in the corresponding super-pixels undergoes a slight change, the basis functions of each spatial graph transform are different and thus incompatible with each others (refer to Figure 3.1), resulting in decreased correlation of the spatial transform coefficients across views. This is shown in the sequel to severely decrease the efficiency of the angular transform.

Basically, during the diagonalization procedure, the eigenfunctions are only defined up to sign flips for Laplacians having a simple spectrum (if the eigenvalues have a multiplicity of 1, for example connected graphs). Therefore, even having the same shape in two different views, we may end up with two opposite eigen-vectors for a specific eigenvalue during the diagonalization.

Moreover, eigenvectors computed independently on two different shapes (i.e. corresponding to two different Laplacians) can be expected to be reasonably consistent only when the shapes are approxi-

mately isometric. Whenever this assumption is violated, it is impossible to expect that the l^{th} eigenvector of a Laplacian $\mathbf{L}_{k,i}$ in view i will correspond to the l^{th} eigenvector of another Laplacian $\mathbf{L}_{k,j}$ in view j. If the basis functions do not behave consistently on the corresponding points of the two shapes, the two signals defined on those two Laplacians will be projected onto incompatible basis functions (see Figure 3.12), and therefore we cannot guarantee any correlation to be preserved after performing the first spatial graph transform.

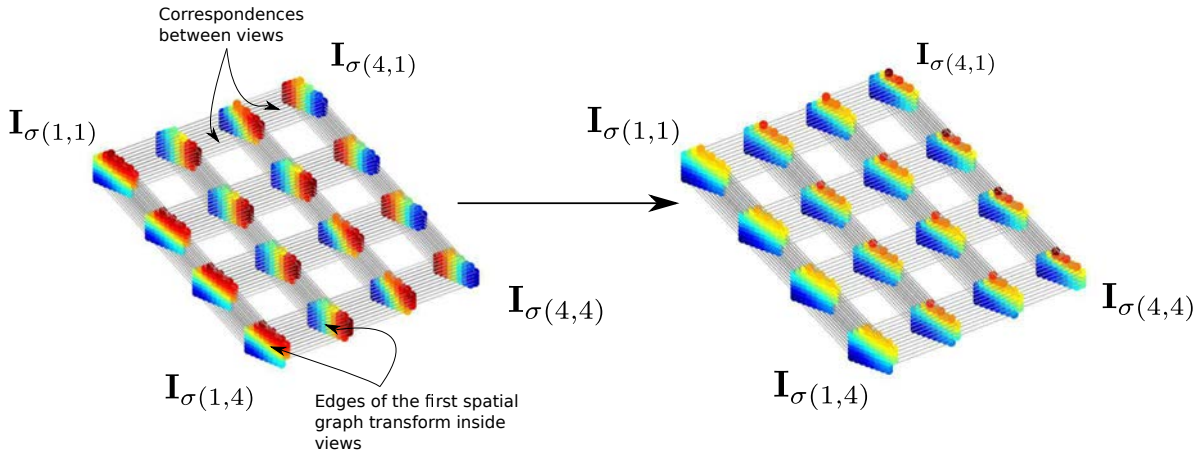


Figure 3.12: Second eigenvector of shape-varying super-pixels belonging to the same super-ray.

Coherent spatial graph transform

In order to overcome those limitations, we consider an approach which aims at finding *coupled* basis functions. More precisely, suppose that, in a super-ray k in a reference view o and a target view i, we have two Laplacians $\mathbf{L}_{k,o}$ and $\mathbf{L}_{k,i}$ with size $(n_o \times n_o)$ and $(n_i \times n_i)$ respectively. They can be diagonalized as:

$$\begin{aligned} \mathbf{L}_{k,o} &= \mathbf{U}_{k,o}\mathbf{\Lambda}_{k,o}\mathbf{U}_{k,o}^{\top} \\ \mathbf{L}_{k,i} &= \mathbf{U}_{k,i}\mathbf{\Lambda}_{k,i}\mathbf{U}_{k,i}^{\top} \end{aligned} \tag{3.8}$$

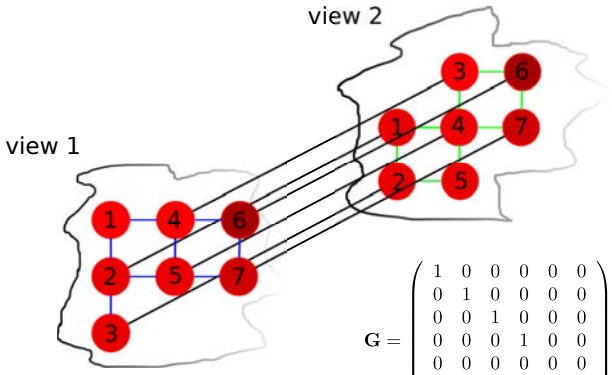
If the two Laplacians are equal, we make sure that their eigenvectors are compatible with sign flips accordingly. We check the first value of the each eigenvector and flip its sign if the value is negative.
In the case where the super-pixel shapes in the sub-aperture images are not isometric, we propose to diagonalize one specific spatial graph Laplacian $\mathbf{L}_{k,o}$ and find $\mathbf{U}_{k,o}$. Then, we search for basis vectors $\hat{\mathbf{U}}_{k,i}$ that approximately diagonalize any other spatial graph Laplacian $\mathbf{L}_{k,i}$ and at the same time preserve correlations after the transform. Inspired by the work of [59], we pose the problem as

$$\begin{aligned} \hat{\mathbf{U}}_{k,i}^{*} = &\min_{\hat{\mathbf{U}}_{k,i}} \; off(\hat{\mathbf{U}}_{k,i}^{\top}\mathbf{L}_{k,i}\hat{\mathbf{U}}_{k,i}) + \alpha \left\| (\mathbf{F}^{\top}\mathbf{U}_{k,o} - \mathbf{G}^{\top}\hat{\mathbf{U}}_{k,i}) \right\|_F^2, \\ &\text{s.t. } \hat{\mathbf{U}}_{k,i}^{\top}\hat{\mathbf{U}}_{k,i} = \mathbf{I}. \end{aligned} \tag{3.9}$$

where we seek to minimize the weighted sum of two terms subject to the orthonormality constraint of the computed basis functions $\hat{\mathbf{U}}_{k,i}$. The first term is a diagonalization term that aims at minimizing

the energy residing on off-diagonal entries ($off(\mathbf{M}) = \sum_{i \neq j} m_{ij}$). The second term aims at enforcing coherence between the two spatial graph transforms and is defined as follows.

Based on the geometry information we have in hand, we can actually define, *a priori*, a set of correspondences between $\mathbf{L}_{k,o}$ and $\mathbf{L}_{k,i}$. More precisely, we suppose that we have a set of p corresponding functions represented by matrices $\mathbf{F}$ and $\mathbf{G}$ of sizes $(n_0 \times p)$ and $(n_i \times p)$ respectively. An example of $\mathbf{F}$ and $\mathbf{G}$ is shown in figure 3.13. Each column of $\mathbf{F}$ and $\mathbf{G}$ can be seen as impulse functions centered on specific vertices of the graphs in both views.



$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 3.13: Example of correspondence functions $\mathbf{F}$ and $\mathbf{G}$ computed for a small shape-varying superpixel. The graph nodes are labeled in both graphs following a vertical scan line. In the second view, we have one disappearing node and another appearing one with respect to the first view.

The basis functions of both Laplacians are supposed to be consistent if the Fourier coefficients of the functions $\mathbf{F}$ and $\mathbf{G}$ on $\mathbf{L}_{k,o}$ and $\mathbf{L}_{k,i}$ are approximately equal i.e. if $\mathbf{F}^\top \mathbf{U}_{k,o} \simeq \mathbf{G}^\top \hat{\mathbf{U}}_{k,i}$. To avoid overdetermining the problem, we use the farthest point sampling technique restricting the correspondence points to a maximum of 15 points.

If we parametrize the new basis functions of $\mathbf{L}_{k,i}$ as being a linear combination of the old basis functions, we can write $\hat{\mathbf{U}}_{k,i} = \mathbf{U}_{k,i}\mathbf{B}_{k,i}$ where $\mathbf{B}_{k,i}$ is a matrix of combination coefficients, that plays a role of reflecting and rotating the original basis vectors in $\mathbf{U}_{k,i}$ so that they will align the best way with $\mathbf{U}_{k,o}$ while almost diagonalizing the Laplacian $\mathbf{L}_{k,i}$. Using the diagonalizing property of $\mathbf{U}_{k,i}$, we can re-write Equation (3.9) as

$$\begin{aligned} \mathbf{B}^*_{k,i} = & \min_{\mathbf{B}_{k,i}} \; off(\mathbf{B}_{k,i}^\top \mathbf{\Lambda}_{k,i} \mathbf{B}_{k,i}) + \alpha \left\| (\mathbf{F}^\top \mathbf{U}_{k,o} - \mathbf{G}^\top \mathbf{U}_{k,i} \mathbf{B}_{k,i}) \right\|_F^2, \\ & \text{s.t. } \mathbf{B}_{k,i}^\top \mathbf{B}_{k,i} = \mathbf{I}, \end{aligned} \tag{3.10}$$

It is important to note that the first term of the above problem does not guarantee a preserved increasing order of the eigenfunctions. It is therefore more convenient to use an alternative penalty equal to

$\left\|\mathbf{B}_{k,i}^{\top}\mathbf{\Lambda}_{k,i}\mathbf{B}_{k,i}-\mathbf{\Lambda}_{k,i}\right\|_F^2$ that relates not only to the diagonalization property, but also to the distribution of the energies across the basis functions after the optimization.

$$\begin{aligned}\mathbf{B}_{k,i}^{*} = &\min_{\mathbf{B}_{k,i}} \left\|\mathbf{B}_{k,i}^{\top}\mathbf{\Lambda}_{k,i}\mathbf{B}_{k,i}-\mathbf{\Lambda}_{k,i}\right\|_F^2+\alpha\left\|(\mathbf{F}^{\top}\mathbf{U}_{k,o}-\mathbf{G}^{\top}\mathbf{U}_{k,i}\mathbf{B}_{k,i})\right\|_F^2,\\ &\text{s.t. } \mathbf{B}_{k,i}^{\top}\mathbf{B}_{k,i}=\mathbf{I},\end{aligned} \tag{3.11}$$

The problem in Equation (3.11) is a non linear optimization problem with an orthogonality constraint, which can be solved by iterative minimization algorithms. In our case, we used Matlab optimization toolbox (interior point method of the *fmincon* function) to solve it.

The gradients of the two terms of the cost function in the optimization of equation 3.11 are provided below:

$$\begin{aligned}&\nabla_B\|\mathbf{B}^{\top}\mathbf{\Lambda}_i\mathbf{B}-\mathbf{\Lambda}_i\|_F^2\\ &=\nabla_B tr\left((\mathbf{B}^{\top}\mathbf{\Lambda}_i\mathbf{B}-\mathbf{\Lambda}_i)^{\top}(\mathbf{B}^{\top}\mathbf{\Lambda}_i\mathbf{B}-\mathbf{\Lambda}_i)\right)\\ &=\nabla_B tr\left((\mathbf{B}^{\top}\mathbf{\Lambda}_i\mathbf{B}-\mathbf{\Lambda}_i^{\top})(\mathbf{B}^{\top}\mathbf{\Lambda}_i\mathbf{B}-\mathbf{\Lambda}_i)\right)\\ &=\nabla_B tr(\mathbf{B}^{\top}\mathbf{\Lambda}_i\mathbf{B}\mathbf{B}^{\top}\mathbf{\Lambda}_i\mathbf{B}-\mathbf{B}^{\top}\mathbf{\Lambda}_i\mathbf{B}\mathbf{\Lambda}_i\\ &\qquad\qquad -\mathbf{\Lambda}_i^{\top}\mathbf{B}^{\top}\mathbf{\Lambda}_i\mathbf{B}+\mathbf{\Lambda}_i^{\top}\mathbf{\Lambda}_i)\\ &=4(\mathbf{\Lambda}_i\mathbf{B}\mathbf{B}^{\top}\mathbf{\Lambda}_i\mathbf{B}-\mathbf{\Lambda}_i\mathbf{B}\mathbf{\Lambda}_i)\end{aligned} \tag{3.12}$$

As for the coupling term, with a similar derivation as the first gradient and using the trace derivation properties in [79], we get:

$$\begin{aligned}&\nabla_B(\left\|(\mathbf{F}^{\top}\mathbf{U}_{s_0}-\mathbf{G}^{\top}\mathbf{U}_{s_i}\mathbf{B})\right\|_F^2)\\ &=2\mathbf{U}_{s_i}^{\top}\mathbf{G}(\mathbf{G}^{\top}\mathbf{U}_{s_i}\mathbf{B}-\mathbf{F}\mathbf{U}_{s_0})\end{aligned} \tag{3.13}$$

Since we are dealing with large datasets and a large number of super-rays, it is convenient to use parallel computing to independently compute eigen-basis for the different super-rays. Also, in order to reduce the complexity of the problem, we propose to split it into smaller problems that are independent: we pick a small number h of eigenvectors to be optimized at a time. Then, for each disjoint group l of h eigenvectors in $\mathbf{U}_{k,i}$, we formulate a sub-problem by expressing h new eigenvectors as a linear combination of h old eigenvectors. Noticing that $\mathbf{U}_{k,i}=[\widetilde{\mathbf{U}}_{k,i_1},\widetilde{\mathbf{U}}_{k,i_2},...,\widetilde{\mathbf{U}}_{k,i_l}]$ and

$$\mathbf{\Lambda}_i=\begin{pmatrix}\widetilde{\mathbf{\Lambda}}_{k,i}^1 & 0 & 0 & 0\\ 0 & \widetilde{\mathbf{\Lambda}}_{k,i}^2 & 0 & 0\\ 0 & 0 & .. & 0\\ 0 & 0 & 0 & \widetilde{\mathbf{\Lambda}}_{k,i}^l\end{pmatrix} \tag{3.14}$$

For each group of h eigenvectors, we find $\widetilde{\mathbf{B}}_{k,i}^l$ of size $(h\times h)$ that will minimize the objective function

on the subset of eigenvectors.

$$\begin{aligned}\widetilde{\mathbf{B}}_{k,i}^{l^*} = &\min_{\widetilde{\mathbf{B}}_{k,i}^{l}} \left\| \widetilde{\mathbf{B}}_{k,i}^{l^\top} \widetilde{\mathbf{\Lambda}}_{k,y}^{l} \widetilde{\mathbf{B}}_{k,i}^{l} - \widetilde{\mathbf{\Lambda}}_{k,y}^{l} \right\|_F^2 + \alpha \left\| (\mathbf{F}^\top \widetilde{\mathbf{U}}_{k,o_l} - \mathbf{G}^\top \widetilde{\mathbf{U}}_{k,i_l} \widetilde{\mathbf{B}}_{k,i}^{l}) \right\|_F^2, \\ &\text{s.t. } \widetilde{\mathbf{B}}_{k,i}^{l^\top} \widetilde{\mathbf{B}}_{k,i}^{l} = \mathbf{I},\end{aligned} \tag{3.15}$$

We examine the performance of our optimization process described above and its effect on the transform coding efficiency. In all the experiments, for each super-ray k we find the super-pixel $\mathbf{L}_{k,o}$ that is on the top-left most of the light field, and fix it as reference for the coupling process. We optimize the maximum number of eigenvectors defined as $floor(\frac{n_{k,0}}{10}) \times 10$ with $n_{k,0}$ being the number of pixels in the reference super-pixel. An example of input and output of the coupling process for a shape-varying super-ray is illustrated in Figure 3.14. We see that the consistency of eigenvectors in the different graphs is much better after our optimization. If we project the light field signal residing in the super-ray on the optimized coupled eigenvectors, the inter-view correlation is better preserved compared to the non optimized eigenvectors.

At the end of the optimization stage, most of the eigenvectors are thereby compatible across views and the transform will necessarily preserve any correlation already observed between views. An example of the second eigenvector of a super-ray before and after optimization is shown in Figure 3.1. While eigenvectors corresponding to higher frequencies are harder to adjust, the low frequency eigenvectors can be easily optimized. In our application, this is not a big problem since we have a high energy compaction in lower frequency bands, and those are the bands that matter the most for reconstruction.

Energy compaction of the spatial transform

Figure 3.15 shows the energy compaction observed in the spatial transform domain, then in the spatio-angular transform domain, *i.e.* after performing the first spatial transform and after performing both spatial and angular transforms on the color signal of the light fields. The energy compaction is computed for both optimized and non optimized cases. It denotes the percentage of energy if we keep some of the coefficients and discard others. For the spatial transform, we gather the transform coefficients of all super-pixels, and then we scan them following the intuitive order increasing order of the Laplacian eingevalues to compute the compaction. For the spatio-angular compaction, we follow the learned sub-optimal scanning order using different observations from the different datasets as explained in section 3.3.4.

If we compare the energy compaction of the spatial transforms only (red and blue curves) for different datasets, we observe that we may loose in terms of energy compaction for some datasets after optimization.

In order to explain such loss, we analyze how the graphs are varying under the new basis functions after optimization. An example is shown in Figure 3.16 where edges between highlighted nodes are added implicitly in the graph after coupling. The new underlying Laplacian is computed as $\hat{\mathbf{L}}_{k,i} = \hat{\mathbf{U}}_{k,i} \mathbf{\Lambda}_{k,i} \hat{\mathbf{U}}_{k,i}^T$.

The underlying assumption behind the optimization procedure is that the signal can be modeled by a modified Gaussian distribution (Gaussian Markov Random Field) with a modified precision matrix

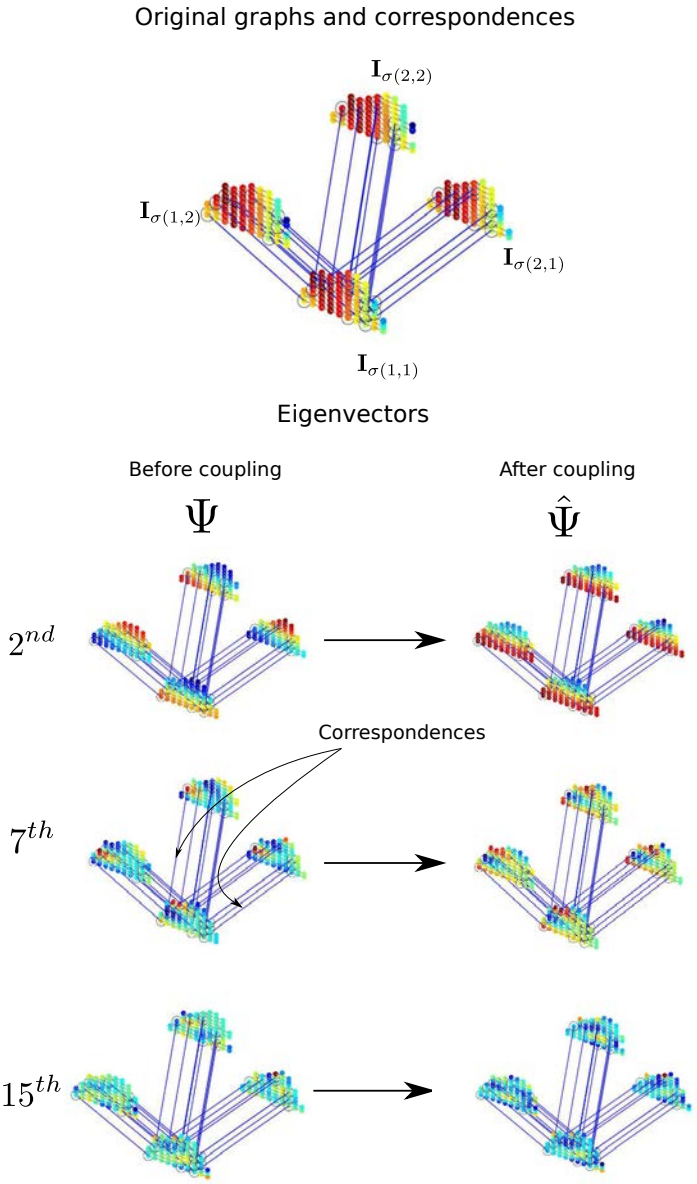


Figure 3.14: Illustration of the output of the optimization process for a super-ray in 4 views. The first row corresponds to a super-ray accross four views of the light field. The signal on the vertices correspond to the color values lying on super-pixels corresponding to the same super-ray and the blue lines denote the correspondences. The second to fourth rows are illustrations of basis functions before and after optimization. The signals on the vertices are the eigenvectors values.

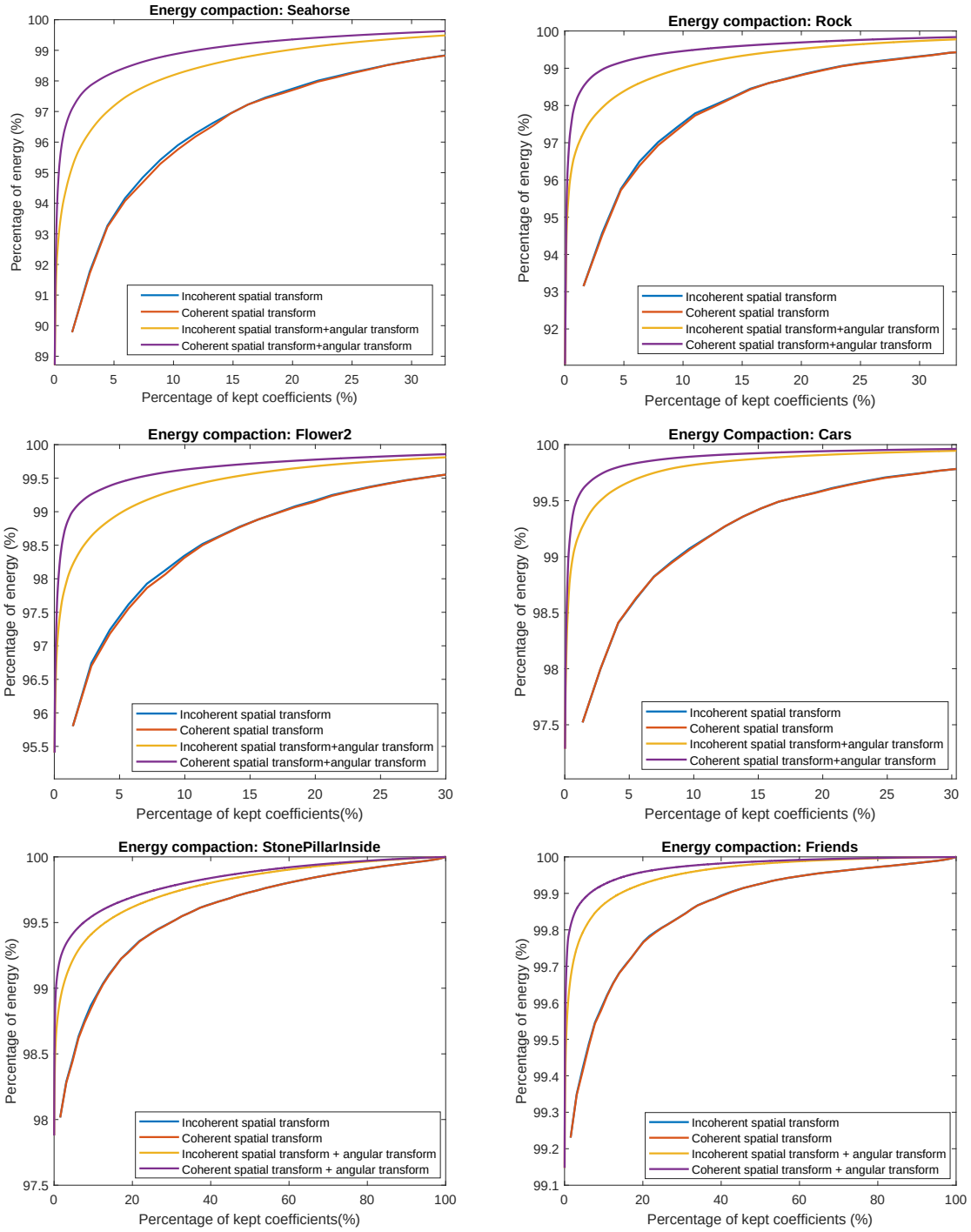


Figure 3.15: Energy compaction with or without optimization of the first spatial transform for four datasets ("*Seahorse*", "*Rock*" , "*Flower2*" and "*Cars*") from the dataset used in [56] and two others ("*Friends*" and "*Stone Pillars Inside*") taken from the datasets in [106].

which is equivalent to the new Laplacian matrix with some added small weights. Since this procedure is modifying the original graph structure, it may, in some cases, bring some high frequencies.

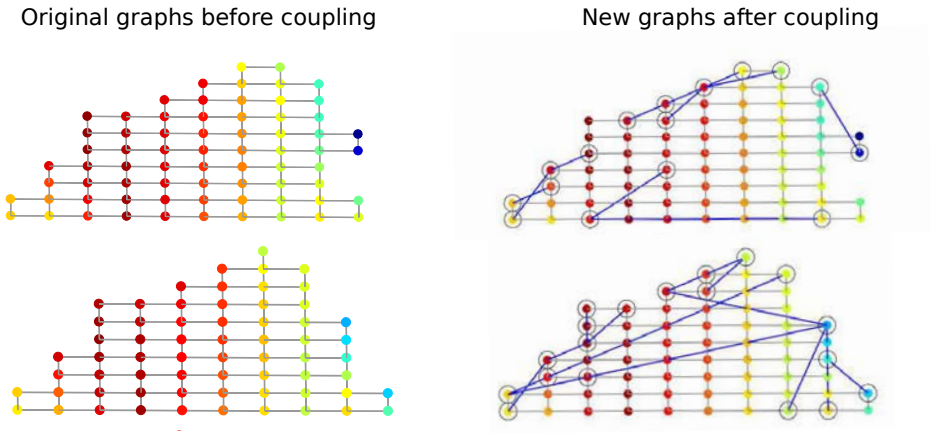


Figure 3.16: Image showing the old graphs before coupling and the new graphs after optimization. New edges with absolute weight values larger than 0.04 are shown as blue lines connecting highlighted nodes.

Angular graph transform As previously detailed in the paragraph 3.2.3, for each super-ray k, and for each band b, the Adjacency $\mathbf{A}_k^b$ and degree $\mathbf{D}_k^b$ matrices are used to compute the inter-view Laplacian as $\mathbf{L}_k^b$.

Moreover, the spatial-band vector is outlined as $\hat{\mathbf{x}}_k^b = [\hat{\mathbf{x}}_{k,v}(b)]_{v\in\{1,2,\ldots,N\}}$, s.t. $_{b<|\mathbf{x}_{k,v}|}$, where N is the total number of views. The angular transform coefficients are obtained by calculating:

$$\hat{\hat{\mathbf{x}}}_k^b = \mathbf{U}_k^{b^\top} \hat{\mathbf{x}}_k^b. \tag{3.16}$$

The inverse angular Graph Transform is then given by

$$\hat{\mathbf{x}}_k^b = \mathbf{U}_k^b \hat{\hat{\mathbf{x}}}_k^b \tag{3.17}$$

Correlation and energy compaction after angular transform

The gain in compaction after the spatio-angular transform is clear in Figure 3.15. This is due to the fact that we are able to preserve angular correlations after the spatial transform, which will be subsequently exploited by the angular transform.

In order to assess the performance of our coupling process in preserving the correlation, we draw in Figure 3.17, the correlation matrices and the covariance matrices for some bands after the first transform with shape-varying super-rays. If we restrict our attention to the first column, We see that after the first transform that is not optimized, we have uncorrelated transform coefficients due to the perturbation of eigenvectors computed on super-pixels having slightly different shapes. This problem is almost resolved with our coupling procedure in the second column, where we can observe more correlation between the coefficients of the same band in neighboring views. Furthermore, the logarithm

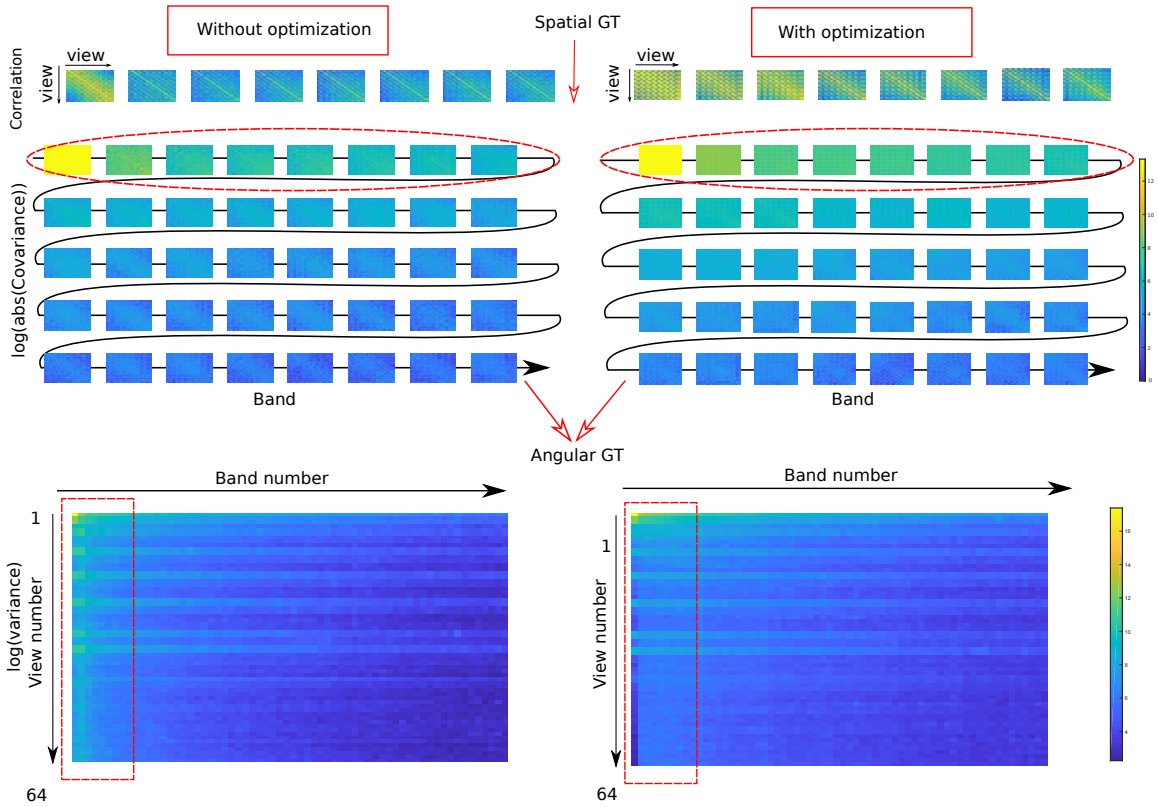


Figure 3.17: Advantage of our optimization in terms of energy compaction. The three rows correspond to (1) correlation matrices of the spatial transformed coefficients of the first ten bands, (2) the log of the absolute value of the covariance matrices of the 64 first bands of the spatial transformed coefficients, and (3) the logarithm of the variance of the coefficients after the angular transform, respectively. The two columns show the two cases: without or with our optimization.

of the variances (values lying on the diagonal in the covariance matrices) being higher in the first low frequency bands and decreasing when moving further from the DC, shows the energy compaction of the first transform. As for the values of the off-diagonal elements of the covariance matrices, they show how correlated are the transformed coefficients after the first transform inside the views. If we observe the off-diagonal values and compare them with or without optimization, we find out that the optimization performs better for low frequencies than for high frequencies and is therefore more able to retrieve coherent basis functions.

After the second angular transform per band, for both cases with or without optimization, we compute the logarithm of coefficients' variances after the second transform and illustrate it in the third row where the x-axis and y-axis correspond to the band number and the view number respectively. A compaction of the energy in fewer coefficients is observed in the optimized case compared to the non-optimized case, especially when we focus on the top-left region. Some inter-view high frequencies are sometimes still there and might be due to the presence of some super-rays are made of super-pixels that adhere well to borders in some views while not adhering in some others due to disparity rounding effects.

Once we have assessed the performance of our optimization and armed with the previously proposed tools, we exploit in the following section, the compaction property to directly code the luminance of the light fields.

3.3.4 Light field color coding scheme

The overall steps of the compression algorithm are shown in Figure 3.18. The top left view of the Light Field is separated into uniform regions using the SLIC algorithm to segment the image into super-pixels [1], and its disparity map is estimated. Using both the segmentation map and the geometry information, we construct consistent super-rays in all views as explained in section 3.3.1. The non separable and separable transforms described above are then locally applied on each super-ray. The transformed coefficients are then quantized and encoded to be stored or transmitted. The segmentation map of the reference view and a disparity value per super-ray also need to be transmitted as side information to the decoder.

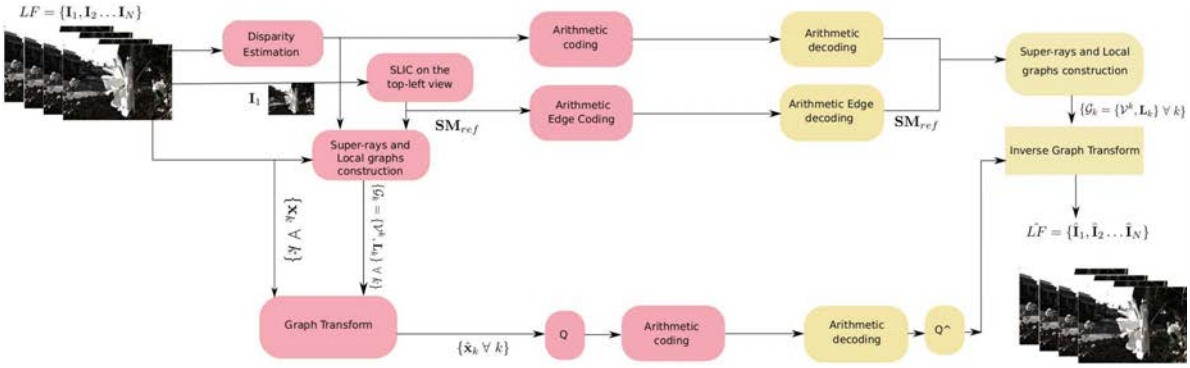


Figure 3.18: Overview of proposed color coding scheme for Light Field Compression

Segmentation map and disparity values coding

The segmentation map of the reference view is encoded using the arithmetic edge coder proposed in [23]. The contours are first represented by differential chaincode [36] and divided into segments. Then, to efficiently encode a sequence of symbols in a segment, *AEC* uses a linear regression model to estimate probabilities, which are subsequently used by the arithmetic coder. Disparity values are encoded using an arithmetic coder.

Grouping and transform coefficients coding

The energy compaction is not the same in all super-rays. This can be explained by the fact, that the segmentation may not well adhere to object boundaries, resulting in high angular frequencies after optimization of the first spatial transform.

To optimize the coding performance, we divide the set of super-rays into four classes, where each class is defined according to an energy compaction criterion.

First, we learn a scanning order. More precisely, at the end of the two graph transform stages, coefficients are grouped into a three-dimensional array $\mathbf{R}$ where $\mathbf{R}(i_{SR}, i_{bd}, v)$ is the v^{th} transformed coefficient of the band i_{bd} for the super-ray i_{SR}. Using the observations on all the super-rays in some training datasets (*Flower1*,*Friends*), we can find the best ordering for scanning and quantization. We sort the variances of coefficients with enough observations in decreasing order and we follow this decreasing order during the scanning process. For a class i, the high frequencies are defined as the last $round(N \times (4-i)/4)$ coefficients where N is the total number of coefficients. Each super-ray belongs to class i if it does not belong to class $i-1$ and the mean energy per high frequency is less than 1. We start by finding the super-rays in the first class than remove them from the search space before finding the other classes, and idem for the following steps. We code a flag with an arithmetic coder to gives the information of the class of super-rays to the decoder side. In class i, the last $round(N \times (4-i)/4)$ coefficients of each super-ray are discarded. The rest of the coefficients are grouped into 32 uniform groups. The quantization step sizes in groups are defined with a rate-distortion optimization taking into account a big number of observed coefficients. At the end of this stage, for each class, each group is coded using the Context Adaptive Binary Arithmetic Coder (*CABAC*) from the HEVC H.265 reference coder.

3.4 Rate-distortion performance evaluation

3.4.1 Experimental setup

For performance evaluation, we test our methods on real light fields captured by plenoptic cameras from [56] and [106]. We consider the 8×8 central sub-aperture images cropped to 364×524 in [56], and 9×9 cropped to 432×624 from [106] in order to avoid the strong vignetting and distortion problems on the views at the periphery of the light field. The disparity map of the top left view of each light field has been estimated using the method in [52]. The estimated disparity map is used to construct super-rays as described in Section 3.3.1.

We assess the compression performance obtained with our graph based transform coding schemes against three state of the art schemes: coding the light field views with *JPEG Pleno VM 1.1* or with HEVC as a video sequence following a lozenge order (*HEVC lozenge*) [83], and according to the scanning order proposed in [65] (*HEVC pseudo*).

The basic configuration files of JPEG Pleno VM 1.1 have been used with small changes in order to be applied on 9×9 views. For *HEVC-lozenge*, the base QPs are set to 20, 26, 32, 38 and a GOP of 4 is used. The HEVC version used in the tests is HM-16.10. The base QPs of *HEVC pseudo* are set to $QP_B = 8$, 14, 20, 26, 32 and 38, and the views at hierarchical layers 2, 3, 4, 5, 6 respectively have QPs equal to $QP_B + 8$, $QP_B + 9$, $QP_B + 10$, $QP_B + 11$ and $QP_B + 12$, as described in [65].

For the color coding scheme, we investigate the performance of the non separable, optimized and non optimized graph transforms which we denote as *Color-NS*, *Color-SO* and *Color-S* respectively. For the predictive coding scheme, two versions are studied: they are based on applying the spatial transform followed by an unweighted (*CNN-uGT*) or weighted (*CNN-wGT*) angular transform. The results are generated by selecting the best pairs of parameters (Q, QP) where Q is the quality parameter used to control the quantization of the transformed residuals and QP is used in the HEVC inter-coding of the four corners used to synthetize the whole light field prediction with CNN. Such selection can be automatically predicted after training a model represented by a function of light field features and target bitrate as in [54].

In Figures 3.19, 3.20 and 3.21, our color coding scheme based on both non separable and separable graph transforms is investigated against *HEVC-lozenge* and *JPEG pleno 1.1* for three light fields with 9×9 views, from the ICIP 2017 Grand Challenge [106].

Further experiments are also depicted in Figures 3.22, 3.23, 3.24 and 3.25 for 8×8 light fields from [56]. Note that *JPEG pleno VM 1.1* can hardly be applied for such data since the number of views is odd. We therefore use *HEVC pseudo*, *HEVC Lozenge*, *CNN* view synthesis (prediction) and *CNN-HEVC* as anchors. The latter is the direct coding of the residuals with HEVC as a video sequence. To show more precisely the gain of the predictive coding scheme at low bitrate we compute the Bjontegaard comparison in Table 3.2.

In Table 3.1, we restrict our attention to the optimized separable graph based transform (*Color-SO*) that can be applied no matter how big the super-rays are. It shows the rate allocation of our color coding scheme, at low and high bitrates, for the different light fields.

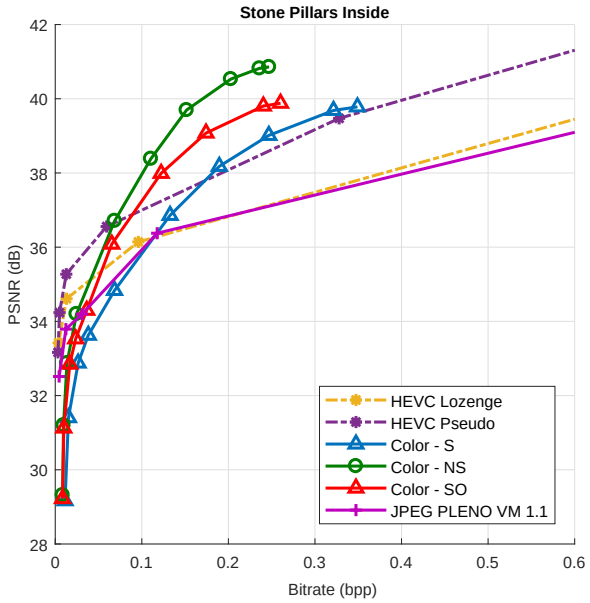


Figure 3.19: Rate distortion performance of our graph based Color coding schemes (*Color-NS*, *Color-S* and *Color-SO*) compared to *HEVC lozenge* and *JPEG Pleno VM 1.1* for "*Stone Pillar Inside*" from [106]

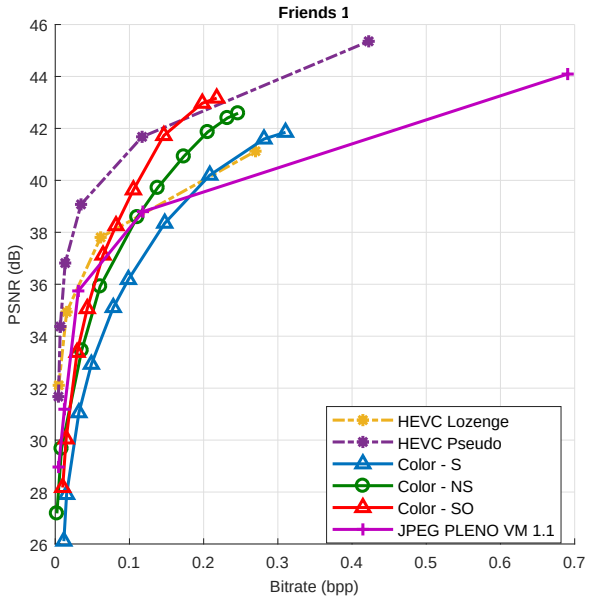


Figure 3.20: Rate distortion performance of our graph based Color coding schemes (*Color-NS*, *Color-S* and *Color-SO*) compared to *HEVC lozenge* and *JPEG Pleno VM 1.1* for "*Friends*" from [106].

3.4.2 The performance of the color coding scheme

A better performance with the optimization We can observe that, for most of the light fields used in our tests, the non separable graph transform (*Color-NS*) yields a better rate-distortion performance

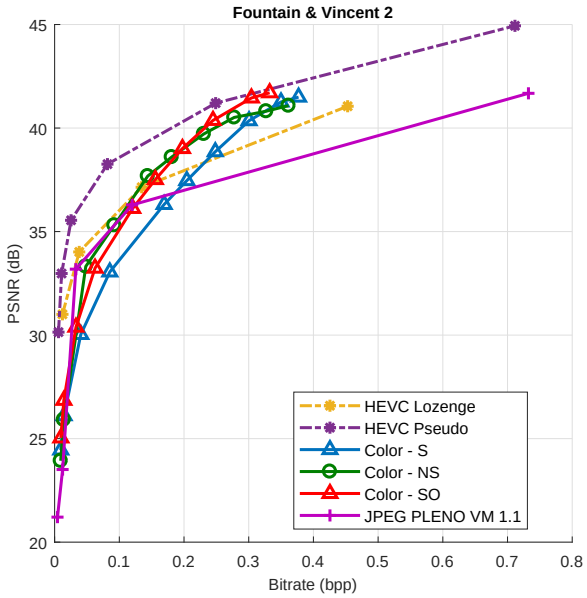


Figure 3.21: Rate distortion performance of our graph based Color coding schemes (*Color-NS*, *Color-S* and *Color-SO*) compared to *HEVC lozenge* and *JPEG Pleno VM 1.1* for "*Fountain Vincent*" from [106]

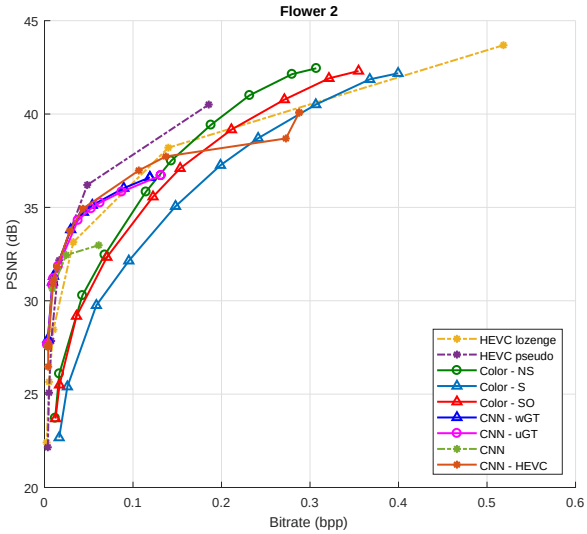


Figure 3.22: Rate distortion performance of our graph based coding schemes (*CNN-uGT*, *CNN-wGT*, *Color-NS*, *Color-S* and *Color-SO*) compared to *CNN*, *CNN-HEVC*, *HEVC lozenge* and *HEVC pseudo* for "*Flower 2*" from [56].

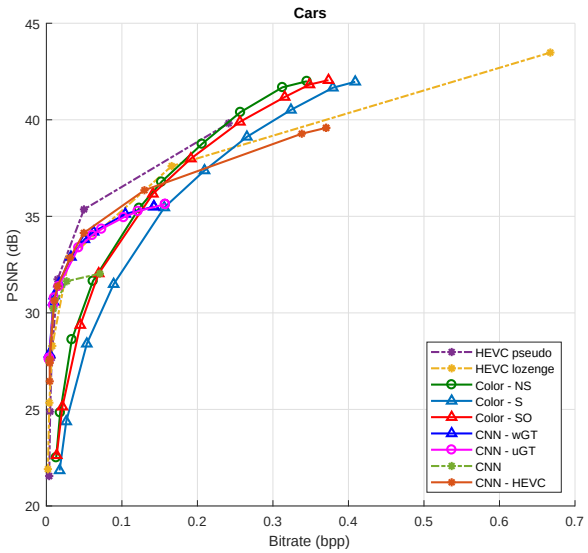


Figure 3.23: Rate distortion performance of our graph based coding schemes (*CNN-uGT*, *CNN-wGT*, *Color-NS*, *Color-S* and *Color-SO*) compared to *CNN*, *CNN-HEVC*, *HEVC lozenge* and *HEVC pseudo* for "*Cars*" from [56].

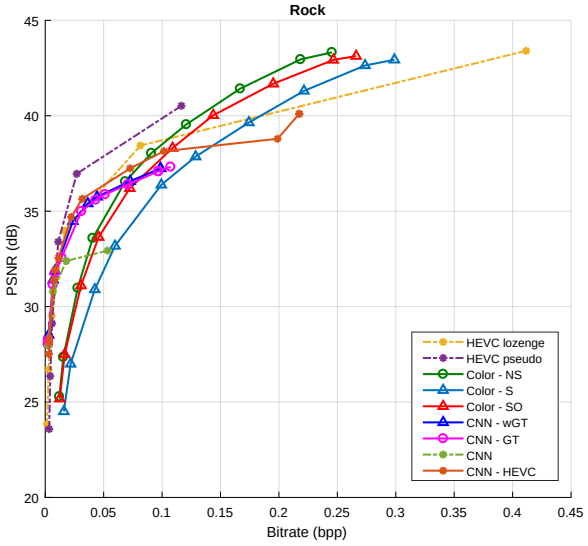


Figure 3.24: Rate distortion performance of our graph based coding schemes (*CNN-uGT*, *CNN-wGT*, *Color-NS*, *Color-S* and *Color-SO*) compared to *CNN*, *CNN-HEVC*, *HEVC lozenge* and *HEVC pseudo* for "*Rock*" from [56].

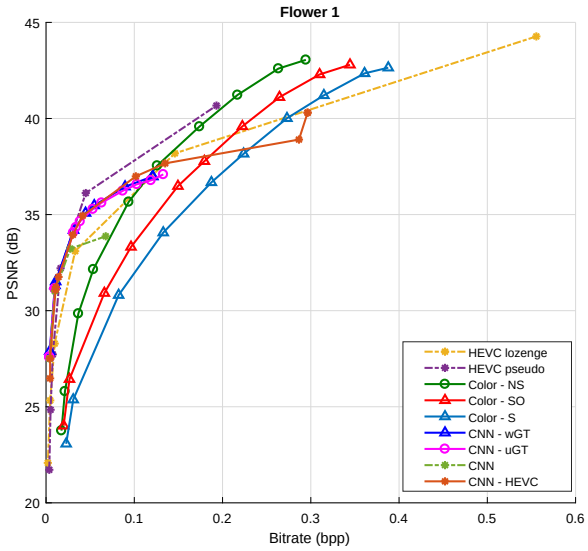


Figure 3.25: Rate distortion performance of our graph based coding schemes (*CNN-uGT*, *CNN-wGT*, *Color-NS*, *Color-S* and *Color-SO*) compared to *CNN*, *CNN-HEVC*, *HEVC lozenge* and *HEVC pseudo* for "*Flower 1*" from [56].

compared to the separable case (*Color-S* and *Color-SO*) for a fixed number of super-rays. While the non optimized graph transform (*Color-S*) fails to compact the energy of the light field, the optimized graph transform (*Color-SO*) is performing better and sometimes almost catches the non separable case. One major advantage of the separable optimized case is that it can be applied on super-rays of large dimensions without facing the basis functions computational complexity issue of the non separable case. Furthermore, the number of eigenvectors to be optimized can be defined by the encoder and does not have to be necessarily large.

A good performance compared with state of the art Moreover, we can observe a better performance of our method at high bitrate compared to *JPEG Pleno VM 1.1* and *HEVC lozenge*. At low bitrate, the prediction in the HEVC and JPEG Pleno based schemes is better than our disparity compensation of super-rays. Also, the bitrate allocated to the segmentation and disparity is very large, especially at low bitrate (almost reaching 30 percent for most datasets in Table 3.1) and could be further reduced.

Impact of disparity errors When the disparity information is not reliable, dis-occluded pixels may be clustered with a wrong super-ray, resulting in high frequencies, hence poor energy compaction, after the spatial transforms in those specific regions. As explained before, we overcome this problem by dividing the super-rays into classes.

Impact of super-rays size The size of super-rays may have an impact on the rate distortion performance especially when the disparity information is reliable and there is a lot of homogeneous objects. If we have large objects, we might want to merge some small super-rays which makes a non separable graph transform practically unfeasible. Here comes the advantage of an optimized separable graph

Light Field	Rate allocation(in %) for the Color - SO scheme			
	Overall bitrate	*Segmentation*	*Disparity*	*Coefficients*
Cars (364×524)	0.2563 bpp (PSNR = 42.24dB)	2.69%	0.55%	96.76%
	0.0212 bpp (PSNR = 25.23dB)	32.55%	6.60%	60.85%
Flower2 (364×524)	0.2710 bpp (PSNR = 40.77dB)	2.69%	0.55%	96.76%
	0.0362 bpp (PSNR = 29.18dB)	20.17%	4.14%	75.69%
Rock (364×524)	0.1951 bpp (PSNR = 41.68dB)	4.00%	0.82%	95.18%
	0.0306 bpp (PSNR = 31.10dB)	25.49%	5.23%	69.28%
Seahorse (364×524)	0.2302 bpp (PSNR = 42.99dB)	2.65%	0.74%	96.61%
	0.0612 bpp (PSNR = 33.88dB)	9.97%	2.78%	87.25%
Friends (432×624)	0.1464 bpp (PSNR = 41.73dB)	3.89%	0.10%	96.01%
	0.0294 bpp (PSNR = 33.38dB)	19.39%	5.10%	75.51%
StonePillarInside (432×624)	0.2204 bpp (PSNR = 39.07dB)	2.59%	0.54%	96.87%
	0.0212 bpp (PSNR = 32.85dB)	26.89%	5.66%	67.45%
FountainVincent (432×624)	0.2448 bpp (PSNR = 40.37dB)	2.12%	0.57%	97.31%
	0.0330 bpp (PSNR = 30.38dB)	15.76%	4.24%	80.00%

Table 3.1: Rate allocation performed by the proposed color coding scheme with the optimized separable graph transform (*Color - SO*). The rate is divided into three parts used for coding the segmentation, disparity and transform coefficients.

transform where one can define the number of eigenvectors to be optimized depending on the homogeneity of the shape-varying super-rays inside the views. In this case, the segmentation and disparity costs will more likely drop also since we also have less contours and values to code.

In our experiments, however, we use a uniform segmentation into super-pixels. We fix the number of super-rays to 2800 for the light fields in [56], and 4000 for the light fields in [106].

We have observed that when we have a small number of super-rays, the disparity errors may have an impact on the compensation and therefore result in a decreased PSNR-Rate performance. On the other hand, having a very large number of super-rays increases the rate needed for segmentation and limits the dimension of each super-ray, resulting in a smaller benefit in terms of de-correlation of the proposed spatio-angular transform.

3.4.3 The performance of the predictive coding scheme

We now restrict our attention to the two versions of the predictive coding scheme *CNN+uGT* and *CNN+wGT* in the Figures 3.22, 3.23, 3.24 and 3.25 and Table 3.2. Our Graph based transform approaches slightly outperform CNN learning based scheme at low bitrate and bring a small improvement to the HEVC based coding of the residues (Table 3.2). For higher bitrates, the compression performance is further enhanced compared to *CNN*, and almost reaching *CNN+HEVC* performance. At low to middle bitrates, both graph-based transform schemes outperform direct use of HEVC inter coding as we can also observe after computing the Bjontegaard metric in Table 3.2. Also, a small improvement is brought

by the weighted angular transform compared to the unweighted version.

Table 3.2: Bjontegaard comparison (ΔPSNR (dB)) at low bitrate (< 0.04 bpp)

	CNN-uGT *vs*			CNN-wGT *vs*
	CNN	HEVC lozenge	CNN-HEVC	CNN-uGT
Car	0.6	0.9	0.3	0.1
Flower 1	0.3	1.7	0.2	0.1
Flower 2	0.4	1.6	0.3	0.2
Rock	-0.1	0.7	-0.1	0.3

3.4.4 The predictive coding scheme vs the color coding scheme

If we restrict our attention to the low bitrate range in the rate-distortion curves, we can actually conclude that the light field prediction with view synthesis is very powerful compared to reconstructing the light field with only low frequencies captured by graph transforms on Color. This is shown in the gap between the dashed green curve (*CNN*) and our Color coding schemes.

Also, we can observe a better energy compaction when we apply the transforms on color with geometry-aware supports than on residues with fixed graph supports. This is clear when we observe the evolution of the curves of both schemes between low and medium to high bitrates. This is mainly due to the fact that the residuals are not smooth on the fixed graph supports, and their spatial angular variations can't be efficiently predicted by the color segmentation. Whereas in the color based coding scheme, geometry-aware graph supports are constructed assuming a good correlation between color values inside super-pixels and across the views, which is rarely violated only when disparity information is very poor.

Another issue in the color based coding scheme is that when we limit our transforms to local graph supports, we do not explore the correlations between super-rays. On the contrary, in the predictive coding scheme, the four corner views are actually coded with HEVC where intra and inter prediction are both applied, *i.e.* where spatial and angular, both short and long term dependencies are captured.

3.4.5 Small note about complexity

One might wonder how complex are our coding schemes especially on the decoder side. Indeed, the decoder needs to compute the optimized basis functions for the non consistent super-rays, inducing some computational complexity. However, the optimization can be performed independently on each super-ray, in a parallel manner. Also, if the super-rays are carefully built and the objects that we have in the scene follow the lambertian assumption, we end up with mostly consistent and some varying super-rays where the energy compaction is expected to be quite high. Hence, only a very few number of eigenvectors need to be optimized, others can be totally discarded. Also, with the advance of parallel computing and the ability of GPU arrays, we believe that the different parallel optimizations can be done at a time, reducing the decoding time.

Chapter 4

Graph-based spatio-angular prediction for light fields

Graph-based transforms have been shown to be powerful tools for compression. However, the computation of the basis functions becomes rapidly intractable when the size of the support increases, i.e. when the data is high dimensional (e.g. Light Fields). To cope with this difficulty, we have investigated the design of local transforms with limited supports in Chapter 3. Nevertheless, the locality of the support does not allow us to capture long term spatial dependencies of the color signal, unlike efficient predictive schemes used in state of the art codecs (e.g. HEVC). More precisely, in the case of local graph based transforms, the correlations between different super-rays are not exploited.

In this Chapter, we aim to tackle this problem. More accurately, the proposed approaches are based on the observation that, when using a local Graph Transform, either non separable or separable, most of the light field energy is packed in the low frequency coefficients. This motivates the design of line of actions that would allow the best compression of these coefficients, e.g. by exploiting spatial correlation beyond the limits of the local graph transform support. To do so, some form of prediction across super-rays would be needed. Nevertheless, the super-rays being of arbitrary shapes, developing inter super-ray prediction mechanisms is not an easy task. The idea we develop here consists instead in encoding a selected set of samples, using powerful prediction mechanisms available in state-of-the-art coders (e.g. HEVC), and then to recover the low frequency coefficients of the local graph transforms from its coded high frequency coefficients and the encoded reference samples as seen in Fig 4.1.

In summary, our contributions are as follow:

- In the non separable case, the ideal way of exploring both the transform's compaction and the long term dependencies would be to define the reference image as a light field view. However, due to matrix conditioning problems explained in the sequel, the recovered low frequencies in that case are very sensitive to high frequencies coefficients quantization, making the prediction very poor. In order to overcome this issue, we find a sub-optimal sampling set in each super-ray and project the samples into one reference image. Although this approach has the best rate distortion results (despite the noisy reference image), the complexity limitation and a high sensitivity to noise stand still for some cases.

- We thus derive our prediction equations in the separable spatio-angular case where no matrix inversion is needed. A light field view is chosen as a reference. This second approach keeps the advantages of both the reduced basis function computational complexity due to the limited support and the structured reference image (easily coded with intra-predictions). It however keeps

only in part the advantage of the energy compaction of the graph transform since the recovered frequencies do not necessarily correspond to the low frequencies.

The proposed methods can be seen as a graph-based prediction deriving low frequency spatio-angular coefficients from one single compressed reference image (e.g. the projected sampling sets in the non-separable case, top-left view in the separable case) and from the high frequency coefficients.

The methods have been assessed in the context of quasi-lossless encoding of light fields. Experimental results show that, when coupled with a powerful intra-prediction tool, the graph-based spatio-angular prediction brings a substantial gain in bitrate reaching almost 30% compared to HEVC coding of the light field as a video sequence.

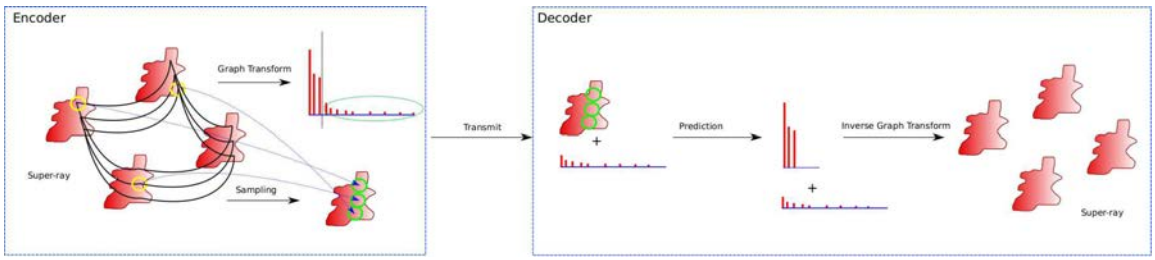


Figure 4.1: Overview of the prediction and sampling with a local graph transform on a specific super-ray. On the encoder side, a sampling is performed to find the best set of samples and a graph transform is applied. The high frequencies along with the reference samples (surrounded in green) are sent to the decoder which predicts the low frequency transform coefficients to then recover the original super-ray by inverse graph transform.

4.1 Spatio-angular prediction based on local non separable graph transform

4.1.1 Notations: Supports and Transform

Recall the second solution proposed in section 3.3 of Chapter 3. In one super-ray, we denote our light field luminance values as $\mathbf{x}_k$. The non separable graph transform is applied on $\mathbf{x}_k$ as follows:

$$\hat{\mathbf{x}}_k = \mathbf{U}_k^\top \mathbf{x}_k \tag{4.1}$$

Where the columns of $\mathbf{U}_k$ are the eigenvectors of the local graph laplacian inside the k^{th} super-ray.

The inverse graph Fourier transform is then given by

$$\mathbf{x}_k = \mathbf{U}_k \hat{\mathbf{x}}_k \tag{4.2}$$

We suppose that we have N_k pixels in one super-ray with N_{k,v_0} being the number of pixels that belong to a reference view v_0. We denote the set of pixel indices in a super-ray k as $\mathcal{S}_k$, those which lie in the sampling set as $\mathcal{S}$ and the set of all other pixels indices of the super-ray as $\mathcal{S}_C = \mathcal{S}_k \setminus \mathcal{S}$. We denote

the first N_{k,v_0} indices of low frequency coefficients as $\mathcal{T}$ and the rest of indices as $\mathcal{T}_C$. We will explain why we chose this number in the sequel.

4.1.2 Background on graph sampling

In this subsection, we introduce the theoretical background and some notations related to graph sampling theory which are crucial for the understanding of the rest of the chapter. Let's consider any graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ made of N vertices associated with a Laplacian $\mathbf{L}$. It has a complete set of eigenvalues $\lambda_l, l \in [1,\ N]$ and eigenvectors $\mathbf{u}_l, l \in [1,\ N]$. A graph signal is bandlimited and has a bandwidth $\omega = \lambda_n$ if it can be expressed as a linear combination of only the first n eigenvectors of $\mathbf{L}$. The space of ω-bandlimited signals is called a Paley-Wiener space and is denoted as $PW(\mathcal{G})_\omega \subset \mathbb{R}$. A subset of vertices $S \subset \mathcal{V}$ is a *uniqueness set* [74] for signals in $PW(\mathcal{G})_\omega \subset \mathbb{R}$ if $\forall f, g \in PW_\omega(\mathcal{G}), f(\mathcal{S}) = g(\mathcal{S}) \implies f = g$.

It can be also inferred that $\mathcal{S}$ is a *uniqueness set* for all signals $f \in PW_\omega(\mathcal{G})$, if and only if $[\mathbf{u}_1(\mathcal{S})\mathbf{u}_2(\mathcal{S}), \ldots, \mathbf{u}_n(\mathcal{S})]$ are linearly independent where λ_n is the n^{th} smallest eigenvalue of $\mathbf{L}$ and $\mathbf{u}_i(S) \in \mathbb{R}^{|S|}$ is a reduced eigenvector. The term reduced implies taking rows corresponding to the indices of the sampling set $\mathcal{S}$. [103]

It can also be shown that for any minimum uniqueness set $\mathcal{S}$ of size n for signals in $PW_\omega(\mathcal{G})$, there is always at least one node $\int_i \notin \mathcal{S}$ such that $\mathcal{S} \bigcup \int_i$ is a uniqueness set of size $n+1$ for signals in $PW_{\omega+1}(\mathcal{G})$. [103]

After building a uniqueness set, a simple way to reconstruct the missing samples is to solve a least-squares problem in the spectral domain [74]. Observing that the signal $\mathbf{f} \in PW_\omega(\mathcal{G})$ can be written as:

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}(\mathcal{S}) \\ \mathbf{f}(\mathcal{S}_c) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{U}}(\mathcal{S}) \\ \tilde{\mathbf{U}}(\mathcal{S}_c) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdots \\ \alpha_n \end{bmatrix} = \left[\begin{array}{cccc} \mathbf{u}_1(\mathcal{S}) & \mathbf{u}_2(\mathcal{S}) & \ldots & \mathbf{u}_n(\mathcal{S}) \\ \hline \mathbf{u}_1(\mathcal{S}_c) & \mathbf{u}_2(\mathcal{S}_c) & \ldots & \mathbf{u}_n(\mathcal{S}_c) \end{array}\right] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdots \\ \alpha_n \end{bmatrix}, \tag{4.3}$$

$[\alpha_1, \alpha_2, \ldots, \alpha_n]^\top$ can be retrieved by the least square solution to the upper part of the system above as:

$$[\alpha_1, \alpha_2, \ldots, \alpha_n]^\top = \left(\tilde{\mathbf{U}}^\top(\mathcal{S})\tilde{\mathbf{U}}(\mathcal{S})\right)^{-1} \tilde{\mathbf{U}}(\mathcal{S})\mathbf{f}(\mathcal{S}), \tag{4.4}$$

then the missing samples are reconstructed as follows:

$$\mathbf{f}(\mathcal{S}_c) = \tilde{\mathbf{U}}(\mathcal{S}_c) \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdots \\ \alpha_n \end{bmatrix} \tag{4.5}$$

where columns of $\tilde{\mathbf{U}}$ are the n first eigenvectors of the $\mathbf{L}$.

In the special case where $\mathcal{S}$ is of size n ($\mathcal{S}$ is therefore a *minimum uniqueness set*[103] for signals $\mathbf{f} \in PW_\omega(\mathcal{G})$), $\tilde{\mathbf{U}}(\mathcal{S})$ is a square invertible matrix. Equipped with the aforesaid arguments, the formulation

in Equation 4.4 can be further simplified to:

$$[\alpha_1, \alpha_2, \ldots, \alpha_n]^\top = \left(\tilde{\mathbf{U}}(\mathcal{S})\right)^{-1} \mathbf{f}(\mathcal{S}), \tag{4.6}$$

While the aforementioned sampling theorem [74] has been proposed for band-limited signals, we extend those equations to our problem in the following section. More precisely, we deal with signals(i.e. Color Signals) that might not be necessarily band-limited on the underlying graph supports (i.e. Super-Rays).

4.1.3 Graph-based spatio-angular prediction

Due to the high level of correlation between the different pixels forming a super-ray k, the energy of the transformed coefficients $\hat{\mathbf{x}}_k$ is highly compacted in the low frequencies $\hat{\mathbf{x}}_k(\mathcal{T})$. However, we might still end up with some non-zero high frequencies $\hat{\mathbf{x}}_k(\mathcal{T}_c)$. If we choose an appropriate uniqueness sampling set $\mathcal{S}$ in the k^{th} super-ray, then the non separable inverse graph transform is defined under appropriate permutation as:

$$\mathbf{x}_k = \mathbf{U}_k \hat{\mathbf{x}}_k \tag{4.7}$$

$$\Leftrightarrow \left[\begin{array}{c} \mathbf{x}_k(\mathcal{S}) \\ \hline \mathbf{x}_k(\mathcal{S}_c) \end{array}\right] = \left[\begin{array}{c|c} \mathbf{U}_k(\mathcal{S},\mathcal{T}) & \mathbf{U}_k(\mathcal{S},\mathcal{T}_c) \\ \hline \mathbf{U}_k(\mathcal{S}_c,\mathcal{T}) & \mathbf{U}_k(\mathcal{S}_c,\mathcal{T}_c) \end{array}\right] \left[\begin{array}{c} \hat{\mathbf{x}}_k(\mathcal{T}) \\ \hline \hat{\mathbf{x}}_k(\mathcal{T}_c) \end{array}\right]. \tag{4.8}$$

If the signal samples are transmitted separately, $\mathbf{x}_k(\mathcal{S})$ is available at the decoder. If we impose $|\mathcal{S}| = |\mathcal{T}|$, then $\mathbf{U}_k(\mathcal{S},\mathcal{T})$ is a square invertible matrix. Furthermore, if we only transmit $\hat{\mathbf{x}}_k(\mathcal{T}_c)$, then we are able to recover $\hat{\mathbf{x}}_k(\mathcal{T})$ from the following equation:

$$\hat{\mathbf{x}}_k(\mathcal{T}) = \Big(\mathbf{U}_k(\mathcal{S},\mathcal{T})\Big)^{-1}\Big(\mathbf{x}_k(\mathcal{S}) - \mathbf{U}_k(\mathcal{S},\mathcal{T}_c)\hat{\mathbf{x}}_k(\mathcal{T}_c)\Big). \tag{4.9}$$

Equation (4.9) is our so-called graph-based spatio-angular prediction. First, $\mathbf{x}_k(\mathcal{S})$ can be seen as a signal composed of a $\lambda_{|\mathcal{S}|}$-band-limited part plus some high frequencies. In this equation, we are actually removing the high frequencies to retrieve the band-limited signal (i.e. $\mathbf{x}_k(\mathcal{S}) - \mathbf{U}_k(\mathcal{S},\mathcal{T}_c)\hat{\mathbf{x}}_k(\mathcal{T}_c)$). Using the least squares reconstruction method in (4.4), we find the low frequency transformed coefficients $\hat{\mathbf{x}}_k(\mathcal{T})$.

Moreover, the high-frequency coefficients $\hat{\mathbf{x}}_k(\mathcal{T}_c)$ can be also seen as prediction coefficients, transmitted to recover the exact light field at the decoder. The basis of the linear prediction is the graph-transform basis, which makes these coefficients low-energetical and thus easy to transmit.

The signal values at $\mathcal{S}_c$ are then retrieved from the following equation:

$$\hat{\mathbf{x}}_k(\mathcal{S}_c) = \mathbf{U}_k(\mathcal{S}_c,\mathcal{T})\hat{\mathbf{x}}_k(\mathcal{T}) + \mathbf{U}_k(\mathcal{S}_c,\mathcal{T}_c)\hat{\mathbf{x}}_k(\mathcal{T}_c)$$

Where the first term is equivalent to the $\lambda_{|\mathcal{S}|}$-band-limited signal recovered on $\mathcal{S}_c$ and the second term is added in order to take into account the high frequency components.

To be able to carry out our graph-based spatio-angular prediction, we should at first determine the

appropriate sampling set. More precisely, we want to find $\mathcal{S}$ that results in the best conditioning of the sub-matrix $\mathbf{U}_k(\mathcal{S}, \mathcal{T})$ that guarantees a small reconstruction error. Simultaneously, we seek a sampling set that can be wrapped onto one single view to be coded with efficient prediction mechanisms. Equipped with the aforementioned statements in the previous subsections, we move forward to find the right $\mathcal{S}$ per super-ray.

4.1.4 Sampling set selection

A first intuitive way to define the sampling set per super-ray is to choose the set of N_{k,v_0} pixels that reside in the reference view v_0 that can be subsequently coded with intra HEVC. In our experiments however, we have found that the resulting sub-matrix $\mathbf{U}_k(\mathcal{S}, \mathcal{T})$ is ill-conditioned for non-consistent super-rays. This drove us to search for more efficient sampling algorithms existing in the literature.

We choose to use an adapted version of the algorithm described in [103]. It has been shown that this method provides a small reconstruction error for noisy signals of different bandwidths.

More precisely, for each super-ray k, we specify the band-limit frequency as λ_n^k with $n_k = N_{k,v_0}$(This number is outlined as such for coding purposes, the argument for this choice is explained in the sequel). We seek to find the optimal sampling set $\mathcal{S}$ that guarantees the exact reconstruction of any signal in $PW(\mathcal{G}_k)_{\lambda_n^k}$. We know that we have a correspondence between the size of the minimum uniqueness set and the signal bandwidth. We therefore want to find a set of N_{k,v_0} samples. In order to find the vertices that belong to this set, we have to find N_{k,v_0} linearly independent rows from the matrix $\tilde{\mathbf{U}}$. We follow the same reasoning as in [103] however with slightly different constraints to adapt it to our coding problem. In summary, the algorithm takes as input the whole super-rays graphs and the number of samples per super-ray. At the output of this stage, we want to wrap all the samples in a reference view to be efficiently coded with HEVC. Inspired by the work in [103], we use Algorithm 1.

While this method allows an optimal sampling per super-ray, yet, it does not guarantee that the output vector is well structured. It is impossible to say that the samples of neighboring super-rays will be efficiently de-correlated using intra-prediction mechanisms of any efficient coder. In extreme cases, we might end up with noisy samples that are very difficult to code. We thus propose to wrap our samples into one reference view taking into account the geometrical information given by our local graph.

We first observe that our non separable graph laplacian is a sum of two laplacians: The first one includes the connections $\mathbf{L}_k^s$ (s for spatial) inside views, and the other $\mathbf{L}_k^a$ (a for angular) made of edges between pixels inside different views. $\mathbf{L}_k^a$ is actually composed of various connected components, each one corresponding to a 3D point in the scene.

Using the angular information provided by $\mathbf{L}_k^a$, we define the matrix $\mathbf{E}$ of size $(N_{k,v_0} \times N_k)$ where each element gives the correspondence between a pixel in of super-ray k in v_0, and any other pixel in the super-ray. Consider a pixel p_1 in the view v_0. If we can access a pixel p_2 from p_1 following the graph connections in $\mathbf{L}_k^a$ then the entry $\mathbf{E}(p_1, p_2) = 1$, otherwise $\mathbf{E}(p_1, p_2) = 0$.

For each sample $\mathcal{S}(i)$ corresponding to a point p, we find the corresponding point p_0 in the set of pixels $\mathcal{S}_0$ belonging to the super-ray in the first view i.e. p_0 such as $\mathbf{E}(p, p_0) = 1$. The best case scenario is when each sample has a correspondence to a different pixel in the first view. In this case, the projection

Algorithm 1: Light Field Super-ray Graph based Sampling Algorithm

Data: The set of graphs for all super-rays, Segmentation map of a reference view, the sampling set size per super-ray: $\{\mathcal{G}_k = \{\mathcal{V}^k, \mathbf{L}_k\}\}$, $\mathbf{SM}_{ref}$,$\{n_k\}$
Result: A reference image made of samples drawn in all super-rays: $\mathbf{I}_{ref}$
foreach *Super-ray* k **do**
  Initialize: $\mathcal{S} \leftarrow \mathcal{V}_i^k$ where $\mathcal{V}_i^k$ is the vertice corresponding to the centroid of the super-pixel residing in the reference view ;
  Compute $\tilde{\mathbf{U}}$;
  for $m = 2 \rightarrow n_k$ **do**
    Define $\mathcal{T} = [1, m]$;
    Compute $z = null(\tilde{\mathbf{U}}(\mathcal{S}, \mathcal{T}))$;
    Normalize rows of $\tilde{\mathbf{U}}(\mathcal{S}_c, \mathcal{T})$;
    Compute $b = \tilde{\mathbf{U}}(\mathcal{S}_c, \mathcal{T})z$;
    $i \leftarrow argmax_i(|b(i)|)$;
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_c(i)$
  end
  Fill $\mathbf{I}_{ref}$ at the right positions : $\mathbf{I}_{ref}(\mathbf{SM}_{ref} = k) = \mathbf{x}_k(\mathcal{S})$;
end

is easy following the graph links. In the worst case, more than one sample might have a correspondence with the same point in the first view. In this case, first found, first served. The others are considered as disocclusions, which along with pixels who have no correspondence in the first view, will be projected into the rest of the available positions. Some images obtained after sampling and projections are shown in Figure 4.2. Despite the non-optimality of this method, we have ascertained that it leads to an acceptable rate with HEVC under lossless settings.

Once we have the samples in hand, they can be sent as prediction information to the decoder side, instead of sending the low frequency coefficients that contain most of the light field energy (about 99 $\%$ for all datasets).

4.2 Spatio-angular prediction based on local separable graph transforms

In this part, we move to deal with the separable graph transform. We propose to code one view as a reference and then recover the whole light field from this view and some high frequencies. We consider the same approach described in section 3.3, to compute the graph spatial and angular supports inside views. The signal considered is the luminance values of the light field. The spatial graph transform coefficients $\hat{\mathbf{x}}_{k,v}$ for each spatial graph $\mathcal{G}_{k,v}$ are obtained as in Chapter 3 by calculating:

$$\hat{\mathbf{x}}_{k,v} = \mathbf{U}_{k,v}^{\top} \mathbf{x}_{k,v}. \tag{4.10}$$

Figure 4.2: Reference Images obtained after projection of the sampling sets in all super-rays for different light fields.

Where $\mathbf{U}_{k,v}$ are the eigenvectors of the spatial laplacian and $\mathbf{x}_{k,v}$ are the luminance values of the super-ray k in view v. Inversely, the luminance values of the pixels belonging to the graph are retrieved from

$$\mathbf{x}_{k,v} = \mathbf{U}_{k,v}\hat{\mathbf{x}}_{k,v}. \tag{4.11}$$

Inside a super-ray k, the spatial transform coefficients $\hat{\mathbf{x}}_{k,v}$ are correlated between the views v. An angular transform is thus used to tract the similarity between the transformed coefficients of each band b, $\hat{\mathbf{x}}_{k,v}(b)$ across the views. The angular transform coefficients are obtained by calculating:

$$\hat{\mathbf{x}}_k^b = \mathbf{V}_k^{b^\top}\hat{\mathbf{x}}_k^b. \tag{4.12}$$

Where $\mathbf{V}_k^b$ is a matrix whose columns are the eigenvectors of the angular laplacian $\mathbf{L}_k^b$ drawn for the band b.

4.2.1 Separable graph-based spatio-angular prediction

Let us consider that the view 1 is coded as a reference. In order to perform the prediction, we follow the same reasoning as in non separable graph case however we apply it to each band that exists in the view 1.

For a given super-ray, the spatial transform in view 1 is $\hat{\mathbf{x}}_{k,1} = \mathbf{U}_{k,1}^{\top}\mathbf{x}_{k,1}$ according to notations introduced before.

We choose one sample for each band. It corresponds to the vertex $\mathcal{V}_i$ that is in the reference view (labeled by 1 in our case). For a given band b, the inverse angular transform is defined as

$$\hat{\mathbf{x}}_k^b = \mathbf{V}_k^b \hat{\hat{\mathbf{x}}}_k^b \tag{4.13}$$

$$\Leftrightarrow \left[\begin{array}{c} \hat{\mathbf{x}}_k^b(1) \\ \hline \hat{\mathbf{x}}_k^b(2) \\ \vdots \\ \hat{\mathbf{x}}_k^b(N_b), \end{array}\right] = \left[\begin{array}{c|ccc} \mathbf{V}_k^b(1,1) & \mathbf{V}_k^b(1,2) & \cdots & \mathbf{V}_k^b(1,N_b) \\ \hline \mathbf{V}_k^b(2,1) & \mathbf{V}_k^b(2,2) & \cdots & \mathbf{V}_k^b(2,N_b) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{V}_k^b(N_b,1) & \mathbf{V}_k^b(N_b,2) & \cdots & \mathbf{V}_k^b(N_b,N_b) \end{array}\right] \left[\begin{array}{c} \hat{\hat{\mathbf{x}}}_k^b(1) \\ \hline \hat{\hat{\mathbf{x}}}_k^b(2) \\ \vdots \\ \hat{\hat{\mathbf{x}}}_k^b(N_b) \end{array}\right] \tag{4.14}$$

where N_b denotes the number of views where the b^{th} band of the k^{th} super-ray is defined. Since the view 1 is transmitted separately, $\hat{\mathbf{x}}_k^b(1)$ is available at the decoder. If we only transmit $\hat{\hat{\mathbf{x}}}_k^b(2), \ldots, \hat{\hat{\mathbf{x}}}_k^b(N_b)$, the we are able to retrieve $\hat{\hat{\mathbf{x}}}_k^b(1)$ from the following equation:

$$\hat{\hat{\mathbf{x}}}_k^b(1) = \frac{1}{\mathbf{V}_k^b(1,1)} \left(\hat{\mathbf{x}}_k^b(1) - \left[\begin{array}{ccc} \mathbf{V}_k^b(1,2) & \cdots & \mathbf{V}_k^b(1,N_b) \end{array}\right] \left[\begin{array}{c} \hat{\hat{\mathbf{x}}}_k^b(2) \\ \vdots \\ \hat{\hat{\mathbf{x}}}_k^b(N_b) \end{array}\right] \right). \tag{4.15}$$

Equation (4.15) is our graph-based spatio-angular prediction for the separable case. The spatial coefficients of all the views are then retrieved from the following equation

$$\left[\begin{array}{c} \hat{\mathbf{x}}_k^b(2) \\ \vdots \\ \hat{\mathbf{x}}_k^b(N_b) \end{array}\right] = \left[\begin{array}{c} \mathbf{V}_k^b(2,1) \\ \vdots \\ \mathbf{V}_k^b(N_b,1) \end{array}\right] \hat{\hat{\mathbf{x}}}_k^b(1) + \left[\begin{array}{ccc} \mathbf{V}_k^b(2,2) & \cdots & \mathbf{V}_k^b(2,N_b) \\ \vdots & \ddots & \vdots \\ \mathbf{V}_k^b(N_b,2) & \cdots & \mathbf{V}_k^b(N_b,N_b) \end{array}\right] \left[\begin{array}{c} \hat{\hat{\mathbf{x}}}_k^b(2) \\ \vdots \\ \hat{\hat{\mathbf{x}}}_k^b(N_b) \end{array}\right]$$

Once the decoder recovers the first spatial graph transform coefficients in all the views, it can reconstruct the whole light field color values by a simple spatial inverse GFT since it has access to the graph supports and coefficients.

4.3 Non separable vs separable graph based prediction

We apply both our methods on real light fields captured by plenoptic cameras from the datasets used in [56] and [106]. To avoid the strong vignetting and distortion problems on the views at the periphery of the light field, we only consider the 8×8 central sub-aperture images cropped to 364×524 in [56], and 9×9 cropped to 432×624 from [106]. Some of the light fields considered are shown in Figure 4.5. The

Figure 4.3: Reference Images being the top-left views of each light field.

full set of light fields considered for the test is: *Flower2*, *Cars* ,*Rock* and *Seahorse* from the dataset in [56] and *StonePillarInside* and *Friends* from the dataset of ICIP challenge 2017 and used in [106]. The method used to estimate the disparity of the top-left views is described in [52]. Examples of the disparity maps provided are shown in Figure 4.6. A sparse set of disparity values and the segmentation maps computed with SLIC [1] are used to construct local graph supports as described in Section 4.2.

4.3.1 Energy compaction

As explained before, we aim at compacting most of the light field energy in few coefficients, and at then predicting these coefficients (i.e. they are not transmitted) from a coded reference image and from the high frequency graph transform coefficients that need to be transmitted at small cost given that they contain little information. Table 4.1 gives the percentage of total energy that resides in the predicted DC spatio-angular bands for both non separable ($\hat{x}_k(\mathcal{T})\ \forall\ k$) and separable ($\hat{\mathbf{x}}_k^b(1)\ \forall k, b$) cases. We can observe that most of the energy is compacted in the DC spatio-angular bands, which shows the efficiency in terms of spatio-angular de-correlation of the graph transforms.

The non separable prediction has the benefit of the low energy of the high frequency coefficients of

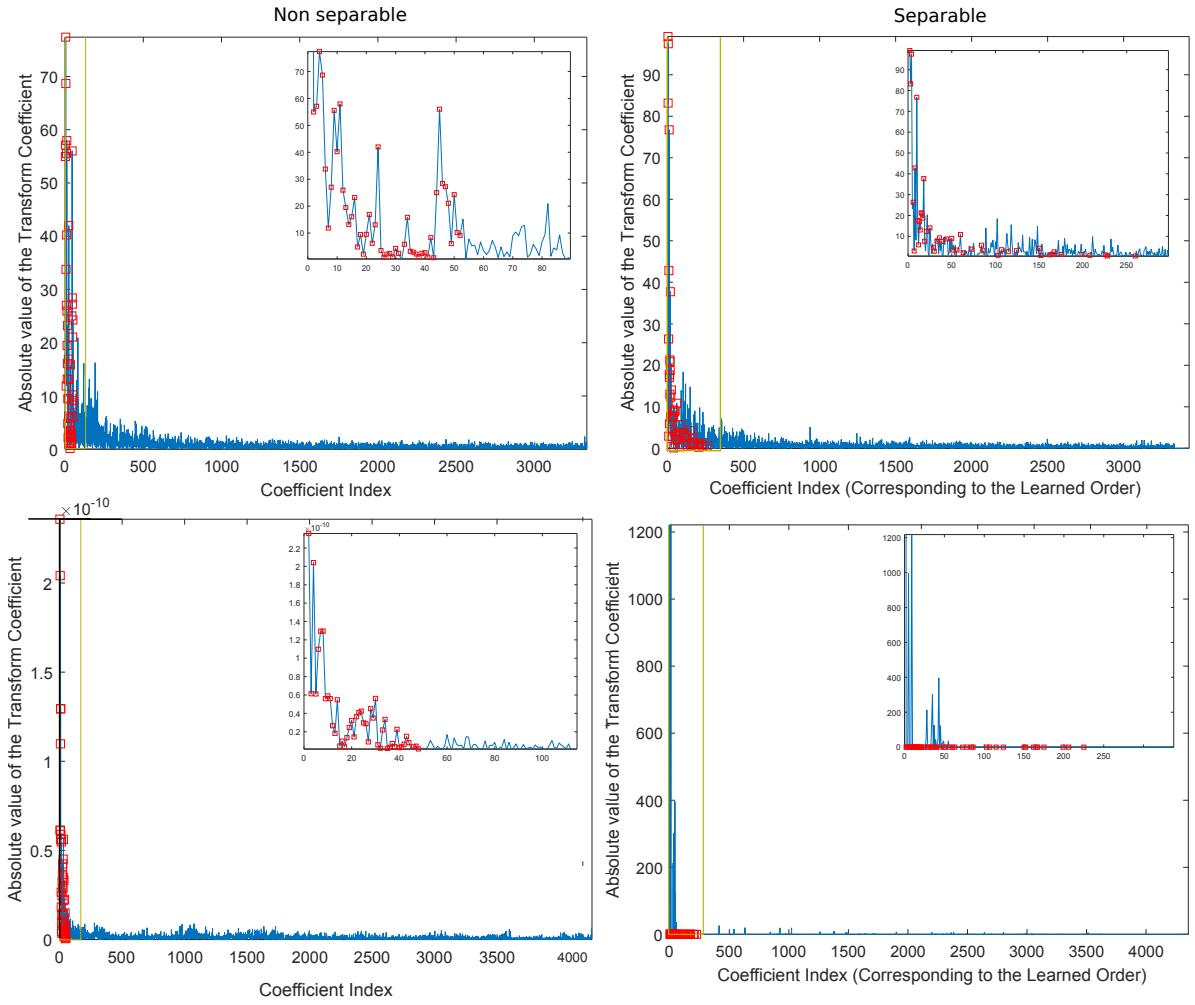


Figure 4.4: Illustration of the energy compaction for two super-rays of *Flower2*. The transform coefficients are ordered with the assumed frequency order. The red squares are the predicted DC values on the decoder side. The two rows correspond to two different super-rays and the two columns are for both cases: Non Separable and Separable respectively.

Figure 4.5: An example set of light fields used in our experiments. Only the top-left view is shown for illustration purpose. From left to right: *Flower2*, *Cars*, *StonePillarsInside* and *Friends*.

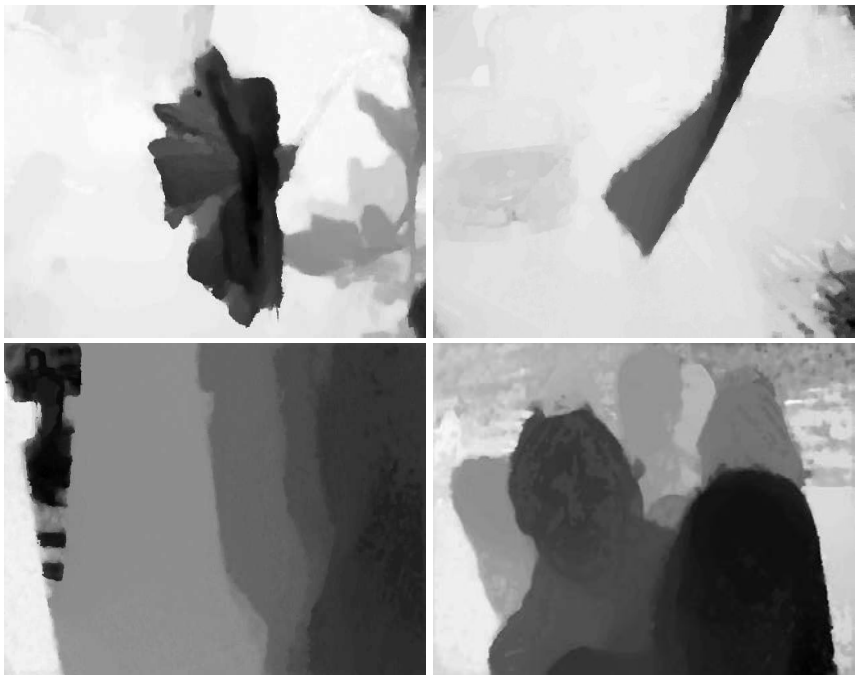


Figure 4.6: An example set of top-left view disparity maps used in our experiments

Table 4.1: Percentage of energy residing in the DC spatio-angular bands in the non separable case $\hat{x}_k(\mathcal{T}) \; \forall \; k$ and in the separable case $\hat{\mathbf{x}}_k^b(1) \; \forall \; k, b$

Light Fields	Energy Percentage in $\hat{x}_k(\mathcal{T}) \; \forall \; k$	Energy Percentage in $\hat{\mathbf{x}}_k^b(1) \; \forall \; k, b$
Flower 2	99.15 %	99.02 %
Cars	99.27 %	99.34 %
Rock	98.63 %	98.45 %
Seahorse	99.17 %	98.73 %
Stone Pillars Inside	98.90 %	98.26 %
Friends	99.76 %	99.80 %

the graph transform that also need to be coded. The separable graph transform, in some cases, loses this benefit as we predict the DC angular(i.e. after the transform across the views) coefficients of all spatial bands. Those low angular frequency coefficients may not contain all the energy otherwise captured by the lower spatio-angular frequency coefficients of the non separable case, although it remains quite efficient in terms of energy compaction as we can see in Table 4.1.

To further illustrate the energy compaction of the transforms, we plot in Fig. 4.4, for two different super-rays, the transform coefficients following the coding order (learned order of frequencies) for both cases: separable and non separable graph transforms. As we can see, in the non separable case, the low frequencies that are predicted on the decoder side (the red dots) correspond to the first frequencies and thus to those who hold most of the energy. However, in the separable case, the coefficients predicted do not necessarily exhibit the highest energy. This is quite clear in the second example, where the red dots in the separable case are assigned to very low values.

4.3.2 Compressibility of the reference view

Thanks to the prediction equations introduced in Section 4.2.1, an efficient encoding of the top-left view in the separable case or the reference view in the non separable case (using any classical encoder with efficient spatial predictors) can be seen as a way to encode those DC spatio-angular frequency coefficients which contain most of the light field energy.

The separable graph transform based prediction takes advantage of the natural structure of the reference view as we can see in Fig. 4.3. It is thus efficiently coded using intra-prediction tools. For the non-separable graph prediction, however, this is not the case since the optimal sampling does not totally guarantee that the samples are well structured in each super-ray of the 2D reference view. Yet, the super-ray segmentation preserves in a certain way the natural structure of the reference view (See Fig. 4.2).

In our experiments, we choose HEVC intra to encode this information (i.e. top left view or reference view). Tables 4.2 and 4.3 give the bit rate obtained when encoding the reference view (from which are derived the DC spatio-angular frequency coefficients) with HEVC-Intra (with QP set to 0). The bit rates are compared with those obtained when using a simple arithmetic coder for directly encoding the spatio-angular DC coefficients. In order to apply the arithmetic coder for each frequency band b, we first group all the coefficients $\hat{\mathbf{x}}_k^b(1) \; \forall k$ of the super-rays in which this band exists, and we code them with an arithmetic coder independently of the other bands. The table shows the rate gain obtained by encoding

the set of reference samples with HEVC intra, thanks to the possibility to capture dependencies between super-rays.

Table 4.2: Bit rate obtained, in the case of the separable graph transform, when using HEVC intra to code the first view (left column), and when using entropy (arithmetic) coding of all the DC spatio-angular bands $\hat{\mathbf{x}}_k^b(1) \; \forall \; k, b$ (right column).

Light Fields	HEVC intra of the first view	Entropy coding of $\hat{\mathbf{x}}_k^b(1) \; \forall \; k, b$
Flower 2	1.0 Mbits	1.48 Mbits
Cars	1.06 Mbits	1.54 Mbits
Rock	1.02 Mbits	1.38 Mbits
Seahorse	0.72 Mbits	1.22 Mbits
Stone Pillars Inside	1.81 Mbits	1.66 Mbits
Friends	1.62 Mbits	2.04 Mbits

Table 4.3: Bit rate obtained, in the case of the non separable graph transform, when using HEVC intra-coding (left column) of the reference set of samples, and entropy (arithmetic) coding of all the DC spatio-angular bands $\hat{\mathbf{x}}_k(\mathcal{T}) \; \forall \; k$ (right column).

Light Fields	HEVC-Intra coding of set of reference samples	Entropy coding of $\hat{\mathbf{x}}_k(\mathcal{T}) \forall k$
Flower 2	1.23 Mbits	1.59 Mbits
Cars	1.45 Mbits	1.64 Mbits
Rock	1.31 Mbits	1.49 Mbits
Seahorse	0.74 Mbits	1.38 Mbits
Stone Pillars Inside	1.92 Mbits	1.84 Mbits
Friends	1.86 Mbits	2.07 Mbits

4.3.3 Robustness of the Prediction

In order to assess the efficiency of our prediction and the light field sampling algorithm for the non separable case, we plot in Fig. 4.7 the condition number in log base 10 of the matrix $\mathbf{U}_k(\mathcal{S}, \mathcal{T})$ for all super-rays k in all the datasets. The condition number is measured to show how much sensible is our prediction in equation 4.9 $\hat{\mathbf{x}}_k(\mathcal{T})$ to a small change in $\Big(\mathbf{x}_k(\mathcal{S}) - \mathbf{U}_k(\mathcal{S}, \mathcal{T}_c)\hat{\mathbf{x}}_k(\mathcal{T}_c)\Big)$. On one hand, the condition numbers are computed without sampling i.e. assuming that the reference samples as those in the top-left view. These are shown in red. While the results in blue correspond to the condition numbers after taking the actual samples found with algorithm 1. A major difference is shown in log scale, where the sampling has reduced the condition number from 10^{15} to a maximum of around 10^2. Without sampling, the prediction fails since a tiny change in the high frequency coefficients (even a small rounding procedure) can result in a huge loss in the reconstruction quality.

For the prediction based on the separable graph transform, we do not need a matrix inversion. We only need to invert a number $\mathbf{V}_k^b(1,1)$ whose minimum corresponds to $1/\sqrt{M \times N}$. This inversion does have a smaller impact than the one in the non separable case. This is a major explanation of the PSNR difference between our schemes for a fixed quantization step size $Q = 1$ in Table 4.4.

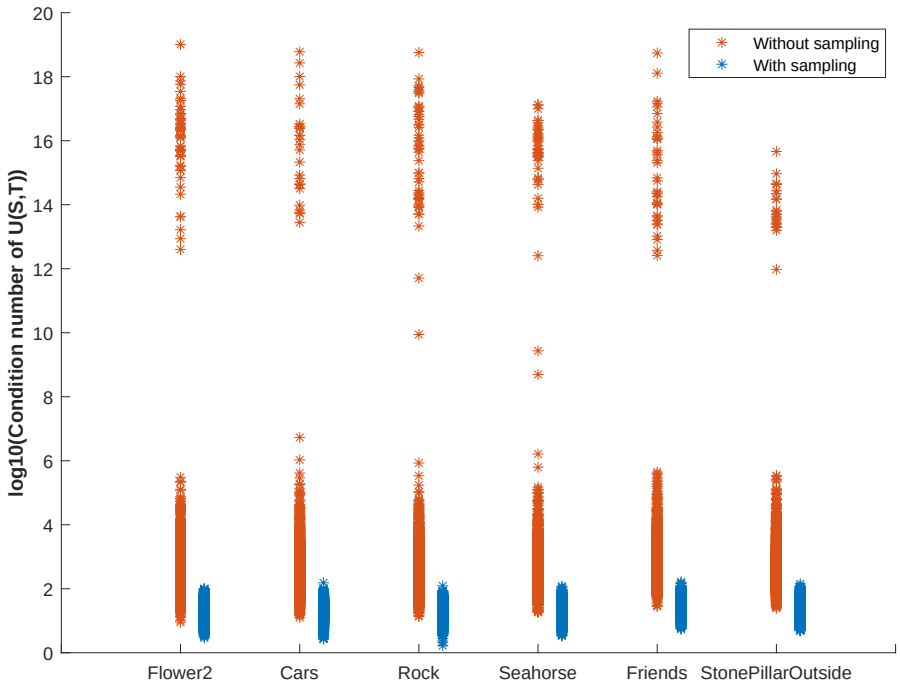


Figure 4.7: Efficiency of the sampling and effect on the condition number of the matrix $\mathbf{U}_k(\mathcal{S}, \mathcal{T})$. We show for each dataset, for all super-rays the log base 10 of the condition number without (red) and with (blue) sampling.

4.4 The proposed coding schemes

4.4.1 Overall description of the coding scheme based on the non separable graph transform

The first proposed quasi-lossless coding scheme is shown in Figure 4.8 and is based on the non separable case.

The top left view $\mathbf{x}_1$ is separated into uniform regions using the SLIC algorithm ([1]) to segment the image into super-pixels, and its disparity map is estimated with [52]. The disparity values are encoded using simple arithmetic coder. The segmentation is coded with edge arithmetic coder (AEC) [23] as in the previous chapter. Using both the segmentation map and the geometrical information, we can build consistent super-rays and graphs in and across all views as explained in section 3.3.3 in both the encoder and decoder sides.

Once the local graphs are computed, we can find the optimal sampling sets (their actual positions in the light field and the corresponding luminance values) as explained in 4.1.4. Those samples are reorganized in a reference image coded with HEVC intra and sent as prediction information to the decoder.

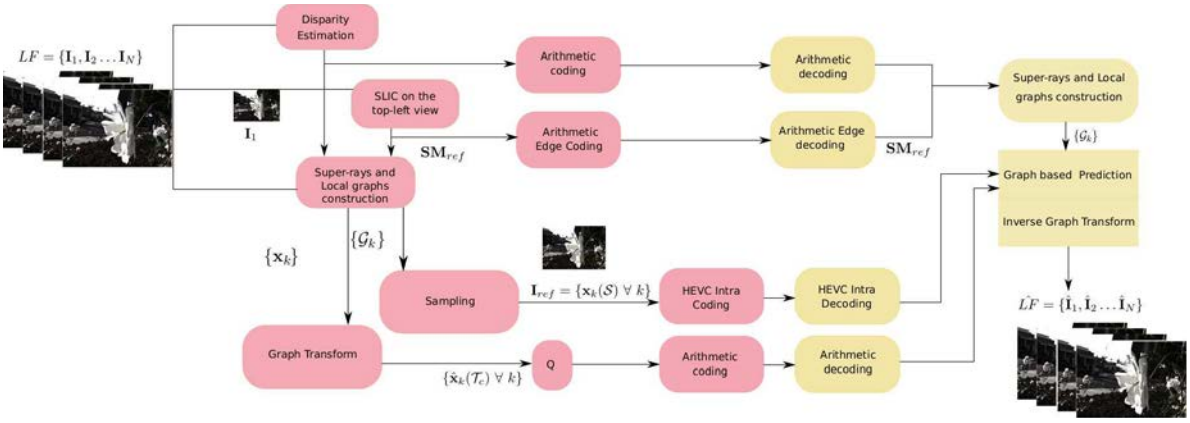


Figure 4.8: Overview of proposed coding scheme based on the non separable graph transform.

We apply the non separable graph transform on the coded version of the reference image (quasi-lossless coding) and the original values of all other samples to compact their energy in fewer coefficients. Since the reference image is coded with very small QP, we are almost sure that we are not adding angular incoherence between the different views. Once we have the graph transforms coefficients, instead of sending the whole spectrum with simple arithmetic coding, we propose to make use of our graph-based prediction and therefore deriving low frequency spatio-angular coefficients in the decoder side from the reference image coded with HEVC-Intra and the high angular frequency coefficients.

We thus send, for all super-rays, the AC coefficients ($N_k - N_{k,1}$) last bands after the non separable graph transform. (N_k and $N_{k,1}$ are the number of pixels belonging to the super-ray k and those only residing in view 1 respectively). Specifically, after applying the spatio-angular graph transforms on all super-rays, all frequency coefficients are grouped into a two-dimensional array $\mathbf{y}$ where $\mathbf{y}(k, v)$ is the v^{th} transformed coefficient for the super-ray k. Using the natural scanning order (increasing order of eigenvalues), we assign a class number to each super-ray. For a class i, the high frequencies are defined as the last $round(N_k \times (4-i)/4)$ coefficients where N_k is the total number of coefficients. Each super-ray belongs to class i if it does not belong to class $i-1$ and the mean energy per high frequency is less than 1. More precisely, we start by finding the super-rays in the first class then remove them from the search space before finding the other classes, and idem for the following steps. We code a flag with an arithmetic coder to give the information of the class of super-rays to the decoder side. In class i, the last $round(N_k \times (4-i)/4)$ coefficients of each super-ray are discarded. Then, the rest of the high frequency spatio-angular coefficients are quantized uniformly with a small step size $Q = 0.5$. Then, they are grouped in 32 uniform groups to enter a simple arithmetic coding.

4.4.2 The proposed coding scheme based on the separable graph transform

In the separable case (Fig 4.9), one major difference resides where the top-left view is the reference view first coded with HEVC intra.

Since the decoder already receives the top left image and the disparity values, with the SLIC algorithm, it can deduce the segmentation map and the super-rays used for local graph transforms design

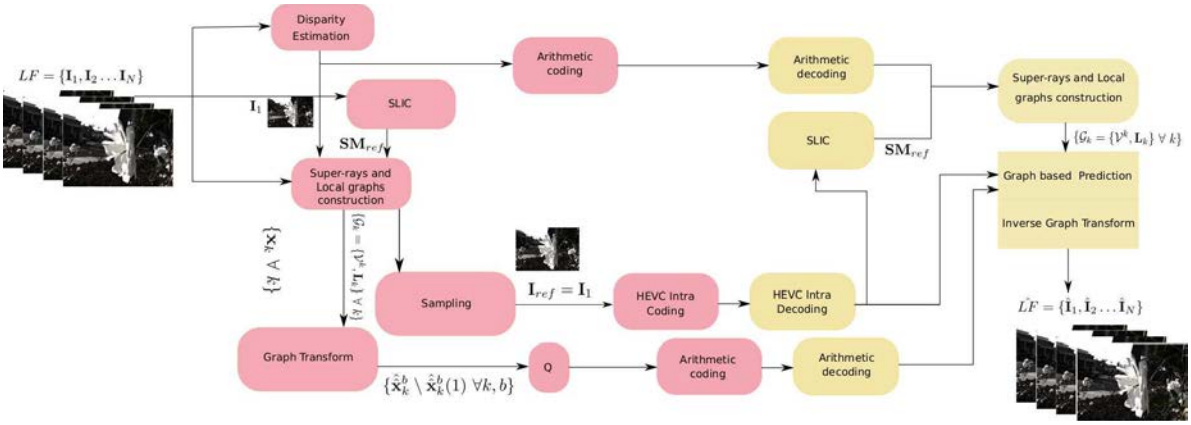


Figure 4.9: Overview of proposed coding scheme based on the separable graph transform.

and application.

Furthermore, the decoder have received the spatio-angular high frequency coefficients. Using those two kinds of information, it can predict the DC spatio-angular components obtained after the angular graph transform and then reconstruct the luminance values of all the views as explained in Section 4.2.1.

4.5 Comparative assessment against State of the Art coders

To assess our Graph-based Spatio-Angular Prediction, we evaluate it in the context of the quasi-lossless coding scheme in Section 4.4 against a complete HEVC based scheme with a QP set to 0 and a GOP of 4. The HEVC version used in the tests is HM-16.10. The light fields are coded following a raster scanning starting with the top-left view as a reference intra-coded frame. Results are reported in Table 4.4 where we compare mainly the rate needed to code a Light Field under quasi-lossless settings (based on visual quality assessment, we consider a PSNR higher that 50 dB as a quasi-lossless compression). A substantial gain in bitrate is observed while preserving a high quality of the reconstructed light fields. This can be justified by the efficiency of our spatio-angular graph transforms in terms of compaction along with the ability of HEVC-intra to effectively exploit spatial correlations in the reference view.

Table 4.4: Rate comparison between our proposed schemes (with both non separable and separable graph transforms) and a scheme using HEVC-inter to code the views in a raster scan order, at high quality (PSNR > 50 dB)

Light Fields	HEVC-Inter (QP=0) Raster Scan	Non Separable Scheme (Q = 0.5)	Non Separable Scheme (Q = 1)	Separable Scheme (Q = 1)
Flower 2	3.3129 bpp (54.2033 dB)	2.4470 bpp (60.4656 dB)	2.4457 bpp (52.9393 dB)	2.4799 bpp (55.1969 dB)
Cars	3.6688 bpp (54.0812 dB)	2.7759 bpp (60.5035 dB)	2.7801 bpp (53.0268 dB)	2.6258 bpp (55.2009 dB)
Rock	3.2700 bpp (53.7601 dB)	2.0423 bpp (60.2994 dB)	2.0545 bpp (52.6230 dB)	2.0162 bpp (54.7765 dB)
Seahorse	2.4751 bpp (54.3804 db)	1.8224 bpp (60.4474 dB)	1.7849 bpp (53.0111 dB)	1.9762 bpp (55.2844 dB)
Stone Pillars Inside	4.9017 bpp (52.1036 dB)	2.5559 bpp (59.7134 dB)	1.5269 bpp (52.3953 dB)	3.3094 bpp (55.0022 dB)
Friends	3.5400 bpp (52.7986 dB)	1.9327 bpp (59.7657 dB)	1.9311 bpp (52.4402 dB)	2.4436 bpp (54.8196 dB)

4.6 Conclusion and perspectives

In this chapter, we have explored local separable spatio-angular graph transforms for light fields compact representation. The limited support of local transforms may not allow us to exploit long term spatial dependencies. To cope with this limitation, we have proposed a novel approach to leverage the good spatial de-correlation properties of traditional codecs (e.g. HEVC intra), making use of efficient predictors, into local spatio-angular graph transforms. A reference view coded with any efficient codec is used to predict low angular frequency transform coefficients that, together with the transmitted high angular transform coefficients, allow recovering the entire graph-based representation. The scheme has been assessed for high quality (quasi-lossless) coding. Note that the prediction mechanism, in case of coarser quantization, tends to amplify the quantization noise on the reconstructed data representation. This issue is left for further study.

Both proposed approaches are very efficient when the quantization noise on the reference view is limited, hence for quasi-lossless compression. If the latter is too coarsely compressed, drift and noise amplification might appear during this prediction step. This is due to the fact that, in Equation (4.15), the prediction uses the spatial transform coefficients estimated on the reference view available at the decoder side. Further study will be dedicated to addressing this problem in the case of lossy compression.

Chapter 5

Rate-distortion optimized graph partitioning for omnidirectional image coding

5.1 Introduction

In this Chapter, we focus on another type of image modalities, namely omnidirectional images captured with 360 cameras. Nowadays, they are widely used for popular applications such as virtual reality and immersive communications. Omnidirectional images are spherical signals captured by cameras with 360-degree field of view. In order to use existing image and video processing algorithms, these signals are usually mapped to planar domain and stored as rectangular lattices. A commonly used planar representation for omnidirectional content is the so-called *equirectangular* representation [12] as explained in section 2.2.2 (Figure 5.1).

Such representation is widely adopted to store and process omnidirectional signals due to its simplicity and its compliance with classical image and video processing chains, designed for rectangular images and videos captured by perspective cameras. Nevertheless, omnidirectional cameras, such as catadioptric or fish-eye cameras, have a specific nature where lines in the 3D space are projected into curves in the image domain. Additionally, this representation presents strong warping distortions around the polar areas and corresponds to an *equi-angular* sample distribution on the spherical surface, which is non-uniform (Figure 5.2).

While the actual sampling patterns of multi-dioptric systems are difficult to model, it is unlikely that the acquisition system would be designed to perform a non-uniform sampling of the surrounding space, with more samples captured around the poles. Therefore, the level of information carried by each pixel is heterogeneous with more informative pixels in the equator and more redundant ones around the poles. This pose problems in real world applications where one might be interested in some viewports on the sphere. An equirectangular image can be fed as input to existing state of the art encoders, however the equirectangular signal statistics differ from those of classical perspective images. Thus, using existing compression algorithms is sub-optimal [94].

In this chapter, similarly to the case of *light fields*, we aim at finding the best supports for local graph based transforms of *omni-directional images*. For that purpose, we first propose to use a graph-based representation which takes into account the spherical geometry and provides a flexible way to efficiently store and compress the omni-directional content. Specifically, we propose to represent an omnidirectional image by a graph, where the graph vertices correspond to the image pixels defined on the spherical surface. The edge weights capture the sampling grid on which the signal is defined. Such a flexible representation permits to go beyond traditional transform coding by moving from classical fixed trans-

(a) Pool (b) Farm

(c) Metro (d) Hotel

Figure 5.1: Omnidirectional images (in equirectangular format) used in our experiments: two outdoor images (a and b) and two indoor images (c and d)[111]

forms such as the Discrete Cosine Transform (DCT) to graph-based transforms that are adapted to the actual signal support, such as the Graph Fourier Transform (GFT) [34] [92].

Due to the high spatial resolution typical of omnidirectional images [12], the graph that we propose to build would have a huge number of vertices (>500K). Consequently, the GFT computation in an actual coding pipeline would be unfeasible. Indeed, one would think about using sampling techniques on the sphere. However such coding schemes involve interpolation which makes the distortion control very complex. To overcome this problem, we propose an efficient graph partitioning strategy, which takes into account the geometrical information in order to optimize the smoothness of the signals on the subgraphs while keeping a small overhead to code the description of the partition. Finally, we propose a complete GFT-based lossy compression scheme using this partitioning and compare its performance to the classical DCT-based JPEG coding [27]. Experimental results show that the partitioning provides an effective tradeoff between the smoothness of signals on the subgraphs and the cost of coding the partition. Moreover, the proposed coding scheme outperforms JPEG coding of planar equirectangular images, in terms of Rate-Distortion (RD) analysis using multiple quality metrics.

5.2 360-degree image as signal on a graph

A 360-degree image I can be represented by a signal $\mathbf{x} \in \mathbb{R}^N$ defined on an undirected, 4-connected, weighted *global graph* $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ (See Figure 5.2), which consists of a finite set of vertices $\mathcal{V}$ defined on the image surface, with $|\mathcal{V}| = N$, a set of edges $\mathcal{E}$, and a weighted adjacency matrix $\mathbf{W}$. For $i = 1, ..., N$, the signal value x_i corresponds to the pixel color value at vertex $i \in \mathcal{V}$. If there is an edge $e = (i, j)$ connecting vertices i and j, the entry $W_{i,j}$ represents the weight of the edge, otherwise, $W_{i,j} = 0$. We define the weight of an edge connecting adjacent vertices i and j via a Gaussian kernel weighting

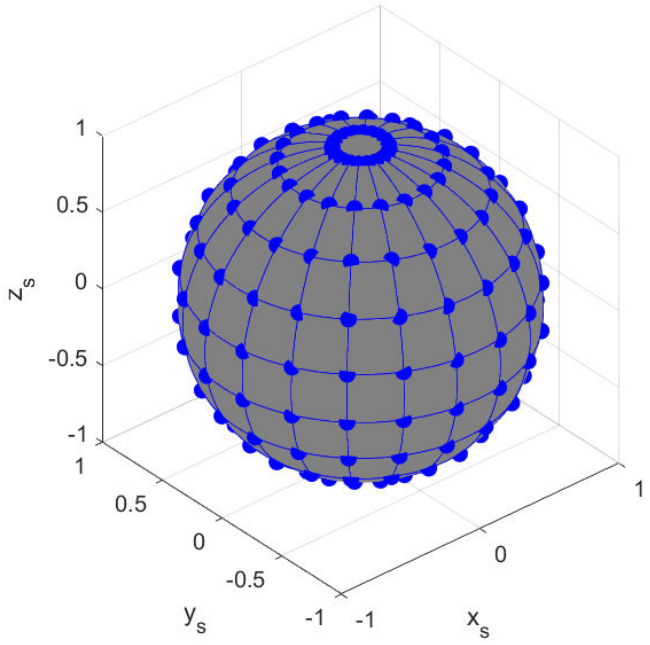


Figure 5.2: Equi-angular (i.e., non uniform) sampling on the sphere corresponding to the planar equirectangular representation. A graph is drawn on the spherical surface with vertices corresponding to pixels and edges connecting each pixel with its four closest neighbors (three closest neighbors at the poles).

function:

$$W_{i,j} = exp\left(-\frac{d_{geo}(i,j)^2}{2\theta_{geo}^2}\right) \tag{5.1}$$

for some parameters θ_{geo} , where $d_{geo}(i,j)$ represents the geodesic distance between vertices i and j capturing the sampling grid on which the vertices are defined (Figure 5.2).

As previously detailed in the previous chapters, once we have a graph and a signal defined on its vertices, the eigenvectors $\mathbf{U}$ of the Laplacian are used to define the graph Fourier transform (GFT) [92] of the signal $\mathbf{x}$ as as follows:

$$\hat{\mathbf{x}} = \mathbf{U}^{\mathrm{T}}\mathbf{x}, \tag{5.2}$$

the inverse graph Fourier transform is given by:

$$\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}. \tag{5.3}$$

A signal $\mathbf{x}$ is considered to be smooth on $\mathcal{G}$ if strongly connected vertices have similar signal values [119]. This is usually quantified in terms of the laplacian quadratic form:

$$S_2(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{L}\mathbf{x}. \tag{5.4}$$

In general, graph-based image compression methods use a graph representation as defined above, and perform a GFT to capture the main characteristics of the signal. The coefficients are then encoded in-

stead of original values. The smoother the signal on a graph (smaller $S_2(\mathbf{x})$), the more its energy is concentrated in the low frequency GFT coefficients and the more it is easily compressible.

5.3 Problem formulation

In our case, the common computational limitation of the *global graph* representation is the maximum acceptable number of vertices in the graph for GFT computation, which limits the resolution of the visual signal that can be supported.

In order to cope with the feasibility of the graph-based transform of the signal in high resolution omnidirectional images, the *global graph* $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ should be separated into several connected components, e.g. M subgraphs $\{\mathcal{G}_1, ..., \mathcal{G}_i, ..., \mathcal{G}_M\}$ by pruning some unreliable edges. The i-th subgraph is $\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i, \mathbf{W}_i\}$ where $\mathcal{V}_i$ are the vertices in the subgraph, with $|\mathcal{V}_i| = N_i < N$, $\mathcal{E}_i$ are their edges, and $\mathbf{W}_i$ is the weights matrix. $\mathbf{x}_i$ is the signal defined on the i-th subgraph. The signals on each of the subgraphs are then independently processed, and transformed separately using their respective local Laplacian $\mathbf{L}_i$.

If the topology and weights of the *global graph* are fixed, in order to obtain a good compression performance, the graph partition should be chosen such that it leads to smooth representations of the signals inside different subgraphs. On the other hand, it should also be easy to encode, since it has to be transmitted to the decoder for signal reconstruction. Our problem is therefore how to split the fixed *global graph* into connected components, so that we achieve optimal RD performance and such that all the connected components contain less than N_{max} nodes..

We first pose the problem as a rate-distortion optimization problem defined as:

$$\begin{aligned} \min_{\tilde{\mathcal{G}}=\{\mathcal{G}_i\}} \quad & \mathcal{D}(\tilde{\mathcal{G}}) + \gamma \mathcal{R}_C(\tilde{\mathcal{G}}) + \beta \mathcal{R}_B(\tilde{\mathcal{G}}) \\ \text{subject to} \quad & N_i < N_{max}, \ \forall i \end{aligned} \tag{5.5}$$

$\tilde{\mathcal{G}} = \{\mathcal{G}_i\}$ is the *global graph* based on the geometry defined a priory using Equation (5.1) , where some edges are removed. $\mathcal{D}(\tilde{\mathcal{G}})$ is the distortion between the original image and the reconstructed one, $\mathcal{R}_C(\tilde{\mathcal{G}})$ is the rate cost of the transform coefficients, and $\mathcal{R}_B(\tilde{\mathcal{G}})$ is the rate cost of the boundaries for the graph partitioning description. Each of these terms possibly depend on the chosen partition of the graph and of the coding scheme envisioned. We detail each one of them in the next section.

5.4 R-D optimized graph partitioning

5.4.1 Distortion estimation

Since the GFT is orthonormal and independant in each subgraph, the distortion term in the above problem $\mathcal{D}(\tilde{\mathcal{G}})$ is equal to the sum of distortions on all subgraphs:

$$\mathcal{D}(\tilde{\mathcal{G}}) = \sum_{i=1}^{M} \mathcal{D}(\mathcal{G}_i) = \sum_{i=1}^{M} \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 = \sum_{i=1}^{M} \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_i\|^2,$$

where x_i and $\tilde{\mathbf{x}}_i$ are the original signal and decoded signal in the i^{th} subgraph respectively. $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_{iq}$ are the original and quantized signal GFT coefficients in the i^{th} subgraph.
If we consider a uniform scalar quantizer with small quantization step q for all N coefficients, $\mathcal{D}(\tilde{\mathcal{G}})$ can be approximated by:

$$\mathcal{D}(\tilde{\mathcal{G}}) = q^2 \frac{N}{12} \tag{5.6}$$

and is thus independent from $\tilde{\mathcal{G}}$. Therefore, the optimization problem (5.5) is reduced to minimizing the rate terms.

5.4.2 Rate approximation of transform coefficients

We can evaluate the rate of the GFT coefficients $\mathcal{R}_C(\mathcal{G}_i)$ in a subgraph i using the approximation in [47]:

$$\mathcal{R}_C(\mathcal{G}_i) = S(\mathbf{x}_i) = \mathbf{x}_i^T \mathbf{L}_i \mathbf{x}_i = \sum_l \lambda_l \hat{\mathbf{x}}_{i,l}^2. \tag{5.7}$$

The parameters λ_l and $\hat{\mathbf{x}}_{i,l}$ are the eigenvalues of the local Laplacian, and the corresponding GFT coefficients of the signal $\mathbf{x}_i$ respectively. Hence, it is an eigenvalue-weighted sum of squared transform coefficients which depends on the underlying local graph $\mathbf{L}_i$.
Such proxy assumes that the bitrate of the transform coefficients increases when the smoothness of a signal on the graph decreases. While the bitrate needed to code the DC component is not captured by this approximation, we assume that it is only dependent on the number of subgraphs, which can be tuned using the N_{max} constraint in our optimization problem. The higher the N_{max}, the lower the bitrate needed to code the DC coefficients.

5.4.3 Rate approximation of the subgraphs boundaries

In fact, in our problem we impose that the pixels of the same subgraph form a connected component. Thus, a common way to code the subgraph membership is to code the boundaries. In order to approximate the coding rate of a boundary B_{ij} between two adjacent subgraphs $\mathcal{G}_i$ and $\mathcal{G}_j$, we use the 4-directional differential freeman chaincodes (*DCC*) [37] and estimate the coding rate of the boundary as its entropy computed as follows:

$$\mathcal{C}_B(ij) = -\#_l \sum_{k=1}^{4} p_k \log_2 p_k, \tag{5.8}$$

where $\#_l$ is the number of chaincodes of the boundary and $p_k, k = 1:4$ are the probabilities of each of the 4 directions.

5.4.4 Minimization of the total coding rate

Using (5.6) (5.7) and (5.8), the optimization problem in (5.5) becomes:

$$\begin{aligned} &\min_{\tilde{\mathcal{G}}=\{\mathcal{G}_i\}} \sum_{i=1}^{M} \mathbf{x}_i^T \mathbf{L}_i \mathbf{x}_i + \alpha \frac{1}{2} \sum_{i=1}^{M} \sum_{j \in \mathcal{N}_i} \mathcal{C}_B(ij), \\ &\text{s.t. } N(\mathcal{G}_i) < N_{max}, \ \forall i \end{aligned} \tag{5.9}$$

where $\mathcal{N}_i$ is the neighborhood of the subgraph $\mathcal{G}_i$. The second term is divided by 2 since we only have to code the boundary between any two neighboring regions once.
Finding this optimal partition is in general a combinatorial task, so we solve it using traditional agglomerative approximation. To initialize our optimization process, we use the Normalized Cut [91] which is well known for favoring the highest smoothness inside partitions. For that, we build a new graph $\mathcal{G}_{\text{NC}}$ with the same connectivity as $\mathcal{G}$ however with weights taking into account both the geodesic distance on the sphere and the euclidean distance in the Y space as:

$$w(i,j) = \exp\left(-\frac{d_{geo}(i,j)^2}{2\theta_{geo}^2}\right) \exp\left(-\frac{d_x(i,j)^2}{2\theta_x^2}\right). \tag{5.10}$$

Note that this graph $\mathcal{G}_{\text{NC}}$ is only used in the Normalized Cut algorithm, and do not serve as a support for GFT thus will not be transmitted. To limit the computation time, the segmentation is performed with recursive 2-way cut algorithm: at each iteration, only the first 2 eigenvectors are computed exploiting the sparse nature of the laplacians. At the output of the initialization, we have an over-segmentation with non-overlapping subgraphs $R = \{\mathcal{G}_1, ..., \mathcal{G}_i, ..., \mathcal{G}_K\}$. To model their spatial locality, we construct a subgraph neighborhood matrix $\mathbf{E}$ where $\mathbf{E}(i,j) = 1$ indicates that the subgraphs $\mathcal{G}_i$ and $\mathcal{G}_j$ are adjacent in the image. In fact, merging any two adjacent subgraphs $\mathcal{G}_i$ and $\mathcal{G}_j$ implies re-considering the connections between adjacent pixels on the boundary between them (from the *global graph*), hence removing the boundary itself. At each iteration of the merging process, we find the two adjacent subgraphs $\mathcal{G}_i^*$ and $\mathcal{G}_j^*$, which if merged, bring the most significant decrease of the criterion in (5.9) while not exceeding N_{max} nodes in the merged region. In other words,

$$\begin{aligned} \{\mathcal{G}_i^*, \mathcal{G}_j^*\} = &\max_{\mathcal{G}_i, \mathcal{G}_j \in R} \Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j) + \lambda \Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j), \\ &\text{s.t. } \mathbf{E}(i,j) = 1, \ N(\mathcal{G}_i \cup \mathcal{G}_j) < N_{max} \end{aligned} \tag{5.11}$$

where

$$\begin{aligned} \Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j) = &\ \mathbf{x}_i^T \mathbf{L}_i \mathbf{x}_i + \mathbf{x}_j^T \mathbf{L}_j \mathbf{x}^j \\ &- \begin{bmatrix} \mathbf{x}_i^T \\ \mathbf{x}_j^T \end{bmatrix} \begin{bmatrix} \mathbf{L}_i + \mathbf{D}_{ij} & \mathbf{W}_{ij} \\ \mathbf{W}_{ji} & \mathbf{L}_j + \mathbf{D}_{ji} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i & \mathbf{x}_j \end{bmatrix} \\ \Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j) = &\ \mathcal{C}_B(ij). \end{aligned}$$

$\Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j)$ and $\Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j)$ essentially capture the difference in the rate needed to code the coefficients and the rate to code the boundaries between the two regions before and after merging, respectively. If $\Delta\mathcal{R}_C(\mathcal{G}_i^*, \mathcal{G}_j^*) + \lambda\Delta\mathcal{R}_G(\mathcal{G}_j^*, \mathcal{G}_j^*) > 0$, we merge $\mathcal{G}_i^*$ and $\mathcal{G}_j^*$ into one subgraph, and repeat the process until the total rate cannot be further reduced.

In the previous formulation and subgraph merging process, we assume that all subgraphs are having the same contribution to the global rate of the whole scheme. However, in our omnidirectional image application, we are interested in giving ideally more rate to the most useful part of the signal, allowing more rate to the subgraphs occupying the biggest surface in the sphere, favoring the merging on smaller surfaces in the spherical domain. Hence, we modify the initial RD gain of Equation (5.11) adding a normalizing factor equal to the area occupied by the merged region on the sphere:

$$\{\mathcal{G}_i^*, \mathcal{G}_j^*\} = \max_{\mathcal{G}_i, \mathcal{G}_j \in R} \frac{\Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j) + \lambda\Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j)}{A_{ij}}, \tag{5.12}$$

where A_{ij} is the area on the sphere that the merged region occupies. Such normalization gives more priority to merging in the poles than in the equator. The final algorithm of the partitioning is detailed in Algorithm 2.

Algorithm 2: Rate-distortion optimized Graph partitioning for omnidirectional image coding

Data: NcutLabels, Maximum number of Nodes in a Subgraph: $\mathcal{G}_{Ncut}$, N_{max}
Result: Labels after merging $\mathcal{G}_{final}$
Initialization: $\mathcal{G} = \{\mathcal{G}_i\} = \mathcal{G}_{Ncut}$;
Construct the region neighborhood matrix $\mathbf{E}$;
Compute $\Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j) + \lambda\Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j)$ for all i, j where $\mathbf{E}(i, j) = 1$ and $N(\mathcal{G}_i \cup \mathcal{G}_j) < N_{max}$;
Repeat
  Find $\{\mathcal{G}_i^*, \mathcal{G}_j^*\}$, Eq. (5.12)
  $\mathcal{G} = \mathcal{G} \setminus \{\mathcal{G}_i^*, \mathcal{G}_j^*\} \cup \{\mathcal{G}_i^* \cup \mathcal{G}_j^*\}$;
  Update $\mathbf{E}$ based on the newly merged region
Until $\max\left(\Delta\mathcal{R}_C(\mathcal{G}_i, \mathcal{G}_j) + \lambda\Delta\mathcal{R}_B(\mathcal{G}_i, \mathcal{G}_j)\right) < 0$;
$\mathcal{G}_{final} = \mathcal{G}$;

5.4.5 Discussion and mathematical interpretation

In this section, we show how the normalized cut algorithm favors smoothness inside partitions, and then we try to justify our intuition in Equation 5.12 by drawing the problem that we are actually solving. Suppose we have an omnidirectional image represented on the sphere by a graph $\mathcal{G}_{\mathrm{NC}}$ with the same connectivity as $\mathcal{G}$ but with weights taking into account both the geodesic distance on the sphere and the euclidean distance in the Y space as in Equation (5.10).

Let's start by reviewing the main notions of the normalized cut. If we fix the number of partitions to k and we consider a partitioning $\mathcal{P} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k\}$. We have k indicator functions $z_l, l \in [1, k]$ of N entries with:

$$z_k(i) = \begin{cases} 1 & \text{if } v_i \in \mathcal{V}_k \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

The entry $z_k(i)$ indicates if a node i belongs to the partition k.

The normalized cut [91] is defined as:

$$Ncut(\mathcal{P}) = \frac{1}{2}\sum_{l=1}^{k}\left(\frac{w(\mathcal{G}_l, \bar{\mathcal{G}}_l)}{\mathrm{vol}(\mathcal{G}_l)}\right) \tag{5.14}$$

With $\mathrm{vol}(\mathcal{G}_l)$ being the sum of degrees of all nodes in $\mathcal{G}_l$, and $w(\mathcal{G}_l, \bar{\mathcal{G}}_l)$ refers to the sum of all weights joining $\mathcal{G}_l$ to any other partition in $\mathcal{P}$. Furthermore, we can observe that the Normalized cut is naturally linked to the sum of weights of edges lying within partitions:

$$\begin{aligned}
2Ncut(\mathcal{P}) &= \sum_{l=1}^{k} \left(\frac{w(\mathcal{G}_l, \bar{\mathcal{G}}_l)}{\mathrm{vol}(\mathcal{G}_l)} \right) \\
&= M - \sum_{l=1}^{k} \left(\frac{w(\mathcal{G}_l, \mathcal{G}_l)}{\mathrm{vol}(\mathcal{G}_l)} \right) \\
&= M - \sum_{l=1}^{k} \frac{1}{\mathrm{vol}(\mathcal{G}_l)} \left(\sum_{i \neq j} (w_{ij} (z_l(i) - (1 - z_l(j)))^2 \right) \\
&= M - \sum_{l=1}^{k} \frac{1}{\mathrm{vol}(\mathcal{G}_l)} \sum_{i \neq j} \left(\exp\left(-\frac{d_{geo}(i,j)^2}{2\theta_{geo}^2} \right) \exp\left(-\frac{d_x(i,j)^2}{2\theta_x^2} \right) (z_l(i) - (1 - z_l(j)))^2 \right) \\
&= M - \sum_{l=1}^{k} \frac{1}{\mathrm{vol}(\mathcal{G}_l)} \sum_{i \neq j} \left(w_{i,j}^{geo} \exp\left(-\frac{(x_i - x_j)^2}{2\theta_x^2} \right) (z_l(i) - (1 - z_l(j)))^2 \right)
\end{aligned} \tag{5.15}$$

Where M is a constant with respect to $\{z_l\}$. Thus, minimizing the normalized cut is equivalent to maximizing the non constant term:

$$\sum_{l=1}^{k} \frac{1}{\mathrm{vol}(\mathcal{G}_l)} \sum_{i \neq j} \left(w_{i,j}^{geo} \exp\left(-\frac{(x_i - x_j)^2}{2\theta_x^2} \right) (z_l(i) - (1 - z_l(j)))^2 \right), \tag{5.16}$$

with respect to z, with z being orthogonal indicator functions defined as in 5.13.

On the other side, maximizing the smoothness inside partitions can be seen as a minimization of the total variation or the laplacian quadratic form $\mathbf{x}^{\mathrm{T}}\mathbf{L}\mathbf{x}$ with $\mathbf{L}$ being the underlying graph used for a transform. In our case, $\mathbf{L}$ is dependent on the geometry and $w_{ij} = w_{geo}$.

If we consider the same partitioning $\mathcal{P}$ as before, the total variation can be re-written as a function of the indicator function z_l as:

$$\begin{aligned}
\mathbf{x}^{\mathrm{T}}\mathbf{L}\mathbf{x} &= \sum_{l=1}^{k} \mathbf{x}_l^T \mathbf{L}_l \mathbf{x}_l \\
&= \sum_{l=1}^{k} \sum_{i \neq j} \left(w_{i,j}^{geo} (x_i - x_j)^2 (z_l(i) - (1 - z_l(j)))^2 \right)
\end{aligned} \tag{5.17}$$

Using the previous equations, we can consider that the normalized cut favors smoothness inside partitions. We tend to group pixels together where the geodesic weight is high, and a small signal variation is observed. The two differences with the direct minimization of the total variation reside in the exponential term of the signal distance, and the normalization factors by the volume of partitions that are crucial for getting a more balanced partitioning. Starting from here, the surface (or area A_{ij}) term in Equation 5.12 can be interpreted as the volume of a partition although in our case, we do not consider the signal variation in computing the surface. The problem that we are actually solving by 5.12

can be considered as a minimization of a normalized version of our rate approximation (normalized total variation and boundary rate) and is henceforth equivalent to:

$$\begin{aligned} &\min_{\tilde{\mathcal{G}}=\{\mathcal{G}_i\}} \sum_{i=1}^{M} \frac{\mathbf{x}_i^{\mathrm{T}} \mathbf{L}_i \mathbf{x}_i}{A_i} + \alpha \frac{1}{2} \sum_{i=1}^{M} \sum_{j \in \mathcal{N}_i} \frac{\mathcal{C}_B(ij)}{A_i}, \\ &\text{s.t. } N(\mathcal{G}_i) < N_{max}, \ \forall i \end{aligned} \tag{5.18}$$

5.5 Experimental validation

We now move to describe how we use the above graph partitioning algorithm in an omnidirectional image compression scheme. As pointed out in the previous sections, once we solve our optimization problem, we have two kind of information to transmit to the decoder side: the GFT coefficients of the signals in all subgraphs and the description of the partitioning. The transform coefficients are quantized using a uniform quantizer with a fixed step size q for all the bands, then coded with a simple entropy coder.

In order to code the partition map, we use the arithmetic edge coder (*AEC*) proposed in [24]. The contours are first represented by differential chaincode (*DCC*) [37] and divided into segments. Then, to efficiently encode a sequence of symbols in a segment, *AEC* uses a linear regression model to estimate probabilities, which will be subsequently used in the arithmetic coder.

5.5.1 Validation of our rate proxy

Although we do not explore the true rate needed to code boundaries using *AEC* in our graph partitioning, we can show the accuracy of our proxy. During the optimization process, we compare the rate needed to code a boundary using *AEC* to our rate proxy using the entropy. Results are shown in Figure 5.3 in which the x-axis corresponds to the rate needed using *AEC*, and the y-axis corresponds to the our proposed rate proxy of $\mathcal{R}_B$. Although our rate proxy of the subgraphs' boundaries has a very small computation time with respect to AEC, the positive linear trend observed in the plot shows that it is a good approximation.

5.5.2 Coding results

We test our method on four grayscale omnidirectional images, namely *Metro*, *Pool*, *Farm* and *Hotel* shown in Figure 5.1. Each omnidirectional image is of size (512×1024).
We test two versions of our scheme that we call *WithoutGeometry* and *WithGeometry*. In the first version, the geodesic distance is not taken into account which comes down to set $\theta_{geo} = \infty$ in the construction of graphs $\mathcal{G}$ and $\mathcal{G}_{\mathrm{NC}}$. Moreover, the merging is done as explained in Equation (5.11). On the other hand, the second version corresponds to our detailed scheme of the previous section taking into account the geometrical information in all stages: normalized cut, merging with equation (5.12) and transform coding. To evaluate the compression performance, we compute the PSNR in three different domains: the equirectangular domain between original and decoded omnidirectional images, in the spherical domain

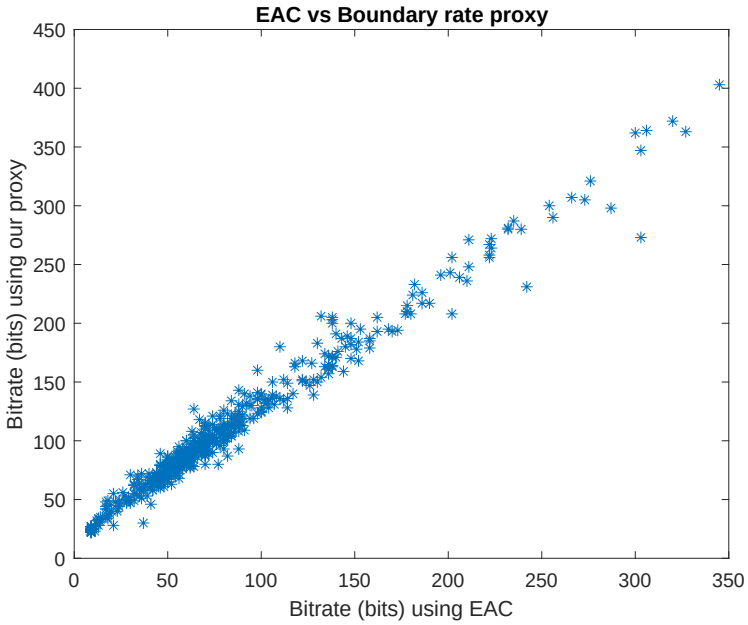


Figure 5.3: Accuracy of the our rate proxy. x-axis: rate needed to code a boundary using *EAC*. y-axis: rate computed using our proxy

after performing a uniform sampling of the spherical surface[112], and in the cube-map domain.
Figure 5.4 shows an example of subgraphs obtained using *WithGeometry* scheme after fixing λ to 600 for the *Pool* image, in the equirectangular and the spherical domains. It is clear that subgraphs are adhering to the objects borders in both domains and larger subgraphs are formed around the poles.
Results in Figures 5.8 and 5.9 show that *WithGeometry* leads to a better rate-distortion performance in all domains, compared to *WithoutGeometry*. There is two major explanations of this behavior. First, the global graph in the first case is more adapted to the omnidirectional signal: more specifically, the signal values in horizontally adjacent pixels around the poles are assumed to be more correlated than those which are horizontally adjacent in the equator. This is the case of most of omnidirectional images where poles usually consist of the floor or the sky. In practice, some images like *Metro* do not totally follow this assumption which explains the comparable performance observed for the two schemes. A second explanation is that in the *WithGeometry* case, the total rate is allocated more carefully taking into account the area occupied on the sphere. Furthermore, the obtained results show that our proposed schemes outperform classical DCT transform coding scheme in JPEG especially in the low bitrate range, although they can be further improved by optimizing the coding step namely in the quantization and arithmetic coding parts.

5.6 Conclusion

In this chapter, we have proposed a new graph-based framework for omnidirectional image compression. We introduced a new R-D optimized graph partitioning to cope with the feasibility of graph

(a)

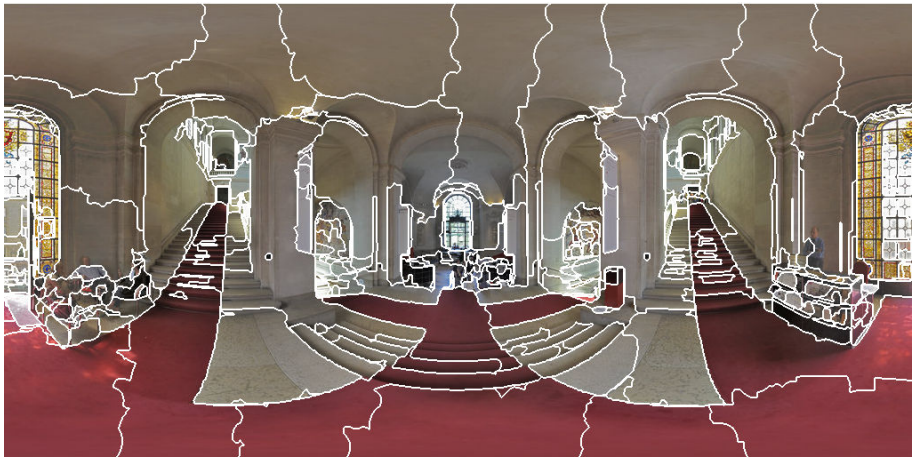
(b)

Figure 5.4: Graph partitions represented in the equi-angular domain (a) and in the spherical domain (b)

(a)

(b)

Figure 5.5: Graph partitions represented in the equi-angular domain (a) and in the spherical domain (b)



(a)

(b)

Figure 5.6: Graph partitions represented in the equi-angular domain (a) and in the spherical domain (b)

(a)

(b)

Figure 5.7: Graph partitions represented in the equi-angular domain (a) and in the spherical domain (b)

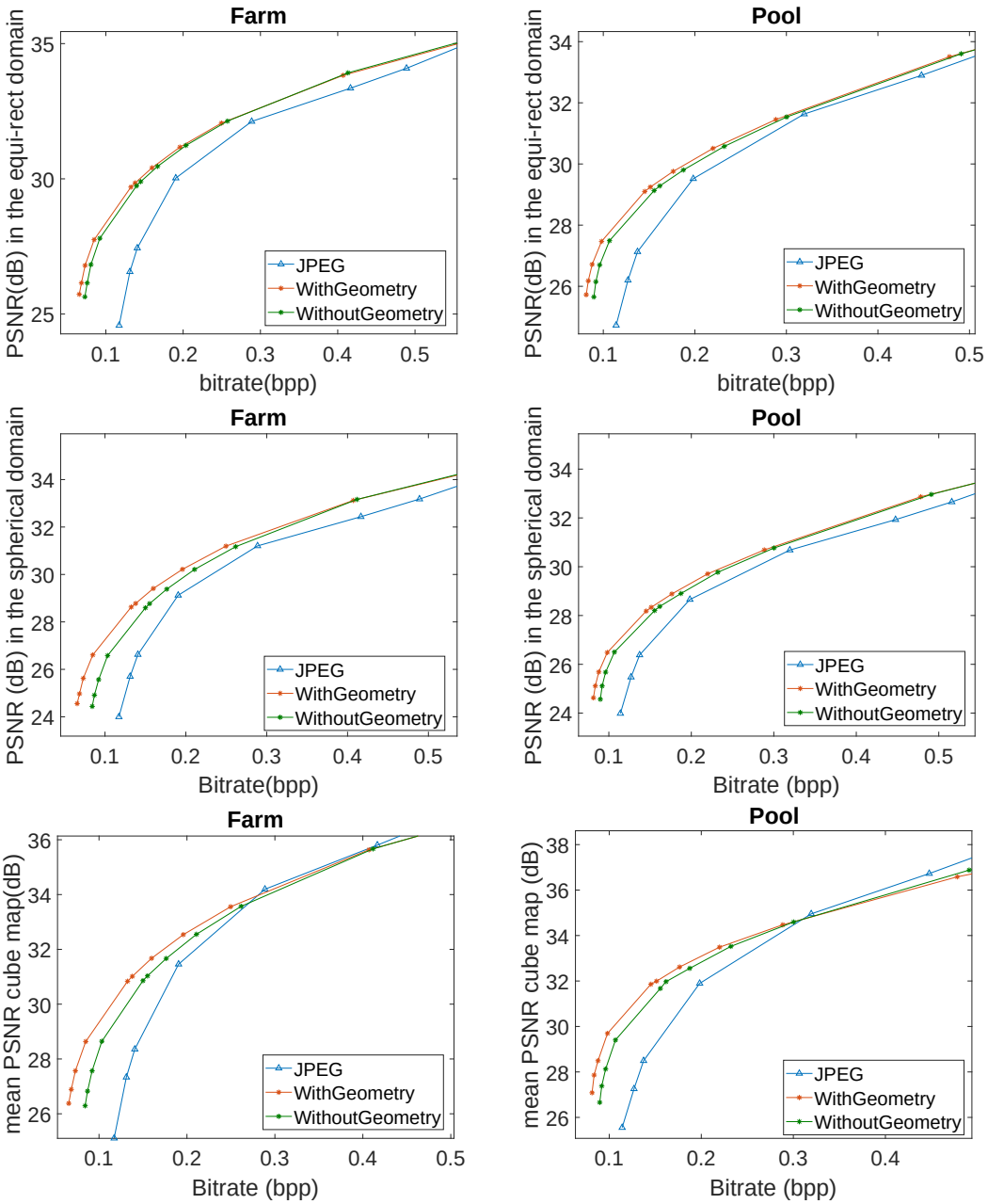


Figure 5.8: Rate-distortion comparison. 1^{st} row: performance in the equirectangular domain. 2^{nd} row : performance in the Spherical domain. 3^{rd} row : performance in the Cubemap domain

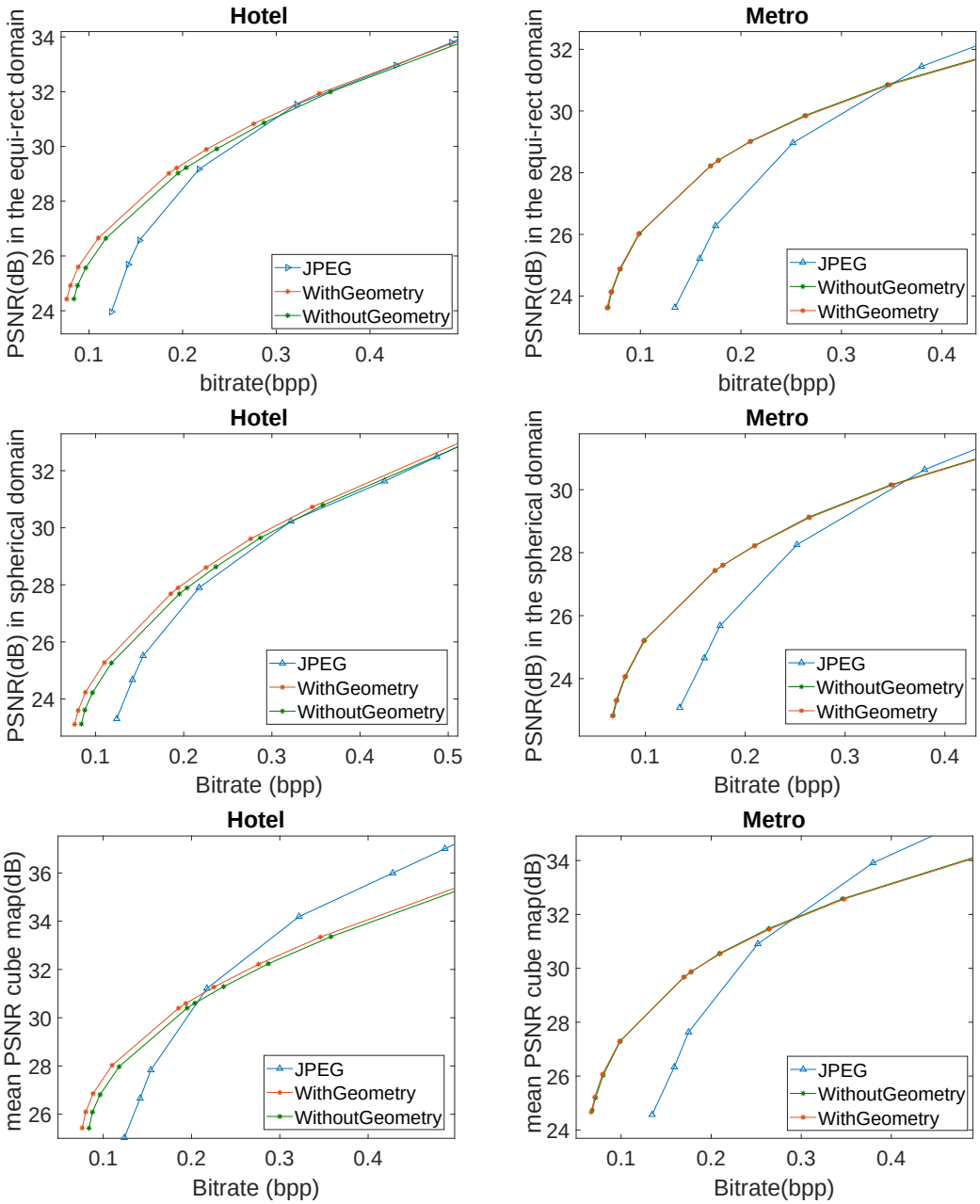


Figure 5.9: Rate-distortion comparison. 1^{st} row: performance in the equirectangular domain. 2^{nd} row : performance in the Spherical domain. 3^{rd} row : performance in the Cubemap domain

Fourier Transform on global graphs defined on high resolution images. The partition obtained provides an effective tradeoff between the smoothness of signals inside subgraphs and the cost of coding the partition description. Also, we showed that our methods outperform traditional DCT coding schemes at low bitrates. As future work, we investigate the use of different forms of laplacians and focus on the adpatation of the quantization and other coding tools, which may lead to further improvements to the coding performance. Comparison with traditional coding of planar representations other than the equirectangular one [13], as well as analysis of RD performance on higher resolution test material, will also be performed.

Chapter 6

Graph based transforms under statistical uncertainties

6.1 Introduction

Suppose we want to provide a more compact representation of a signal $\mathbf{x}$ living on a nodes set $\mathcal{V}$ embedded in a 2D (images) or 3D euclidean domain (point clouds or 3D scenes).

As already mentioned in the previous chapter, an intuitive way to define the graph Adjacency matrix $\mathbf{A}$ is to use the neighborhood structure of the underlying domain. For example, one pixel is connected to its four or eight nearest neighbors in an 2D image, or a 3D point is connected to its neighbors within a specific distance in the 3D scene. Using $\mathbf{A}$, an unweighted Laplacian or any alternative topology-based matrix $\mathcal{L}_a$ can be further computed. The eigenvectors matrix $\mathbf{\Psi} = [\psi_1 \psi_2 ... \psi_n]$ of the laplacian serves as the unweighted Graph Fourier Transform (*uGT*) basis.

One other approach is to model the vertices similarities and thus characterize the signal by a weight matrix $\mathbf{W}$ where each weight is either learned under some probabilistic assumptions or heuristically defined as a function of distance measures (as done for example in Chapter 5 in the *With Geometry* case). A lot of weights models are adopted in the literature such as inverse-distance, auto-regressive and squared exponential. [15][115][69]. The weight matrix is then used to compute the precision matrix of the signal as the weighted Laplacian or any alternative model based matrix $\mathbf{Q} = \mathcal{L}_{\mathbf{w}}$ [69][115]. The precision matrix is mainly useful for characterizing a stationary Gaussian Markov Random Field (GMRF), in which case the precision matrix will have only a finite number of non-zero elements. Besides, another way is to suppose that the signal is a Gaussian Process and directly compute an approximated covariance matrix $\mathbf{\Sigma}$ by assigning to each element $\mathbf{\Sigma}(i,j)$ between the nodes i and j a function of the distance such as the Ornstein Uhlenbeck or the squared exponential functions, or an approximation learned from training data block [15]. The eigenvectors matrix $\mathbf{\Phi} = [\phi_1 \phi_2 ... \phi_n]$ of the assumed model precision or covariance matrices form the transform basis of what we call: a Model-Based Graph Transform (*mGT*).

In practice, the signals that we are dealing with might not exactly follow the data models we assume. After all, statistically speaking, we almost never have enough observations to compute an optimal data model. For example, a high weight learned or assumed for the edge $\mathcal{E}\{i,j\}$ between neighboring nodes i and j in an image, does not necessarily mean that the signal values on the two nodes are highly correlated. An edge appearing at object boundaries leads to two different pixel values on two adjacent vertices. Typically, uncertainty appears when the weights of the true signal model, i.e the covariance of the signal in hand, differ from the assumed or learned ones.

In this chapter, we aim at understanding the effect of this uncertainty on the compression efficiency

of both the transforms evoked previously: *uGT* and *mGT*. Specifically, we want how much uncertainty can the model-based transform handle while being more efficient than the topology-based transform?

6.2 When does the model based transform outperform the topology based transform?

The true distribution of the signal $\mathbf{X}$ residing on the nodes of the graph $\mathcal{G}$ can be thought as a multivariate normal distribution or Gaussian Markov Random Field $\mathcal{N}(\mu, \hat{\mathbf{\Sigma}})$ with mean μ and covariance $\hat{\mathbf{\Sigma}}$.

Intuitively, in order to achieve the best decorrelation of the signal, the assumed graph model $\mathbf{W}$ should well approximate the true model $\hat{\mathbf{W}}$ of the graph signals, since undoubtedly, the true graph which stands for the true precision matrix has edges and weights that are consistent with inter-node correlations. Defining $\mathbf{W} = \hat{\mathbf{W}}$ leads to signal smoothness with respect to the graph and to the highest energy compaction in the low frequencies. However, with real data, this is generally not the case, and we mostly deal with uncertain models specifying signal distributions as $\mathcal{N}_{\mathcal{M}}(\mu_M, \mathbf{\Sigma} = (\mathcal{L}_{\mathbf{w}} + \delta\mathbf{I})^{-1})$ [1]. with $\hat{\mathbf{\Sigma}} = \mathbf{\Sigma} + \xi$. Suppose that we are dealing with distributions with the same mean $\mu_M = \mu$.

Moreover, a reliable topology $\mathbf{A}$ has only relevant information related to the edge positions. It tells whether the signal value on node i is conditionally dependent or not of the signal on node j given all the other values, without giving any additional information on the strength of this dependency. The graph topology specifies a signal distribution $\mathcal{N}_{\mathcal{L}}(\mu_L, \mathbf{\Sigma_L} = (\mathcal{L}_{\mathbf{a}} + \delta\mathbf{I})^{-\mathbf{1}})$.

To examine when relying on the topology outperforms the model based transforms in terms of energy compaction and decorrelation efficiency, we will use the *Kullback-Leibler (KL)* divergence as a metric to measure how close the two transforms are to the best energy compaction. The use of such metric is justified by two facts: First, it has always been used in graph learning [30] [28]. The minimization of the KL divergence is equivalent to maximizing the log-likelihood and thus leads to compact representations which is a desirable property of signal transforms. Second, it theoretically measures the expected number of extra bits required to code samples from a true distribution P using a code optimized for Q rather than the code optimized for P.

We start by finding the expressions of the two *KL* divergences that we are interested in. The expressions in Equations 6.1 and 6.2 denote the KL divergences between the true model distribution $\mathcal{N}$ and either the assumed model $\mathcal{N}_{\mathcal{M}}$ or the model specified by the graph topology $\mathcal{N}_{\mathcal{L}}$, respectively.

[1] δ is a small positive value to preserve positive definitiveness. The eigenvectors of $\mathcal{L}_{\mathbf{a}}$ and $\mathcal{L}_{\mathbf{a}} + \delta\mathbf{I}$ are the same, only the eigenvalues σ_i of the latter are equal to a shifted version of the eigenvalues values of the former $\lambda_i + \delta$

$$
\begin{aligned}
D_{KL}(\mathcal{N}||\mathcal{N}_{\mathcal{M}}) &= \frac{1}{2}\Bigg(\mathrm{Tr}(\mathbf{\Sigma^{-1}}\hat{\mathbf{\Sigma}}) + (\mu_M - \mu)\mathbf{\Sigma}^{-1}(\mu_M - \mu) \\
&\quad - N + \log\Big(\frac{|\mathbf{\Sigma}|}{|\hat{\mathbf{\Sigma}}|}\Big)\Bigg) \\
&= \frac{1}{2}\Bigg(\mathrm{Tr}(\mathbf{\Sigma^{-1}}\hat{\mathbf{\Sigma}}) - \mathrm{Tr}(\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{\Sigma}}) - \log\Big(|\mathbf{\Sigma}^{-1}\hat{\mathbf{\Sigma}}|\Big)\Bigg) \\
&= \frac{1}{2}\Bigg(\mathrm{Tr}\Big((\mathcal{L}_{\mathbf{w}} + \delta\mathbf{I} - \hat{\mathbf{Q}})\hat{\mathbf{\Sigma}}\Big) \\
&\quad - \log\Big(|(\mathcal{L}_{\mathbf{w}} + \delta\mathbf{I})\hat{\mathbf{\Sigma}}|\Big)\Bigg)
\end{aligned}
\tag{6.1}
$$

$$
\begin{aligned}
D_{KL}(\mathcal{N}||\mathcal{N}_{\mathcal{L}}) &= \frac{1}{2}\Bigg(\mathrm{Tr}(\mathbf{\Sigma_L^{-1}}\hat{\mathbf{\Sigma}}) + (\mu_L - \mu)\mathbf{\Sigma_L}^{-1}(\mu_L - \mu) \\
&\quad - N + \log\Big(\frac{|\mathbf{\Sigma_L}|}{|\hat{\mathbf{\Sigma}}|}\Big)\Bigg) \\
&= \frac{1}{2}\Bigg(\mathrm{Tr}(\mathbf{\Sigma_L^{-1}}\hat{\mathbf{\Sigma}}) - \mathrm{Tr}(\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{\Sigma}}) - \log\Big(|\mathbf{\Sigma_L}^{-1}\hat{\mathbf{\Sigma}}|\Big)\Bigg) \\
&= \frac{1}{2}\Bigg(\mathrm{Tr}\Big((\mathcal{L}_{\mathbf{a}} + \delta\mathbf{I} - \hat{\mathbf{Q}})\hat{\mathbf{\Sigma}}\Big) \\
&\quad - \log\Big(|(\mathcal{L}_{\mathbf{a}} + \delta\mathbf{I})\hat{\mathbf{\Sigma}}|\Big)\Bigg)
\end{aligned}
\tag{6.2}
$$

Where Tr stands for the trace operator, N is the number of nodes of the graph $\mathcal{G}$ and $|\mathbf{A}|$ is the determinant of a matrix $\mathbf{A}$.

Theoretically, a better transform performance in terms of energy compaction and a lower extra cost is expected using the code optimized for the distribution with the smallest *KL* divergence with respect to the true model distribution. So a sufficient condition for the Model-based Transform(*mGT*) to outperform the Graph-based Transform(*uGT*) is that $D_{KL}(\mathcal{N}||\mathcal{N}_{\mathcal{M}}) \leq D_{KL}(\mathcal{N}||\mathcal{N}_{\mathcal{L}})$. Applying this condition to Equations 6.1 and 6.2,and replacing $\hat{\mathbf{\Sigma}}$ by $\mathbf{\Sigma} + \xi$ we get the following.

$$
\begin{aligned}
D_{KL}(\mathcal{N}||\mathcal{N}_{\mathcal{M}}) &\leq D_{KL}(\mathcal{N}||\mathcal{N}_{\mathcal{L}}) \\
\frac{1}{2}\Bigg(\mathrm{Tr}\Big((\mathcal{L}_{\mathbf{w}} + \delta\mathbf{I} - \hat{\mathbf{Q}})\hat{\mathbf{\Sigma}}\Big) - \log\Big(|(\mathcal{L}_{\mathbf{w}} + \delta\mathbf{I})\hat{\mathbf{\Sigma}}|\Big)\Bigg) &\leq \\
\frac{1}{2}\Bigg(\mathrm{Tr}\Big((\mathcal{L}_{\mathbf{a}} + \delta\mathbf{I} - \hat{\mathbf{Q}})\hat{\mathbf{\Sigma}}\Big) - \log\Big(|(\mathcal{L}_{\mathbf{a}} + \delta\mathbf{I})\hat{\mathbf{\Sigma}}|\Big)\Bigg) & \\
\mathrm{Tr}\Big((\mathcal{L}_w - \mathcal{L}_a)(\mathbf{\Sigma} + \xi)\Big) &\leq -\log\det\Big(\frac{\mathcal{L}_a + \delta\mathbf{I}}{\mathcal{L}_w + \delta\mathbf{I}}\Big) \\
\mathrm{Tr}\Big((\mathcal{L}_w - \mathcal{L}_a)\xi\Big) &\leq \mathrm{Tr}((\mathcal{L}_a + \delta\mathbf{I})\mathbf{\Sigma} - \mathbf{I}) - \log\det\Big(\frac{\mathcal{L}_a + \delta\mathbf{I}}{\mathcal{L}_w + \delta\mathbf{I}}\Big)
\end{aligned}
\tag{6.3}
$$

The left term of inequation 6.3 can be equivalently written as a multiplication of the vectorized forms

[2] of the two matrices $(\mathcal{L}_w - \mathcal{L}_a)^T$ and ξ , where T denotes the transpose operator. Using the symmetry property of $(\mathcal{L}_w - \mathcal{L}_a)$ and since ξ is a centered noise, we get the following.

$$\begin{aligned} vec(\mathcal{L}_w - \mathcal{L}_a)vec(\xi) &\leq \mathrm{Tr}\left((\mathcal{L}_a + \delta\mathbf{I})\mathbf{\Sigma} - \mathbf{I}\right) - \log\det\left(\frac{\mathcal{L}_a + \delta\mathbf{I}}{\mathcal{L}_w + \delta\mathbf{I}}\right) \\ N^2 E\Big((\mathcal{L}_w - \mathcal{L}_a)\xi\Big) &\leq \mathrm{Tr}\left((\mathcal{L}_a + \delta\mathbf{I})\mathbf{\Sigma} - \mathbf{I}\right) - \log\det\left(\frac{\mathcal{L}_a + \delta\mathbf{I}}{\mathcal{L}_w + \delta\mathbf{I}}\right) \\ N^2\Big(Cov(\mathcal{L}_w - \mathcal{L}_a, \xi) + E(\xi)E(\mathcal{L}_w - \mathcal{L}_a)\Big) &\leq \\ &\mathrm{Tr}\left((\mathcal{L}_a + \delta\mathbf{I})\mathbf{\Sigma} - \mathbf{I}\right) - \log\det\left(\frac{\mathcal{L}_a + \delta\mathbf{I}}{\mathcal{L}_w + \delta\mathbf{I}}\right) \\ N^2 Cov(\mathcal{L}_w - \mathcal{L}_a, \xi) &\leq \mathrm{Tr}\left((\mathcal{L}_a + \delta\mathbf{I})\mathbf{\Sigma} - \mathbf{I}\right) - \log\det\left(\frac{\mathcal{L}_a + \delta\mathbf{I}}{\mathcal{L}_w + \delta\mathbf{I}}\right) \end{aligned} \tag{6.4}$$

Where E denotes the expectation and N is the total number of nodes in the graph. Since the covariance of two random variables is equal to their correlation multiplied by the product of their standard deviations, we can get a sufficient condition on the uncertainty ξ as

$$\begin{aligned} \sigma_\xi corr(\xi, \mathcal{L}_w - \mathcal{L}_a) &\leq \frac{\mathrm{Tr}\left((\mathcal{L}_a + \delta\mathbf{I})\mathbf{\Sigma} - \mathbf{I}\right) - \log\det\left(\frac{\mathcal{L}_a + \delta\mathbf{I}}{\mathcal{L}_w + \delta\mathbf{I}}\right)}{N^2\sigma_{\mathcal{L}_w - \mathcal{L}_a}} \\ \sigma_\xi corr(\xi, \mathcal{L}_w - \mathcal{L}_a) &\leq \frac{2D_{KL}(\mathcal{N}_\mathcal{M}||\mathcal{N}_\mathcal{L})}{N^2\sigma_{\mathcal{L}_w - \mathcal{L}_a}} \\ f(\xi) &\leq \mathcal{B} \end{aligned} \tag{6.5}$$

The second line of the inequality can be found by noticing that $\mathrm{Tr}(\mathbf{I}) = \mathrm{N}$ and that the resulting term of the nominator is hence equal to double of the $D_{KL}(\mathcal{N}_\mathcal{M}||\mathcal{N}_\mathcal{L})$.

6.3 Discussion

The decision to "whether use a model-based or a topology-based transform" is now well defined in Eq. 6.5 as a function of the model uncertainty. The bound $\mathcal{B}$ shows that the ability of Model Based Transforms to handle uncertainty is dependent on both the graph where the signal resides(in the term $\mathcal{L}_a$), and the assumed model itself (in the terms $\mathcal{L}_w$ and $\mathbf{\Sigma}$). Interestingly, the theoretical limit is very dependent on the *KL* divergence between the two distributions defined by the topology and the model (in the term $D_{KL}(\mathcal{N}_\mathcal{M}||\mathcal{N}_\mathcal{L})$). Since the *KL* divergence is a positive term, then the resulting theoretical limit $\mathcal{B}$ is always positive.

The correlation existing between the noise matrix (ξ) and the mismatch between the model and the topology ($\mathcal{L}_w - \mathcal{L}_a$) is also a factor that plays a very important role. If the correlation is negative, then we are in a case where the model based transform is always outperforming a topology based transform no matter how high the level of uncertainty is. On the other hand, if the noise and the mismatch are

[2] We mean by vectorized form stacking the columns of the matrix in a vector.

positively correlated, then as much as the correlation increases, we tend to have a lower bound making the model-based transform less resistant to uncertainties.

6.4 Experimental validation

6.4.1 Experimental setup

In our experiments, we follow the steps depicted in Figure 6.1.

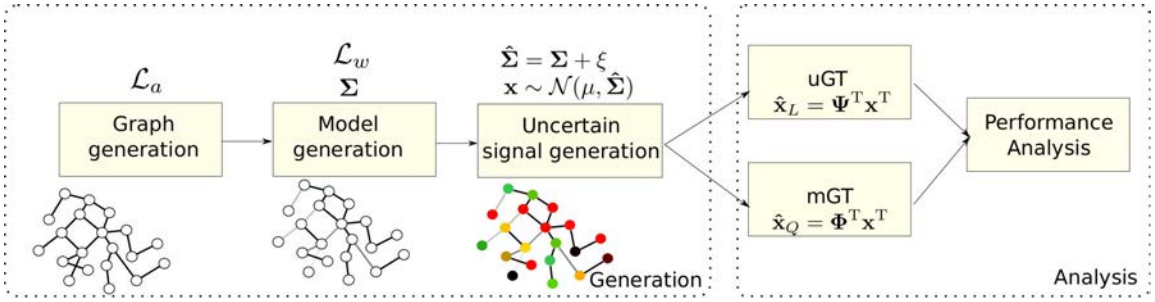


Figure 6.1: Followed scheme in order to study the impact of the model uncertainty on both *uGT* and *mGT*.

The signal $\mathbf{x}$ is an $n \times m$ matrix made of n columns of m values corresponding to the m observations of the n different variables $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n$ assigned to the nodes of the graph $\mathcal{G}$. Two transforms are applied leading to two different transformed signals: on one hand, the signal $\mathbf{x}$ is projected into the eigenspace of the unweighted normalized graph Laplacian as in equation 6.6. This makes up the *uGT* transform. On the other hand, the Model based transform (*mGT*) consists of projecting the signal into the eigenspace of the assumed model precision matrix or covariance matrix as in equation 6.7.

$$\hat{\mathbf{x}}_L = \mathbf{\Psi}^{\mathrm{T}}\mathbf{x}^{\mathrm{T}} \tag{6.6}$$

$$\hat{\mathbf{x}}_Q = \mathbf{\Phi}^{\mathrm{T}}\mathbf{x}^{\mathrm{T}} \tag{6.7}$$

Note that the signal of interest $\mathbf{x}$ has a true covariance matrix $\hat{\mathbf{\Sigma}}$ which, more likely will be different from the previously mentioned model covariance matrix and also does not necessarily fall in the space spanned by the eigenvectors of the unweighted normalized Laplacian of the underlying graph. We will model this difference as a gaussian centered noise added to the model covariance $\mathbf{\Sigma}$ (Equation 6.8). This choice can be justified by the fact that if we are wrong about high weights at $z1$ positions $\{p_1, p_2 \dots p_{z1}\}$, most probably we would also be wrong about low weights at another $z2$ positions $\{q_1, q_2 \dots q_{z2}\}$. The extra value given to the edges in the first set would be required to rectify low weights assigned to edges in the second set.

$$\hat{\mathbf{\Sigma}} = \mathbf{\Sigma} + \xi \tag{6.8}$$

6.4.2 Synthetic topologies and models

We perform different tests on synthetic data with various weights distributions while also changing the degrees of our graphs. More precisely, the construction of the graph $\mathcal{G}$ is done in two steps. We first determine the graph structure (i.e connectivity). We generate regular graphs with different degrees d, where each node is exactly connected to d other nodes. In the second step, edge weights related to the assumed models are randomly selected based on three different distributions:

1. **Gaussian distribution** with a mean μ_w and variance σ_w^2
2. **Uniform distribution** drawn between two values w_a and w_b
3. **Bimodal distribution** centered on two different values μ_a and μ_b

The selection of those options can be intuitively explained. For example, the use of a regular graph and bimodal weight distribution does intuitively model the case when depth values lying on a 2D grid are compressed assuming some bimodal weights. Each edge is assigned a small or high weight depending whether there is an border detected in the depth map or not to model the degree of dependency between the connected pixels.

Uniform and Gaussian weights appear when the model is highly heterogeneous or approximately homogeneous respectively. To illustrate the case, consider an inverse distance model where each weight W_{ij} is assigned a value of $\frac{1}{d_{ij}}$ where d_{ij} is the euclidean distance between two nodes i and j. If in the considered connected node set, the number of $\{i, j\}$ pairs is the same for all d in the distance range, then we have a uniform weight distribution. On the contrary, if the distance is more or less constant between connected nodes then the weights follow a Gaussian distribution.

Once the graph has been created and model weights have been assigned, the signals to be entering the transform blocks of *mGT* and *uGT* are computed as random observations of a Gaussian Markov Random Field defined by its precision matrix $\hat{\mathbf{Q}}$ with modified weights $\hat{\mathbf{W}}$ that are generated following the method described below.

Uncertainty generation

A fair method to generate the uncertainty of the signal covariance is to modify the precision matrix elements corresponding to edges positions in the graph before inversion hence restricting the uncertainty to the normalized weights $\mathbf{W}$ assigned to graph edges instead of adding or removing edges. The uncertain covariance matrix can be directly computed as the inverse of the noisy precision matrix $\hat{\mathbf{Q}} = \mathbf{I} - \hat{\mathbf{W}}$. More precisely, we propose to modify the distribution of the non-zeros elements of the model normalized weights matrix $\mathbf{W}$ by adding a controlled zero-mean Gaussian noise $\boldsymbol{\epsilon}$. We first create n random unit vectors $\{\mathbf{u}_1\mathbf{u}_2 \dots \mathbf{u}_n\}$ of dimension $d = 25$ and create the $d \times n$ matrix where each column corresponds to one unit vector. The noise is then generated as in Equation 6.9.

$$\begin{aligned} \hat{\mathbf{W}} &= \mathbf{W} + \boldsymbol{\epsilon}; \\ \boldsymbol{\epsilon} &= \beta(\mathbf{U}^{\mathbf{T}}\mathbf{U} \cdot \mathbf{A}) \end{aligned} \tag{6.9}$$

Where β is a value that serves to tune the noise variance, $\mathbf{A}$ is the graph Adjacency matrix and $(\cdot)$ denotes the dot product between two matrices. The resulting noise is restricted to the non-zero elements of the Laplacian matrix. The unreliable weights are then shifted and rescaled to preserve the same mean and variance as original normalized weights as in Equation 6.10.

$$\hat{\mathbf{W}}_{ij} = \frac{\sigma^2_{\mathbf{W}}}{\sigma^2_{\mathbf{W}} + \sigma^2_{\boldsymbol{\epsilon}}}(\mathbf{W}_{ij} + \boldsymbol{\epsilon}_{ij} - \mu_{\mathbf{W}}) + \mu_{\mathbf{W}} \tag{6.10}$$

$\sigma^2_{\mathbf{W}}$ and $\sigma^2_{\boldsymbol{\epsilon}}$ denote the variances of the normalized weights and of the added noise ϵ respectively. $\mu_{\mathbf{W}}$ is the mean of the normalized weights. By changing β, we can generate uncertain models with different covariance matrices each corresponding to a certain level of uncertainty on the weights, or equivalently to a certain level of unreliability on the covariance matrix elements.

Transform efficiency metrics

To quantify the efficiency of the transforms and practically study the impact of uncertainty, we will use two measures namely the decorrelation efficiency and the transform coding gain which are directly related to the energy compaction property. The decorrelation efficiency η_c compares the sum of absolute off-diagonal terms in the correlation matrix of the signal before (z_x) and after the transform ($z_{\hat{x}}$) as in Equation 6.11. A higher value of η_c denotes a higher decorrelation efficiency.

$$\eta_c = 1 - \frac{z_{\hat{x}}}{z_x} \tag{6.11}$$

The transform coding gain TC is defined from the variances of the transformed signal coefficients as

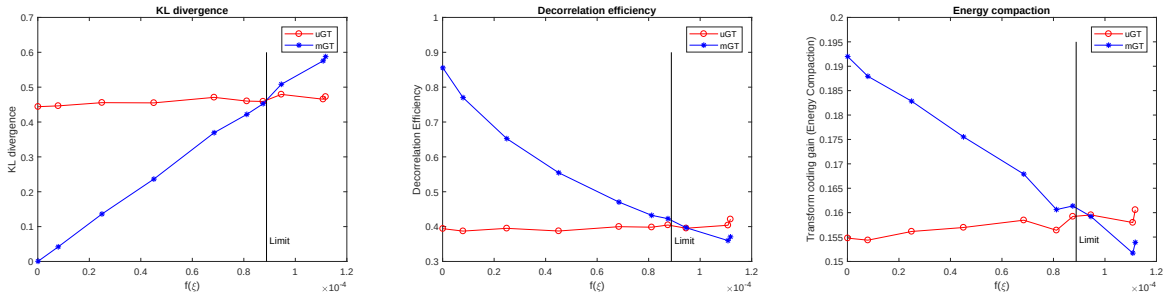
$$TC = 10\log_{10}\left(\frac{\frac{1}{N}\sum_{i=1}^{N}\sigma_i^2}{\left(\prod_{i=1}^{N}\sigma_i^2\right)^{\frac{1}{N}}}\right) \tag{6.12}$$

where N denotes the number of signal coefficients and σ_i refers to the variance of the i^{th} transformed coefficient. The transform with the higher coding gain packs more energy into a fewer number of coefficients.

KL divergence: a reasonable metric

In order to see how the *KL* divergence relates to the coding efficiency, we first perform different tests and plot the *KL divergence* as well as the decorrelation efficiency and transform coding gain as a function of the SNR of the covariance matrix elements. For a clear illustration, we only show an example with a regular graph consisting of $N = 100$ nodes with $deg = 8$. Weights are drawn from a bimodal distribution centered on 0.2 and 0.7. Results of Figure 6.2 show that the *KL divergence* is a good metric for denoting the energy compaction and decorrelation properties of a transform; the crossing point of the *mGT* and *uGT* curves is roughly the same ($f(\xi)$ around 0.9×10^{-4}) in the three plots 6.2a, 6.2b and 6.2c. It also appears that the *uGT* efficiency is constant with different amount of uncertainty as long as the mean and the variance of the true signal model does not vary. This can be mostly justified by the fact that the uncertain weights have the same statistical moments as those associated with the true data model. On

the other hand, the transform coding gain of the Model-based transform is getting lower when the $f(\xi)$ increases.



(a) (b) (c)

Figure 6.2: Impact of the uncertainty when the graph consists of 100 nodes, is regular with d = 8, and the weights follow a bimodal distribution centered around 0.2 and 0.7. The Limit shown in the plot is computed as in Eq. 6.5

Impact of the graph degree and the data statistical model on the uncertainty theoretical bound

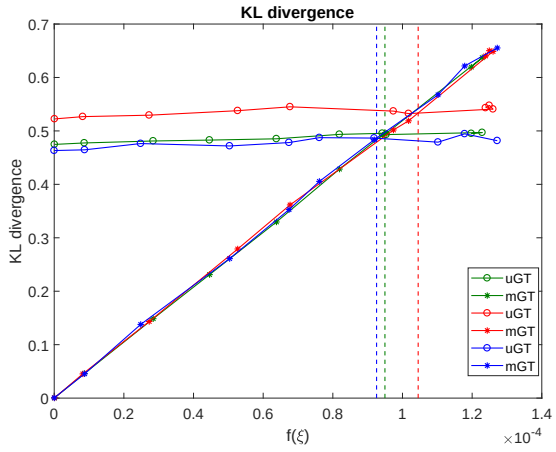
For a comprehensive evaluation of our theoretical bound $\mathcal{B}$ found in section 6.2 and in order to provide a fair study of the impact of the graph structure (degree) and data model on this bound, we will consider three different scenarios.

For the first scenario, we fix the graph degree and the two first statistical moments of the precision model weights. We then generate them from the three different distributions. For the second scenario, we fix the graph degree, the statistical distribution and mean of the weights while only varying the weight variance. For the last scenario, we fix the model i.e. the weights statistical distribution, mean and variance and vary the graph degree. Results are shown in Figure 6.3, 6.4 and 6.5 respectively for the three scenarios.

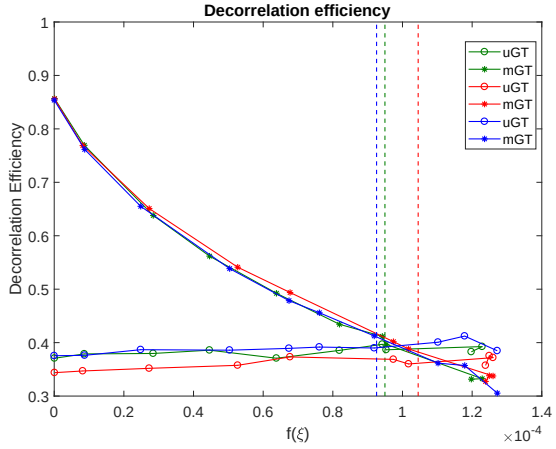
For the sake of clarity, in the following analysis, we are mainly interested in the crossing point (the bound) and the gap between the two curves of a same color (representing both transform *uGT* and *mGT*) in the plots. However, comparing the curves with different colors does not make sense since we are not dealing with same references.

As shown in Figure 6.3, the unweighted graph transform is performing more closely to *mGT* in the case when the true precision model weights follow a Gaussian distribution. The uniform distribution is the case when it has the lowest performance compared to the *mGT*. The bimodal distribution lies between the two. Furthermore, the crossing point between the two transform efficiencies corresponds exactly to our previous analysis and theoretical bound. Our theoretical limit $\mathcal{B}$ provides a sufficient condition for the Model-Based transform to outperform the unweighted Graph Transform.

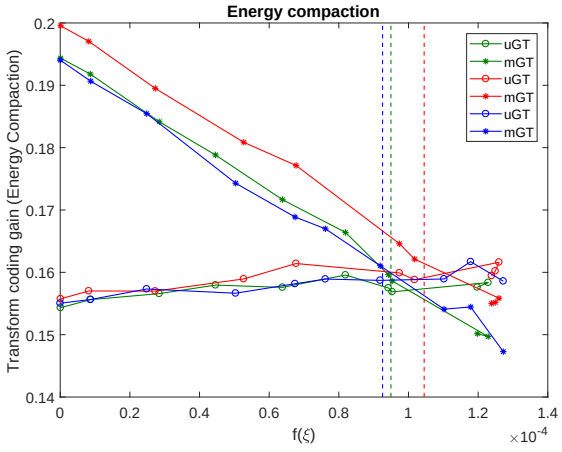
The results of the second scenario in Figure 6.4 provide a clear evidence that the ability of the unweighted graph transform to compete with the *mGT* in terms of transform coding efficiency is highly dependent on the variance of the precision model weights. A higher variance results in a larger gap between the *uGT* and *mGT* decorrelation efficiency and transform coding gain (Figure 6.4b , 6.4c). This



(a)

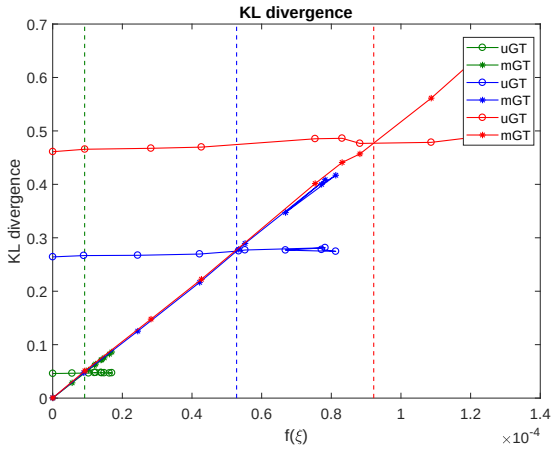


(b)

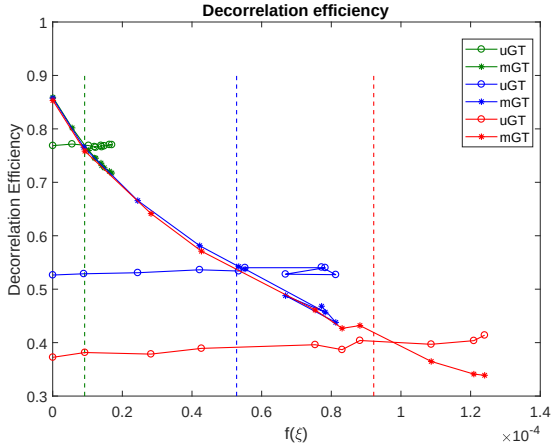


(c)

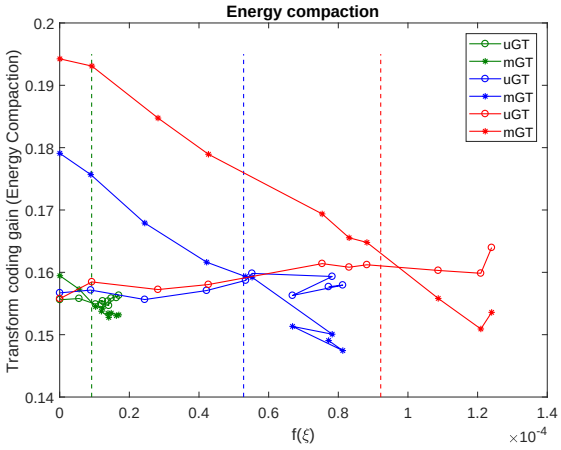
Figure 6.3: Impact of the uncertainty when the graph consists of 100 nodes, is regular with $deg = 8$, and the weights follow uniform(red), bimodal(green) or gaussian(blue) distributions with a fixed variance 0.0671 and mean 0.45



(a)

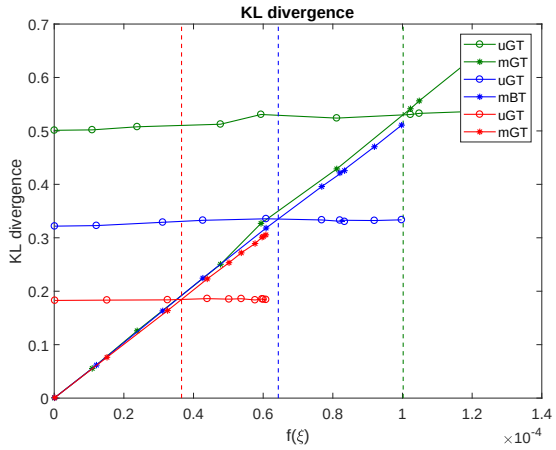


(b)

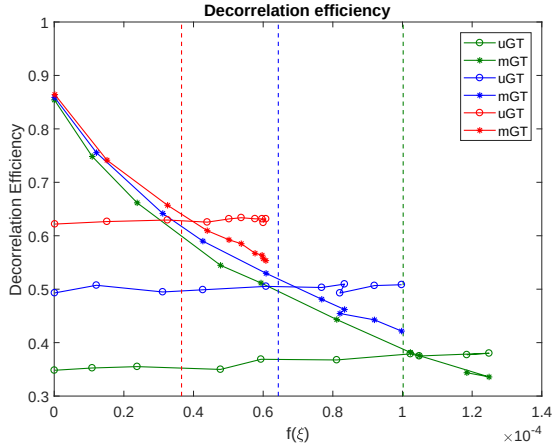


(c)

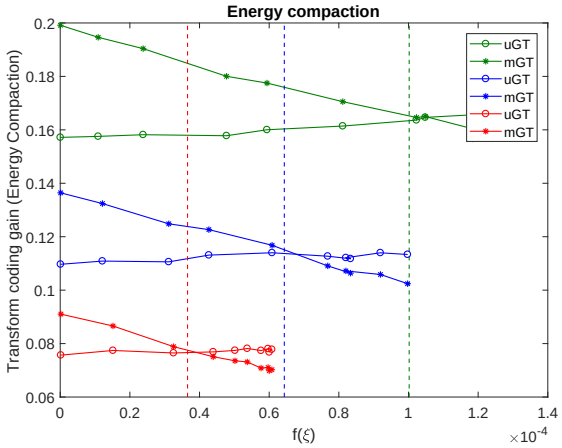
Figure 6.4: Impact of the uncertainty when the graph consists of 100 nodes, is regular with $deg = 8$, and the weights follow a uniform distribution with $dist = \{0.3, 0.7, 0.89\}$ that correspond to green, blue and red respectively.



(a)



(b)



(c)

Figure 6.5: Impact of the uncertainty when the graph consists of 100 nodes, is regular with $deg = \{8, 12, 20\}$(green, blue and red respectively), and the weights follow a uniform distribution centered on 0.45 with $dist = 0.89$

is also apparent in the large gaps in the crossing points of the three colored pairs of curves. It must be noted that the theoretical limit in this case is also well aligned with the observations and moves to the right when the variance of the model weights is getting smaller meaning that the Model based transform is more vulnerable against uncertainty when the model is getting more homogeneous.

Moreover, in Figure 6.5, the graph structure, more precisely the degree of a regular graph plays a crucial role in determining the amount of vulnerability of the *mGT* against the model uncertainty. This can be shown by the theoretical bound varying between 24 and 27 dB when the graph degree is 8, 12 or 20 with fixed model weights. the higher the degree of a graph, the higher the sensitivity of the *mGT* to uncertainty.

6.5 Conclusion

In this chapter, we showed that under some statistical uncertainties, relying on the unweighted graph laplacian to transform the signal could outperform the model-based transforms in terms of compression efficiency. Experimental results show that by using the theoretical bound $\mathcal{B}$, which does only require a graph laplacian matrix and a signal model, one can decide whether to apply the unweighted graph transform or the model-based transform to the graph signal to get a better compression efficiency. We showed that the sparsity of the graph, the variance of the model weights and the model weights distributions can have an impact on the model-based transform vulnerability against uncertainty.

Conclusion and perspectives

General conclusion

The ever-growing development of new camera types (whether Light Field or Omnidirectional cameras) that aim at capturing extra geometrical information, has led to a tremendous amount of redundant data to be stored and delivered. This expects constant innovation in the compression domain. An important step in the development of efficient coding schemes is to delineate effective transforms that are capable of successfully decorrelating the signals and thus result in signals much easier to code.

In this thesis, we were interested in defining optimal transforms for the new imaging modalities mentioned above. Albeit exclusive to such data, the dense color information resides on irregular domains. This irregularity arises from the underlying complex geometry of the scene in the case of light fields, and the non-uniformity of the sampling pattern in omni-directional captures. We rely on graph based representations, where we build graphs linking different pixels to represent their dependencies. Using those graphs, we design efficient graph based transforms for the compression of two emerging imaging modalities: light fields and omni-directional images. The efficiency is measured founded on the energy compaction property of the transform, its complexity and the resulting rate distortion performance.

In Chapter 3, we have targeted the problem of local graph transform design for light field energy compaction and compact representation. We have proposed two solutions. The first solution consists of using geometry-blind graph supports which are fixed for all light field sub-aperture images coupled with a powerful prediction mechanism based on Convolutional Neural Networks (CNN). In the second solution, the transform supports are based on super-rays (geometry-aware) built in a way that their shape remains coherent across the different views. We have first considered non separable graph transforms. Despite the limited size of the transform support, the Laplacian matrix of such graph remains of high dimension and its diagonalization to compute the transform eigenvectors is computationally expensive. To solve this problem, we then considered a separable spatio-angular transform. We have shown that, when the shape of corresponding super-pixels in the different views abide small changes, the basis functions of the spatial transforms are not coherent, consequent in a decreased correlation between spatial transform coefficients. We hence proposed a novel transform optimization method that aims at preserving angular correlation even when the shapes of corresponding super-pixels (i.e. forming one super-ray) are not isometric. This procedure has been shown to increase energy compaction of the separable spatio-angular graph transforms and bring substantial rate-distortion performance gains compared to a non optimized case. The proposed optimized spatio-angular graph transforms can be applied on both color or residual signals and can be easily parallelized to reduce the complexity on the decoder side.

While in the previous chapter, the limited support of local transforms may not allow us to exploit long term spatial dependencies, in Chapter 4, we cope with this limitation. We propose a novel approach to leverage the good spatial decorrelation properties of traditional codecs (e.g. HEVC intra), making use

of efficient predictors, into local spatio-angular graph transforms. A reference view coded with any efficient codec is used to predict low angular frequency transform coefficients that, together with the transmitted high angular transform coefficients, allow recovering the entire graph-based representation. The scheme has been assessed for high quality (quasi-lossless) coding.

We then tackle the compression of the second modality, namely the *omni-directional* images in chapter 5. We have proposed a new graph-based framework for omnidirectional image compression. We introduced a new R-D optimized graph partitioning to cope with the feasibility of graph Fourier Transform on global graphs defined on high resolution images. The partition obtained provides an effective trade-off between the smoothness of signals inside subgraphs and the cost of coding the partition description. Also, we showed that our methods outperform traditional DCT coding schemes at low bitrates.

In the previously evoked methods, we are mainly relying on the topologies of the underlying graphs to define the transforms. In the following chapter (chapter 6), we tend to give more flexibility to the weights on the edges of the graph, prior to transform design. Specifically, We develop a theoretical study to give a deeper understanding of the impact of the model uncertainty on the graph transforms.

Perspectives and future work

Several approaches could be considered to extend the work presented in this thesis.

In chapter 3, we have designed two solutions for lossy compression that work well for light fields with comparatively small baselines. Concerning the first solution based on the super-pixel over-segmentation and CNN based prediction, an efficient bit allocation can be pictured with a careful choice of the QP of the four references and the Q parameter. If we restrict our attention to the second solution, with the geometry aware supports, instead of adding a prediction step, we can incorporate a merging process as in [98]. This method allows us to exploit long term dependencies by merging similar neighboring super-rays with a rate distortion optimization and hence reducing the overall coding cost. An alternative way to exploit the long term dependencies is to perform a third graph transform on the low spatio-angular frequency coefficients with adapted weights depending on the super-rays similarity. In what concerns the spatio-angular transform itself, the complexity can be further reduced using fast graph Fourier transforms [60]. Moreover, future work can tackle the coding step itself by adjusting the quantization step sizes and tailoring the arithmetic coding to the statistical properties of the coefficients.

The same merging process can also be applied in the schemes proposed in chapter 4. This appends an extra prediction mechanism that reduces correlations between neighboring super-rays. Furthermore, in the case of lossy compression and a coarser quantization, noise and drift amplification is a subject of a future study. The conditioning problem that we faced could maybe be resolved with an efficient optimization procedure that is sufficiently regularized to retrieve the exact reconstruction of the light fields.

As a common future direction for both chapters, the methods need to be tuned for light fields with wider baselines, adapting the number of references and/or samples. Also, one could think of extending all coding schemes to dynamic light fields. Coupled with an efficient scene flow estimation, the proposed graph-based transforms can similarly be considered to best de-correlate the signal along super-rays and motion trajectories. In order to account for the temporal dimension, we can think of using the

notion of time-varying graphs [10], where the edges change over time.

In chapter 5, we have developed a rate-distortion optimized graph partitioning to define optimal sub-graphs. We have chosen the parameter λ in an intuitive manner. There is still no exact way of defining it in an optimal manner. Thus, it might be interesting to see how we can cautiously define this parameter for a better Rate-distortion optimization. Also, the coding of the segmentation map (i.e. the contours of sub-graphs) can be ameliorated with new efficient coders [118] that can be customized for the spherical geometry of the scene. Conforming the quantization step sizes of the sub-graphs frequency coefficients to the geometry of the sphere is also an interesting future direction. An extension to the RGB color and to omni-directional videos space can be envisioned.

Giving more degree of freedom to the weights, in chapter 6, we can broaden our study to different data models and apply our theoretical bound on real data such as 2D images or 3D point clouds. However, the main difficulty resides in the estimation for each 2D or 3D partition of a correlation term that is dependent on both the model uncertainty and the mismatch between model and topology. Statistically speaking, we want to find a plausible estimate of the true signal model of the signals inside blocks. Specific classes of images where the true model is approximately known will therefore be considered in our future work.

Author's publications

Conference papers

M. Rizkallah, T. Maugey, C. Guillemot, "Graph-based Spatio-angular Prediction for Quasi-Lossless Compression of Light Fields", *Data Compression Conference DCC*, Snowbird, United States, 2019

M. Rizkallah, F. de Simone, T. Maugey, C. Guillemot and P. Frossard "Rate Distortion Optimized Graph Partitioning for Omnidirectional Image Coding" , *European Signal Processing Conference EUSIPCO*, Rome, Italy, 2018 (BEST PAPER AWARD)

X. Su, M. Rizkallah, T. Maugey and C. Guillemot,"Rate-Distortion Optimized Super-Ray Merging for Light Field Compression", *European Signal Processing Conference EUSIPCO*, Rome, Italy, 2018

M. Rizkallah, X. Su, T. Maugey and C. Guillemot, "Graph-based Transforms for Predictive Light Field Compression based on Super-Pixels", *International Conference on Acoustics Speech and Signal Processing ICASSP*, Calgary, Canada, 2018

X. Su, M. Rizkallah, T. Maugey and C. Guillemot,"Graph-based Light Fields Representation and Coding using Geometry Information", *International Conference on Image Processing ICIP*, Beijing, China, 2017

M Rizkallah, T Maugey, C Yaacoub, C Guillemot, "Impact of Light Field Compression on Focus Stack and Extended Focus Images", *Signal Processing Conference EUSIPCO*, Budapest, Hungary, 2016

Journal papers

M. Rizkallah, X. Su, T. Maugey, C. Guillemot, "Geometry-Aware Graph Transforms for Light Field Compact Representation", *IEEE Trans. on Image Processing*, TIP (submitted)

M. Rizkallah, T. Maugey, C. Guillemot, "Light Field Prediction and Sampling with Local Graph Transforms for Quasi-Lossless Compression", *IEEE Trans. on Image Processing*, TIP(submitted)

List of Figures

List of Tables

List of Algorithms

Appendix A

Impact of light field compression on post-capture functionalities

Impact of Light Field Compression on Focus Stack and Extended Focus Images

Mira Rizkallah*, Thomas Maugey†, Charles Yaacoub‡ and Christine Guillemot†
*IRISA/Université Rennes 1
†INRIA Rennes Bretagne-Atlantique
‡Holy Spirit University Of Kaslik (USEK)

***Abstract*—Light Fields capturing all light rays at every point in space and in all directions contain very rich information about the scene. This rich description of the scene enables advanced image creation capabilities, such as re-focusing or extended depth of field from a single capture. But, it yields a very high volume of data which needs compression. This paper studies the impact of Light Fields compression on two key functionalities: refocusing and extended focus. The sub-aperture images forming the Light Field are compressed as a video sequence with HEVC. A focus stack and the scene depth map are computed from the compressed light field and are used to render an image with an extended depth of field (called the extended focus image). It has been first observed that the Light Field could be compressed with a factor up to 700 without significantly affecting the visual quality of both refocused and extended focus images. To further analyze the compression effect, a dedicated quality evaluation method based on contrast and gradient measurements is considered to differentiate the natural geometrical blur from the blur resulting from compression. As a second part of the experiments, it is shown that the texture distortion of the in-focus regions in the focus stacks is the main cause of the quality degradation in the extended focus and that the depth errors do not impact the extended focus quality unless the light field is significantly distorted with a compression ratio of around 2000:1.**

I. Introduction

During the last two decades, there has been a growing interest in Light Fields. Many acquisition and sampling techniques were envisioned in order to capture the light information present in a scene using arrays of cameras, plenoptic cameras or moving cameras [1], [2], [3], [4]. Essentially, a captured light field comprises geometrical information of the scene along with texture information. The light field representation enables various applications such as digital refocusing, depth estimation, changing perspective and viewpoints, simulating captures with different depth of fields and 3D reconstructions.

This comes at the expense of collecting large volumes of high-dimensional data, which appears to be the key downside of light fields. For example, a modest four dimensional light field, captured by a plenoptic camera comprising a 256x256 array of microlenses with 32x32 photosensors behind each microlens, yields a storage footprint of around 200 Mbytes, which is significantly large for a photograph. Therefore, to realize practical applications of light fields, it is essential to efficiently compress this data without compromising the ultimate quality and more importantly, without leaving an undesirable effect on the targeted post-capture processing.

Some research effort has been dedicated in the past years to the design of light fields compression schemes based on vector quantization [5], transform coding [6] [7], statistical representations [8], multiview video compression and disparity compensation techniques [9], or adaptations of HEVC [10], [11], [12]. However, very little attention has been given to the impact of compression on the image creation functionalities.

In this paper, we aim at analyzing the impact of compression on two post-capture image rendering functionalities: refocusing and extended focus. The extended focus image is characterized by the fact that all its pixels are in-focus whereas the focus stack images have in-focus and out-of-focus pixels in different regions of the scene.

Toward this goal, we first compress the light field using HEVC, considering the set of sub-aperture images as a video sequence. The focus stack images are then computed by shifting and adding the compressed sub-aperture images. Afterwards, the scene depth map is estimated from the compressed sub-aperture images and the computed focus stack using the method described in [13]. The observed quality of the focus stack and the extended focus images show that one can vary the QP parameter up to 32, and decrease the light field data volume from 50 Mbytes down to 75 kbytes without visually impacting the quality of the rendered images.

Looking at a refocused image after compression, the blur due to compression is unnoticed in out-of focus regions where the quantization blur is mixed to natural geometry blur. As for the extended focus, the compression blur is visible in the entire image. Comparing with the refocused and extended focus images computed from the original light field, it appears that the refocused images are visually more robust to compression than the extended focus image. Traditional metrics such as PSNR fail to accurately reflect this unbalanced robustness. For this reason, a dedicated metric naturally differentiating the geometrical blur present in out-of-focus regions from the blur introduced by compression is then considered based on contrast and gradient measurements. Furthermore, it is shown that the depth map used for creating the extended focus image does not impact the extended focus even for a compression factor of 2000, and that the extended focus image quality degradation essentially results from the texture distortion of the in-focus pixels of the focus stack images.

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

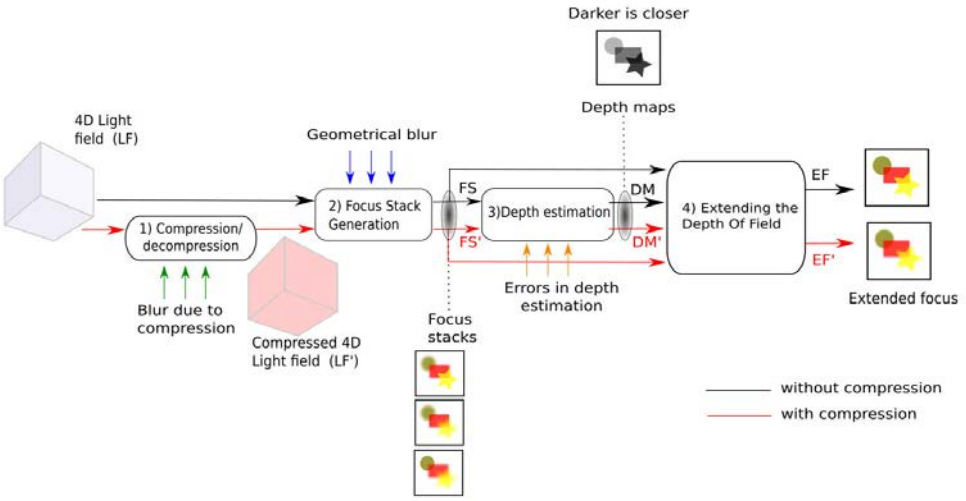


Fig. 1: Processing chain considered in this work with four main stages: 1) Compression, 2) Focus stack generation, 3) Depth estimation and 4) Extending the depth of field.

II. Light Fields processing chain

The light field processing chain considered in this work is depicted in the scheme of Fig. 1. It proceeds in four phases: compression, generating the focus stack, depth estimation and finally extending the depth of field. We explain each of these steps separately in the following.

The light field sub-aperture images are first assembled, following a spiral scan line (see Fig. 2), as a video sequence which is coded with HEVC. The spiral scan line is justified by the fact that the luminosity of the sub-aperture images varies progressively going from the center outwards due to the spherical shape of the camera main lens. The method is compared to previous multiview and disparity compensation techniques [9] in Fig. 3 for the Buddha light field[1]. Given the significant improvement in the rate-distortion efficiency, this method is used in our analysis in the following sections.

Once the light field has been compressed, 32 photographs focused at different depths, forming the so-called focus stack, are computed by shifting and adding the sub-aperture images [1] [2] as in Eq. (1)

$$E_{\alpha_i}(x,y) = \frac{1}{(\alpha_i F)^2} \sum_u \sum_v L^{(u,v)}(X_s + x, Y_s + y) \qquad (1)$$

$$X_s = u(1 - \frac{1}{\alpha_i}); Y_s = v(1 - \frac{1}{\alpha_i}),$$

where $E_{\alpha_i}(x,y)$ denotes the pixel value at position (x,y) in the i^{th} refocused image E_{α_i} at depth α_i. The value F represents the distance between the lens and the reference film planes. Focusing at different depths α_i corresponds to changing the separation between the lens and the film plane, hence the multiplication $\alpha_i F$.

In the experiments reported in the paper, the parameter α_i takes 32 different values in the interval [0.6, 1.8]. This parameter actually controls the position of the focus plane.

[1]buddha4.tar.gz at http://graphics.stanford.edu/software/lightpack/lifs.html

$L^{(u,v)}$ is the sub-aperture image from the (u,v) position on the main lens aperture. X_s and Y_s are the shift amounts.

The image creation process is equivalent to shearing the 4D light field varying the slope of epipolar lines. At that stage, a natural geometrical blur appears in out-of-focus regions.

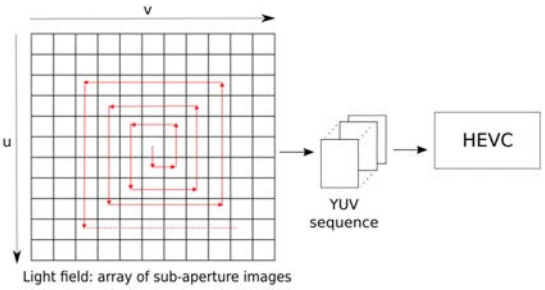


Fig. 2: Light field coding using HEVC: The sub-aperture images are coded in a video sequence following a spiral scan line.

Afterwards, the scene depth map is also estimated from compressed sub-aperture images and the computed focus stack using the method detailed in [13] which combines both defocus and correspondence cues. For each pixel, we seek the optimal contrast within a patch along refocused images thus the highest defocus response, and the minimum correspondence response minimizing the angular variance along the sheared light fields. The output of this phase is a depth map with each pixel value pointing out to one image of the focus stack. Once the depth map and the focus stack have been computed, the depth of field can be extended, focusing on all the scene at once leading to an image with no geometrical blur. The all in-focus image $E(x,y)$ is constructed from the depth map and the focus stack images as follows. For each pixel at position (x,y) and its corresponding depth α_i, the extended focus image $E(x,y)$ is formed by taking the pixel at position (x,y) in

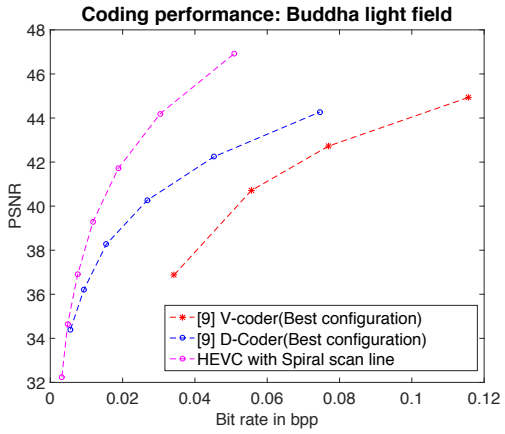


Fig. 3: Comparison of the coding efficiency of HEVC-Spiral (using QP =24, 28, 32, 36, 40, 44, 48) with the methods in [9] based on disparity compensated multiview coding techniques.

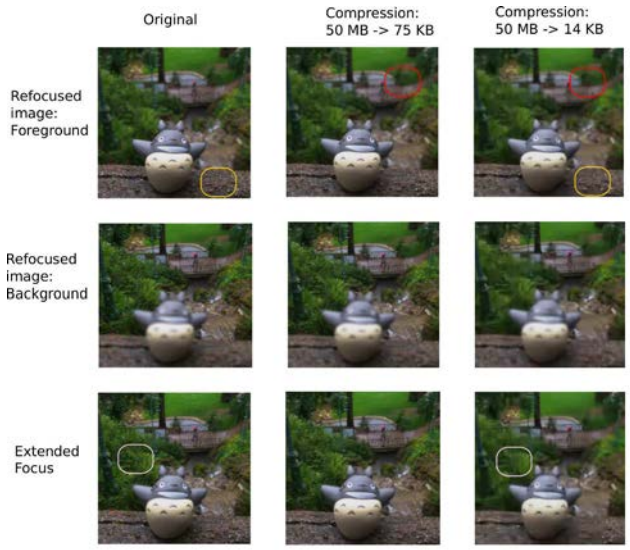


Fig. 4: Evolution of the quality of the refocused and extended focus images' while varying the compression ratio. The red patches refer to out-of focus regions of a refocused image, the orange ones to the in-focus regions without and with light field compression and the white ones to a region in the extended focus without and with light field compression.

the focus stack image $E_{\alpha_i}(x, y)$ ($E(x, y) = E_{\alpha_i}(x, y)$).

III. Subjective results and analysis

As depicted in Fig. 1, the four stages involve two types of blur: a geometrical blur caused by the refocusing procedure which is, by nature, good for image quality, and a quantization blur which decreases the perceived quality due to texture compression. Besides, some errors may occur in the depth estimation process, where a wrong depth is assigned to some of the pixels.

The added amount of blur can be observed in Fig. 4 where results of the Refocusing and Extended Focusing algorithms before and after compression of a natural Light Field are shown. The two types of blur are apparent in the refocused image. On the one hand, the red patch refers to an out-of-focus region in the background where the geometrical blur is mixed with compression blur after compression. In these regions, the compression blur is not visually perceptible. On the other hand, the orange ones refer to an in-focus region in the foreground only affected by the compression blur that thus becomes visible at high compression ratios. When combining the in-focus regions of all the refocused images in an extended focus image, only the blur due to compression remains such as in the white patches. Experimentally, we observed that we can achieve high compression ratios, attaining a range of 500:1 to 700:1 without altering the visual quality of the extended focus image. Afterwards, for very high compression ratios, the quality drops significantly leading to a fully blurred image.

We remark that the perceived quality of the refocused images is naturally higher compared to the extended focus. This is due to the fact that some parts of the refocused images are naturally blurred, and the probability of adding blur to in-focus regions is relatively small compared to the extended focus. Moreover, the extended focus is built from the focus stack, therefore the compression blur propagates from in-focus regions of the focus stack to the extended focus. However, this subjective quality trend does not exactly align with primary PSNR measurements depicted in Fig. 5. More precisely, we plot the PSNR of the rendered images as a function of the PSNR of the compressed Light field and its size after compression while varying the QP from 24 to 44. A decrease of 1 dB of the light field's PSNR leads to a decrease of 2 dB in terms of PSNR for both functionalities. In other words, the PSNR does not reflect the fact that the extended focus is more affected by compression than the focus stack[2]. This is mainly due to the fact that the PSNR does not differentiate the geometry blur from the compression blur.

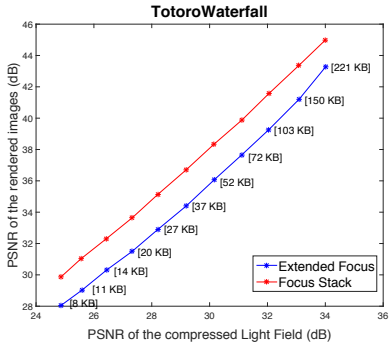


Fig. 5: The evolution of the refocused and extended focus images' quality as a function of the compressed Light Field quality.

For this reason, we propose in section IV, a dedicated metric based on gradient and contrast measurements that evaluates the amount of compression blur added to in-focus regions. With

[2]There, we are interested by the PSNR evolution rather than the PSNR value since the reference from which the MSE is computed is not the same.

this metric, we study the impact of compression on the amount of blur in the focus stack and the extended focus images. Afterwards, we examine the major cause of the extended focus quality degradation: errors in depth estimation or texture compression.

IV. Proposed metric

As explained in the previous section, the proposed metric aims at measuring the amount of compression blur added in the regions that are supposed to be in-focus in the focus stack and extended focus images. Let n_i and n_i' denote the numbers of in-focus pixels in a refocused image E_{α_i} (at depth α_i), before and after compression of the light field respectively. The percentage of pixels that become blurred after compression is expressed by:

$$\psi = \frac{n_i - n_i'}{N} \times 100 \tag{2}$$

where N denotes the total number of pixels in the evaluated image. n_i' is estimated from the difference in the gradient response of in-focus pixels before and after compression. The gradient responses are evaluated using the following gradient operator:

$$G(x,y) = \frac{1}{|W_D|} \sum_{(x',y') \in W_D} \Delta E_{\alpha_i}(x',y') \tag{3}$$

where E_{α_i} is the evaluated refocused image in the focus stack. Considering only pixels at positions (x,y) which were in-focus before compression, $G(x,y)$ is the gradient response averaged within a patch to improve robustness. W_D represents the window around the current pixel (x,y) with its size $|W_D|$ and Δ stands for the spatial gradient operator.

The gradient's variation is compared to a predefined threshold value T, to determine whether the pixel is still in-focus or not after compression.

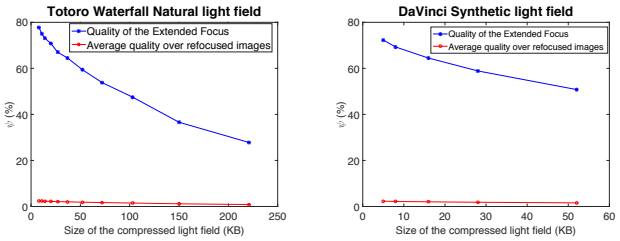
The calculated ratio ψ is averaged over the entire focus stack. For the extended focus image, ψ is computed under the perspective that everything is supposed to be in-focus (n_i is thus equal to N).

V. Experiments

In our experiments, HEVC Test Model (HM) reference software[3] was used to code the sub-aperture images. A GOP size of 4 was picked with a 'IBBBP' encoding scheme. Only the central viewpoint image is intra-coded. The QP was varied from 24 to 44. PSNR, SSIM and ψ are measured taking as reference the refocused and extended focus images computed from the original uncompressed light field.

A. Quality Evaluation: Focus Stack vs. Extended Focus

We used the ψ (Eq. (2)) to measure the percentage of blur added in the refocused and extended focus images after compression of the light field. For the experiments, we use two light fields: a natural one captured by a plenoptic camera and a synthetic one. *Totoro Waterfall* (available at [14]) is captured by a Lytro plenoptic camera and consists of 11x11 subaperture images containing 379x379 RGB pixels each. *Da Vinci* (available at [15]) is a synthetic light field comprising 9x9 Multiview images with 768x768 pixels each. Fixing T to 0.003 and the window radius to 3, ψ is plotted in Fig. 6 as a function of the size of the compressed light field. It shows the difference in the amount of blur added due to compression between the refocused and extended focus images.



(a) (b)

Fig. 6: Percentage of blurred in-focus pixels due to compression (ψ) for both the extended focus and refocused images. (a) Natural light field (Original size = 50 MB), (b) Synthetic light field (Original size = 136 MB)

After compressing both light fields, an average maximum of 3% of initially in-focus pixels become out-of focus in each refocused image. This is a very small amount compared to the extended focus image where a very high percentage of pixels (up to 80% at high compression ratios) becomes blurred due to compression.

The probability of adding blur in the in-focus regions is relatively small which makes the focus stack visually more robust to compression than the extended focus.

B. Impact of the Texture and Depth Estimation on the Extended Focus Quality

There are two possible causes of quality degradation of the extended focus image resulting from compression of the light field: depth map errors and focus stack texture distortion. For the purpose of investigating the impact of those two types of errors, we evaluate the quality of the extended focus image under different conditions. More precisely, three combinations of inputs to the extended focus estimation algorithm are tested separately: an original depth map DM with the focus stacks after compression FS'; a reconstructed depth map after compression DM' with original focus stacks FS and finally DM' with FS' which is the natural combination. We then plot, for each one of them, the quality evolution as a function of the size of the compressed light field. The quality is estimated with PSNR (Fig. 7 (a)(d)), SSIM (Fig. 7 (b)(e)), and ψ (Fig. 7 (c)(f)).

We see that using an original focus stack FS, the depth map DM'can be estimated from a natural light field compressed with a factor 3600 or a synthetic one with a factor of 8000 without altering the quality of the extended focus. On the other side, no additional gain in quality is observed with an original depth map DM and a reconstructed focus stack FS'. This

[3]Reference software for ITU-T H.265 high efficiency video coding Version 10/14 available at http://www.itu.int/rec/T-REC-H.265.2

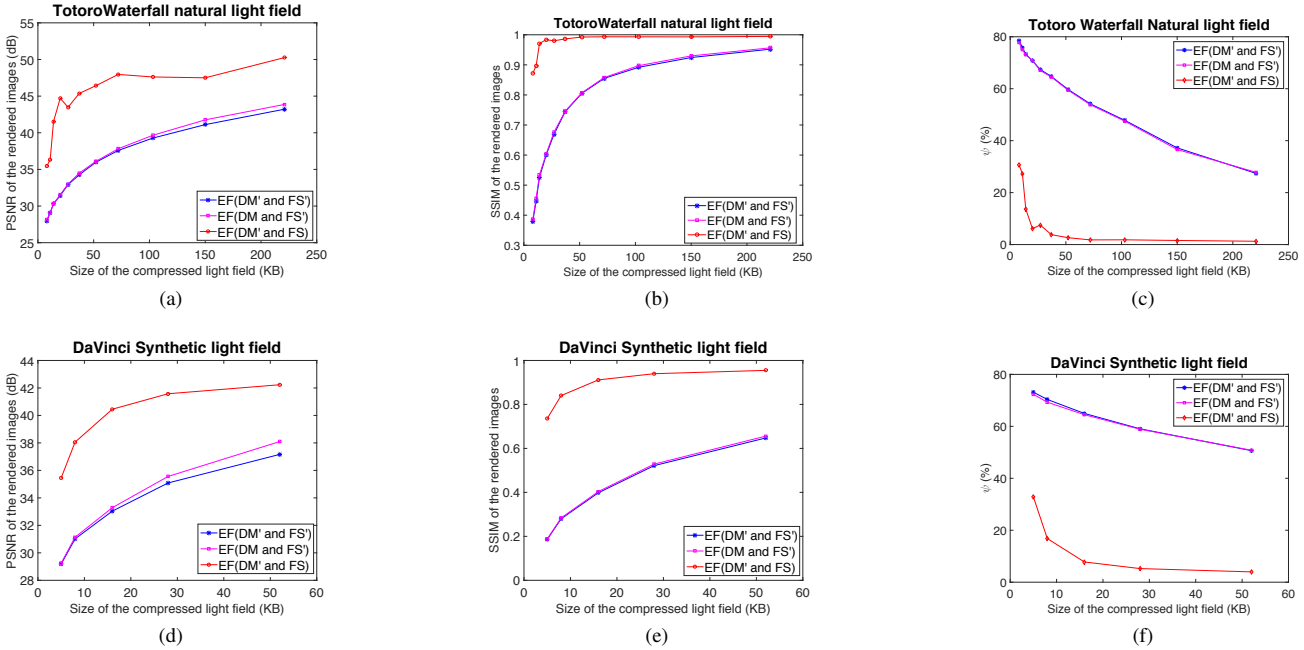


Fig. 7: Evolution of the quality (PSNR, SSIM and the dedicated metric ψ) of the Extended Focus (EF) with different input combinations: An original Depth Map (DM) and a reconstructed Focus Stack (FS'); an original Focus Stack (FS) and a reconstructed Depth Map (DM'); DM' and FS'. (a)(b)(c): A natural light field (Original size = 50 MB). (d)(e)(f): A synthetic light field (Original size = 136 MB).

shows that the extended focus quality degradation essentially results from the focus stack texture distortion.

VI. Conclusion

In this paper, we analyze the impact of light field compression on the quality of the focus stack and the extended focus images. It has been observed that a light field can be compressed by a factor of around 700 without altering the visual quality of both considered functionalities. Based on a dedicated quality metric adapted to the problem, we showed the focus stack is more robust to compression than the extended focus since already some parts of it are blurred natively. It has been also shown that the major cause of quality degradation in the extended focus is the texture distortion of in-focus regions in a focus stack while the depth estimation errors do not have a significant impact on the rendering quality. This study might be used to develop new performing coding schemes for Light Fields taking into account the quality of targeted applications.

References

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 31–42.

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

[3] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 765–776, Jul. 2005.

[4] T. Georgiev, G. Chunev, and A. Lumsdaine, "Superresolution with the focused plenoptic camera," pp. 78 730X–78 730X–13, 2011.

[5] A. C. Beers, M. Agrawala, and N. Chaddha, "Rendering from compressed textures," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 373–378.

[6] G. Miller, R. S., and D. Ponceleon, "Lazy decompression of surface light fields for precomputed global illumination," in *Rendering Techniques '98*, 1998, pp. 281–292.

[7] M. Magnor, A. Endmann, and B. Girod, "Progressive compression and rendering of light fields," in *Vision, Modelling and Visualization*, 2000, pp. 199– 203.

[8] D. Lelescu and F. Bossen, "Representation and coding of light field data," *Graph. Models*, vol. 66, no. 4, pp. 203–225, Jul. 2004.

[9] M. Magnor and B. Girod, "Data compression for light-field rendering," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 10, no. 3, pp. 338–343, Apr. 2000.

[10] C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod, "Light field compression using disparity-compensated lifting and shape adaptation," *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 793–806, April 2006.

[11] Y. Li, M. Sjstrm, R. Olsson, and U. Jennehag, "Efficient intra prediction scheme for light field image compression," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 539–543.

[12] Y. Li, M. Sjstrm, R. Olsson, and U. Jennehag, "Scalable coding of plenoptic images by using a sparse set and disparities," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 80–91, Jan 2016.

[13] M. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *2013 IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, 2013, pp. 673–680.

[14] A. Mousnier, E. Vural, and C. Guillemot, "Partial light field tomographic reconstruction from a fixed-camera focal stack," https://www.irisa.fr/temics/demos/lightField/index.html.

[15] S. Wanner, S. Meister, and B. Goldluecke, "Datasets and benchmarks for densely sampled 4d light fields," http://hci.iwr.uni-heidelberg.de/HCI/Research/LightField/lfbenchmark.php.

Appendix B

Graph based compression of light fields

GRAPH-BASED LIGHT FIELDS REPRESENTATION AND CODING USING GEOMETRY INFORMATION

Xin Su, Mira Rizkallah, Thomas Maugey and Christine Guillemot

INRIA, Campus de Beaulieu, Rennes 35042, France

ABSTRACT

This paper describes a graph-based coding scheme for light fields (LF). It first adapts graph-based representations (GBR) to describe color and geometry information of LF. Graph connections describing scene geometry capture inter-view dependencies. They are used as the support of a weighted Graph Fourier Transform (wGFT) to encode disoccluded pixels. The quality of the LF reconstructed from the graph is enhanced by adding extra color information to the representation for a sub-set of sub-aperture images. Experiments show that the proposed scheme yields rate-distortion gains compared with HEVC based compression (directly compressing the LF as a video sequence by HEVC).

Index Terms— Graph Based Representation (GBR), Graph Fourier Transform (GFT), Compression, Light fields (LF).

1. INTRODUCTION

Light fields (LF) have emerged as a representation of light rays emitted by a 3D scene and received by an observer at a particular point (x, y, z) in space, along different orientations. A variety of capturing devices have been designed based on camera arrays [1], on single cameras mounted on moving gantries, or on arrays of micro-lenses placed in front of the photosensor to obtain angular information about the captured scene [2, 3].

The problem of LF compression rapidly appeared as quite critical given their significant demand in terms of storage capacity. Classical block-based coding schemes such as JPEG applied for each image of the 2D array of images forming the lumigraph have been quite naturally considered yielding however limited compression performances (compression factors not exceeding 20 for an acceptable quality) [4]. A method based on video compression is presented in [5] where a few views are encoded in Intra while the other views are encoded as P-images in which each block can be predicted from one of the neighboring Intra views with or without disparity compensation, the choice of the prediction mode being made to optimize a rate-distortion measure. A second scheme is presented where several predictions of a view are computed from neighboring views using disparity maps, and averaged to give the final predicted view. The prediction residue is then encoded using classical coding tools (DCT, quantization). Multiview video compression and disparity compensation techniques are considered in [5, 6], and intra coding modes have also been proposed in [7] for LF compression using HEVC. The authors of [8] exploits inter-view correlation by using a homography-based low rank approximation of the LF, showing significant gains compared to HEVC Inter-coding for real LF captured by micro-lenses based devices.

In this paper, we explore the use of GBR for LF. GBR has been proposed for describing the geometry of multi-view images, first for horizontally aligned cameras [9] and more recently for complex camera configurations [10]. Here, we consider GBR to represent LF using 3D geometry information. The graph connections are derived from the disparity and hold just enough information to synthesize other sub-aperture images from one reference image of the LF. Based on the concept of epipolar segment, the graph connections are sparsified (less important segments are removed) by a rate-distortion optimization. The graph vertices and connections are compressed using HEVC [11]. The graph connections capturing the inter-view dependencies are used as the support of a Graph Fourier Transform [12] used to encode disoccluded pixels.

However, the graph mostly represents scene geometry. Texture information is limited to a reference view and disoccluded pixels, which is not sufficient for reaching a high reconstructed LF quality. The bitrate distribution between texture and geometry (i.e. depth) is a key issue in view synthesis from multi-view data and depends on the camera configuration [13]. To enhance the quality of the reconstructed LF, the residuals of a subset of views are added to the graph representation. Experiments with synthetic LF from the dataset in [14] rendered with Blender [15] show that the proposed scheme achieves higher reconstruction quality at low rates compared with traditional video compression by HEVC.

2. LIGHT FIELDS GEOMETRY

We consider the simplified 4D representation of LF describing the radiance along rays by a function $L(x, y, u, v)$ of 4 parameters at the intersection of the light rays with 2 parallel planes. This representation can be seen as an array of multi-view images $\{\mathcal{I}_{u,v}\}$. Each view $\mathcal{I}_{u,v} \in \mathbb{R}^{X \times Y \times 3}$ at position (u, v) is an RGB image with $X \times Y$ pixels. Given a pixel (x, y) in $\mathcal{I}_{u,v}$, its *corresponding* pixel in $\mathcal{I}_{u',v'}$ (the pixel corresponding to the same 3D point in the real world), should have the same color values under the Lambertian assumption. In principle, multiple views of a scene can be rendered from one unique view with the help of scene geometry. This is the core idea of depth image based rendering (DIBR). For instance, given a LF dataset with available depth images $\{\mathcal{Z}_{u,v}\}$, pixel (x', y') in $\mathcal{I}_{u',v'}$ corresponding to the same 3D point as the pixel (x, y) in $\mathcal{I}_{u,v}$ can be located by

$$\begin{aligned} (x', y') &= (x + d_x, y + d_y)\ , \\ d_x = \tfrac{B*(u-u')*f}{Z_{u,v}(x,y)}\ , d_y &= \tfrac{B*(v-v')*f}{Z_{u,v}(x,y)}\ \ , \end{aligned} \tag{1}$$

where B is the distance between neighboring cameras, f is the focal length, $Z_{u,v}(x, y)$ is the depth of pixel (x, y) in $\mathcal{I}_{u,v}$. View $\mathcal{I}_{u',v'}$

thus can be rendered pixel by pixel by Eq.(1). (d_x, d_y) is also known as disparity. In the tests, we consider synthetic LF [14] for which depth information is available. For real LF, depth has to be estimated using for example the methods in [16, 17].

Pixels in different views corresponding to the same 3D point have same or similar color values. In this paper, we represent inter-view dependencies in LF with a graph using geometry information, and use the graph as a support to encode the color information using graph-based transform coding.

3. GRAPH REPRESENTATION

3.1. Graph construction

Let us denote the graph by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where vertices $\mathcal{V} = \{v_i\}$ correspond to each pixel in sub-aperture images $\{\mathcal{I}_{u,v}\}$, and edges $\mathcal{E} = \{e_{ij}\}$ connect pairs of pixels across two images.

Graph connections for reference image. As shown in Fig.1.a, image $\mathcal{I}_{1,1}$ (left bottom corner image marked in red) is selected as the reference view. Pixels on each row of $\mathcal{I}_{1,1}$ are grouped into a set of *straight* horizontal segments based on their depth. One segment has a constant depth. As shown in Fig.1.b, one row in $\mathcal{I}_{1,1}$ has been divided into 3 segments. Every segment in $\mathcal{I}_{1,1}$ is connected to one segment in every sub-aperture image by one graph edge, since the two segments correspond to the same straight segment in the real 3D world. For instance in the toy example in Fig.1.a, the reference view $\mathcal{I}_{1,1}$ is connected with every sub-aperture image $\mathcal{I}_{u,v}$ by graph edges. However, for one straight segment in $\mathcal{I}_{1,1}$, all its connections to other sub-aperture images can be deduced from each other by Eq.(1). Therefore, for one segment in $\mathcal{I}_{1,1}$, only one of its connections is necessary in the final graph structure. In our GBR, we only keep the graph connections between $\mathcal{I}_{1,1}$ and $\mathcal{I}_{1,2}$ (the right sub-aperture image of $\mathcal{I}_{1,1}$), as shown in Fig.1.a, the connections marked as red solid line are kept and the other connections marked as black doted lines are redundant and removed. Fig.1.c gives an illustration of the kept graph connections between $\{\mathcal{I}_{1,1}\}$ and $\{\mathcal{I}_{1,2}\}$.

To simplify the graph representation, each graph connection is represented by a one-dimensional metric namely unidimensional disparity based on the *epipolar segment* concept [10]. The epipolar segment is a line segment consisting of all possible projections of a pixel with varying depth. The unidimensional disparity actually is the distance between the start point of the epipolar segment and the position of the true projection.

Graph connections for disoccluded pixels. Besides the reference image $\mathcal{I}_{1,1}$, the disoccluded pixels which are not visible in $\mathcal{I}_{1,1}$ are also considered in the graph construction. For the sake of simplicity, we only consider the disoccluded pixels in $\mathcal{I}_{U,V}$ (the top right corner image in Fig.1.a), since most of the disoccluded pixels in images $\{\mathcal{I}_{u,v}\}$ $(1 < u < U, 1 < v < V)$ are visible in $\mathcal{I}_{U,V}$. To construct the graph connections for the disoccluded pixels, the same strategy has been applied here. In other words, these disoccluded pixels are treated as "*reference pixels*" for other sub-aperture images.

3.2. Graph sparsification

As presented in [10, 18], the constructed graph in section 3.1 is sparsified based on a rate-distortion model,

$$\mathcal{J}(\mathcal{E}) = \mathcal{D}(\mathcal{E}) + \alpha\mathcal{R}(\mathcal{E})\,, \tag{2}$$

where $\mathcal{J}$ is the Lagrangian cost (smaller $\mathcal{J}$ values mean better optimal status), $\mathcal{D}$ is the distortion of rendered sub-aperture images and $\mathcal{R}$ is the modeled bitrate cost for coding the graph connections. α is the Lagrangian multiplier which represents the relation between bitrate and rendering quality (distortion). To decrease the computational cost we compute the rendering distortion on only a subset of views. Edges are removed based on the shortest path optimization of

$$\mathcal{E} = \underset{\mathcal{E}}{\operatorname{argmin}}\ \mathcal{J}(\mathcal{E}). \tag{3}$$

Graph sparsification does not only reduce bitrate cost but also corrects errors in the depth, since the optimization modifies graph connections regarding rendering distortion. For real LF with estimated depth, it is very useful due to noise or errors in the estimated depth.

3.3. Graph with Residuals

So far, the constructed graph contains minimum amount of color information, since only the reference view $\mathcal{I}_{1,1}$ and the disoccluded pixels in $\mathcal{I}_{U,V}$ are kept. To enhance the quality of the reconstructed views, residues $r_{m,n}$ between a subset of M rendered images (from the graph) $\tilde{\mathcal{I}}_{m,n}$ and the original true images $\mathcal{I}_{m,n}$, computed as $r_{m,n} = \mathcal{I}_{m,n} - \tilde{\mathcal{I}}_{m,n}$ are added to the graph.

At the decoder, these selected sub-aperture images are also treated as "*reference images*" to render the remaining sub-aperture images. The depth of each straight segment in the reference image $\mathcal{I}_{1,1}$ is estimated from the corresponding graph connections by Eq.(1). Then, the depth of the selected images $\mathcal{I}_{m,n}$ is computed by projection from the estimated depth of $\mathcal{I}_{1,1}$. We compute each remaining sub-aperture image $\mathcal{I}_{m,n}$ by combining $M+1$ rendered images, one image recovered from the graph and M images warped from the selected *reference images*.

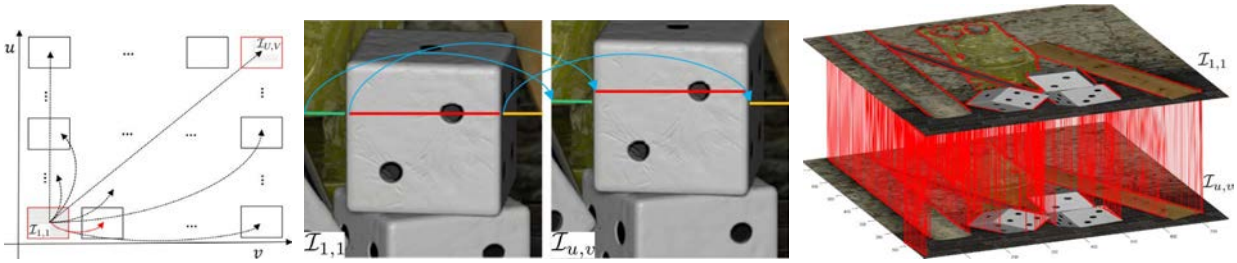
$$\hat{\mathcal{I}}_{u,v} = \frac{1}{\sum w_i}\left(w_0\tilde{\mathcal{I}}_{u,v} + \sum_{i=1}^{M} w_i\ \tilde{\mathcal{I}}_{u,v}\Big|_{\hat{\mathcal{I}}_{m,n}}\right),$$

$$[w_0, w_1, ...w_M]^T = \mathbf{Rxy}\left(\tilde{\mathcal{I}}_{u,v}\ |_{\hat{\mathcal{I}}_{m,n}}, \mathcal{I}_{u,v}\right)\mathbf{Rxx}\left(\tilde{\mathcal{I}}_{u,v}\ |_{\hat{\mathcal{I}}_{m,n}}\right)^{-1}$$

where weights $[w_0, w_1, ...w_M]^T$ are computed using the minimum mean square error estimation theory. $\mathbf{Rxy}\left(\tilde{\mathcal{I}}_{u,v}\ |_{\hat{\mathcal{I}}_{m,n}}, \mathcal{I}_{u,v}\right)$ is the cross-correlation of the $M+1$ rendered images and the original image $\mathcal{I}_{u,v}$, and $\mathbf{Rxx}\left(\tilde{\mathcal{I}}_{u,v}\ |_{\hat{\mathcal{I}}_{m,n}}\right)$ is the autocorrelation of the $M+1$ rendered images.

4. CODING SCHEME

The proposed encoder is shown in Fig. 2. As explained in Section 3, from two sub-aperture images, namely the corner images $\mathcal{I}_{1,1}$ and $\mathcal{I}_{U,V}$, we construct the LF graph representation $(\mathcal{G} = (\mathcal{V}, \mathcal{E}))$. The graph edges $\mathcal{E}$ are stored in a grey-level image which is coded using HEVC (More details about the graph edges coding can be found in [10]). The vertices are pixels in the images $\mathcal{I}_{1,1}$ and $\mathcal{I}^o_{U,V}$ (the parts of $\mathcal{I}_{U,V}$ that do not appear in $\mathcal{I}_{1,1}$). A part of the graph is depicted in Fig. 3 where blue segments are edges with a small weight (0.5) whereas red ones are edges with high weight(1). While $\mathcal{I}_{1,1}$ is classically compressed using HEVC, the arbitrarily shaped $\mathcal{I}^o_{U,V}$ requires dedicated tools. We propose to compress it using a graph-based compression scheme as follows.



(a) GBR for LF. (b) example of graph connections between $\mathcal{I}_{1,1}$ and $\mathcal{I}_{u,v}$. (c) Graph connections between $\mathcal{I}_{1,1}$ and $\mathcal{I}_{u,v}$.

Fig. 1. Graph based representation (GBR) adapted to LF.

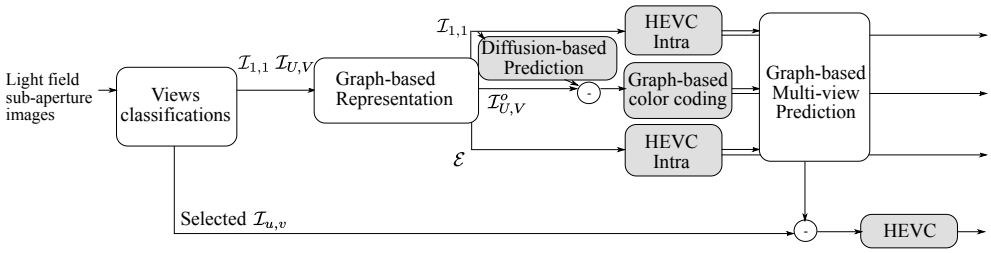


Fig. 2. Proposed encoder

Let $S = [S_1 S_2]$ be the vector of all color values in $\mathcal{V}$ to be coded, where S_1 comprises the color values of the reference image $\mathcal{I}_{1,1}$ (separately coded with HEVC), and S_2 the color values of the disoccluded pixels $\mathcal{I}^o_{U,V}$. The color values in the reference view are initially propagated to the disoccluded pixels using an iterative diffusion method. More precisely, at the first iteration, the pixels at the borders of the disocclusion areas are predicted by computing a weighted average of their $1-$hop neighborhood in the reference image. For example, the prediction of a disoccluded pixel p_1 connected to four pixels in the reference view (p_2, p_3, p_4, p_5) is computed as

$$\frac{w_{12}p_2 + w_{13}p_3 + w_{14}p_4 + w_{15}p_5}{w_{12} + w_{13} + w_{14} + w_{15}}$$

where w_{ij} denotes the weight of the connection between the pixels i and j. In practice, the weight values are always 1 except where the depth difference exceeds threshold $\frac{Z_{\max} - Z_{\min}}{20}$ ($Z_{\max}$ and $Z_{\min}$ are maximum and minimum values of the depth image). In that case, a lower weight is assigned to attenuate the color propagation. The predicted pixels are then used to predict other disoccluded pixels in the following iteration.

To code the prediction residuals of the disoccluded pixels, i.e., $R = S_2 - E(S_2/S_1)$, we use the weighted Graph Fourier Transform (wGFT) [12] . The target disocclusion image is divided in 8×8 pixel blocks. In each block, we use the 4-neighbors graph which connects the disoccluded pixels to transform the residuals. More specifically, given the weight matrix W, we define the diagonal degree matrix D, where $D_{ii} = \sum_j w_{ij}$. Lastly, the graph normalized weighted Laplacian matrix L_{norm} is computed as $L_{\text{norm}} = I - D^{-1/2} W D^{-1/2}$. Let Ψ be the matrix whose columns contain the wGFT basis i.e., the eigenvectors of the graph normalized laplacian. The residuals are thus projected on the wGFT basis as $\hat{R} = \Psi R$. The coefficients are quantized for various quality factors following the method in [19], entropy coded then sent to the decoder side.

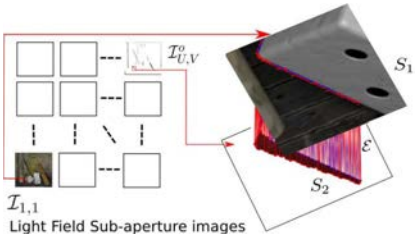


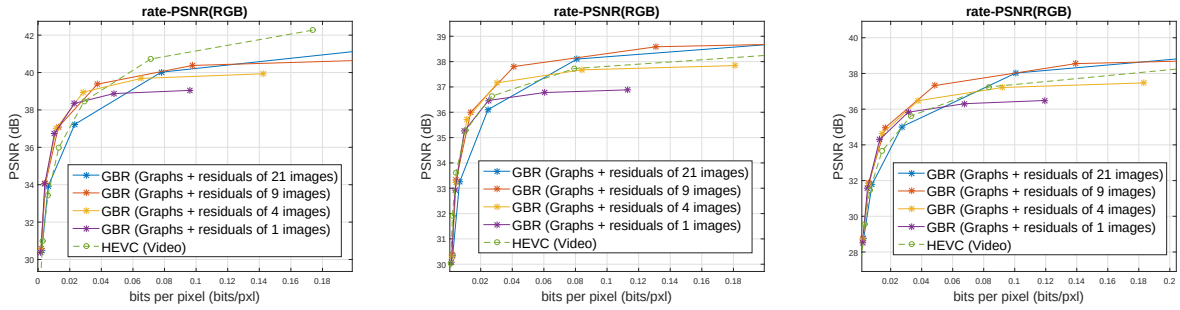
Fig. 3. A part of the graph drawn between the pixels of the reference image $\mathcal{I}_{1,1}$ and the disocclusions image $\mathcal{I}^o_{U,V}$. Red and Blue connections have 1 and 0.5 as weights respectively

Because the decoder already received the disparity information in the graph-based representation, it can deduce the exact same locations of disoccluded pixels in the target image as the encoder. It builds the same 4-neighbors graph connecting the disoccluded pixels, computes the edge weights using the disparity information and derives the same transform basis.This computation is required only for few blocks containing the disoccluded pixels. Also, there is no need to send additional side information as done in edge-adaptive approaches [20, 21].

Finally, the remaining views $\mathcal{I}_{u,v}$ are coded as follows. They are first predicted using the graph-based representation. Then, a residual is computed with the true $\mathcal{I}_{u,v}$. This residual is further compressed with HEVC.

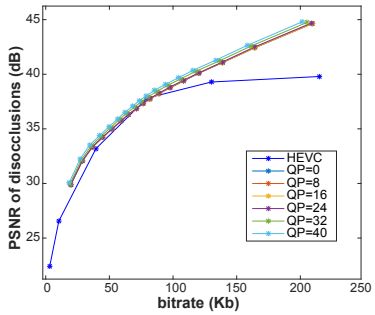
5. EXPERIMENTS

We test our GBR on synthetic LF (with $U = 9$, $V = 9$, $X = 768$ and $Y = 768$) from the dataset in [14] rendered with Blender [15]. Three datasets, called *Buddha*, *butterfly* and *monasRoom*, have been tested here.

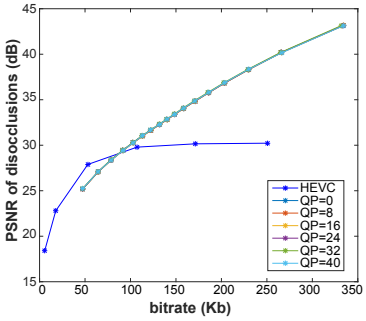


(a) *buddha* (b) *butterfly* (c) *monasRoom*

Fig. 5. PSNR-rate performance of the proposed GBR on different datasets, (a) *buddha*, (b) *butterfly* and (c) *monasRoom*.



(a) *Buddha*



(b) *monasRoom*

Fig. 4. Results of coding the disoccluded pixels using a Graph-based approach(QF from 10 to 90) vs HEVC (QP 0 to 40) for *Buddha* and *monasRoom*

5.1. Evaluation of GFT

To show the interest of exploiting inter-view neighboring relations(i.e graph edges) in coding the disocclusions, we first compare the performance of our graph-based compression scheme against HEVC inter-coding. We first code the disoccluded parts along with the reference view as a video sequence using HEVC. We vary the QP from 0 to 40. For each QP, a prediction of the disocclusions is computed(Sec. 4), then the residuals are coded while varying the quality factor from 10 to 90. The bitrate is the one needed to code the disocclusions. The PSNR is measured taking as reference the original disocclusions color values. From the results (Fig. 4), we notice that our approach outperforms HEVC with a higher PSNR for most QP values while preserving acceptable bitrates. Our diffusion method yields a good prediction with *Buddha* since the background mostly consists of smooth regions, and that explains the better coding performance. Whereas for *monasRoom*, the background is made of texture and wrong color values are propagated to the disoccluded areas resulting in residuals harder to code.

5.2. Light field representation and compression

We perform the GBR representation with fixed Lagrangian multiplier $\alpha = 0.5$ in Eq.(2). In this case, the graph sparsification highly depends on the distortion term $\mathcal{D}(\mathcal{E})$. The number of sub-aperture images selected to add residuals is chosen as $\{1, 4, 9, 21\}$ with a regular sub-sampling pattern. The baseline method is the scheme which directly compresses the whole LF dataset as a video sequence with HEVC. Fig.5 shows the PSNR-rate performance of the proposed GBR on different datasets. At low bitrate, the proposed GBR can yield PSNR-rate gain. However, at high bitrate, the GBR scheme is outperformed by HEVC, due to the limited number of selected sub-aperture images. More results (including visual results of rendered views) can be found on the web page `https://www.irisa.fr/temics/demos/lightField/GBR/GBR_LF_2017.html`. The proposed method is only tested on the synthetic light fields data, since the accurate depth or disparity information is needed.

6. CONCLUSION

In this paper, we have adapted the graph based representation (GBR) [10] to represent light fields (LF). The weighted Graph Fourier Transform (wGFT) is applied on the constructed graph to code the disoccluded pixels. To improve the rendering quality, the residuals of a sub-set of views are added into the graph and further used to render the other views of the LF. Experimental results show rate-distortion gain compared with HEVC based compression. For future work, we will focus on the application of our method to the real light fields.

7. ACKNOWLEDGEMENT

This project has been supported in part by the EU H2020 Research and Innovation Programme under grant agreement No 694122 (ERC advanced grant CLIM).

8. REFERENCES

[1] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy, "High performance imaging using large camera arrays," in *ACM Transactions on Graphics (TOG)*. ACM, 2005, vol. 24, pp. 765–776.

[2] Ren Ng, *Light field photography*, Ph.D. thesis, Stanford University, 2006.

[3] Todor Georgiev, Georgi Chunev, and Andrew Lumsdaine, "Superresolution with the focused plenoptic camera," in *Computational Imaging*, 2011, pp. 78 730X–78 730X–13.

[4] Gavin Miller, Steven Rubin, and Dulce Ponceleon, "Lazy decompression of surface light fields for precomputed global illumination," in *Rendering Techniques 98*, pp. 281–292. Springer, 1998.

[5] Marcus Magnor and Bernd Girod, "Data compression for light-field rendering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 3, pp. 338–343, 2000.

[6] Chuo-Ling Chang, Xiaoqing Zhu, Prashant Ramanathan, and Bernd Girod, "Light field compression using disparity-compensated lifting and shape adaptation," *IEEE transactions on image processing*, vol. 15, no. 4, pp. 793–806, 2006.

[7] Yun Li, Marten Sjostrom, Roger Olsson, and Ulf Jennehag, "Efficient intra prediction scheme for light field image compression," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 539–543.

[8] Xiaoran Jiang, Mikaël Le Pendu, Reuben A Farrugia, Sheila S Hemami, and Christine Guillemot, "Homography-based low rank approximation of light fields for compression," in *IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

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

[10] Xin Su, Thomas Maugey, and Christine Guillemot, "Rate-Distortion Optimized Graph-Based Representation for Multiview Images With Complex Camera Configurations," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2644–2655, 2017.

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

[12] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[13] Emilie Bosc, Vincent Jantet, Muriel Pressigout, Luce Morin, and Christine Guillemot, "Bit-rate allocation for multi-view video plus depth," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*. IEEE, 2011, pp. 1–4.

[14] Sven Wanner, Stephan Meister, and Bastian Goldluecke, "Datasets and Benchmarks for Densely Sampled 4D Light Fields," in *VMV*. Citeseer, 2013, pp. 225–226.

[15] Blender, "Blender," `https://www.blender.org/`, [Online].

[16] Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 673–680.

[17] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon, "Accurate depth map estimation from a lenslet light field camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1547–1555.

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

[19] Jesse D Kornblum, "Using JPEG quantization tables to identify imagery processed by software," *Digital Investigation*, vol. 5, pp. S21–S25, 2008.

[20] Godwin Shen, W-S Kim, Sunil K Narang, Antonio Ortega, Jaejoon Lee, and Hocheon Wey, "Edge-adaptive transforms for efficient depth map coding," in *Picture Coding Symposium (PCS), 2010*. IEEE, 2010, pp. 566–569.

[21] Hilmi E Egilmez, Amir Said, Yung-Hsuan Chao, and Antonio Ortega, "Graph-based transforms for inter predicted video coding," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3992–3996.

Rate-Distortion Optimized Super-Ray Merging for Light Field Compression

Xin Su
INRIA
Rennes, FRANCE
Xin.Su@inria.fr

Mira Rizkallah
IRISA
Rennes, FRANCE
Mira.Rizkallah@irisa.fr

Thomas Maugey
INRIA
Rennes, FRANCE
Thomas.Maugey@inria.fr

Christine Guillemot
INRIA
Rennes, FRANCE
Christine.Guillemot@inria.fr

***Abstract*—In this paper, we focus on the problem of compressing dense light fields which represent very large volumes of highly redundant data. In our scheme, view synthesis based on convolutional neural networks (CNN) is used as a first prediction step to exploit inter-view correlation. Super-rays are then constructed to capture the inter-view and spatial redundancy remaining in the prediction residues. To ensure that the super-ray segmentation is highly correlated with the residues to be encoded, the super-rays are computed on *synthesized residues* (the difference between the four transmitted corner views and their corresponding synthesized views), instead of the synthesized views. Neighboring super-rays are merged into a larger super-ray according to a rate-distortion cost. A 4D shape adaptive discrete cosine transform (SA-DCT) is applied per super-ray on the prediction residues in both the spatial and angular dimensions. A traditional coding scheme consisting of quantization and entropy coding is then used for encoding the transformed coefficients. Experimental results show that the proposed coding scheme outperforms HEVC-based schemes at low bitrate.**

***Index Terms*—Super-Ray (SR) Merging, Rate-Distortion Minimization, Light Field (LF) Compression, Shape-Adaptive DCT (SA-DCT)**

I. Introduction

Light fields (LF) are defined as the representation of radiance of light rays emitted along several directions by the different points in a 3D scene. Several devices have been developed for light fields capture, either based on camera arrays [1], on single moving cameras, or on arrays of microlenses [2], *etc*. Light fields have recently gained in popularity due to the variety of potential applications in computational photography and computer vision, however they represent very large volumes of data with challenges in terms of storage, transmission and processing.

In this paper, we focus on the problem of compressing dense light fields captured by plenotic cameras. First methods for compressing synthetic light fields appeared late 90's essentially based on classical coding tools as vector quantization or using JPEG coding for each sub-aperture view, yielding however limited compression performances. It is only recently that compression solutions have been proposed for dense real light fields captured by plenoptic cameras. The proposed solutions can be classified into two categories: either coding the array of sub-aperture images extracted from the lenslet image as a pseudo video sequence in [3], [4], or directly encoding the lenslet images captured by plenoptic cameras In [4]–[11], with extensions of HEVC with dedicated prediction modes. Multiview video compression and disparity compensation techniques are considered in [7]. A homography-based low rank approximation [12] is used to exploit angular correlation of LF. Besides being represented and encoded as images or videos, the LF is represented by 4D Gaussian mixture models in [13] and by graphs containing minimum amount of color and disparity information in [14].

In this paper, we propose a compression scheme based on view synthesis. Four corner views of the LF are first encoded by HEVC-Inter and transmitted. The whole LF is then synthesized from the four corner views using the convolutional neural networks (CNN) based architecture proposed in [15]. The prediction residues are then transformed using a 4D-shape adaptive Discrete Cosine Transform (4D SA-DCT) which exploits both spatial and angular correlation remaining in the residue signals. The support of the 4D SA-DCT is defined by a segmentation of the light field into super-rays. Super-rays can be seen as a set of super-pixels that are coherent across all light field views, taking into account disparity information. Note that local transforms have also been investigated for light fields compression in [16], however, the support of the local transform was defined by co-located super-pixels in the different views, not taking into account disparity.

To ensure that the super-ray segmentation is highly correlated with the residual signals, the super-rays are computed on *synthesized residues* (the difference between the four transmitted corner views and their corresponding synthesized views). Neighboring super-rays with similar homogeneous residues are merged into a larger super-ray to have a better spatial energy compaction by optimizing a rate-distortion cost. Experimental results show that the proposed coding scheme yields rate-distortion gains at low bitrates (e.g. < 0.04 bpp corresponding to a PSNR quality up to 35 dB) compared with HEVC-based coding schemes, while being comparable or slightly worse at higher bitrates.

II. Light Field Coding Scheme

A. Scheme Overview

Fig.1 depicts the proposed coding scheme. Let $\mathbf{LF} = \{I_{u,v}\}$ denote a light field, where $u = 1, \ldots, U$ and $v = 1, \ldots, V$ are the view indices. Four views at the corners $\mathbf{LF}^{\text{cor}} = \{I_{1,1}, I_{1,V}, I_{U,1}, I_{U,V}\}$ are encoded using HEVC-Inter and used to synthesize the whole light field with the

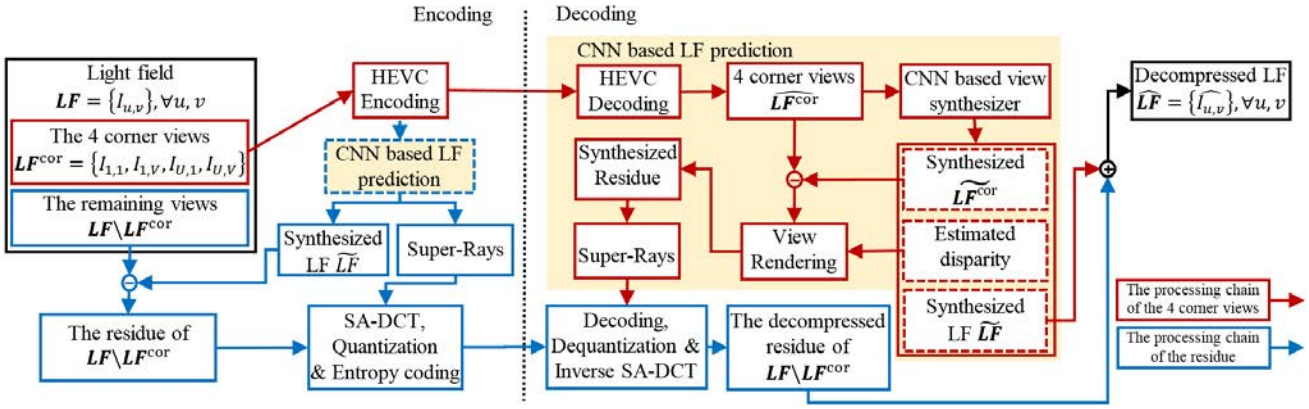


Fig. 1. Overview of proposed coding scheme.

CNN based synthesis method [15], as shown in Fig.1 (red arrows). To improve the quality of the synthesized light field, the residuals between the synthesized and original views are encoded by applying local super-ray based shape adaptive DCT (SA-DCT) (see Fig.1, blue arrows).

1) The processing chain of the four corner views: At the decoder, the decompressed four corner views are used to synthesize the whole LF using the CNN-based architecture of [15], as shown by the yellow region in Fig.1. The first CNN is trained to model the disparity from the four input views, while the second CNN is used to estimate the color of the synthesized views. The synthesis quality depends on the QP value of the HEVC-inter coder.

As shown by the yellow region in Fig.1, we compute the residual signals as the difference between the decompressed four corner views $\widehat{\mathbf{LF}^{\text{cor}}}$ and their corresponding synthesized views $\widetilde{\mathbf{LF}^{\text{cor}}}$. The four images of residues are then warped onto the other views using the disparity estimated by the CNN. The super-ray segmentation is then computed by applying the SLIC algorithm [17] on the set of residue images, but also taking into account disparity when performing the clustering. Computing the segmentation on the residue images rather than on the synthesized views, similar residues signals are more likely to be grouped into one segment which can benefit the following energy compaction in transform domain.

2) The processing chain of the residues: The synthesized views $\widetilde{\mathbf{LF}}$ and the super-ray segmentation are computed in the same manner at the encoder. We apply a local spatial SA-DCT on the residuals for each view using the super-ray as a support of the transform. Spatial SA-DCT coefficients of each super-ray corresponding to the same frequency form a $U \times V$ block in the angular domain, on which a second angular SA-DCT is applied to capture angular dependencies. A traditional coding scheme consisting of quantization and entropy coding is then used for encoding the transformed coefficients. At the decoder side, the decoded residuals are added to the synthesized views to obtain the final decompressed LF.

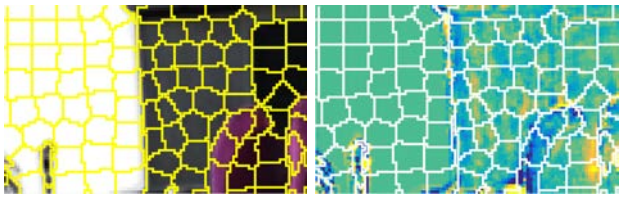
B. Super-Ray Segmentation on Residues

The concept of super-ray has been introduced in [18] as an extension of super-pixels [17] to group light rays coming from the same 3D object, i.e. to group pixels having similar color values and being close spatially in the 3D space. While the authors in [18] estimate disparity only at the centroid of the super-rays, here we consider a scheme using dense disparity maps to synthesize the entire light field from a sparse set of views. The disparity maps used in the tests reported below, have been estimated by the first CNN of the view synthesis architecture of [15] from the four corner views which are available at both the encoder and decoder. Having a dense disparity map for each view, the pixels in all the views are clustered using a method similar to SLIC [17] with a weighted combination of a color distance, a spatial distance and in terms of depth.

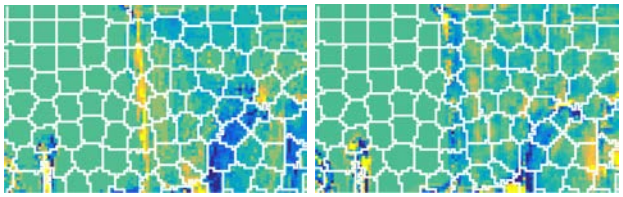
Fig.2 (a) shows that the super-rays computed on the set of synthesized color images are well aligned with the edges of the objects, but are not really suited for the residue images they are supposed to represent (see the edge of the building). Indeed the residues generally lie on both sides of object edges. Since the real residue signal is not available at the decoder, we generate an estimation of it (called synthesized residue), that is used to build a more accurate segmentation. For that purpose, the residue images at the four corner views are first computed as the difference between the coded/decoded corner views and their synthesized versions (see Fig.1). The residue images at the other viewpoints are then obtained by warping the corner residue images to the other positions using the estimated disparity. We see in Fig.2 (b) that the super-ray segmentation computed on these synthesized residue images has better correlation with the real residual signals, which is proved by the energy compaction comparison between Fig.2 (a) and (b) in section IV-A. Since the synthesized residues are available at both the encoder and decoder, we can obtain the same super-ray segmentation on both sides, and do not need to transmit it.

C. 4D Shape Adaptive DCT (SA-DCT)

While 2D DCT applied on a square or rectangular support may fail to capture correlation at image edges, we consider here a separable 4D shape-adaptive DCT with a support defined by the super-ray segmentation. Fig.3 illustrates how a 4D SA-DCT is applied on the i-th super-ray $\mathbf{SR}_i$ in the LF.



(a) Super-ray segmentation computed on synthesized color is shown with corresponding color image (left) and real residual image (right).



(b) Super-ray segmentation computed on synthesized residues is shown with corresponding synthesized residual image (left) and real residual image (right).

Fig. 2. Super-ray segmentation computed on (a) synthesized color and (b) synthesized residues. Only the center view ($u = 4, v = 4$) has been shown here. Segments in (a) are well aligned with the edges of color image, however, not aligned with the discontinuities within the residue images, see the border of the building. Segments in (b) obtained using synthesized residues better follow the discontinuities of the real residue signals.

A spatial 2D SA-DCT is first applied per view on each super-ray (i.e. on the super-pixel of the view which belongs to the considered super-ray). In each view, the obtained coefficients form a rectangular block with non-zero coefficients only in the top-left area with the DC component located at the left-top corner, as shown in the middle of Fig.3. These coefficients are sorted from low to high frequency by following a Zig-Zag order. Coefficients corresponding to the same frequency band but from different views form an $U \times V$ block in angular domain, as shown on the right of Fig.3. Another SA-DCT, *i.e.* an angular SA-DCT, is then applied on this block. Note that some values may be missing in the $U \times V$ block, since the size of $\mathbf{SR}_i$ varies in different views. Generally, the 4D DCT $\mathfrak{T}(\cdot)$ can be computed as

$$\begin{aligned} \{X_{i,b}\} &= \mathfrak{T}(\mathbf{SR}_i) \ , \\ \mathfrak{T}(\cdot) &= \underbrace{\mathbf{DCT}_u \otimes \mathbf{DCT}_v}_{\text{Spatial SA-DCT}} \otimes \underbrace{\mathbf{DCT}_y \otimes \mathbf{DCT}_x}_{\text{Angular SA-DCT}} \end{aligned} \quad (1)$$

where $\{X_{i,b}\} \in \mathbb{R}^{U \times V \times N}$, N is the maximum size of $\mathbf{SR}_i$ in different views, $b = 1, 2, \cdots, U \times V \times N$. For some values of b, $X_{i,b}$ may be missing due to the non-regular shape of $\mathbf{SR}_i$. The positions of missing elements in $\{X_{i,b}\}$ are available at both encoder and decoder, since the super-ray segmentation is known. $\mathbf{DCT}_* \in \mathbb{R}^{n \times n}$ corresponds to a n-point DCT (n is the number of elements of $\mathbf{SR}_i$ in corresponding coordinate), and $\otimes$ denotes a Kronecker product operator.

D. Quantization and Entropy Coding

At the end of those two transform stages, coefficients are grouped into a 2-dimensional array $\mathbf{X}$ where $\mathbf{X}(i, b)$ is the b-th band in super-ray $\mathbf{SR}_i$. Using the observations on all the super-rays in a training dataset (*Rose* [12]), we can find the best ordering for quantization. We first sort the variances of coefficients with enough observations in decreasing order. We then split them into 64 groups $\mathbf{g}$: $\{\mathbf{X}_{i,b}, \forall i, b \in \mathbf{g}\}$. All the remaining coefficients with less observations will be considered in the last group. We use the zigzag ordering of the *JPEG* quantization matrix to assign the quantization step size Q for each. A simple rounding procedure then results in the quantized coefficients $\mathbf{X}_q(i, b) = round(\mathbf{X}(i, b)/\boldsymbol{Q}(\mathbf{X}(i, b))$. that are further coded using an arithmetic coder.

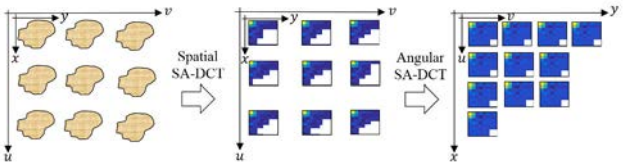


Fig. 3. An illustration of 4D SA-DCT performed on super-ray.

III. Rate-Distortion Optimized Super-Ray Merging

In order to increase compression performances, we improve the super-ray segmentation with a rate-distortion optimized super-ray merging. Four initial segmentations are performed using different initial numbers of clusters leading to different super-ray sizes as shown in Fig.4.a. The resulting segmentations are referred to as layers. Note that we modify the super-ray segmentation in layer l respecting to the boundaries of super-rays in layer $l-1$, to make sure the boundaries are coherent at different layers. For instance, if super-ray $\mathbf{SR}_i^{l-1}$ in layer $l-1$ is across two (or maybe more) super-rays in layer l, one of these super-rays in layer l is enlarged to completely contain $\mathbf{SR}_i^{l-1}$, and the other super-rays are reduced correspondingly. We choose to enlarge the super-ray that initially contains most parts of $\mathbf{SR}_i^{l-1}$.

The merging results $\{SR_i\}$ are initialized by the super-rays $\{SR_i^{l=0}, \forall i\}$ at layer 0 and the merging starts from layer 1 to layer 3. At each time, we only consider one super-ray SR_i^l at layer l, which consists of several super-rays $\{SR_j^{l-1}\}$ at layer $l-1$, *i.e.* $SR_i^l = \{SR_j^{l-1}\} = \{SR_j^{l-1} \in SR_i^l, \forall j\}$, as shown in Fig.4 (b). $\{SR_j^{l-1}\}$ will be merged into SR_i^l, *i.e.* $\{SR_j^{l-1}\} \Rightarrow SR_i^l$, if and only the rate-distortion cost $\mathcal{J}$ reduces after merging.

The rate cost of the 4D SA-DCT coefficients (after quantization) is modeled by their entropy computed per coefficient group as

$$R = \sum_{\mathbf{g}} \text{entropy}\left(\{Q(X_{i,b}), \forall i, b \in \mathbf{g}\}\right). \quad (2)$$

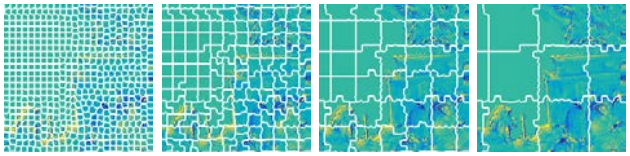
The distortion is computed as the distortion of SA-DCT coefficients before and after quantization as

$$D = \sum_i \sum_b \left[X_{i,b} - Q(X_{i,b})\right]^2 \ . \quad (3)$$

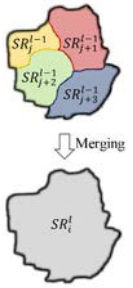
Thus, the rate-distortion cost is

$$\mathcal{J} = D + \lambda R \ , \quad (4)$$

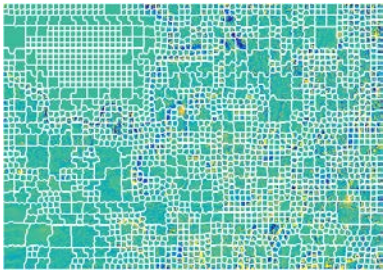
where $\lambda = 1$ represents the relation between rate and distortion. The merging procedure based on the minimization of the rate-distortion cost $\mathcal{J}$ is detailed in Algorithm 1. Fig.4 (c) gives an illustration of merging results using *Flower1* dataset in [15].



(a) Super-rays in different layers.



(b) Merging.



(c) Merging result of *Flower1* [15].

Fig. 4. An illustration of super-ray merging. (a) Super-rays at different layer $l = 0, 1, 2, 3$ from left to right. (b) Super-rays $\{SR_j^{l-1}\} = \{SR_j^{l-1} \in SR_i^l, \forall j\}$ in layer $l-1$ are merged into SR_i^l, *i.e.* $\{SR_j^{l-1}\} \Rightarrow SR_i^l$, if the rate-distortion cost $\mathcal{J}$ has been reduced. (c) A merging result of *Flower1* [15].

Algorithm 1: Merging based on rate-distortion minimization

Data: Super-rays at different layer $\{SR_i^{l=0}\}$, $\{SR_i^{l=1}\}$, $\{SR_i^{l=2}\}$, $\{SR_i^{l=3}\}$
Result: Merged super rays $\{SR_i\}$
Initialization: $\{SR_i\} = \{SR_i^{l=0}\}$;
for *Layer* $l = 1$ *to* 3 **do**
  for *Each super-ray* SR_i^l *in layer* l **do**
    Compute $\mathcal{J}$ with $\{SR_i\}$ **Eq.4**;
    Find super-rays $\{SR_j^{l-1}\} = \{SR_j^{l-1} \in SR_i^l, \forall j\}$ in layer $l-1$;
    Compute $\mathcal{J}'$ with $\{\{SR_i\} \setminus \{SR_j^{l-1}\}\} \bigcup SR_i^l$, *i.e.* $\{SR_j^{l-1}\}$ are replaced by SR_i^l **Eq.4**;
    if $\mathcal{J} > \mathcal{J}'$ **then**
      Merge, $\{SR_j^{l-1}\} \Rightarrow SR_i^l$;
      Update $\{SR_i\}$;
    end
  end
end

IV. Experiments

We test our coding scheme on four real LF with 8×8 sub-aperture images of size $(X = 536, Y = 376)$ from the dataset used in [15], called *Flower1*, *Flower2* and *Cars*.

A. Energy Compaction

We first evaluate the effectiveness of our 4D segmentation, by analyzing the compaction of the energy after transformation. Therefore, 4D SA-DCT is applied on: 1) super-pixels used in [16] (computed on synthesized color without disparity compensation or merging) 2) super-rays computed on synthesized residues without disparity compensation or merging 3) super-rays computed on synthesized residues with disparity compensation but without merging 4) super-rays computed on synthesized residues with disparity compensation and merging. Fig. 5 shows the percentage of energy carried by a given percentage of coefficients obtained by SA-DCT on these segmentations. The blue curve is the baseline method [16]. The red curves show the impact of using the synthesized residues to compute the segmentation. However, due the error in the synthesized residue, the improvement is limited. The Yellow curve shows the impact of using the disparity information, while the purple curve measures the effect of the merging. Thanks to the merging operation which compensates the errors in the super-ray segmentation, the proposed contributions bring a significant increase in terms of energy compaction ($\sim$10%) with respect to a direct use of a super-pixel segmentation per view [16].

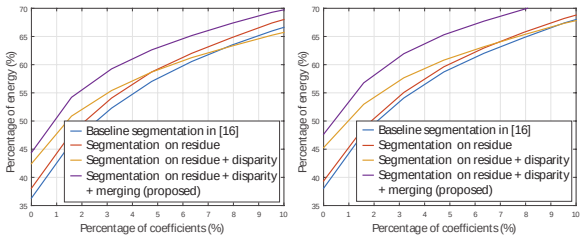


Fig. 5. Energy compaction of the transformed coefficients of *Flower1* and *Cars* using different super-ray computations. The super-rays computed on synthesized residues with merging and disparity compensation yield higher energy compaction.

B. Rate-PSNR results

The final rate-distortion performance of the proposed scheme is evaluated in comparison with three baseline methods: 1) *HEVC-lozenge* [3], the whole LF is considered as a video and compressed by HEVC with a lozenge sequence, 2) *CNN+HEVC* [3], the same CNN based view synthesis is applied here, while the residues are compressed by HEVC, 3) *CNN+SA-DCT (no merging, no disparity)* [16], our previous coding scheme presented in [16] using the same CNN based view synthesis, however, there is no disparity compensation or super-merging strategy. Note that the coding methods using CNN based prediction are performed with best pairs of parameters (Q, QP) where Q is the quality parameter used to compress the residues and QP is used in the HEVC inter-coding of the four corners. The obtained rate-distortion curves are shown in Fig. 6.

The proposed CNN+SA-DCT coding scheme yields better or comparable rate-PSNR performance at low bitrate than HEVC based reference methods. The improvement of the proposed CNN+SA-DCT compared with the baseline method in [16] indicates the effectiveness of super-ray merging. It allows the proposed coding scheme to capture more information with fewer bits (at low bitrate), compared with HEVC

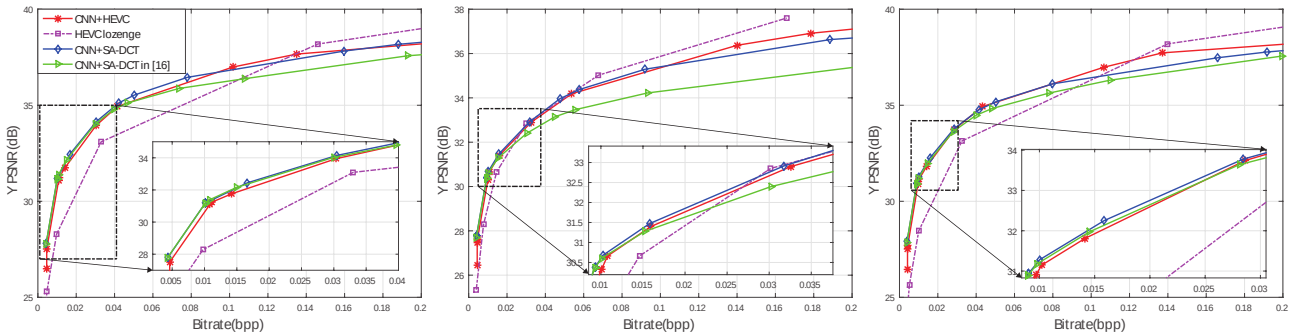


Fig. 6. Rate-distortion comparison. From left to right: *Flower 1*, *Cars* and *Flower 2*.

based encoders. Tab. I shows the improvement in terms of bjontegaard metric at low bitrate (< 0.04 bpp corresponding to a PSNR quality up to 35 dB) obtained by our coding scheme. However, as shown in Tab. I, at bitrates higher than 0.04 bpp, the HEVC based encoders (HEVC lozenge and CNN+HEVC) generally outperform the proposed coding scheme at high bitrates. This is due to the fact that the proposed scheme does not have very complex and high quality prediction strategies in residue coding which is useful at high bitrate.

TABLE I
BJONTEGAARD COMPARISON (ΔPSNR (DB)) AT *low* BITRATE (< 0.04 BPP) AND *high* BITRATE (> 0.04 BPP)

	Our CNN+SA-DCT *vs*					
	CNN+HEVC		HEVC lozenge		CNN+SA-DCT in [16]	
	Low	High	Low	High	Low	High
Flower 1	0.22	0.05	1.92	-0.19	0.03	0.55
Cars	0.21	-0.17	0.48	-0.50	0.17	1.08
Flower 2	0.1	-0.26	1.77	-0.85	0.09	0.38

V. CONCLUSION

In this paper, we have presented a rate-distortion optimized super-ray merging to exploit the correlation in the spatial and angular dimensions of light fields. After the CNN-based view synthesis, the residue inside each super-ray is compacted into a few coefficients using 4D shape-adaptive DCT transform. The experimental results show that the proposed light field coding scheme can yield rate-distortion gains compared with HEVC based compression, especially at low bitrate.

ACKNOWLEDGMENT

This work has been supported in part by the EU H2020 Research and Innovation Programme under grant agreement No 694122 (ERC advanced grant CLIM).

REFERENCES

[1] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 765–776, 2005.

[2] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report CSTR*, vol. 2, no. 11, pp. 1–11, 2005.

[3] M. Rizkallah, T. Maugey, C. Yaacoub, and C. Guillemot, "Impact of light field compression on focus stack and extended focus images," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 898–902.

[4] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression," in *Multimedia & Expo Workshops (ICMEW), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–4.

[5] C. Conti, P. Nunes, and L. D. Soares, "HEVC-based light field image coding with bi-predicted self-similarity compensation," in *Multimedia & Expo Workshops (ICMEW), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–4.

[6] C. Conti, L. D. Soares, and P. Nunes, "HEVC-based 3D holoscopic video coding using self-similarity compensated prediction," *Signal Processing: Image Communication*, vol. 42, pp. 59–78, 2016.

[7] Y. Li, M. Sjostrom, R. Olsson, and U. Jennehag, "Efficient intra prediction scheme for light field image compression," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 539–543.

[8] R. Monteiro, L. Lucas, C. Conti, P. Nunes, N. Rodrigues, S. Faria, C. Pagliari, E. da Silva, and L. Soares, "Light field HEVC-based image coding using locally linear embedding and self-similarity compensated prediction," in *Multimedia & Expo Workshops (ICMEW), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–4.

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

[10] W. Ahmad, R. Olsson, and M. Sjöström, "Interpreting Plenoptic Images as Multi-View Sequences for Improved Compression," in *IEEE International Conference on Image Processing, Beijing, China 17-20 September 2017*, 2017.

[11] C. Jia, Y. Yang, X. Zhang, X. Zhang, S. Wang, S. Wang, and S. Ma, "Optimized Inter-View Prediction Based Light Field Image Compression With Adaptive Reconstruction," in *IEEE International Conference on Image Processing, Beijing, China 17-20 September 2017*, 2017.

[12] X. Jiang, M. Le Pendu, R. A. Farrugia, and C. Guillemot, "Light Field Compression With Homography-Based Low-Rank Approximation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1132–1145, 2017.

[13] R. Verhack, T. Sikora, L. Lange, R. Jongebloed, G. Van Wallendael, and P. Lambert, "Steered mixture-of-experts for light field coding, depth estimation, and processing," in *Multimedia and Expo (ICME), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1183–1188.

[14] X. Su, M. Rizkallah, T. Maugey, and C. Guillemot, "Graph-based light fields representation and coding using geometry information," in *IEEE International Conference on Image Processing (ICIP)*, 2017.

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

[16] M. Rizkallah, X. Su, T. Maugey, and C. Guillemot, "Graph-based Transforms for Predictive Light Field Compression based on Super-Pixels," in *IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

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

[18] M. Hog, N. Sabater, and C. Guillemot, "Superrays for Efficient Light Field Processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1187–1199, 2017.

Bibliography

[1] Radhakrishna Achanta et al., "SLIC superpixels compared to state-of-the-art superpixel methods", *in*: *IEEE transactions on pattern analysis and machine intelligence* 34.*11* (2012), pp. 2274–2282.

[2] Edward H Adelson, James R Bergen, et al., "The plenoptic function and the elements of early vision", *in*: ().

[3] Edward H Adelson and John Y. A. Wang, "Single lens stereo with a plenoptic camera", *in*: *IEEE Transactions on Pattern Analysis & Machine Intelligence 2* (1992), pp. 99–106.

[4] Waqas Ahmad, Roger Olsson, and Mårten Sjöström, "Interpreting Plenoptic Images as Multi-View Sequences for Improved Compression", *in*: *IEEE International Conference on Image Processing, Beijing, China 17-20 September 2017*, 2017.

[5] Patrice Rondao Alface, Jean-François Macq, and Nico Verzijp, "Interactive omnidirectional video delivery: A bandwidth-effective approach", *in*: *Bell Labs Technical Journal* 16.*4* (2012), pp. 135–147.

[6] I. Bauermann, "H.264 BASED CODING OF OMNIDIRECTIONAL VIDEO", *in*: *International Conference on Computer Vision and Graphics*, 2004.

[7] Mikhail Belkin, Irina Matveeva, and Partha Niyogi, "Regularization and semi-supervised learning on large graphs", *in*: *COLT*, vol. 3120, Springer, 2004, pp. 624–638.

[8] Robert C Bolles, H Harlyn Baker, and David H Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion", *in*: *International journal of computer vision* 1.*1* (1987), pp. 7–55.

[9] James C Bremer et al., "Diffusion wavelet packets", *in*: *Applied and Computational Harmonic Analysis* 21.*1* (2006), pp. 95–112.

[10] Arnaud Casteigts et al., "Time-varying graphs and dynamic networks", *in*: *International Journal of Parallel, Emergent and Distributed Systems* 27.*5* (2012), pp. 387–408.

[11] Shenchang Eric Chen, "Quicktime VR: An image-based approach to virtual environment navigation", *in*: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM, 1995, pp. 29–38.

[12] Zhenzhong Chen, Yiming Li, and Yingxue Zhang, "Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation", *in*: *Signal Processing* 146 (2018), pp. 66–78, ISSN: 0165-1684.

[13] Zhenzhong Chen, Yiming Li, and Yingxue Zhang, "Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation", *in*: *Signal Processing* 146 (2018), pp. 66–78.

[14] G. Cheung et al., "Graph Spectral Image Processing", *in*: *Proceedings of the IEEE* 106.*5* (May 2018), pp. 907–930, ISSN: 0018-9219, DOI: 10.1109/JPROC.2018.2799702.

[15] Philip A. Chou and Ricardo L. de Queiroz, "GAUSSIAN PROCESS TRANSFORMS", *in*: *Submitted for possible publication in Int'l Conf. on Image Processing (ICIP)*, IEEE Institute of Electrical and Electronics Engineers, Sept. 2016, URL: http://research.microsoft.com/apps/pubs/default.aspx?id=260803.

[16] Fan RK Chung, *Spectral graph theory*, vol. 92, American Mathematical Soc., 1997.

[17] Ronald R Coifman and Mauro Maggioni, "Diffusion wavelets", *in*: *Applied and Computational Harmonic Analysis* 21.1 (2006), pp. 53–94.

[18] Caroline Conti, Paulo Nunes, and Luis Ducla Soares, "New HEVC prediction modes for 3D holoscopic video coding", *in*: *2012 19th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2012, pp. 1325–1328.

[19] Caroline Conti, Paulo Nunes, and Luís Ducla Soares, "HEVC-based light field image coding with bi-predicted self-similarity compensation", *in*: *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, 2016, pp. 1–4.

[20] Caroline Conti, Luís Ducla Soares, and Paulo Nunes, "HEVC-based 3D holoscopic video coding using self-similarity compensated prediction", *in*: *Signal Processing: Image Communication* 42 (2016), pp. 59–78.

[21] Mark Crovella and Eric Kolaczyk, "Graph wavelets for spatial traffic analysis", *in*: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, IEEE, 2003, pp. 1848–1857.

[22] Łukasz Dąbała et al., "Efficient Multi-image Correspondences for On-line Light Field Video Processing", *in*: *Computer Graphics Forum*, vol. 35, 7, Wiley Online Library, 2016, pp. 401–410.

[23] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression", *in*: *2012 19th IEEE International Conference on Image Processing*, Sept. 2012, pp. 1541–1544, DOI: 10.1109/ICIP.2012.6467166.

[24] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression", *in*: *2012 19th IEEE International Conference on Image Processing*, Sept. 2012, pp. 1541–1544, DOI: 10.1109/ICIP.2012.6467166.

[25] Pierre David, Mikaël Le Pendu, and Christine Guillemot, "White lenslet image guided demosaicing for plenoptic cameras", *in*: *IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), 2017*, IEEE, 2017, pp. 1–6.

[26] Ronald A DeVore, Björn Jawerth, and Bradley J Lucier, "Image compression through wavelet transform coding", *in*: *IEEE Transactions on information theory* 38.2 (1992), pp. 719–746.

[27] *Digital compression and coding of continuoustone still images*, ISO/IEC IS 10918-1 ITU-T.

[28] Xiaowen Dong et al., "Learning Graphs from Data: A Signal Representation Perspective", *in*: *CoRR* abs/1806.00848 (2018), arXiv: 1806.00848, URL: http://arxiv.org/abs/1806.00848.

[29] Xiaowen Dong et al., "Learning Laplacian Matrix in Smooth Graph Signal Representations", *in*: *arXiv preprint arXiv:1406.7842* (2014).

[30] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph Learning From Data Under Laplacian and Structural Constraints", *in*: *IEEE Journal of Selected Topics in Signal Processing* 11.6 (Sept. 2017), pp. 825–841, ISSN: 1932-4553, DOI: 10.1109/JSTSP.2017.2726975.

[31] Hilmi E. Egilmez, Eduardo Pavez, and Antonio Ortega, "Graph Learning from Data under Structural and Laplacian Constraints", *in*: *CoRR* abs/1611.05181 (2016), URL: http://arxiv.org/abs/1611.05181.

[32] Hilmi E Egilmez et al., "Graph-based transforms for inter predicted video coding", *in*: *Image Processing (ICIP), 2015 IEEE International Conference on*, IEEE, 2015, pp. 3992–3996.

[33] Giulia Fracastoro and Enrico Magli, "Predictive graph construction for image compression", *in*: *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2015, pp. 2204–2208.

[34] Giulia Fracastoro, Dorina Thanou, and Pascal Frossard, "Graph transform learning for image compression", *in*: *2016 Picture Coding Symposium, PCS 2016, Nuremberg, Germany, December 4-7, 2016*, 2016, pp. 1–5, DOI: 10.1109/PCS.2016.7906368.

[35] Giulia Fracastoro et al., "Superpixel-driven graph transform for image compression", *in*: *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2015, pp. 2631–2635.

[36] H. Freeman, "On the Encoding of Arbitrary Geometric Configurations", *in*: *IRE Transactions on Electronic Computers* EC-10.2 (June 1961), pp. 260–268, ISSN: 0367-9950, DOI: 10.1109/TEC.1961.5219197.

[37] H. Freeman, "On the Encoding of Arbitrary Geometric Configurations", *in*: *IRE Transactions on Electronic Computers* EC-10.2 (June 1961), pp. 260–268, ISSN: 0367-9950, DOI: 10.1109/TEC.1961.5219197.

[38] Chi-Wing Fu et al., "The rhombic dodecahedron map: An efficient scheme for encoding panoramic video", *in*: *IEEE Transactions on Multimedia* 11.4 (2009), pp. 634–644.

[39] Tarek El-Ganainy and Mohamed Hefeeda, "Streaming virtual reality content", *in*: *arXiv preprint arXiv:1612.08350* (2016).

[40] Todor Georgiev, Georgi Chunev, and Andrew Lumsdaine, "Superresolution with the focused plenoptic camera", *in*: *Computational Imaging IX*, vol. 7873, International Society for Optics and Photonics, 2011, p. 78730X.

[41] Todor Georgiev and Andrew Lumsdaine, "Reducing plenoptic camera artifacts", *in*: *Computer Graphics Forum*, vol. 29, 6, Wiley Online Library, 2010, pp. 1955–1968.

[42] Vivek K Goyal, Jun Zhuang, and M Veiterli, "Transform coding with backward adaptive updates", *in*: *IEEE Transactions on Information Theory* 46.4 (2000), pp. 1623–1633.

[43] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval, "Wavelets on graphs via spectral graph theory", *in*: *Applied and Computational Harmonic Analysis* 30.2 (2011), pp. 129–150.

[44] Philipp Helle et al., "Block merging for quadtree-based partitioning in HEVC", *in*: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (2012), pp. 1720–1731.

[45] M. Hog, N. Sabater, and C. Guillemot, "Super-rays for Efficient Light Field Processing", *in*: *IEEE J. on Selected Topics in Signal Processing, special issue on light field image processing* (Oct. 2017).

[46] Matthieu Hog, Neus Sabater, and Christine Guillemot, "Superrays for Efficient Light Field Processing", *in*: *IEEE Journal of Selected Topics in Signal Processing* 11.7 (2017), pp. 1187–1199.

[47] W. Hu et al., "Multiresolution Graph Fourier Transform for Compression of Piecewise Smooth Images", *in*: *IEEE Transactions on Image Processing* 24.1 (Jan. 2015), pp. 419–433, ISSN: 1057-7149, DOI: 10.1109/TIP.2014.2378055.

[48] Wei Hu et al., "Multiresolution graph Fourier transform for compression of piecewise smooth images", *in*: *IEEE Transactions on Image Processing* 24.1 (2015), pp. 419–433.

[49] Anil K Jain, "A sinusoidal family of unitary transforms", *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence 4* (1979), pp. 356–365.

[50] Tommy R Jensen and Bjarne Toft, *Graph coloring problems*, vol. 39, John Wiley & Sons, 2011.

[51] Chuanmin Jia et al., "Optimized inter-view prediction based light field image compression with adaptive reconstruction", *in*: *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, pp. 4572–4576.

[52] Xiaoran Jiang, Mikaël Le Pendu, and Christine Guillemot, "Depth estimation with occlusion handling from a sparse set of light field views", *in*: *IEEE International Conference on Image Processing (ICIP)*, 2018.

[53] Xiaoran Jiang, Mikaël Le Pendu, and Christine Guillemot, "Light Fields Compression Using Depth Image Based View Synthesis", *in*: *Hot3D workshop held jointly with IEEE Int. Conf. on Multimedia and Expo, ICME*, IEEE, July 2017.

[54] Xiaoran Jiang et al., "Homography-based low rank approximation of light fields for compression", *in*: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, IEEE, 2017, pp. 1313–1317.

[55] Xiaoran Jiang et al., "Light field compression with homography-based low-rank approximation", *in*: *IEEE Journal of Selected Topics in Signal Processing* 11.7 (2017), pp. 1132–1145.

[56] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi, "Learning-based view synthesis for light field cameras", *in*: *ACM Transactions on Graphics (TOG)* 35.6 (2016), p. 193.

[57] Changil Kim et al., "Scene reconstruction from high spatio-angular resolution light fields.", *in*: *ACM Trans. Graph.* 32.4 (2013), pp. 73–1.

[58] Woo-Shik Kim, Sunil K Narang, and Antonio Ortega, "Graph based transforms for depth video coding", *in*: *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, IEEE, 2012, pp. 813–816.

[59] Artiom Kovnatsky et al., "Coupled quasi-harmonic bases", *in*: *Computer Graphics Forum*, vol. 32, 2pt4, Wiley Online Library, 2013, pp. 439–448.

[60] Luc Le Magoarou, Rémi Gribonval, and Nicolas Tremblay, "Approximate fast graph fourier transforms via multilayer sparse approximations", *in*: *IEEE transactions on Signal and Information Processing over Networks* 4.2 (2018), pp. 407–420.

[61] Marc Levoy, "Light fields and computational imaging", *in*: *Computer* 39.8 (2006), pp. 46–55.

[62] Marc Levoy and Pat Hanrahan, "Light field rendering", *in*: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, 1996, pp. 31–42.

[63] Marc Levoy et al., "Synthetic aperture confocal imaging", *in*: *ACM Transactions on Graphics (ToG)*, vol. 23, 3, ACM, 2004, pp. 825–834.

[64] Yun Li, Roger Olsson, and Mårten Sjöström, "Compression of unfocused plenoptic images using a displacement intra prediction", *in*: *IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2016*, IEEE, 2016, pp. 1–4.

[65] D. Liu et al., "Pseudo-sequence-based light field image compression", *in*: *IEEE Int. Conf. on Multimedia Expo Workshops (ICMEW)*, July 2016, DOI: 10.1109/ICMEW.2016.7574674.

[66] Dong Liu et al., "Pseudo-sequence-based light field image compression", *in*: *Multimedia & Expo Workshops (ICMEW), 2016 IEEE International Conference on*, IEEE, 2016, pp. 1–4.

[67] T. Maugey, O. Le Meur, and Z. Liu, "Saliency-based navigation in omnidirectional image", *in*: *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, Oct. 2017, pp. 1–6, DOI: 10.1109/MMSP.2017.8122229.

[68] Leonard McMillan and Gary Bishop, "Plenoptic modeling: An image-based rendering system", *in*: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM, 1995, pp. 39–46.

[69] B. Motz, G. Cheung, and P. Frossard, "Graph-based representation and coding of 3D images for interactive multiview navigation", *in*: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 1155–1159, DOI: 10.1109/ICASSP.2016.7471857.

[70] Sunil K Narang, Yung-Hsuan Chao, and Antonio Ortega, "Critically sampled graph-based wavelet transforms for image coding", *in*: *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, IEEE, 2013, pp. 1–4.

[71] Sunil K Narang and Antonio Ortega, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs", *in*: *IEEE Transactions on Signal Processing* 61.*19* (2013), pp. 4673–4685.

[72] Sunil K Narang and Antonio Ortega, "Lifting based wavelet transforms on graphs", *in*: *Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference*, Asia-Pacific Signal, Information Processing Association, 2009 Annual Summit, and Conference, International Organizing Committee, 2009, pp. 441–444.

[73] Sunil K Narang and Antonio Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data", *in*: *IEEE Transactions on Signal Processing* 60.*6* (2012), pp. 2786–2799.

[74] Sunil K Narang et al., "Localized iterative methods for interpolation in graph structured data", *in*: *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, IEEE, 2013, pp. 491–494.

[75] R. Ng, "Light Field Photography", PhD thesis, Stanford University, 2006.

[76] Ren Ng et al., "Light field photography with a hand-held plenoptic camera", *in*: *Computer Science Technical Report CSTR* 2.*11* (2005), pp. 1–11.

[77] Bastien Pasdeloup et al., "Translations on graphs with neighborhood preservation", *in*: *CoRR* abs/1709.03859 (2017), arXiv: 1709.03859, URL: http://arxiv.org/abs/1709.03859.

[78] Eduardo Pavez et al., "GTT: Graph template transforms with applications to image coding", *in*: *Picture Coding Symposium (PCS), 2015*, IEEE, 2015, pp. 199–203.

[79] Kaare Brandt Petersen, Michael Syskind Pedersen, et al., "The matrix cookbook", *in*: *Technical University of Denmark* 7.15 (2008), p. 510.

[80] Robert Prevedel et al., "Simultaneous whole-animal 3D imaging of neuronal activity using light-field microscopy", *in*: *Nature methods* 11.7 (2014), p. 727.

[81] Markus Puschel and José MF Moura, "Algebraic signal processing theory: Foundation and 1-D time", *in*: *IEEE Transactions on Signal Processing* 56.8 (2008), pp. 3572–3585.

[82] Idan Ram, Michael Elad, and Israel Cohen, "Redundant wavelets on graphs and high dimensional data clouds", *in*: *IEEE Signal Processing Letters* 19.5 (2012), pp. 291–294.

[83] Mira Rizkallah et al., "Impact of light field compression on focus stack and extended focus images", *in*: *Signal Processing Conference (EUSIPCO), 2016 24th European*, IEEE, 2016, pp. 898–902.

[84] Ivano Rotondo et al., "Designing sparse graphs via structure tensor for block transform coding of images", *in*: *APSIPA ACS, Hong Kong, China* (2015).

[85] Raif Rustamov and Leonidas J Guibas, "Wavelets on graphs via deep learning", *in*: *Advances in Neural Information Processing Systems*, 2013, pp. 998–1006.

[86] Aliaksei Sandryhaila and Jose MF Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure", *in*: *IEEE Signal Processing Magazine* 31.5 (2014), pp. 80–90.

[87] Aliaksei Sandryhaila and José MF Moura, "Discrete signal processing on graphs", *in*: *IEEE Transactions on Signal Processing* 61.7 (2013), pp. 1644–1656.

[88] Daniel Scharstein and Richard Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", *in*: *International journal of computer vision* 47.1-3 (2002), pp. 7–42.

[89] Godwin Shen and Antonio Ortega, "Optimized distributed 2D transforms for irregularly sampled sensor network grids using wavelet lifting", *in*: *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008.* IEEE, 2008, pp. 2513–2516.

[90] Godwin Shen et al., "Edge-adaptive transforms for efficient depth map coding", *in*: *Picture Coding Symposium (PCS), 2010*, IEEE, 2010, pp. 566–569.

[91] Jianbo Shi and J. Malik, "Normalized cuts and image segmentation", *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (Aug. 2000), pp. 888–905, ISSN: 0162-8828, DOI: 10.1109/34.868688.

[92] David I Shuman et al., "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains", *in*: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 83–98.

[93] Thomas Sikora and Bela Makai, "Shape-adaptive DCT for generic coding of video", *in*: *IEEE Transactions on Circuits and Systems for Video Technology* 5.1 (1995), pp. 59–62.

[94] F. De Simone et al., "Geometry-driven quantization for omnidirectional image coding", *in*: *2016 Picture Coding Symposium (PCS)*, Dec. 2016, pp. 1–5, DOI: 10.1109/PCS.2016.7906402.

[95] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi, "The JPEG 2000 still image compression standard", *in*: *IEEE Signal processing magazine* 18.*5* (2001), pp. 36–58.

[96] Gilbert Strang, "The discrete cosine transform", *in*: *SIAM review* 41.*1* (1999), pp. 135–147.

[97] Xin Su et al., "Graph-based light fields representation and coding using geometry information", *in*: *IEEE International Conference on Image Processing (ICIP)*, 2017.

[98] Xin Su et al., "Rate-Distortion Optimized Super-Ray Merging for Light Field Compression", *in*: *European Signal Processing Conference (EUSIPCO)*, 2018.

[99] Ioan Tabus, Petri Helin, and Pekka Astola, "Lossy compression of lenslet images from plenoptic cameras combining sparse predictive coding and JPEG 2000", *in*: *Image Processing (ICIP), 2017 IEEE International Conference on*, 2017, pp. 4567–4571.

[100] Michael W Tao et al., "Depth from combining defocus and correspondence using light-field cameras", *in*: *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 673–680.

[101] Dorina Thanou, David I Shuman, and Pascal Frossard, "Learning parametric dictionaries for signals on graphs", *in*: *IEEE Transactions on Signal Processing* 62.*15* (2014), pp. 3849–3862.

[102] Ivana Tosic and Pascal Frossard, "Low bit-rate compression of omnidirectional images", *in*: *Proceedings of PCS*, EPFL-CONF-130365, 2009.

[103] Dion EO Tzamarias, Pinar Akyazi, and Pascal Frossard, "A Novel Method for Sampling Bandlimited Graph Signals", *in*: *Proceedings of EUSIPCO*, CONF, 2018.

[104] Francesco Verdoja and Marco Grangetto, "Directional graph weight prediction for image compression", *in*: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, IEEE, 2017, pp. 1517–1521.

[105] Ruben Verhack et al., "Steered mixture-of-experts for light field coding, depth estimation, and processing", *in*: *Multimedia and Expo (ICME), 2017 IEEE International Conference on*, IEEE, 2017, pp. 1183–1188.

[106] Irene Viola et al., "A graph learning approach for light field image compression", *in*: *Applications of Digital Image Processing XLI*, vol. 10752, International Society for Optics and Photonics, 2018, 107520E.

[107] Gregory K Wallace, "The JPEG still picture compression standard", *in*: *IEEE transactions on consumer electronics* 38.*1* (1992), pp. xviii–xxxiv.

[108] Wei Wang and Kannan Ramchandran, "Random multiresolution representations for arbitrary sensor network graphs", *in*: *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings.* Vol. 4, IEEE, 2006, pp. IV–IV.

[109] Sven Wanner and Bastian Goldluecke, "Globally consistent depth labeling of 4D light fields", *in*: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 41–48.

[110] Bennett Wilburn et al., "High performance imaging using large camera arrays", *in*: *ACM Transactions on Graphics (TOG)*, vol. 24, 3, ACM, 2005, pp. 765–776.

[111] Jianxiong Xiao et al., "Recognizing scene viewpoint using panoramic place representation", *in*: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 2695–2702.

[112] M. Yu, H. Lakshman, and B. Girod, "A Framework to Evaluate Omnidirectional Video Coding Schemes", *in*: *ISMAR*, 2015.

[113] Cha Zhang and Tsuhan Chen, "A self-reconfigurable camera array", *in*: *ACM SIGGRAPH 2004 Sketches*, ACM, 2004, p. 151.

[114] Cha Zhang and Dinei Florêncio, "Analyzing the optimality of predictive transform coding using graph-based models", *in*: *IEEE Signal Processing Letters* 20.1 (2013), pp. 106–109.

[115] Cha Zhang, Dinei Florêncio, and Charles Loop, "Point cloud attribute compression with graph transform", *in*: *2014 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2014, pp. 2066–2070.

[116] Xuan Zhang, Xiaowen Dong, and Pascal Frossard, "Learning of structured graph dictionaries", *in*: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2012, pp. 3373–3376.

[117] Shengyang Zhao and Zhibo Chen, "Light field image coding via linear approximation prior", *in*: *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, pp. 4562–4566.

[118] Amin Zheng, Gene Cheung, and Dinei Florencio, "Context tree-based image contour coding using a geometric prior", *in*: *IEEE Transactions on Image Processing* 26.2 (2017), pp. 574–589.

[119] D. Zhou and B. Scholkopf, "A regularization framework for learning from graph data", *in*: *ICML workshop on statistical relational learning and Its connections to other fields*, 2004.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre: Transformées basées graphes pour la compression de nouvelles modalités d'image

Mot clés : Transformées à base de graphes, champs de lumière, images omnidirectionelles

Resumé : En raison de la grande disponibilité de nouveaux types de caméras capturant des informations géométriques supplémentaires, ainsi que de l'émergence de nouvelles modalités d'image telles que les champs de lumière et les images omnidirectionnelles, il est nécessaire de stocker et de diffuser une quantité énorme de hautes dimensions. Les exigences croissantes en matière de streaming et de stockage de ces nouvelles modalités d'image nécessitent de nouveaux outils de codage d'images exploitant la structure complexe de ces données. Cette thèse a pour but d'explorer de nouvelles approches basées sur les graphes pour adapter les techniques de codage de transformées d'image aux types de données émergents où les informations échantillonnées reposent sur des structures irrégulières. Dans une première contribution, de nouvelles transformées basées sur des graphes locaux sont conçues pour des représentations compactes des champs de lumière. En tirant parti d'une conception minutieuse des supports de transformées locaux et d'une procédure d'optimisation locale des fonctions de base , il est possible d'améliorer considérablement le compaction d'énergie. Néanmoins, la localisation des supports ne permettait pas d'exploiter les dépendances à long terme du signal. Cela a conduit à une deuxième contribution où différentes stratégies d'échantillonnage sont étudiées. Couplés à de nouvelles méthodes de prédiction, ils ont conduit à des résultats très importants en ce qui concerne la compression quasi sans perte de champs de lumière statiques. La troisième partie de la thèse porte sur la définition de sous-graphes optimisés en distorsion de débit pour le codage de contenu omnidirectionnel. Si nous allons plus loin et donnons plus de liberté aux graphes que nous souhaitons utiliser, nous pouvons apprendre ou définir un modèle (ensemble de poids sur les arêtes) qui pourrait ne pas être entièrement fiable pour la conception de transformées. La dernière partie de la thèse est consacrée à l'analyse théorique de l'effet de l'incertitude sur l'efficacité des transformées basées graphes.

Title: Graph Based Transforms for Compression of new Imaging Modalities

Keywords : Graph based transforms, Light Fields, Omni-directional images

Abstract : Due to the large availability of new camera types capturing extra geometrical information, as well as the emergence of new image modalities such as light fields and omnidirectional images, a huge amount of high dimensional data has to be stored and delivered. The ever growing streaming and storage requirements of these new image modalities require novel image coding tools that exploit the complex structure of those data. This thesis aims at exploring novel graph based approaches for adapting traditional image transform coding techniques to the emerging data types where the sampled information are lying on irregular structures. In a first contribution, novel local graph based transforms are designed for light field compact representations. By leveraging a careful design of local transform supports and a local basis functions optimization procedure, significant improvements in terms of energy compaction can be obtained. Nevertheless, the locality of the supports did not permit to exploit long term dependencies of the signal. This led to a second contribution where different sampling strategies are investigated. Coupled with novel prediction methods, they led to very prominent results for quasi-lossless compression of light fields. The third part of the thesis focuses on the definition of rate-distortion optimized sub-graphs for the coding of omni-directional content. If we move further and give more degree of freedom to the graphs we wish to use, we can learn or define a model (set of weights on the edges) that might not be entirely reliable for transform design. The last part of the thesis is dedicated to theoretically analyze the effect of the uncertainty on the efficiency of the graph transforms.