Omnidirectional Light Field Analysis and Reconstruction

THÈSE N^{O} 5483 (2012)

PRÉSENTÉE LE 19 OCTOBRE 2012

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

LABORATOIRE DE TRAITEMENT DES SIGNAUX

PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Luigi BAGNATO

acceptée sur proposition du jury:

Prof. J.-Ph. Thiran, président du jury
Prof. P. Vandergheynst, Prof. P. Frossard, directeurs de thèse
Prof. K. Daniilidis, rapporteur
Dr L. Jacques, rapporteur
Prof. S. Süsstrunk, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2012

All intelligent thoughts have already been thought;
what is necessary is only to try to think them again.
...Johann Wolfgang von Goethe

Life is like riding a bicycle,
to keep balance you need to keep going.
...Albert Einstein

Acknowledgments

A Phd thesis is not a solo journey. Surely it was not for me. During the five years that led to the reduction of this manuscript the beautiful interactions I had with people influenced my work and changed deeply my vision of the world.

I firstly thank my advisors, Pierre and Pascal, for giving me the opportunity of doing a Phd in EPFL. I also thank them for keeping me in track and, at the same time, for giving me enough freedom to find my own interests and path through research.

I would also like to thank the members of my Jury, for their valuable feedbacks and comments. A special thanks goes to Dr. Laurent Jacques for the detailed correction of the manuscript and all the fruitful conversations about science.

A big thanks goes to all the members of the signal processing laboratories family. Through the years you have been colleagues and friends. Thanks to Karin for the valuable advices during the first steps of the my Phd and for the coffee breaks with Zafer. Thanks to Meri, Nawal, Benoit, Elda, Francesca, Ashkan, Anna, Alia, Alex, Julien, Yann, Florian, Mathieu, David for the unforgettable baby-foot moments, dinners and dancing in the Candy Box. Laurent and Andrea, you made the tennis player I am now! Thanks to all the members of the LTS4: Dorina, Sofia, Xiaowen, Jacov, Nikos, Ivana, Tamara, Veejay, Elif, Eirini, Thomas, David for being the best lab mates ever, for the lunches together and the hiking and the swiss fondues. And thanks to the LTS2 boys: Momo for being a great officemate, Emmanuel for the fun on twitter and the brainstorming coffee moments on computer vision applications, Gilles for the patient discussions about convex optimization, Yves for the discussion on spherical harmonics and then Kirell, Simon and Mahdad. Alex you are a friend and a source of inspiration.

I have to admit it: I am lucky. Since my first day in Lausanne I never felt alone. My gratitude goes to Claudia for all the love and support she gave me. You kept my hand when I most needed it.

I will never thank enough all my friends for the exciting moments spent together. In Lausanne I found all the stimuli I needed to feed my curious nature.

Thanks to Baicomen and all the Esperienza Viaggi crew for initiating me to the wonderful world of cyclo-turism: it changed my entire perspective on traveling. How can I forget the fun I had the first two years partying with Karin, Nino, Harm, Veronica, German, Maria, Denisa, Marius? The trips to the mountains and the nights in la ruche? And I will never forget the fun I had with my travel mates Chris, Evy, Marzia and Anja, the surfing in Portugal, the cuttlefish moments, the new year eves drinking Scotch Whiskey.

Life was never boring in Lausanne, my friends where there. A true family in Lausanne: Mattia, Gp, Marzia, Ludovica, Sanja, Livia, Eleonora, Alessandro, Federico, Eugenio,

Roberto, Annamaria. Dinners, movies, theatre, opera, ballet, discussions about politics, art music and architecture, what could I ask more? And indeed there was more, the unforgettable events organized with all the 2MoreRaw crew! Mattia I hope we will still keep rocking behind the console.

Anche se separato da una enorme distanza fisica ho sempre sentito in questi anni il calore della mia terra, la Calabria. E devo ringraziare per questo i miei amici Cesare e Bubu e le miriadi di emails scambiate e le telefonate e l'esserci nel momento del bisogno. È bello poter ridere insieme ancora oggi a distanza di 15 anni.

Infine un grazie infinito alla mia famiglia per il supporto che mi hanno sempre offerto, per la meravigliosa educazione che ho ricevuto e per i valori che sono riusciti a trasmettermi, valori che ancora mi accompagnano e mi accompagneranno sempre.

Abstract

Digital photography exists since 1975, when Steven Sasson attempted to build the first digital camera. Since then the concept of digital camera did not evolve much: an optical lens concentrates light rays onto a focal plane where a planar photosensitive array transforms the light intensity into an electric signal.

During the last decade a new way of conceiving digital photography emerged: a photography is the acquisition of the entire light ray field in a confined region of space. The main implication of this new concept is that a digital camera does not acquire a 2-D signal anymore, but a 5-D signal in general. Acquiring an image becomes more demanding in terms of memory and processing power; at the same time, it offers the users a new set of possibilities, like choosing dynamically the focal plane and the depth of field of the final digital photo.

In this thesis we develop a complete mathematical framework to acquire and then reconstruct the omnidirectional light field around an observer. We also propose the design of a digital light field camera system, which is composed by several pinhole cameras distributed around a sphere. The choice is not casual, as we take inspiration from something already seen in nature: the compound eyes of common terrestrial and flying insects like the house fly.

In the first part of the thesis we analyze the optimal sampling conditions that permit an efficient discrete representation of the continuous light field. In other words, we will give an answer to the question: how many cameras and what resolution are needed to have a good representation of the 4-D light field? Since we are dealing with an omnidirectional light field we use a spherical parametrization. The results of our analysis is that we need an irregular (i.e., not rectangular) sampling scheme to represent efficiently the light field. Then, to store the samples we use a graph structure, where each node represents a light ray and the edges encode the topology of the light field. When compared to other existing approaches our scheme has the favorable property of having a number of samples that scales smoothly for a given output resolution.

The next step after the acquisition of the light field is to reconstruct a digital picture, which can be seen as a 2-D slice of the 4-D acquired light field. We interpret the reconstruction as a regularized inverse problem defined on the light field graph and obtain a solution based on a diffusion process. The proposed scheme has three main advantages when compared to the classic linear interpolation: it is robust to noise, it is computationally efficient and can be implemented in a distributed fashion.

In the second part of the thesis we investigate the problem of extracting geometric information about the scene in the form of a depth map. We show that the depth information is encoded inside the light field derivatives and set up a TV-regularized inverse problem, which efficiently calculates a dense depth map of the scene while respecting the discontinuities at

the boundaries of objects. The extracted depth map is used to remove visual and geometrical artifacts from the reconstruction when the light field is under-sampled. In other words, it can be used to help the reconstruction process in challenging situations. Furthermore, when the light field camera is moving temporally, we show how the depth map can be used to estimate the motion parameters between two consecutive acquisitions with a simple and effective algorithm, which does not require the computation nor the matching of features and performs only simple arithmetic operations directly in the pixel space.

In the last part of the thesis, we introduce a novel omnidirectional light field camera that we call Panoptic. We obtain it by layering miniature CMOS imagers onto an hemispherical surface, which are then connected to a network of FPGAs. We show that the proposed mathematical framework is well suited to be embedded in hardware by demonstrating a real time reconstruction of an omnidirectional video stream at 25 frames per second.

Keywords: Computational Photography, Spherical Light Field Camera, Panoptic, Omnidirectional, Manifold, Sphere, Spectral Graph Photography, Structure-From-Motion, Depth Estimation, Variational, Distributed Processing, Plenoptic Sampling, Graph Diffusion

Riassunto

Da quando il padre della fotografia digitale, Steven Sasson, invento la prima fotocamera numerica nel 1975, il concetto di macchina fotografica non è evoluto: un'ottica focalizza la luce su una superficie fotosensibile che la trasforma in un segnale elettrico. Nell' ultimo decennio sta emergendo un nuovo modo di concepire la fotografia digitale, grazie all'ausilio di computer sempre più potenti: non ci si limita più a catturare una semplice foto, bensì un intero volume di luce. Il segnale catturato diventa così 5-dimensionale. Catturare una immagine diventa così molto più oneroso in termini di memoria e potenza di calcolo, ma permette all' utente finale di beneficiare di possibilità finora impensabili, come la possibilità di mettere a fuoco la foto dopo la sua acquisizione o cambiare la profondità di campo.

In questa tesi abbiamo sviluppato una intera teoria matematica per rappresentare in modo efficiente l' intero campo luminoso intorno ad un osservatore. Mostriamo quale sia il design di una fotocamera che permette di conseguire questo scopo in modo ottimo: una superficie sferica ricoperta di microsensori ottici. La scelta è anche inspirata a qualcosa di già visto in natura: gli occhi di alcuni insetti come la comune mosca sono infatti composti di migliaia di piccole superfici fotosensibile distribuite su una superficie sferica.

Nella prima parte della tesi ci occupiamo di analizzare un problema di grande importanza nel design della telecamera: di quante telecamere abbiamo bisogno per avere una adeguata rappresentazione del campo luminoso? Il risultato della nostra analisi è che un modo efficiente di disporre le fotocamere intorno alla sfera è di rispettare una distribuzione uniforme. Questo si traduce in un campionamento dello spazio che non è regolare, non è definito, cioè, su una classica griglia rettangolare. Noi proponiamo di rappresentare questa struttura irregolare usando dei grafi, dove ogni nodo rappresenta un raggio di luce, mentre gli archi ne definiscono la topologia.

Il passo successivo è quello di utilizzare il campo luminoso per formare una fotografia, che in effetti non è altro se non una sezione del volume di luce catturato. La ricostruzione dell' immagine è interpretata come un processo di diffusione sul grafo che rappresenta il campo luminoso. Questa soluzione ha dei vantaggi rispetto a soluzione più classiche di interpolazione: è insensibile al rumore e molto efficiente da implementare in modo distribuito.

Nella seconda parte della tesi investighiamo il problema di estrarre una mappa di profondità direttamente dal campo luminoso. In effetti nascoste dentro le variazioni del campo luminoso, ci sono utilissime informazioni sulla struttura della scena. Noi proponiamo di estrarle mediante la formulazione di un problema inverso con un termine di regolarizzazione che tende a minimizzare la variazione totale della mappa di profondità. L'informazione così estratta permette di migliorare la formazione di fotografie dal campo luminoso, soprattutto quando il segnale è sotto-campionato. Se la telecamera è in movimento, parametri come la

velocità e la rotazione dell'apparecchio possono anche essere estratti dalle immagini acquisite. Noi proponiamo un semplice ed efficace algoritmo che non richiede l'estrazione di punti di interesse, operazione di solita lenta e soggetta ad errori.

Nell' ultima parte della tesi documentiamo la costruzione di una vera fotocamera omnidirezionale: Panoptic. La fotocamera è realizzata posizionando dei sensori CMOS miniaturizzati su una superficie emisferica in alluminio. I sensori sono poi connessi ad una rete di FPGA, che realizzano, in tempo reale, l'elaborazione numerica necessaria per ricostruire una fotografia omnidirezionale.

Keywords: Fotografia Computazionale, Fotocamera, Campo lumisoso, Panoptic, Omnidirezionale, Varieta, Sfera, Fotografia Spettrale, Grafi, Structure-From-Motion, Mappa di Profondita, Ottimizzazione Variazionale, Elaborazione Distribuita, Campionamento, Diffusione

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The space around us is filled with light rays. This concept was clear already to Leonardo da Vinci who wrote: *The body of the air is full of an infinite number of radiant pyramids caused by the objects located in it.* [49]. Much later, in a 1936 paper by Gershun [50], the concept of light field was formalized as the amount of light traveling in every direction through every point in space.

Why are we interested in light fields? Because much of the information about the appearance and the structure of the scene is encoded in the complex structure of light fields.

This thesis is dedicated to the study of a camera model that captures what we call the *omnidirectional light field*, *i.e.*, the full set of light rays traveling in every direction through every point in a sphere of finite radius. We also study the algorithms that can extract information from the captured omnidirectional light field. How can we form an omnidirectional image from the acquired light field? How can we extract the geometrical structure of the scene directly from the light measurements? We will give an answer to these fundamental questions, addressing at the same time the problem of finding efficient computational solutions.

A Quick Tour of Light Field Imaging Despite that light fields are been known for a long time, the theory of light field imaging is quite recent. In fact, what enabled the acquisition of light fields was the availability of inexpensive digital imagers in recents years and the rapid growth of the processing power of digital computers. Adelson in his pioneering paper [1] represents the light field with a five-dimensional function that he calls the plenoptic function. What we usually capture with a conventional camera is only a limited portion of the light field, *i.e.*, a two-dimensional slice of the plenoptic function. Therefore, a lot of information is lost during the acquisition of a conventional photograph. This simple observation is the motivation behind the recent research efforts to find solutions to measure and process the light field. Adelson is the first to understand [2] that the light field could be captured by a planar array of micro lenses placed in front of the light sensor. Few years later, Levoy [51] and Gortler [36] introduces the concept of light field camera within the computer graphics community. Levoy, using a planar array of conventional cameras, shows how virtual perspective views of the scene could be created from the captured light field by a simple rearrangement of a proper selection of pixels from the original views. Gortler uses a series of images coming from a handheld camera to achieve the same goal. Light field cameras have also been investigated in the computational photography community. An interesting result emerging from these

studies is the observation that a planar array of cameras behaves as an optical lens. This idea is behind the Fourier slice photography theory developed by Ng [62], who suggests the use of light field cameras to perform digital refocusing of photographs in post-processing. The theory has been recently transferred into a commercial product called Lytro camera [55]. The same idea is being used by another emerging company, Pelican Imaging [43], to reduce the dimension of standard camera modules inside mobile phones, using a planar array of micro imagers.

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. Surprisingly, only limited efforts have been made to address the problem of acquiring the omnidirectional light field. The concentric mosaics model [72] proposed by Shum uses a camera mounted on a rotating level beam. The acquired light field is not omnidirectional but contains all the directions around the equator. More recently Taguchi [79] proposes a catadioptric system composed by an array of several mirrors in front of a camera. The system offers an increased field of view but the resulting light field is not fully omnidirectional.

A trend with the development of omnidirectional cameras has been running in parallel to the study of light field cameras. Omnidirectional imagers begun 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 a property that is advantageous in many applications like autonomous navigation, surveillance, and 3D modeling of environments. The construction of omnidirectional imagers has been of interest to the scientific community for over a decade, since the obvious solution of systematic placement of photodiodes over a spherical structure is not currently possible with available silicon fabrication technologies. Most of the research efforts have been devoted to single effective viewpoint omnidirectional cameras [8], *i.e.*, cameras that capture the omnidirectional light field in a single point in space. Cameras designed with such a principle have the advantage of preserving linear perspective geometry, which is one of the fundamental assumptions in most of the algorithms in computer vision. Omnidirectional catadioptric devices were first introduced by Nayar [59]. They are composed by a combination of curved mirrors that reflect light onto a classic planar sensor. Catadioptic cameras became popular because they have a one-to-one mapping with the sphere, as shown in [33], hence they have a single effective viewpoint, *i.e.*, the optical rays reflected by the mirror surface intersect into a unique point. The images produced by catadrioptric lenses suffer from severe distortions that result in a trade-off between spatial and angular resolution. To achieve higher spatial resolution, Swarninathan [78] proposes to build a cluster of wide-angle cameras arranged in circle. This is one of the few works that uses a spatial arrangement of cameras to provide an omnidirectional vision. In more recent years, Foote presented the FlyCam [32], a system of 8 miniature cameras arranged on a ring to produce panoramas, which is one of the first attempt to build a cheap integrated system with off-the-shelf components. The field of view of the system remains confined, though, around a limited portion around the equator.

While most of the research on omnidirectional vision was devoted to single effective viewpoint cameras, Neumann has been the first to suggest in [60] that all the information about the 3D camera motion and the structure of the scene is encoded into the differential information of the time varying plenoptic function. Furthermore, he suggests that the ideal device to estimate structure and motion is what he calls a "full field of view polydioptric camera", *i.e.*, a closed surface where each point represents a pinhole imager. In fact, this can be considered as the first attempt to model the capture of an omnidirectional light field. Neumann does not

develop further the idea of full field of view polydioptric camera, which remains a theoretical tool to analyze the performance of structure-from-motion algorithms.

1.1 Thesis Contributions

Inspired by Neumann's intuition, we propose a camera model that captures the omnidirectional light field on a spherical surface layered with hundreds of pinhole imagers. Our design assumptions are supported by the existence of a similar model in nature: the compound eyes of common terrestrial and flying insects, like the house fly, are composed by thousands of sensors, called ommatidia, placed on a spherical surface.

If the imagers are placed closed enough on the spherical surface, then the full system can be interpreted as a spherical optical lens, which suggests that such a camera model could be used to design miniature omnidirectional cameras.

The idea has a concrete technological foundation. Current silicon technology makes possible the fabrication of small and low cost imagers that behave like pinhole cameras: the camera modules conceived for modern smart phones measure only few millimeters in size and have a very short focal. However, the tremendous amount of data generated by hundreds of these camera modules would be too high to be transferred to a central unit for processing. For example, 100 imagers with resolution of 1 Mpixels and 8bits of color depth, would produce a data rate of 20Gbits/sec at a frame rate of 25 fps. These numbers are too high for the currently available transfer protocols. To overcome this limitation we consider that every imager is equipped with some computing power and, following a recent trend in signal processing [14], we interpret the light field processing as a distributed operation over a network. This imposes a radically different thinking in the design of imaging algorithms.

To validate our design, the proposed model has been used in the construction of an omnidirectional light field camera prototype that we call the *Panoptic Camera*. It consists of 100 miniature Complementary metal–oxide–semiconductor (CMOS) imagers layered around an hemispherical support. The cameras are connected through a network of field-programmable gate arrays (FPGAs), which collect the video streams from the cameras and process them in real time.

The list of contributions in this thesis can be summarized as follows:

1. **Spherical Light Field Camera.** We design a novel camera model to capture the omnidirectional light field.

2. **Fourier Analysis on $S^2 \times S^2$ through a small angle approximation.** We develop a new approximate Fourier analysis to determine the optimal sampling conditions for the captured omnidirectional light field.

3. **Linear omnidirectional light field interpolation.** We define the concept of an Omnidirectional Photography Operator as an extension of the photography operator defined in [62] and propose to form an omnidirectional photo as a kernel interpolation process for the omnidirectional light field with radial basis functions.

4. **Graph-based omnidirectional light field interpolation.** We give the definition of light field graph as the mathematical structure that approximates the continuous 4D-plenoptic function and then interpret the formation of omnidirectional photos as a

regularized diffusion process on the light field graph. We also make a formal connection between the kernel-based interpolation and a certain class of diffusion processes.

5. **Light Field Depth Estimation.** We propose a novel algorithm to compute the depth map of the scene, using all the information contained in one light field acquisition.

6. **Omnidirectional structure-from-motion.** We propose a variational framework on graphs to jointly extract the structure of the scene and the motion parameters from two omnidirectional images.

7. **The Panoptic Camera.** We develop a real camera prototype called The Panoptic Camera where we test our novel imaging algorithms.

We envisage that the contributions of this thesis will have a strong impact on the following applications in robotics and computer vision:

Autonomous Navigation The ability of our camera to calculate the motion parameters leads to a direct application in visual odometry.

3D Environment Modeling Our camera can acquire at the same time the appearance and the structure of the scene. This can be used to model complex environment with applications in architecture, art, virtual tourism, augmented reality in gaming or assisted driving.

Omnidirectional Vision Our camera can acquire omnidirectional images and videos, which makes it suitable for video-surveillance or video-conferencing. It is a light field camera that can be used in panoramic photography, immersive videos, cinematography and stereo cinematography.

1.2 Thesis Roadmap

In a nutshell, the thesis is oriented along three main axes:

1. Sampling theory and light field reconstruction methods in Chapter 2 and Chapter 3.

2. Scene structure and camera motion estimation in Chapter 4 and Chapter 5.

3. Practical implementation of a real spherical light field camera in Chapter 6.

In more details, the thesis is organized as follows.

Chapter 2 We formalize the concept of Spherical Light Field Camera. Using a Fourier Analysis based on a small angle approximation, we find the optimal sampling conditions that prevent the aliasing of the capture light field. We also introduce the concept of Omnidirectional Photography Operator and present the Kernel Interpolation algorithm.

Chapter 3 We describe the graph-based interpolation algorithms. First, we formalize the concept of Light Field Graph. Then we interpret the interpolation process as the operation of an inpainting operator on the nodes of the light field graph. The solution of the inpainting problem is found through a diffusion process on the light field graph, which can

be implemented in a completely distributed fashion through the local exchange of information among neighbor nodes. Two different diffusion processes are described, one based on Tikhonov regularization and the other one based on Total-Variation regularization. We make a formal connection between the diffusion process based on Tikhonov regularization and the kernel interpolation algorithm, showing how classic linear filtering can be implemented in a distributed fashion.

Chapter 4 We describe a novel algorithm to calculate a dense depth map from a single light field measurement. The algorithm is based on the minimization of the squared pairwise light field intensity errors, with a total-variation regularization term to promote piecewise smooth depth maps.

Chapter 5 We describe a novel structure-from-motion framework on graphs, which uses two omnidirectional images to extract the camera ego motion and a dense depth map.

Chapter 6 We finally present a light field camera prototype called the Panoptic Camera.

Chapter 2

Omnidirectional Light Field Sampling and Representation

Massive changes in the way we acquire and consume video content has recently led to the raise of a new paradigm for the representation of 3D scenes, called Image Based Rendering (IBR). In applications such as 3DTV or free-viewpoint TV the user is able to choose the viewpoint and the orientation of the camera. While reconstructing the full 3D model of a scene is still a technological challenge under realtime constraints, image based rendering solutions appear promising as long as the light field is finely sampled. The problem is in fact to determine the number of samples necessary to represent the content of interest. This chapter addresses this fundamental question for the omnidirectional light field around an observer.

The concept of light fields was first devised by Leonardo da Vinci [49]. Further studies conducted by Adelson and Bergen [1] resulted in the definition of the *plenoptic* function as the most complete representation of a scene over time. The plenoptic function $\mathcal{L}(\mathbf{x}, \omega, \mathrm{t}, \lambda_\mathrm{c})$, is the function that represents the intensity (or radiance) that a perfect observer inside a given 3-D scene records at any position $\mathbf{x} \in \mathbb{R}^3$, in any direction $\omega \in S^2$, at time t and wavelength λ_c

In the last decade several parametrizations have been proposed for the plenoptic function: the light field [51] and the lumigraph [36] are among the most popular ones. In these works a 4-D parametrization of the plenoptic function is proposed under the assumption that the scene is confined to a cubic bounding box, or, in other words, that each light ray is represented by the intersection with two parallel planes. One of the biggest advantages of such a parametrization lies in its simple analytic description and efficient implementation. The authors in [18] propose an approximate spectral analysis of the light field showing that the bounds of the spectrum of the light field for a two-plane parametrization are determined by the minimum and maximum depth of the scene. However, as pointed out in [17], the use of the representation based on two planes has a major drawback: the limited visual angle of the representation. Several couples of planes could be used to cover the convex hull of an object, but artifacts are unfortunately visible in the reconstruction at the boundaries. The authors in [17] refer to this problem as the *disparity problem*. In order to solve it, they propose to use a spherical surface for the parametrization of the light field and choose a sampling scheme based on a polyedra approximation of the spherical surface. While their approach offers a way to sample uniformly the light field, their scheme is difficult to apply in practical situations and no clear

indication is provided on the number of samples necessary for a good rendering. Furthermore, a good knowledge of the scene geometry is required. A rather different approach, called concentric mosaics [72], consists in capturing the scene with a camera mounted at the end of a horizontal rotating level beam. As the beam rotates, regular images are acquired. This is a 3D representation of the light field, since the camera motion during the acquisition is constrained to lie on a plane. The lack of vertical sampling causes a distortion in the rendered image when the vertical field of view tends to get farther from the image acquisition plane.

The main contribution in this chapter is a novel sampling scheme that overcomes the limitations of the approaches described earlier. We also propose an efficient algorithm to reconstruct the light field with predictable performance. Inspired by the efficient visual system of flying insects we propose a new 4-D sampling scheme of the light field. The common fly has two faceted eyes (see Figure 2.1) composed of several thousands simple sensors called ommatidias [88] that provide plenty of planar overlapping views of the world. Mimicking the faceted eye concept, a spherical light field camera can be realized by layering perspective imagers over a spherical surface.

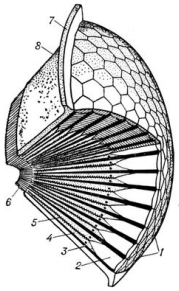


Figure 2.1: *Compound eyes of insects.*

The sphere is a closed convex surface with constant curvature, which enables a complete angular coverage while keeping a simple topology. One direct consequence is that it can be efficiently parametrized in spherical coordinates, using only the zenithal and azimuthal angles. Assuming that the air is transparent and that there is no radial fall-off of the light intensity, we propose a 4D representation of the light field based on a two-sphere parametrization of the plenoptic function. We also develop a spectral analysis of the captured light field at a constant depth from the center of the sphere and propose a constructive sampling strategy that guarantees the absence of aliasing during the reconstruction of the visual scene. We finally validate our model with experiments in a synthetic environment.

The rest of the chapter is organized as follows: In Section 2.1 we describe the proposed light field parametrization and the scene modeling. In Section 2.3 we introduce the Spherical Light Field Camera model whose design is based on the spectral analysis of the plenoptic function. In Section 2.4 we describe an algorithm to render omnidirectional images. Finally we show in Section 2.5 some experimental results in a synthetic environment.

2.1 Omnidirectional Light Field Parametrization

The light field can be represented mathematically through the plenoptic function $\mathcal{L}(\mathbf{x}, \omega, \mathrm{t}, \lambda_\mathrm{c})$. In the rest of the Chapter, for the sake of simplicity of notation and without loss of generality, we will drop the temporal variable t and the variable λ_c. Let us consider a convex surface

layered with a discrete set of pinhole cameras. Assuming an ideal behavior of the cameras, *i.e.*, the cameras can be modeled as pure perspective imagers with a single focal point, the system is an ideal plenoptic sampler if the convex surface contains a sufficient number of pinhole cameras. It means that we can reconstruct the plenoptic function anywhere on the surface and also inside the space delimitated by the surface under certain assumptions. In the rest of the chapter we assume that the convex surface is a sphere. To start formalizing the problem and introduce the notation, we assume that we have N_c pinhole cameras distributed around a spherical surface of radius R, in positions $\mathbf{q} = R\hat{\mathbf{q}} \in \mathbb{R}^3$, where with the notation $\hat{\mathbf{q}}$ we indicates a unit vector. A camera c is positioned at $\mathbf{q}_c$ and can capture light in all the directions ω up to a maximum angle of α_c with the surface normal in $\mathbf{q}_c$. The angle α_c is called the field-of-view (FOV) of the pinhole imager. If we keep the FOV in the range $\alpha_c \in [0, \pi/2]$ and we assume to have a convex positional surface, we automatically exclude the possibility of self-visual-occlusions, *i.e.*, cameras are not able to look at each other. The full set of directions that each pinhole camera is able to acquire are thus contained in the positive half space defined by the tangent plane of the surface at the point $\mathbf{q}_c$. Since we are dealing with a spherical geometry, we choose to work with spherical coordinates. A point on the positional sphere is then parametrized as $\mathbf{q} = (\phi_{\mathbf{x}}, \theta_{\mathbf{x}}, \mathrm{R})$, while a direction is parametrized by $\omega = (\phi_\omega, \theta_\omega)$ as shown in Figure 2.2. In the following we assume that the coordinates are specified with respect to a common reference system.

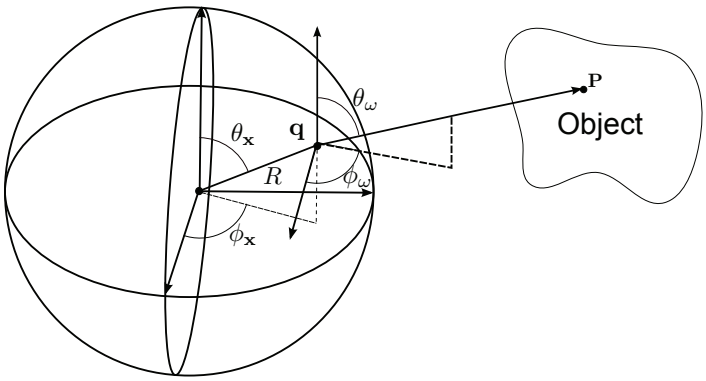


Figure 2.2: *Schematic representation of the chosen parametrization of the plenoptic function.*

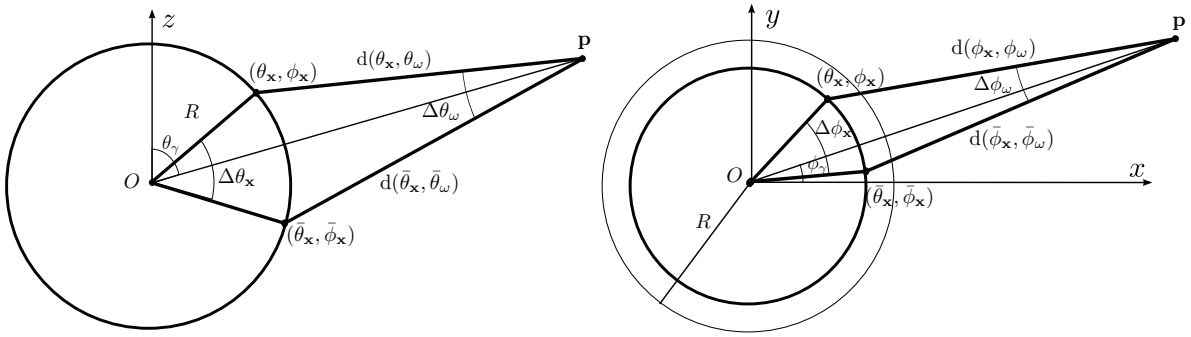
The plenoptic function $\mathcal{L}$ is continuous so what we acquire in practice is a sampled version:

$$\mathcal{L}_T(\mathbf{x}, \omega) = \sum_{\mathbf{q}_i} \delta(\mathbf{x} - \mathbf{q}_i)\mathcal{L}(\mathbf{q}_i, \omega), \tag{2.1}$$

If we sample finely enough we can reconstruct the original light field $\mathcal{L}$ from the sampled version $\mathcal{L}_T$ through a convolution in $\mathbb{R}^3$ with an interpolation function $g(\mathbf{x})$:

$$\mathcal{L}(\mathbf{x}, \omega) = g(\mathbf{x}) * \mathcal{L}_T(\mathbf{x}, \omega) = \sum_{\mathbf{q}_i} g(\mathbf{x} - \mathbf{q}_i)\mathcal{L}(\mathbf{q}_i, \omega). \tag{2.2}$$

The problem of finding an optimal sampling scheme can be then formulated as finding the minimum number of samples necessary for the perfect reconstruction of the continuous function $\mathcal{L}$ from its sampled version $\mathcal{L}_T$. Fourier analysis has been proven as a useful tool to study the plenoptic function [27, 61, 89]. In the next section we apply it to the study of the omnidirectional light field.



a) Vertical plane with $\phi_{\mathbf{x}} = \bar{\phi}_{\mathbf{x}}$ b) Horizontal plane with $\theta_{\mathbf{x}} = \bar{\theta}_{\mathbf{x}}$

Figure 2.3: *Geometrical relationship between angular disparities.*

2.2 Spectral Analysis

The spectrum of the captured light field is given in principle by the spherical harmonics expansion of $\mathcal{L}$ with coefficients h:

$$h(m_{\mathbf{x}}, l_{\mathbf{x}}, m_{\omega}, l_{\omega}) = \int_{S^2} d\Omega(\phi_{\mathbf{x}}, \theta_{\mathbf{x}}) Y_{l_{\mathbf{x}} m_{\mathbf{x}}}(\phi_{\mathbf{x}}, \theta_{\mathbf{x}}) \int_{S^2} d\Omega(\phi_{\omega}, \theta_{\omega}) \mathcal{L}(\phi_{\mathbf{x}}, \theta_{\mathbf{x}}, \phi_{\omega}, \theta_{\omega}) Y_{l_{\omega} m_{\omega}}(\phi_{\omega}, \theta_{\omega}), \tag{2.3}$$

for natural $l_{\mathbf{x}}, l_{\omega} \in \mathbb{N}$ and integer $m_{\mathbf{x}}, m_{\omega} \in \mathbb{Z}$, $m_{\mathbf{x}} \leqslant l_{\mathbf{x}}$ and $m_{\omega} \leqslant l_{\omega}$. The functions $Y_{l_{\omega} m_{\omega}}$ and $Y_{l_{\mathbf{x}} m_{\mathbf{x}}}$ are the spherical harmonic function of degree respectively l_{ω} and $l_{\mathbf{x}}$, and order respectively m_{ω} and $m_{\mathbf{x}}$. The spectrum on the sphere is discrete because it is defined on a closed domain. If we discretize the sphere using an equiangular grid, we can use the theorem from Driscoll and Healy [28] or the more recent one from McEwan and Wiaux [58]: if the function is bandlimited at $(L_{\omega}, L_{\mathbf{x}})$, we can calculate exactly the spherical harmonic coefficients using the available light field as long as we collect at least $((L_{\omega} - 1) \times (L_{\omega} - 1)) \times ((L_{\mathbf{x}} - 1) \times (L_{\mathbf{x}} - 1))$ samples.

Although spherical harmonics would be in principle an appropriate instrument for analyzing the spectrum of the omnidirectional light field we choose to follow a rather different approach, using a Fourier analysis on the tangent bundle of the manifold where the light field lives. Our choice is motivated by two reasons:

1. By construction, the imagers have a FOV limited to one hemisphere.

2. The sampling theorems that use spherical harmonics induce the use of equiangular grids, which in turn determine a non-uniform sampling on the sphere.

If we assume that the light from objects in the scene is scattered almost uniformly in all directions, then there is some geometrical redundancy in the captured light field that we can use to estimate bounds for the support of the spectrum. With reference to Figure 2.3, given a point $\mathbf{p} = (\theta_{\gamma}, \phi_{\gamma}, \mathrm{r})$ in the scene and a reference point $(\bar{\theta}_{\mathbf{x}}, \bar{\phi}_{\mathbf{x}})$ on the sphere, we can derive

the following geometric relationship among angles on a vertical plane when $\phi_{\mathbf{x}} = \bar{\phi}_{\mathbf{x}}$:

$$\theta_\gamma - \bar{\theta}_\omega = \sin^{-1}\left(\frac{R}{\mathrm{d}(\bar{\theta}_{\mathbf{x}}, \bar{\theta}_\omega)} \sin\left(\bar{\theta}_{\mathbf{x}} - \theta_\gamma\right)\right) \quad (2.4)$$

$$\theta_\omega - \theta_\gamma = \sin^{-1}\left(\frac{R}{\mathrm{d}(\theta_{\mathbf{x}}, \theta_\omega)} \sin\left(\theta_\gamma - \theta_{\mathbf{x}}\right)\right) \quad (2.5)$$

$$\Delta\theta_\omega = \theta_\omega - \bar{\theta}_\omega = \sin^{-1}\left(\frac{R}{\mathrm{d}(\theta_{\mathbf{x}}, \theta_\omega)} \sin\left(\theta_\gamma - \theta_{\mathbf{x}}\right)\right) + \sin^{-1}\left(\frac{R}{\mathrm{d}(\bar{\theta}_{\mathbf{x}}, \bar{\theta}_\omega)} \sin\left(\bar{\theta}_{\mathbf{x}} - \theta_\gamma\right)\right). \quad (2.6)$$

Similarly, if we consider the orthogonal projection onto an horizontal plane where $\theta_{\mathbf{x}} = \bar{\theta}_{\mathbf{x}}$, as shown in Figure 2.3, we will have:

$$\Delta\phi_\omega = \sin^{-1}\left(\frac{R\sin\theta_{\mathbf{x}}}{\mathrm{d}(\phi_{\mathbf{x}}, \phi_\omega)\sin\theta_\omega} \sin\left(\phi_\gamma - \phi_{\mathbf{x}}\right)\right) + \sin^{-1}\left(\frac{R\sin\theta_{\mathbf{x}}}{\mathrm{d}(\bar{\phi}_{\mathbf{x}}, \bar{\phi}_\omega)\sin\theta_\omega} \sin\left(\bar{\theta}_{\mathbf{x}} - \theta_\gamma\right)\right). \quad (2.7)$$

Eq. (2.7) includes some dependencies in $\theta_{\mathbf{x}}$ and θ_ω, which are due to the specific parametrization of the sphere.

The equations (2.4) to (2.7) describe a non-linear relationship between the angular disparities, the radius of the spherical surface that contains the cameras and the distance to the scene. The use of these geometrical relationships in the analysis of the optimal plenoptic sampling is not straightforward, since they do not simplify the expression of the integral in Eq. (2.3).

We decide to linearize Eq. (2.7) assuming that $\mathrm{d}(\phi_{\mathbf{x}}, \phi_\omega) \simeq \mathrm{d}(\phi_\gamma, \phi_\gamma)$ and $\mathrm{d}(\bar{\phi}_{\mathbf{x}}, \bar{\phi}_\omega) \simeq \mathrm{d}(\phi_\gamma, \phi_\gamma)$. It can be shown that the approximation holds when the depth is much larger than the radius of the sphere. For example, the approximation error is about 2% if $\mathrm{d} > 3R$ or, equivalently, if the field of view of each camera is small. Eq. (2.7) then simplify to:

$$\Delta\phi_\omega = \frac{R\sin\theta_{\mathbf{x}}}{\mathrm{d}(\phi_\gamma, \phi_\gamma)\sin\theta_\omega}\Delta\phi_{\mathbf{x}}, \quad (2.8)$$

where $\Delta\phi_{\mathbf{x}} = \phi_{\mathbf{x}} - \bar{\phi}_{\mathbf{x}}$. Similarly, for Eq. (2.4) we will have:

$$\Delta\theta_\omega = \frac{R}{\mathrm{d}(\theta_\gamma, \theta_\gamma)}\Delta\theta_{\mathbf{x}}, \quad (2.9)$$

with $\Delta\theta_{\mathbf{x}} = \theta_{\mathbf{x}} - \bar{\phi}_{\mathbf{x}}$. We now derive the spectrum support for a simple scene at constant depth from the camera surface, i.e., we take $\mathrm{d} = \mathrm{d}_0 = \textit{constant}$. Although on a spherical domain a compact representation of the light field would be obtained using spherical harmonics with the coefficient expressed in Eq. (2.3), under the assumption we made of small field of view (FOV) we can use the classic Fourier analysis where the complex exponentials will be a fairly good basis for the light field. In other words, since we do not integrate over all the sphere, due to the small FOV assumption, we can locally approximate the sphere with its tangent plane. We can further notice that, if we perform the following bijective mapping:

$$s_\omega(\phi_\omega, \theta_\omega) = \phi_\omega \sin\theta_\omega \quad (2.10)$$

$$s_{\mathbf{x}}(\phi_{\mathbf{x}}, \theta_{\mathbf{x}}) = \phi_{\mathbf{x}} \sin\theta_{\mathbf{x}} \quad (2.11)$$

which describe the arc lengths $s_\omega(\phi_\omega, \theta_\omega)$ and $s_\omega(\phi_{\mathbf{x}}, \theta_{\mathbf{x}})$ on the sphere, then Eq. (2.8) simplify as $\Delta s_\omega = \frac{R}{\mathrm{d}(\phi_\gamma, \phi_\gamma)}\Delta s_{\mathbf{x}}$. In practice the previous mapping introduces the elements of the metric

tensor on the sphere, $\sin\theta_\omega$ and $\sin\theta_{\mathbf{x}}$ inside the parametrization such that, *e.g.*, for small variations of $\theta_{\mathbf{x}}$ and $\phi_{\mathbf{x}}$, $\Delta s_{\mathbf{x}}$ and $\Delta\theta_{\mathbf{x}}$ represent the coordinates of an orthonormal basis on the tangent plane in $(\theta_{\mathbf{x}}, \phi_{\mathbf{x}})$. Let us assume that the scene is lambertian, i.e., it reflects light isotropically. If, without loss of generality, we take as reference the camera in $(\bar{\theta}_{\mathbf{x}} = 0, \bar{\phi}_{\mathbf{x}} = 0)$ and use the lambertian property of the scene, we can write $\mathcal{L}_s(s_{\mathbf{x}}, \theta_{\mathbf{x}}, s_\omega, \theta_\omega) = \mathcal{L}_s(0, 0, s_\omega + \frac{R}{\mathrm{d}_0} s_{\mathbf{x}}, \theta_\omega - \frac{R}{\mathrm{d}_0}\theta_{\mathbf{x}})$, where $\mathcal{L}_s(s_{\mathbf{x}}, \theta_{\mathbf{x}}, s_\omega, \theta_\omega) = \mathcal{L}(\phi_{\mathbf{x}}, \theta_{\mathbf{x}}, \phi_\omega, \theta_\omega)$. The local Fourier transform of the light field around the position $(\bar{\theta}_{\mathbf{x}}, \bar{\phi}_{\mathbf{x}})$ generated by a scene at constant depth d_0 is:

$$\begin{aligned} L(\Omega_{s_{\mathbf{x}}}, \Omega_{\theta_{\mathbf{x}}}, \Omega_{s_\omega}, \Omega_{\theta_\omega}) = \\ \int d\theta_\omega e^{-j\Omega_{\theta_\omega}\theta_\omega} \int ds_\omega e^{-j\Omega_{s_\omega}s_\omega} \int ds_{\mathbf{x}} \int d\theta_{\mathbf{x}} \mathcal{L}_s(s_{\mathbf{x}}, \theta_{\mathbf{x}}, s_\omega, \theta_\omega) e^{-j\Omega_{s_{\mathbf{x}}}s_{\mathbf{x}}} e^{-j\Omega_{\theta_{\mathbf{x}}}\theta_{\mathbf{x}}} = \\ \int d\theta_\omega e^{-j\Omega_{\theta_\omega}\theta_\omega} \int ds_\omega e^{-j\Omega_{s_\omega}s_\omega} \mathcal{L}_s(0, 0, s_\omega, \theta_\omega) \int ds_{\mathbf{x}} e^{-j(\Omega_{s_{\mathbf{x}}} + \frac{R}{\mathrm{d}_0}\Omega_{s_\omega})s_{\mathbf{x}}} \int d\theta_{\mathbf{x}} e^{-j(\Omega_{\theta_{\mathbf{x}}} + \frac{R}{\mathrm{d}_0}\Omega_{\theta_\omega})\theta_{\mathbf{x}}} = \\ = F\{\mathcal{L}_s(0, 0, \theta_\omega, \phi_\omega)\}\delta(\Omega_{\theta_{\mathbf{x}}} + \frac{R}{\mathrm{d}_0}\Omega_{\theta_\omega})\delta(\Omega_{s_{\mathbf{x}}} + \frac{R}{\mathrm{d}_0}\Omega_{s_\omega}) \end{aligned} \tag{2.12}$$

The analysis in Eq.(2.12) neglects the windowing effect due to a finite domain of integration under the assumption that it is negligible (this assumption is common in the literature [18, 89]). What emerges is that, if we consider a slice of the spectrum on the plane $(\Omega_{s_{\mathbf{x}}}, \Omega_{s_\omega})$, we observe that the spectrum is constrained to a family of lines given by $\Omega_{s_{\mathbf{x}}} + \frac{R}{\mathrm{d}_0}\Omega_{s_\omega} = 0$ (and similarly $\Omega_{\theta_{\mathbf{x}}} + \frac{R}{\mathrm{d}_0}\Omega_{\theta_\omega} = 0$), whose slope depends on the ratio R/d_0. A similar results was obtained with a different procedure in [89] but only applied to concentric mosaics. Let us now consider a uniform sampling lattice made of samples spaced by quantities Δs_ω, $\Delta s_{\mathbf{x}}$, $\Delta\theta_\omega$, $\Delta\theta_{\mathbf{x}}$, the sampled light field spectrum will consist in equally spaced replicas of the original spectrum L:

$$L_s(\Omega_{s_{\mathbf{x}}}, \Omega_{\theta_{\mathbf{x}}}, \Omega_{s_\omega}, \Omega_{\theta_\omega}) = \sum_{m_{\mathbf{x}}, l_{\mathbf{x}}, m_\omega, l_\omega} L(\Omega_{s_{\mathbf{x}}} - \frac{2\pi m_{\mathbf{x}}}{\Delta s_{\mathbf{x}}}, \Omega_{\theta_{\mathbf{x}}} - \frac{2\pi l_{\mathbf{x}}}{\Delta\theta_{\mathbf{x}}}, \Omega_{s_\omega} - \frac{2\pi m_\omega}{\Delta s_\omega}, \Omega_{\theta_\omega} - \frac{2\pi l_\omega}{\Delta\theta_\omega}), \tag{2.13}$$

where $m_{\mathbf{x}}, l_{\mathbf{x}}, m_\omega, l_\omega \in \mathrm{d}$ are integer indexes. Let us also assume that the scene is confined between two spheres with radius $\mathrm{r}_{min} = R + \mathrm{d}_{min}$ and $\mathrm{r}_{min} = R + \mathrm{d}_{max}$, then the spectrum will be confined between two lines $\Omega_{s_{\mathbf{x}}} + \frac{R}{\mathrm{d}_{min}}\Omega_{s_\omega} = 0$ and $\Omega_{s_{\mathbf{x}}} + \frac{R}{\mathrm{d}_{max}}\Omega_{s_\omega} = 0$ (for the sake of simplicity we only consider the plane $(\Omega_{s_{\mathbf{x}}}, \Omega_{s_\omega})$). If we assume that the input angular images for each camera are sampled at their Nyqvist rate, i.e. they are bandlimited in the range $[-\frac{1}{2\Delta\theta_\omega}, \frac{1}{2\Delta\theta_\omega}]$ and $[-\frac{1}{2\Delta s_\omega}, \frac{1}{2\Delta s_\omega}]$, we find, using geometric reasoning, that the following sampling condition must be respected for $\theta_{\mathbf{x}}$:

$$\Delta\theta_{\mathbf{x}} \leq \frac{2\Delta\theta_\omega}{R(\frac{1}{\mathrm{d}_{min}} - \frac{1}{\mathrm{d}_{max}})} \tag{2.14}$$

and similarly for $s_{\mathbf{x}}$:

$$\Delta s_{\mathbf{x}} \leq \frac{2\Delta s_\omega}{R(\frac{1}{\mathrm{d}_{min}} - \frac{1}{\mathrm{d}_{max}})}. \tag{2.15}$$

The above conditions permit to avoid aliasing of the spectral replicas, user the assumptions that depth is much larger than the radius of the sphere and that cameras have a small FOV.

2.3 The Spherical Light Field Camera Model

In Section 2.2 we derive a criterion to sample an omnidirectional 4-D light field using perspective imagers positioned around a sphere. The intuition behind the derivation is to approximate the positional sphere $(\theta_{\mathbf{x}}, \phi_{\mathbf{x}})$ with its tangent bundle, using the assumption that the imagers have a small FOV, such that we can assume that locally on the sphere the Fourier integral is well defined. Using a first order Taylor approximation of the geometric relationships that govern the light field in spherical coordinates, we show that the light field is bounded in the transformed Fourier domain and the bounds are determined by the depth distribution in the scene. We also show that we can avoid the aliasing of the spectral replicas if the sampling conditions given in Eq. (2.14) and Eq. (2.15) are respected. The use of the mapping in Eq. (2.10) dictates the use of non-rectangular sampling lattice in spherical coordinates. Let us consider the directional sphere $(\theta_\omega, \phi_\omega)$, from Eq. (2.10) we have that $\phi_\omega \in [0, 2\pi]$ and $s_\omega \in [0, 2\pi \sin\theta_\omega]$, which means that the number of samples in the s_ω direction depends on the actual position of the sphere, *i.e.*, $N_{s_\omega}(\theta_\omega) = \lfloor \frac{2\pi \sin\theta_\omega}{\Delta s_\omega} \rfloor$ and similarly $N_{s_{\mathbf{x}}}(\theta_{\mathbf{x}}) = \lfloor \frac{2\pi \sin\theta_{\mathbf{x}}}{\Delta s_{\mathbf{x}}} \rfloor$. We propose here a sampling scheme that determines an approximately uniform distribution of samples over the sphere, which is an advantage of the chosen lattice when compared to the equiangular grid. The algorithm is summarized in `Algorithm 2.1`. We provide in Figure 2.4 an example of sampling pattern generated by the proposed scheme on a sphere, showing with small circles the samples on a geographic projection of an equiangular grid and on the sphere itself.

Input: $N_\theta, \Delta_\theta, s.t.\, N_\theta = \frac{\pi}{\Delta_\theta}$
Output: The positions $\{\theta^k, k = 1, 2, \cdots, N_\theta\}$ and $\{\phi^j(\theta^k), k = 1, 2, \cdots, N_\phi(\theta^k)\}$
Divide the sphere in N_θ layers: $\theta^k = \frac{\pi}{(N_\theta+1)}k, k = 1, 2, \cdots, N_\theta$
for $k = 1, 2, \cdots N_\theta$ **do**
  $N_\phi(\theta^k) = \lfloor \frac{2\pi}{\Delta\theta} \sin\theta^k \rfloor$
  $\phi^j(\theta^k) = \frac{2\pi}{N_\phi(\theta^k)}j, j = 0, 1, \cdots, N_\phi - 1$
end for

Algorithm 2.1: Generation of the spherical sampling pattern

On the basis of these considerations we define the Spherical Light Field Camera:

Definition 2.3.1. Spherical Light Field Camera *The Spherical Light Field Camera is a spherical surface parametrized by* $\mathbf{q} = (\theta_{\mathbf{x}}, \phi_{\mathbf{x}}, R)$ *which is covered by pinhole imagers* c_i *positioned in* $\mathbf{q}_i$*. Each pinhole imager has the optical axis oriented as the surface normal in* $\mathbf{q}_i$*, and sample uniformly the light field in the cone identified by its field of view* α_i*. The positions of the pinhole imagers are determined according to* `Algorithm 2.1`*.*

We simulate the spherical light field camera inside the 3D modeling environment Blender. In the simulated model, each perspective camera in the model has a planar sensor with a resolution of 288x216 pixels and an horizontal field of view of 54°, which corresponds to an equivalent directional angular resolution $\Delta\theta_\omega = \pi/960$ rad and a subdivision of the directional sphere in 960 layers. We use a spatial angular sampling with $\Delta\theta_{\mathbf{x}} = \pi/10$ rad, which determines a subdivision of the positional sphere into 10 layers for a total number of 126 imagers as shown in Figure 2.5.

The reason behind our choices is that such an arrangement of cameras can be implemented using the current technology, as we will show in Chapter 6.

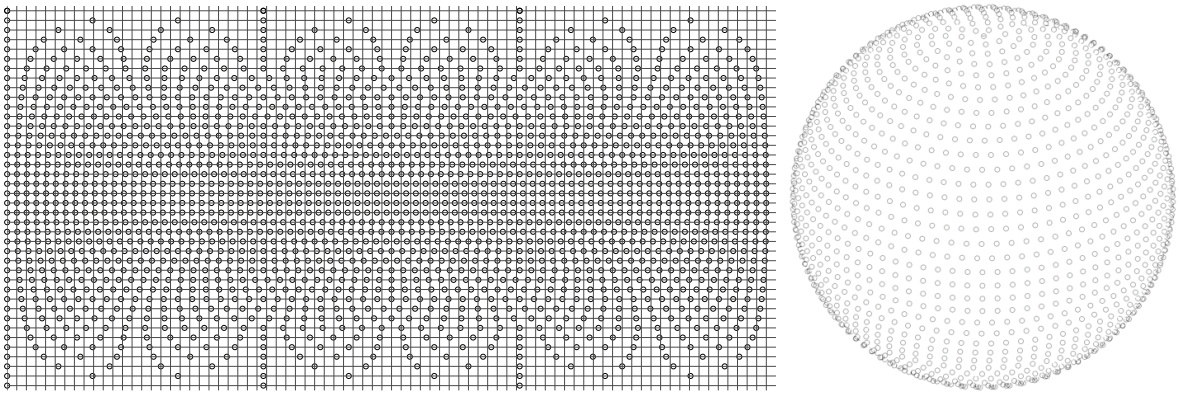


Figure 2.4: *Sampling pattern of* `Algorithm 2.1` *shown on a equiangular grid (left) and on the sphere (right). Samples are represented with small circles. We observe that on the sphere the samples are approximately uniform distributed.*

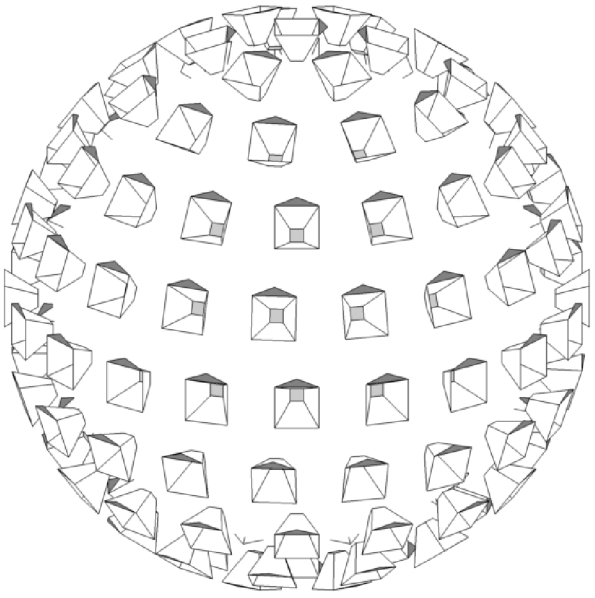


Figure 2.5: *The simulated Spherical Light Field Camera obtained with* `Algorithm 2.1`*. This model can be realized using current technology as we show in Chapter 6.*

2.4 Kernel-Based Light Field Reconstruction

Similarly to [44] and [62] we can interpret the interpolation of an omnidirectional image as the formation of a photograph through lenses. We can construct an omnidirectional image with a unique focal point $\mathbf{o}$ inside the positional sphere by observing that $\mathcal{L}(\mathbf{o},\omega) = \mathcal{L}(R\frac{\mathbf{o}+\omega}{\|\mathbf{o}+\omega\|},\omega)$. In other words, along a given direction ω we always observe the same value for $\mathcal{L}$, so the value of the plenoptic function in $\mathbf{o}$ can be interpolated on some point of the positional surface. This is true because we assume the absence of attenuation of light inside the positional sphere. If $\mathbf{o} = 0$ the expression simplify into $\mathcal{L}(\mathbf{o},\omega) = \mathcal{L}(R\omega,\omega)$. In the following we assume that $\mathbf{o} = 0$, without loss of generality, in order to keep a light notation.

Before defining the omnidirectional photography operator we define the focus operator $\mathcal{M}_\mathrm{r}$ as:

$$\mathcal{M}_\mathrm{r}[\mathcal{L}](\mathbf{x},\omega) = \mathcal{L}(\mathbf{x}, M(\mathbf{x},\omega,r)), \tag{2.16}$$

where the mapping M is defined as:

$$M(\mathbf{q},\omega,r) = \frac{r\hat{\omega} - \mathbf{q}}{\|r\hat{\omega} - \mathbf{q}\|} \tag{2.17}$$

Intuitively the action of the operator $\mathcal{M}_\mathrm{r}$ is to perform a change of coordinates such that, for a given direction ω observed from the focal point $\mathbf{o}$, $\mathcal{M}_\mathrm{r}[\mathcal{L}](\mathbf{x},\omega)$ gives the value of the light ray that originates from the point in space $\mathbf{x} = R\hat{\omega}$ and passes through the point $\mathbf{x}$. In other words, the operator $\mathcal{M}$ re-projects the light field on the focal surface defined by $\mathbf{o}$ and r.

Definition 2.4.1. *Omnidirectional Photography Operator*
The omnidirectional photography operator for a scene at distance r *from the focal point* $\mathbf{o}$ *is is defined by:*

$$\mathcal{P}_o[\mathcal{L}](\omega) = \mathcal{L}(R\hat{\omega},\omega) = (g(\mathbf{x}) * \mathcal{M}_\mathrm{r}[\mathcal{L}_T])(R\hat{\omega},\omega), \tag{2.18}$$

for some interpolation kernel $g(\mathbf{x})$*.*

In the definition of the omnidirectional photography operator, the function $g(\mathbf{x})$ can be interpreted as the response function of an optical lens. In the simple case where $g(\mathbf{x}) = 1$ then Eq. (2.18) reads

$$\mathcal{P}_o[\mathcal{L}](\omega) = \sum_{\mathbf{q}_i} \mathcal{L}(\mathbf{q}_i, M(\mathbf{q}_i,\mathrm{r},\omega)) \tag{2.19}$$

and the effect of $g(\mathbf{x})$ is the same of a wide aperture lens. In the general case, the convolution reads

$$\mathcal{P}_o[\mathcal{L}](\omega) = \sum_{\mathbf{q}_i} g(R\hat{\omega} - \mathbf{q}_i)\mathcal{L}(\mathbf{q}_i, M(\mathbf{q}_i,\mathrm{r},\omega)) \tag{2.20}$$

The combined use of the focus operator $\mathcal{M}$ and the filter $g(\mathbf{x})$ leads to the definition of the synthetic aperture for an omnidirectional system.

If we use a Gaussian interpolant

$$g(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right), \tag{2.21}$$

and if we notate with ω_j the direction in which we want to interpolate, then the value $\mathcal{L}(0, \omega_j)$ is readily found as:

$$\mathcal{L}(0, \omega_j) = \mathcal{P}_o[\mathcal{L}](\omega_j) = \frac{\sum_i g(\|\mathbf{q}_i - R\hat{\omega}_j\|)\mathcal{L}(\mathbf{q}_i, M(\mathbf{q}_i, \mathrm{r}, \omega_j))}{\sum_i g(\|\mathbf{q}_i - R\hat{\omega}_j\|)} \tag{2.22}$$

The interpolation kernel implements the reconstruction filter for the sampled light field. As suggested in [44] the linear interpolation described in Eq. (2.22) can be interpreted as the filtering operated by an optical lens. By varying the parameter σ we are able to change the aperture of the system: a small σ corresponds to a narrow aperture and consequently a large depth-of-field, while a big σ corresponds to a large aperture and a narrow depth-of-field. Differently from an optical system, changing the synthetic aperture does not affect the exposure of the reconstructed image, since we normalize the weights in the reconstruction formula in Eq. (2.22). While other choices of the interpolating kernel are possible, the choice of a Gaussian interpolant permits an easy control over the bandwidth of the reconstruction filter in the frequency domain: the Fourier transform of a Gaussian function with standard deviation σ is again a Gaussian with standard deviation $1/2\sigma$

$$F\{g(d)\}(\Omega_d) \propto \exp\left(-2\sigma^2\Omega_d^2\right).$$

This is a practical advantage in the design of the reconstruction filter. For example, let us suppose that the light field has been sampled finely enough to guarantee the absence of aliasing. Then, in order to perfectly reconstruct the light field, we can focus at optimal distance d_{opt}, defined by:

$$\frac{1}{\mathrm{d}_{opt}} = (\frac{1}{\mathrm{d}_{min}} + \frac{1}{\mathrm{d}_{max}})/2 \tag{2.23}$$

and then use a Gaussian reconstruction filter with $\sigma \leq \Delta\theta_{\mathbf{x}}/2$.

From Eq. (2.14) and Eq. (2.15) we can also derive the concept of hyperfocal distance of the Spherical Light Field Camera. In optics and photography, the hyperfocal distance is a distance beyond which all objects can be brought into an acceptable focus. In our model the hyperfocal distance can be computed from Eq. (2.23) and Eq. (2.14). From Eq. (2.14) we have:

$$\Delta\theta_{\mathbf{x}} = \frac{2\Delta\theta_\omega}{R(\frac{1}{\mathrm{d}_{min}} - \frac{1}{\mathrm{d}_{max}})}$$

then for $\mathrm{d}_{max} \to \infty$ we obtain:

$$\frac{1}{\mathrm{d}_{min}} = \frac{2\Delta\theta_\omega}{R\Delta\theta_{\mathbf{x}}}$$

We can define the hyperfocal distance d_h as:

$$\frac{1}{\mathrm{d}_h} = \frac{1}{2\mathrm{d}_{min}} = \frac{\Delta\theta_\omega}{R\Delta\theta_{\mathbf{x}}} \tag{2.24}$$

2.5 Experimental Results

In this last section we present some experimental results to validate the proposed sampling scheme. The Spherical Light Field Camera is positioned approximatively in the middle of

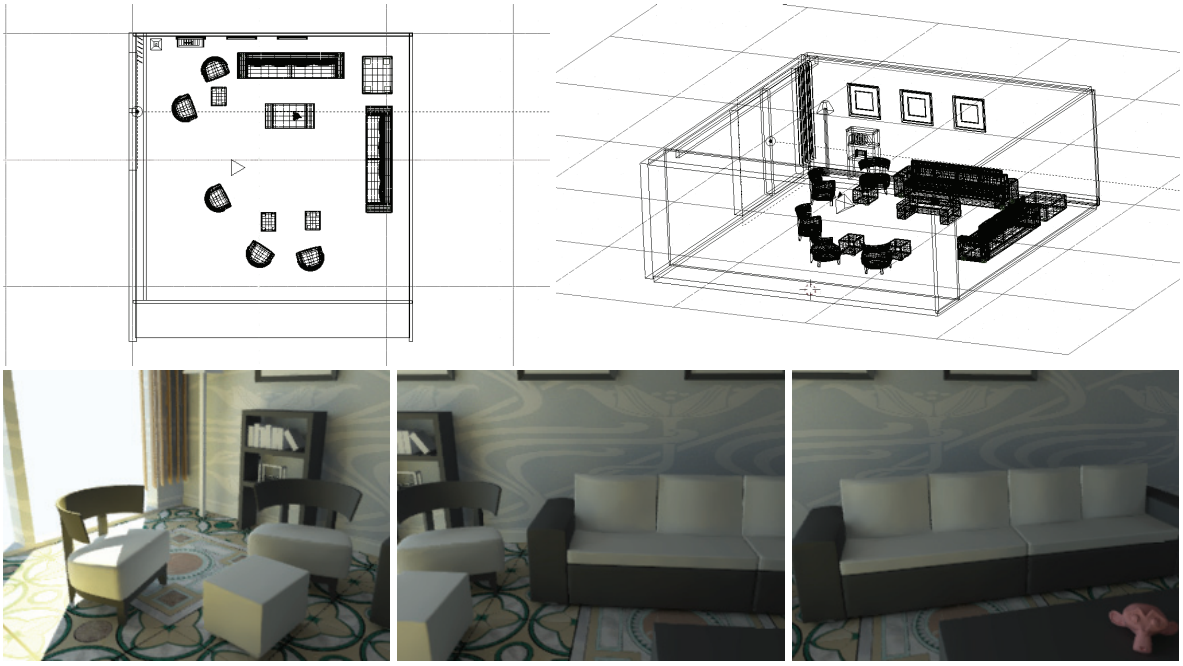


Figure 2.6: *The 3D Blender model for the room scene . On the bottom raw we show three renderings from different cameras on the sphere.*

a living room of size 20x20x3 as shown in Figure 2.6. In Figure 2.7, we show the ground truth for the reconstructed light field at the center of the positional sphere when the rendered image has a vertical angular resolution of $\pi/480$. In the same figure we show four different representations of the spherical image on the plane, that we will use in the thesis. One of the most common representations of the sphere on the plane is obtained by the equirectangular mapping, shown in the top row of Figure 2.7, where we show the image generated with the proposed irregular grid on the left and the image represented onto a equiangular grid of size 960x480 on the right. The equiangular grid offers a more comfortable visual representation, so we will prefer this representation whenever the equirectangular mapping is used. On the bottom row, we show a perspective view of the image mapped on a 2-sphere and a pseudocylindrical sinusoidal projection. We use the sinusoidal projection to show the reconstruction results, because it has the nice property of being an equal-area projection and to nicely fit our irregular sampling scheme. We now test the proposed sampling scheme by comparing the ground truth with the rendered images in the center of the positional sphere $\mathbf{x} = 0$ for different values of the radius of the positional sphere R and the distance d. The images are rendered with an angular resolution of $\pi/480$, *i.e.*, $\Delta\theta_\omega = \Delta s_\omega = \pi/480$, which is a bit lower than the one of the planar cameras. We use the PSNR to measure the quality of the reconstruction and the results are shown in Figure 2.9. In the experiments we used a fixed standard deviation for the gaussian kernel of $\sigma = \Delta_{\mathbf{x}}/2$. As expected the best results are obtained when the sampling conditions are respected, which happens for $R = 0.1$, and when the system is focused close to the optimal focus plane. When we get far from these two conditions the performances degrade and artifacts in the form of blur and double edges become visible in the rendered images as shown in Figure 2.10. The images are produced by interpolating the light field in about 138240 directions, which is a saving of 30% with respect

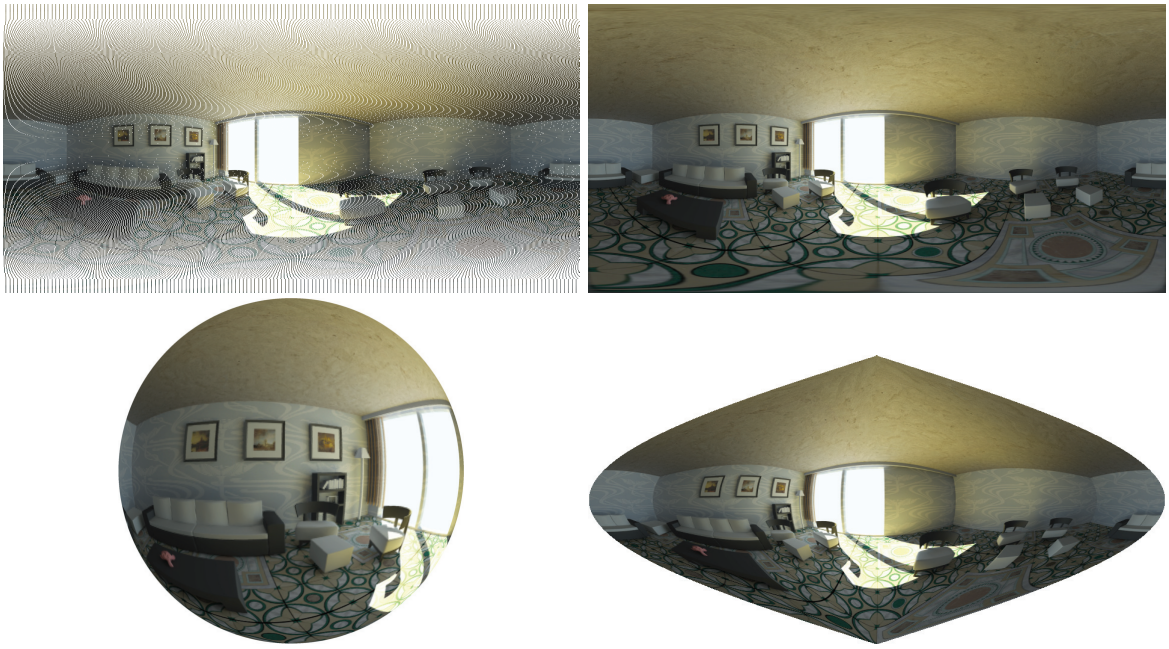


Figure 2.7: *Different representations for the spherical image. Top: equirectangular projection on the proposed irregular grid (left) and on a regular equiangular grid (right). Bottom: a perspective rendering of the image mapped on a 2-Sphere (left), a pseudocylindrical map (right).*

to the equiangular grid of 960x480 for an equivalent quality of the output image. Finally in

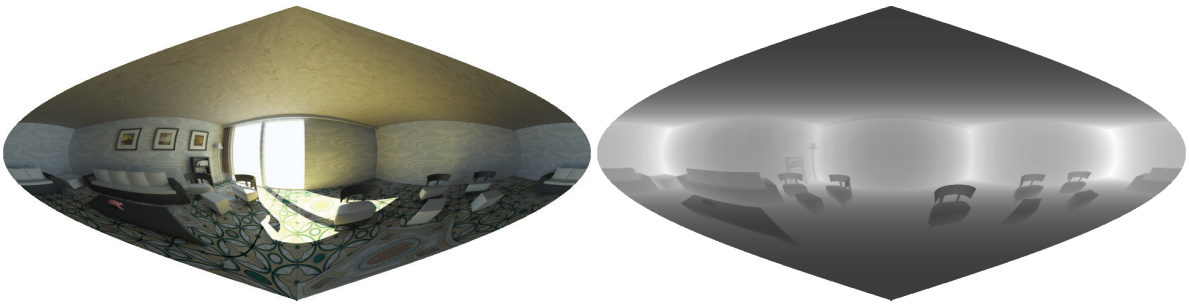


Figure 2.8: *Ground truth spherical image and depth map rendered in the middle of the positional sphere.*

Figure 2.11 we show how we can use the camera as a photographic lens by focusing at two different depths in the scene.

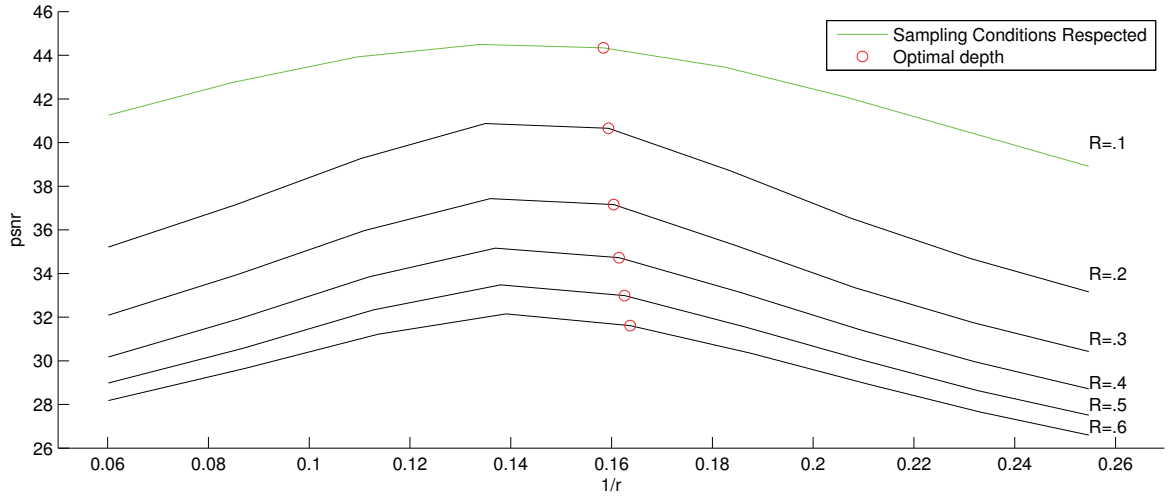


Figure 2.9: *Reconstruction PSNR as function of the inverse distance to the scene radius* $(1/\mathrm{r})$ *and of the radius* R *of the positional sphere.*

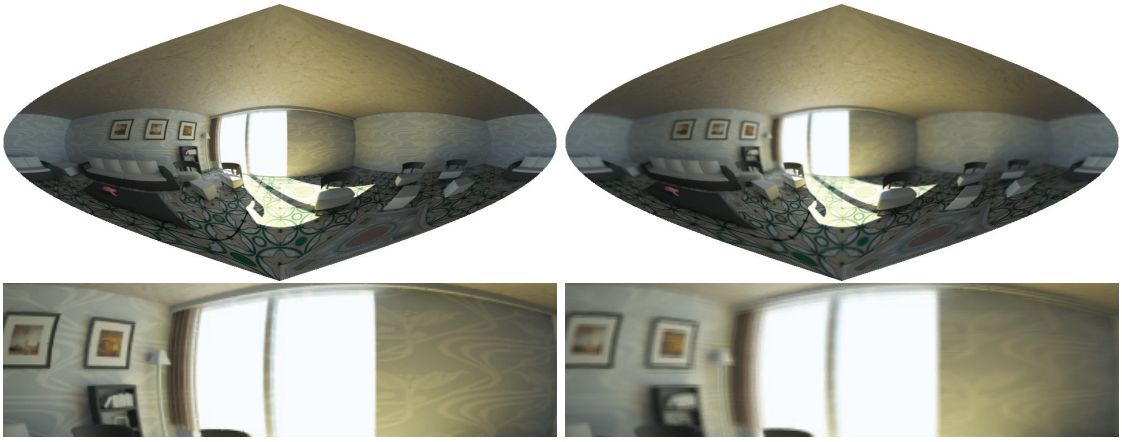


Figure 2.10: *Reconstruction results with focus plane at the optimal distance* d_{opt}*. Left:* $R = 0.1$*, the sampling conditions are respected. Right:* $R = 0.6$*, the sampling conditions are not respected and artifacts are visible. On the bottom a zoom on a detail of the respective reconstructed images.*

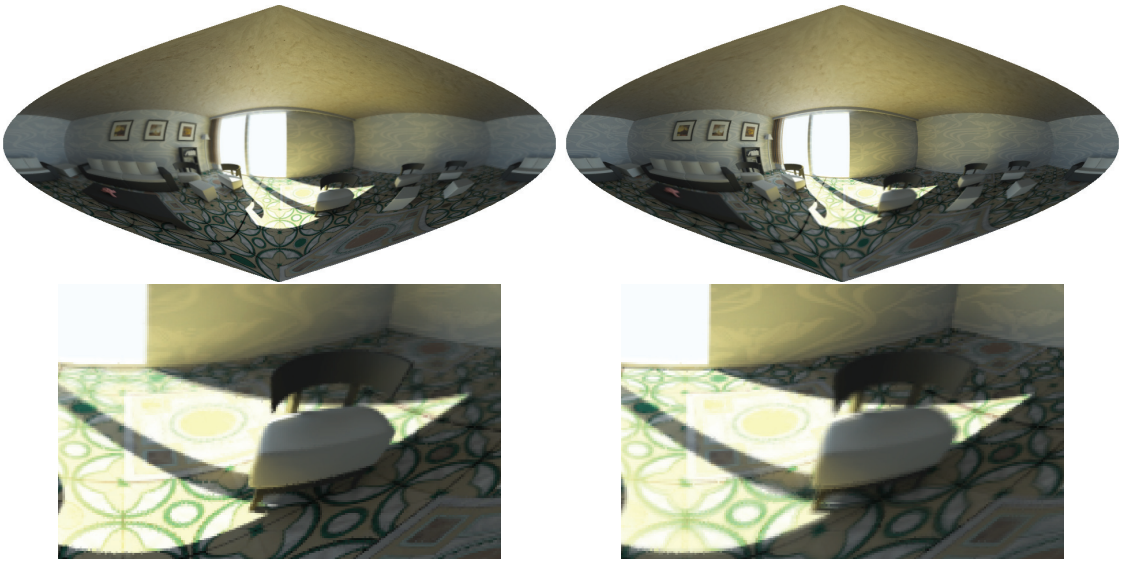


Figure 2.11: *Example of rendering with camera focused at two different depths. On the left the focus is close; on the right the focus is further away. On the bottom a zoom on a close object in the scene is given for each respective image.*

Chapter 3

Spherical Light Field Reconstruction

In Chapter 2 we derive a criterion to sample the omnidirectional 4-D light field using perspective imagers positioned around a sphere. The proposed scheme has the advantage of inducing a uniform sampling of the 2-sphere. The samples do not fall on the regular grid, while the design of digital filters and the Fast Fourier Transform algorithm assume that signals have been sampled on a regular lattice. Existing state-of-the-art techniques to process the light field, like the Fourier Slice Photography theory developed in [62], strongly rely on efficient computation of the Fourier transform. Unfortunately, our samples are non-uniformly distributed and a classical matrix does not efficiently represent the data. In this Chapter we propose an alternative approach to process the light field, which relies on variational techniques defined on undirected graphs. Graphs are very flexible structures that perfectly adapt to the irregular nature of our signal. In recents years transductive graph algorithms [11, 90] have gained much attention as a general way to perform manifold learning, with interesting applications in image processing [29, 67]. The advantage of graph-based methods is that they do not rely on the specific surface parametrization, while the specific geometry can be modeled by an appropriate choice of the graph connectivity. In [47] it is observed that, under certain discretized structures, such as triangle meshes, surfaces can be interpreted as weighted graphs and therefore discrete graph methods can be used instead of intrinsic geometry method. While some works (*e.g.*, [76, 11, 29, 10]) address the study of the convergence of the graph to the approximated manifold, still it is not straightforward to develop theoretical analysis by viewing surfaces as graphs.

The main contributions of this chapter are the definition of an embedding of the light field into what we call a *Light Field Graph* and a formal connection between the linear filtering scheme proposed in Chapter 2 and a class of diffusion processes on graphs.

After giving the definition of the light field graph, we propose two methods to perform a regularized transductive interpolation of the light field on a graph. The interpolation of new values of the plenoptic function is interpreted as an inpainting problem defined on the nodes of the light field graph and the solution is found through a diffusion process. The chosen regularization makes the interpolation robust to noise, as shown in the experimental results, while the algorithms can be implemented in a distributed fashion.

3.1 Graph Representation of the Light Field

Formally, a weighted undirected graph $\Gamma = (V, E, w)$ consists of a set of vertices V, a set of vertices pairs $E \subseteq V \times V$, and a weight function $w : E \mapsto \mathbb{R}^+$ satisfying $w(u, v) > 0$ and $w(u, v) = w(v, u)$, $\forall (u, v) \in E$. In the following, we assume that the graph is connected (for every vertex there exists a path to any other vertex of the graph), and that the graph Γ has no self-loops.

We can define a light field graph by representing every light ray with a node in the graph. We define two disjoint set of vertices, V_D with cardinality N_D and V_I with cardinality N_I, such that the total set of vertices is given by $V = V_D \bigcup V_I$ with cardinality $N_T = N_I + N_D$. The vertices in V_D represent the light rays acquired by the pinhole sensors, while the vertices in V_I represent the light rays that we want to interpolate.

Before giving a formal definition of light field graph let us define a light ray as the vector $\mathbf{l} \in \mathbb{R}^3$ which generates from the scene at point $\mathbf{p} \in \mathbb{R}^3$ and hits an observer at point $\mathbf{x} \in \mathbb{R}^3$. It can be written as : $\mathbf{l} = \mathbf{p} - \mathbf{x}$. It is immediate to see that the intensity of the light ray $\mathbf{l}$ is given by $\mathcal{L}(\mathbf{x}, \hat{\mathbf{l}})$, where the notation $\hat{\mathbf{l}}$ stands for the unit vector associated to $\mathbf{l}$, *i.e.*, $\mathbf{l} = \|\mathbf{l}\|\hat{\mathbf{l}}$.

Definition 3.1.1. The Light Field Graph *The light field graph is defined on the set of nodes V, where each node represent a light ray* $\mathbf{l}$.

Two light rays $\mathbf{l}_1$ *and* $\mathbf{l}_2$ *are connected in the light field graph if one of these two conditions are satisfied:*

1. *They share the same positional coordinates* $\mathbf{x}$. *The distance between the two light rays in* $\mathbb{R}^3$ *is given by definition with* $\|\mathbf{l}_1 - \mathbf{l}_2\| = \|\mathbf{p}_1 - \mathbf{x} - \mathbf{p}_2 + \mathbf{x}\| = \|\mathbf{p}_1 - \mathbf{p}_2\|$.
2. *They originate from the same position* $\mathbf{p}$. *Then* $\|\mathbf{l}_1 - \mathbf{l}_2\| = \|\mathbf{p} - \mathbf{x}_1 - \mathbf{p} + \mathbf{x}_2\| = \|\mathbf{x}_2 - \mathbf{x}_1\|$.

We associate a weight to the edge (u, v) *representing the connection between* $\mathbf{l}_1$ *and* $\mathbf{l}_2$, *using a thresholded Gaussian kernel weighting function as follows:*

$$w(u, v) = \begin{cases} e^{-\|\mathbf{l}_1 - \mathbf{l}_2\|^2/\sigma^2} & \text{if } \|\mathbf{l}_1 - \mathbf{l}_2\| < \tau \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

for some parameter σ *and* τ.

It has been shown in [76] that the chosen definition of the weights $w(u, v)$ has interesting properties in terms of convergence of the embedded graph to the Riemaniann manifold. More specifically, the graph Laplacian asymptotically converges to the continuous manifold Laplacian when the number of samples increases. In practice, however, the choice of an optimal value for σ is not easy and in the following we will consider σ as a parameter.

In a typical construction of the light field graph for interpolation in the origin of the reference system, the set of vertices V_D represents all light rays $\mathbf{l}_D = \mathbf{p} - \mathbf{q}$, while the set of vertices V_I will represent all light rays $\mathbf{l}_I = \mathbf{p} - \frac{\mathbf{p}}{\|\mathbf{p}\|}$.

3.2 Differential Operators on Graphs

In this section we provide the definition of all necessary discrete graph differential operator used in our work. We denote the Hilbert space of a real function on vertices with $\mathcal{H}(V)$, where $f : V \to \mathbb{R}^+$ assigns a real number to each vertex of the graph. A function on the

graph edges is denoted by $F \in \mathcal{H}(E)$, where $F : E \to \mathbb{R}$ assigns a real value to each edge. Following Elmoataz et al [29] we define the gradient $\nabla^w : \mathcal{H}(V) \to \mathcal{H}(E)$ and the divergence $div^w : \mathcal{H}(E) \to \mathcal{H}(V)$ over Γ, $\forall (u, v) \in E$ as:

$$(\nabla^w f)(u, v) = \sqrt{w(u, v)} f(u) - \sqrt{w(u, v)} f(v) \tag{3.2}$$

and

$$(div^w F)(u) = \sum_{v \sim u} \sqrt{w(u, v)} \left(F(v, u) - F(u, v) \right), \tag{3.3}$$

where $v \sim u$ denotes all vertices v connected to u. From the definition of the gradient and the divergence operator we also extrapolate the graph Laplacian operator $\Delta^w : \mathcal{H}(V) \to \mathcal{H}(V)$, defined by:

$$\Delta^w f = -\frac{1}{2} div^w (\nabla^w f), \tag{3.4}$$

that we can write explicitly, using the definitions in Eq. (3.2) and Eq. (3.3), as

$$\Delta^w f(u) = \sum_{v \sim u} w(u, v) \left(f(u) - f(v) \right). \tag{3.5}$$

In the literature, other definitions for the graph operators are found, like the normalized version of Zhou et al [90], [91]:

$$(\nabla^w f)(u, v) = \sqrt{\frac{w(u, v)}{d(u)}} f(u) - \sqrt{\frac{w(u, v)}{d(v)}} f(v) \tag{3.6}$$

and

$$(div^w F)(u) = \sum_{v \sim u} \sqrt{\frac{w(u, v)}{d(v)}} \left(F(v, u) - F(u, v) \right), \tag{3.7}$$

where $d : V \mapsto \mathbb{R}^+$ is the degree function defined as $d(v) = \sum_{u \sim v} w(u, v)$. As remarked in [29], this definition of the gradient operator is not null when the function f is locally constant, which is not a desirable property in image processing applications, since natural images have typically many flat areas and a sparse gradient. Hence, we prefer the definitions given in Eq. (3.2) and Eq. (3.3). We can also define the local isotropic variation of f at vertex v as a measure of roughness of a function around the vertex:

$$\|\nabla_v^w f\| = \sqrt{\sum_{u \sim v} \left[\left(\nabla^w f \right)(u, v) \right]^2}. \tag{3.8}$$

3.3 Interpolation as a Diffusion Process on Graphs

The interpolation problem can be interpreted as finding the function $f : V \to \mathbb{R}$ from an incomplete (noisy) measurement $y \in \mathbb{R}^{N_D}$ of the plenoptic function $\mathcal{L}$ on the nodes in V_D. We also use the following notation: $f = f(V) \in \mathbb{R}^{N_T}$. We formulate the variational inverse problem where we look for a regularized solution f^* s.t.

$$f^* = \underset{f}{\operatorname{argmin}} \|y - \mathrm{A} f\|^2 + J(f) \tag{3.9}$$

where A is a matrix defined as:

$$\mathrm{A} = \mathbb{I}_{V_D} \in \mathbb{R}^{N_D \times N_T}. \tag{3.10}$$

In other words, it is obtained from the identity matrix $\mathbb{I} \in \mathbb{R}^{N_T \times N_T}$ by keeping only the rows corresponding to the vertices in V_D, which restricts f to the available data. If the functional J is convex, we can use proximal iterations to solve it. In recent years many algorithms have been proposed to find a solution to the proposed problem. We refer to the paper of Combettes et al. [25], which contains a review of the most popular ones. Here we restate the forward-backward splitting scheme proposed in [25] as we use it later in this chapter. First we recall the definition of the proximity operator of a convex functional J:

$$\mathrm{Prox}_J(y) = \underset{f}{\operatorname{argmin}} \|y - f\|^2 + J(f). \tag{3.11}$$

An estimate of the function f^* can be obtained by the proximal iteration steps:

$$f^{k+1} = \mathrm{Prox}_{J/\mu}(f^k + \frac{1}{\mu}\mathrm{A}^T(y - \mathrm{A}f^k)), \tag{3.12}$$

where μ is a step size that must be chosen in order to have $\|\mathrm{A}^T\mathrm{A}\| < \mu$. Since the matrix A has only ones on the main diagonal, we consider $\mu = 1$ in our application. The idea of adopting a similar functional for non-local image processing has been recently proposed by Peyre in [67], but, to the best of our knowledge, this is the first time that a similar framework is proposed to interpolate the light rays of a 4D light field. The choice of the regularizer Jdetermines the performance of the algorithm. In the following, we discuss two different regularizers which have the form:

$$J(f) = \lambda \sum_v \|\nabla_v^w F\|^p, \tag{3.13}$$

where λ is a parameter that controls the amount of smoothness in the solution, while for different values of p we control the roughness of the function over the graph. We discuss and show results for $p = 1, 2$. While the proposed algorithms have different performance, they have a common behavior: the solution is obtained through a diffusion process, *i.e.*, the values of the function over the graph evolve through time by local exchange of information between neighbor nodes.

3.3.1 Tikhonov Regularization

We first analyze the case of the regularized problem with $p = 2$. The full minimization problem reads

$$f^* = \underset{f}{\operatorname{argmin}} \|y - \mathrm{A}f\|^2 + \lambda \sum_v \|\nabla_v^w f\|^2. \tag{3.14}$$

Such a regularization encourages the reconstruction of smooth signals; in fact if we write the full regularization term we have

$$J(f) = \sum_v \|\nabla_v^w f\|^2 = \sum_v \sum_{u \sim v} w(u, v)(f(u) - f(v))^2, \tag{3.15}$$

it is easy to see that the energy is small only when f has similar values over close neighbors. The functional defined in Eq. (3.14) is smooth and convex, so a solution f^* can be obtained by the Euler-Lagrange equation

$$\mathrm{A}^T(\mathrm{A}f - y) + \lambda \mathrm{L} f = 0 \tag{3.16}$$

where L is the graph laplacian in matrix form, *i.e.*,

$$\mathrm{L}_{i_u, i_v} = \begin{cases} \sum_{u \sim v} w(u,v) & \text{if } i_u = i_v \\ -w(u,v) & \text{if } u \sim v \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

The Eq. (3.16) defines a linear system of equations, given by:

$$(\lambda \mathrm{L} + \mathrm{A}^T\mathrm{A}) f = \mathrm{A}^T y. \tag{3.18}$$

Since the matrix $(\lambda \mathrm{L} + \mathrm{A}^T\mathrm{A})$ is positive-definite and symmetric, a solution to Eq. (3.18) can be obtained by the conjugate gradient method [69]. Furthermore, since the Laplacian matrix L is sparse, the solution is computed in linear time $O(N_T)$.

Alternatively the functional can be solved by the forward-backward splitting scheme described in Eq. (3.12). To implement one iteration of the algorithm we need to solve the proximal operator:

$$\mathrm{Prox}_J(y) = \underset{f}{\operatorname{argmin}} \, \|y - f\|^2 + \lambda f^T \mathrm{L} f. \tag{3.19}$$

As earlier, if we differentiate with respect to f, the estimate f^* of the proximal operator must be the solution of:

$$(\lambda \mathrm{L} + I) f = y. \tag{3.20}$$

Again, Eq. (3.20) describes a linear system of equations that can be solved with known algorithms, like the conjugate gradient method. An interesting property of Eq. (3.20) is that it can be solved by filtering in the spectral domain as shown in [73, 74]. The authors in [73] propose a solution based on Fourier multiplier operators. Using notions from spectral graph theory [22] the authors develop a way to implement the analogous of linear filters on regular domains. As described in [37], Fourier multiplier operators can be implemented very efficiently using an iterative distributed algorithm based on a truncated series of shifted Chebyshev polynomials. We refer to [73] and references therein for more details.

Spectral Graph Photography We now prove a result that makes an important connection between the diffusion process defined by Eq. (3.14) and the omnidirectional photography operator defined in Section 2.4.

Lemma 3.3.1. *If nodes in the set V_I are only connected to nodes in the set V_D,* i.e., *$\nexists (u,v) \in E, s.t.\, u \in V_I$ and $v \in V_I$, $\forall u, v \in V_I$, the solution of the minimization problem in Eq. (3.14) for $\lambda \to 0$, is given by*

$$f(u) = \begin{cases} \dfrac{\sum_{v \in V_D} w(u,v) y(v)}{\sum_{v \in V_D} w(u,v)} & \text{if } u \in V_I \\ y(u) & \text{if } u \in V_D \end{cases} \tag{3.21}$$

Proof. When $\lambda \to 0$, $\lambda\ f^T \mathrm{L} f \to 0$ and the minimum of Eq. (3.14) is reached for $\mathrm{A}f = y$, which proves that

$$f(u) = y(u), \forall\, u \in V_D. \tag{3.22}$$

We observe that the equation $\mathrm{A}f = y$ can be written as:

$$f = \mathrm{A}^T y + (I - \mathrm{A}^T \mathrm{A}) f. \tag{3.23}$$

Taking the derivative of the functional Eq. (3.14) with respect to f gives

$$\mathrm{A}^T(\mathrm{A}f - y) + \lambda \mathrm{L} f = 0,$$

which using Eq. (3.23) transforms into

$$\lambda \mathrm{L}(\mathrm{A}^T y + (I - \mathrm{A}^T \mathrm{A}) f) = 0 \tag{3.24}$$

The Lemma is proved once we observe that if we consider the i^{th} row of the matrix in Eq. (3.24) corresponding to the node $u_i \in V_I$:

$$\mathrm{L}_i \mathrm{A}^T y = -\sum_{v \in V_D} w(u_i, v) y(v) \tag{3.25}$$

$$\mathrm{L}_i (I - \mathrm{A}^T \mathrm{A}) f = \sum_{v \in V_D} w(u_i, v) f(v_i) \tag{3.26}$$

□

Proposition 3.3.1. Spectral Graph Photography.
The Omnidirectional Photography Operator defined in Eq. (2.18) can be implemented as the solution of the diffusion process defined in Eq. (3.14) on the Light Field Graph Γ, when the nodes in the set V_I are only connected to nodes in the set V_D, and when $\lambda \to 0$.

Proof. The proof comes directly from Lemma 3.3.1 and the observation that Eq. (2.20) is equivalent to Eq. (3.21) when

$$g(\mathbf{x}) = \frac{w(u, v)}{\sum_{v \in V_D} w(u, v)} \tag{3.27}$$

□

Remark 3.3.1. *The proposed graph interpolation method based on Tikhonov regularization can be interpreted as linear filtering of the 4-D light field. If the graph is sparse the algorithm runs in linear time $O(N_T)$. Furthermore the algorithm is fully distributed, since each node of the graph needs only to communicate with its neighbors and to perform simple arithmetic computations.*

3.3.2 Total Variation Regularization

For $p = 1$ in the regularized inverse problem of Eq. (3.9) the functional J becomes:

$$J(f) = \lambda \sum_v \|\nabla_v^w f\|. \tag{3.28}$$

It can be interpreted as the total variation of the light field, since a small energy of the regularizer imply sparse gradients along edges with high weights. The same definition of

Total Variation for graphs is found in [91] and [29], where it is used to perform denoising of signals defined on graphs. Total variation denoising has been extensively used in image processing over the last twenty years after the pioneer work of Rudin and Osher [70] as one of the most effective tools to remove noise. The global functional is given by:

$$f^* = \underset{f}{\operatorname{argmin}} \|y - \mathrm{A}f\|^2 + \lambda \sum_v \|\nabla_v^w f\|, \tag{3.29}$$

and the scheme boils down to a total variation inpainting scheme [21] on graph. Differently from the typical applications of inpainting in imaging, we are not seeking for missing information, but rather looking for a non-linear regularized way to integrate light rays. An interesting interpretation of the bounded variation model applied to the light field is obtained by considering the operation of rendering a spherical perspective image in the center of the sphere: the integration of light rays can be thought of as the filtering by an optical lens. The kernel interpolation simulates the effect of a linear system, since it takes a weighted sum of the collected light rays. This is in fact the behavior of most photographic lenses. Imposing a model of bounded variations for the light ray corresponds then to simulating a non-linear lens that chooses a light ray based on the closest strongest edge. When we get far from the ideal sampling conditions described in Chapter 2, we expect that the total variation model might help to suppress artifacts like double edges or ghosts. We also expect some robustness to noise. These assumptions are confirmed in the experimental results reported in Section 3.4.

To find a solution f^* we have to specify the proximal operator for the total variation norm:

$$\mathrm{Prox}_J(y) = \underset{f}{\operatorname{argmin}} \|y - f\|^2 + \lambda \sum_v \|\nabla_v^w f\|. \tag{3.30}$$

The minimization in Eq. (3.30) corresponds to the total variation image denoising model. An efficient solution, proposed by Chambolle in [19], consists in an iterative fixed point algorithm. The advantage of the algorithm is that it only requires localized operations on the graph, namely divergence and gradient. As most TV denoising algorithms, it is iterative and both gradient and divergence will be computed at each iteration. Chambolle's iterations are defined on a regular rectangular domain; we adapt them to our graph representation:

$$\mathbf{p}^{n+1}(u,v) = \frac{\mathbf{p}^n(u,v) + \tau\nabla^w(div^w\mathbf{p}^n(u) - y(u)/\lambda)(u,v)}{1 + \tau\|\nabla_u^w(div^w\mathbf{p}^n(u) - y(u)/\lambda)\|}, \tag{3.31}$$

where $\mathbf{p}$ is a dual variable which is defined on the edges of the graph, *i.e.*, $\mathbf{p} \in \mathcal{H}(E)$. It has been shown in [19] that for τ small enough and $n \to \infty$ the solution of the proximal operator is obtained by:

$$f(u) = y(u) - \lambda div^w\mathbf{p}^n(u), \tag{3.32}$$

The solution is typically achieved with a linear convergence rate $O(1/k)$; it is further shown in [20] how it is possible to integrate Nesterov accelerations [86] to reach quadratic convergence $O(1/k^2)$.

3.4 Experimental Results

We test the proposed algorithms on the *room scene* using the camera model defined in Section 2.3. During the experiments we choose a rendering angular resolution of $\Delta_\omega = \pi/480$ rad.

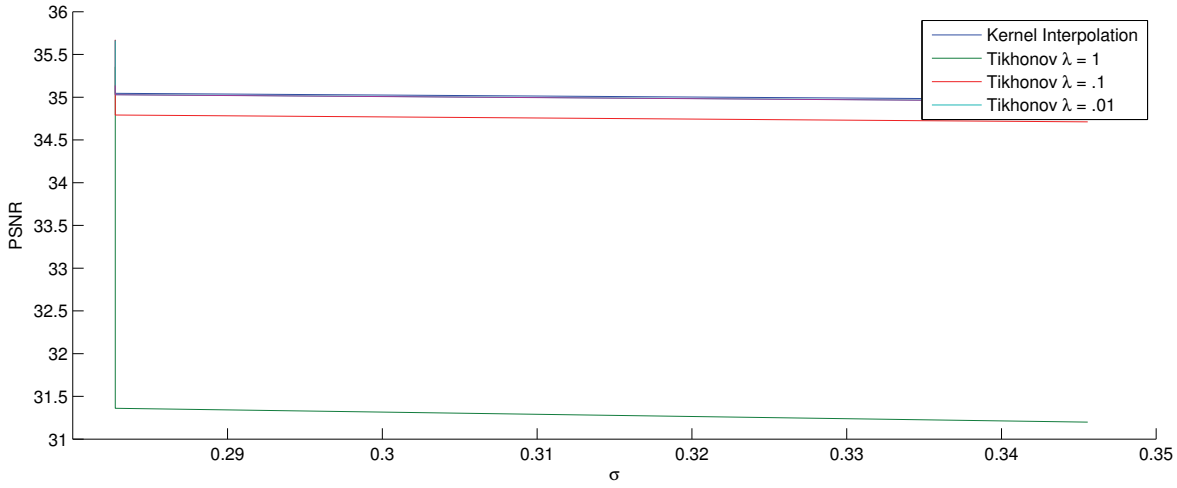


Figure 3.1: *Tuning of λ for the Tikhonov scheme.*

The first experiment in Figure 3.1 shows the influence of the parameter λ on the convergence of the Tikhonov interpolation scheme to the Kernel Interpolation as shown in Section 3.3. In the experiment we used $R = 0.1$. We observe that, if $\lambda < 0.01$ the two schemes produce the same result for a small variation of the parameter σ. In Figure 3.2 we show the influence of the parameter σ on the three different schemes. In the experiments we used $R = 0.2$, $\lambda_{TV} = 0.005$ and $\lambda_{Tik} = 0.1$. We observe that the performance of the TV interpolation scheme is relatively independent of the choice of σ. This is something that we could expect, since the TV interpolation behave as a non-linear lens, which depends much on the intensity values of the closest neighbors, rather than on their absolute distance. The other interesting outcome is that, although the Tikhonov scheme does not perform exactly as the Kernel Interpolation, it reaches the maximum in the PSNR curve for the same value of σ. In Figure 3.4, we show a different scenario: the algorithms are tested for increasing values of

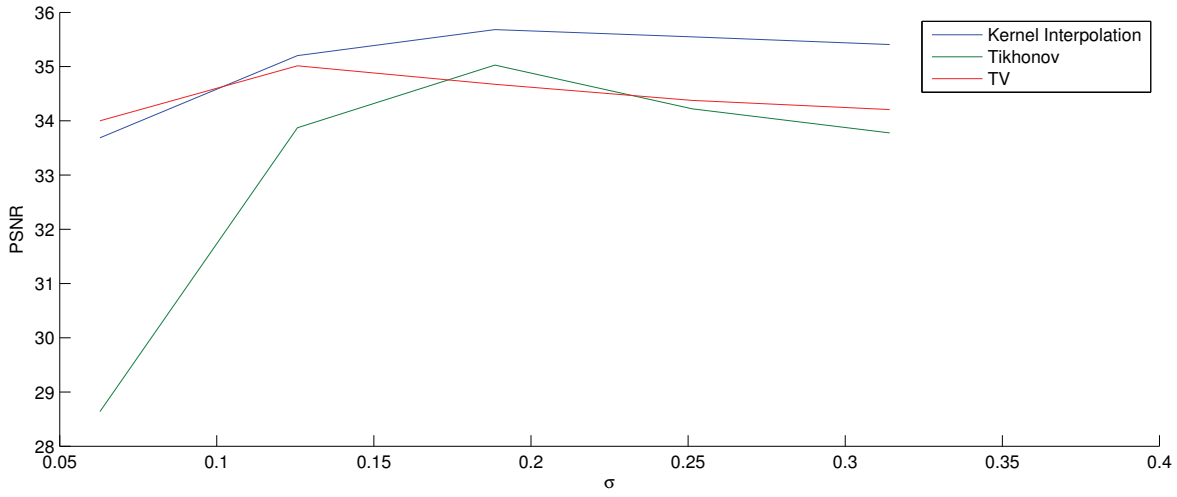


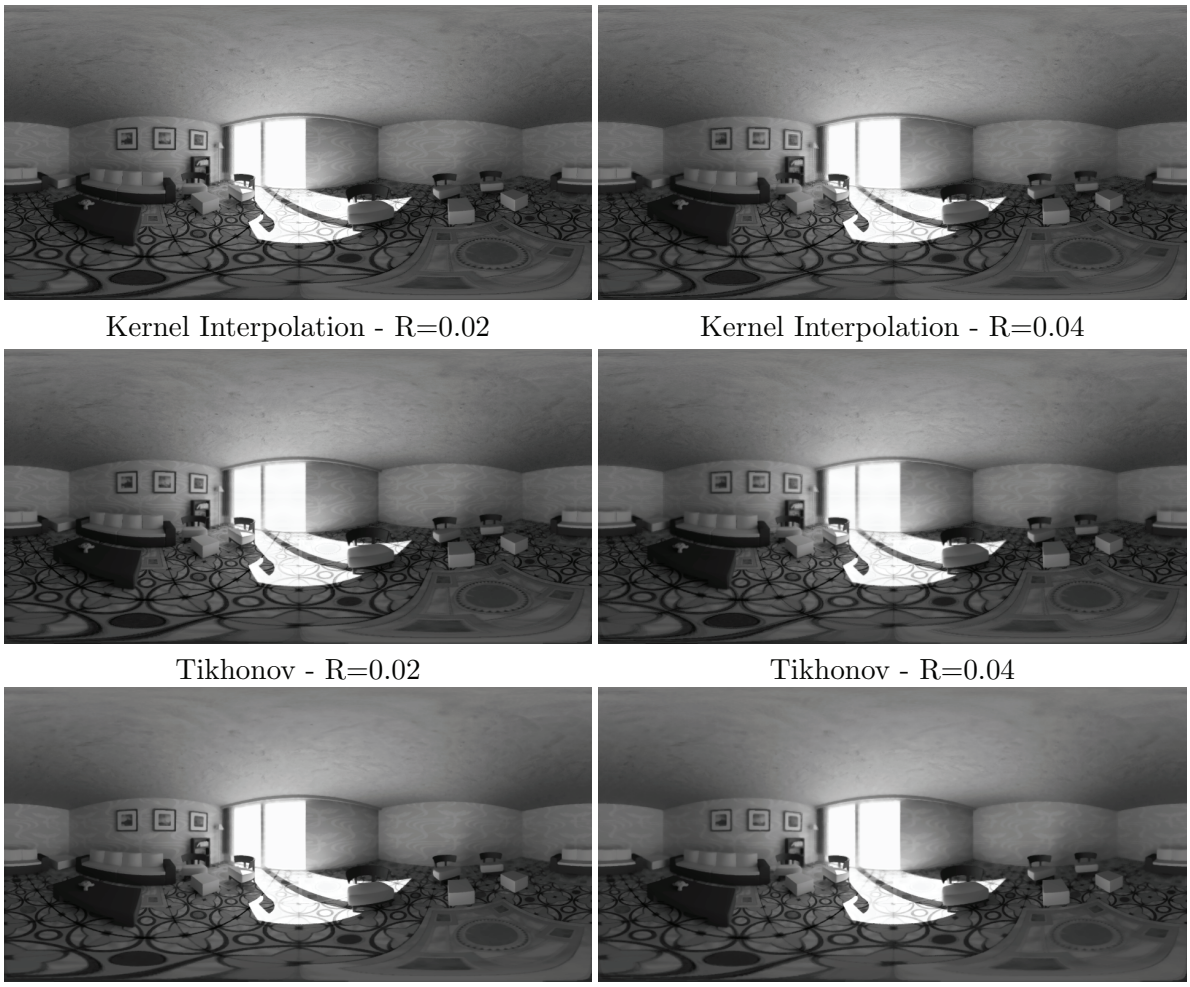
Figure 3.2: *Influence of the choice of σ on the reconstruction methods.*

R while the system has a focus at the hyperfocal distance. For values of the radius $R > 0.3$ double edges begin to appear in the images due to aliasing. As visible in Figure 3.3 the TV scheme renders sharper edges compared to the other methods.

The last test is to verify the robustness to white Gaussian noise. As expected, the regularized solutions (TV and Tikhonov), are more resilient and provide a better PSNR as shown in Figure 3.6 and Figure 3.7. Better results could be obtained by tuning the parameters λ and λ_{Tik}, that we fix instead to a constant value $\lambda_{TV} = 0.005$ and $\lambda_{Tik} = 0.1$.

Kernel Interpolation Tikhonov TV

Figure 3.3: *A zoom on a detail when $R = 0.4$. Visually the TV reconstruction achieves pleasant results, removing some aliasing artifacts*



Kernel Interpolation - R=0.02 | Kernel Interpolation - R=0.04

Tikhonov - R=0.02 | Tikhonov - R=0.04

TV - R=0.02 | TV - R=0.04

Figure 3.4: *Behavior of the algorithms for increasing values of R.*

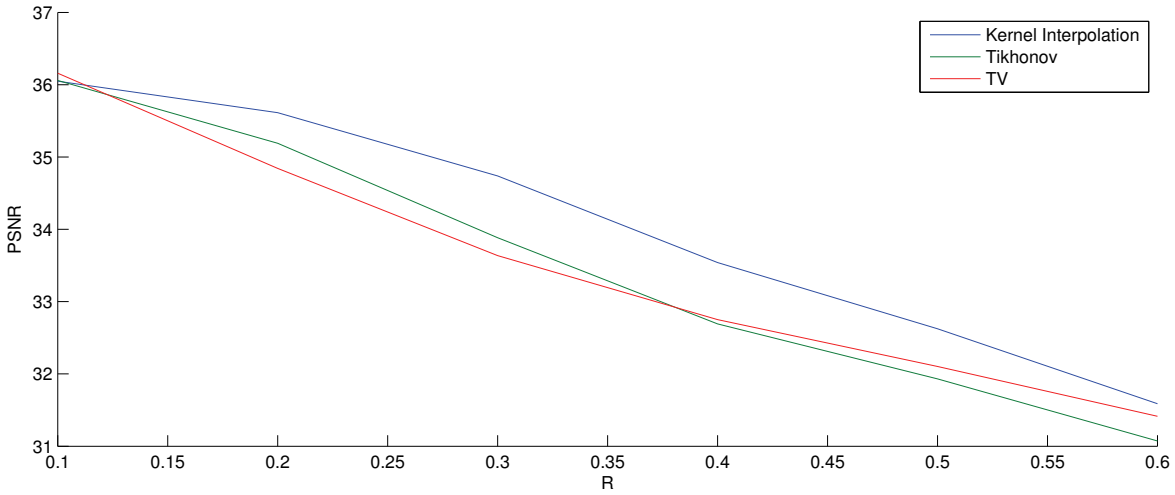


Figure 3.5: *PSNR curve showing the performance of the interpolation algorithms for increasing values of R.*

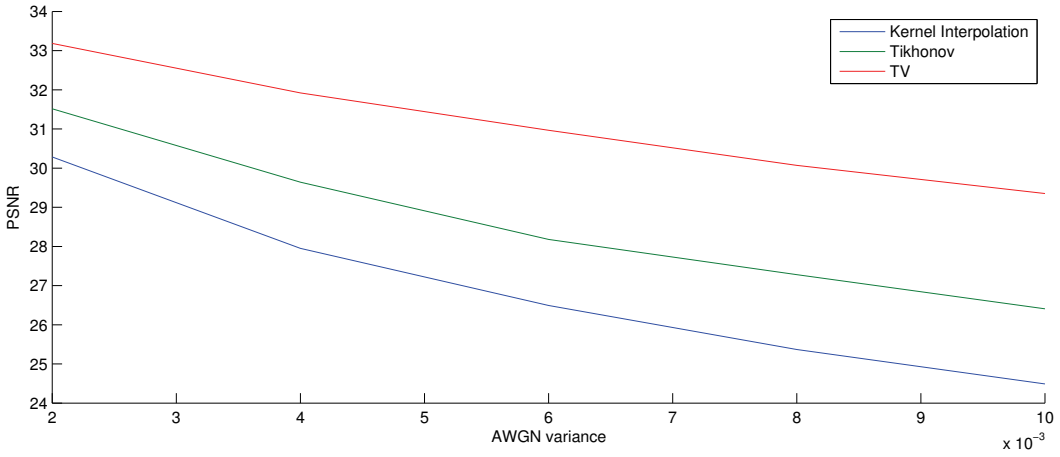


Figure 3.6: *PSNR curve showing the performances of the interpolation schemes when the input image are corrupted with additive white Gaussian noise.*

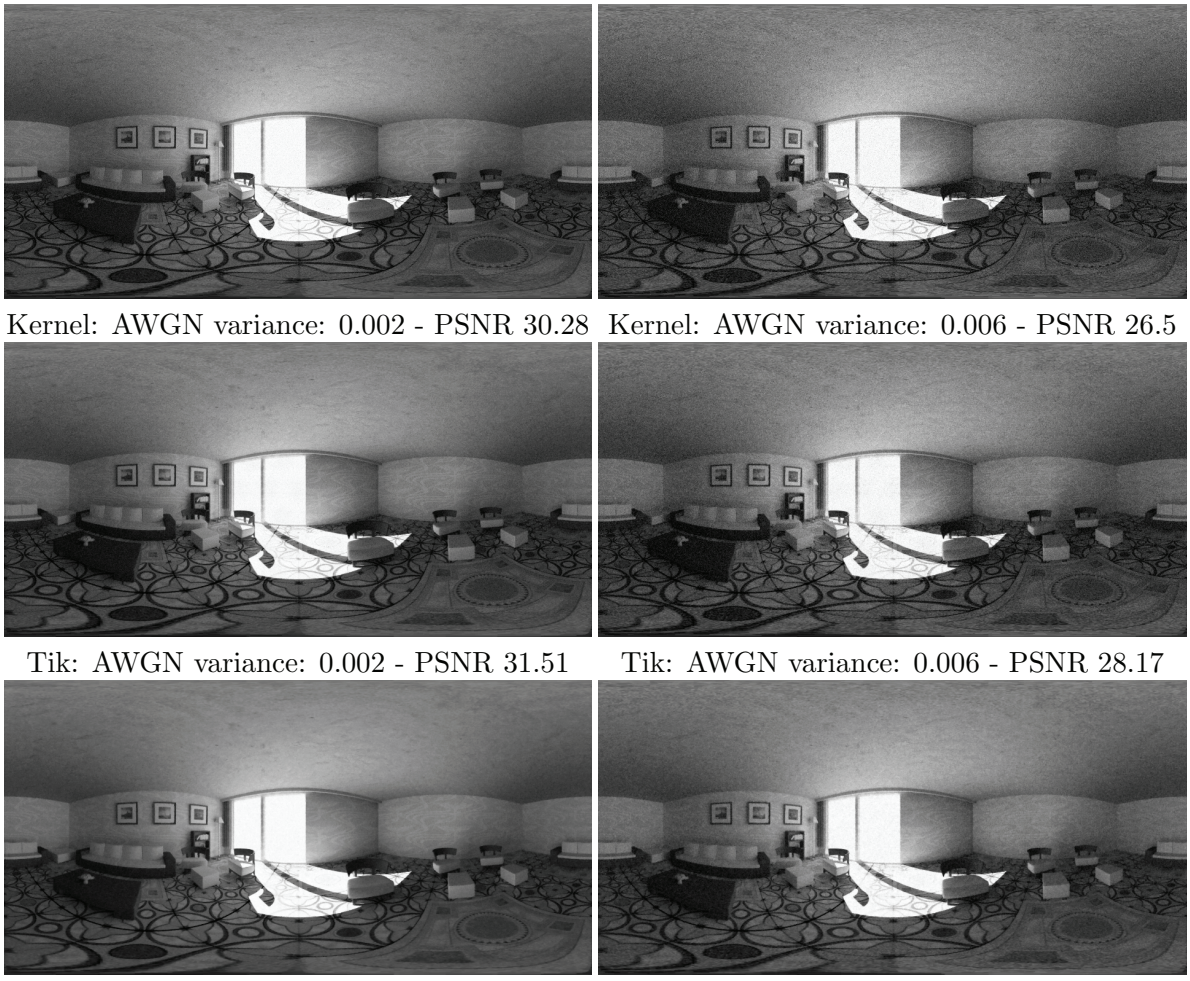


Figure 3.7: *Comparison of interpolation schemes in the presence of additive white Gaussian noise.*

Chapter 4

Depth Extraction from the Light Field

All practical multi-aperture systems need to calculate some structure of the scene in order to operate correctly. For example, if we know the minimum and maximum depths of the scene, we can calculate the optimal focal distance that permits to have all the part of the omnidirectional image in focus, or we could use a depth map to extend the Depth-Of-Field (DOF) limits of the system.

In one of the first works on plenoptic cameras, [2] Adelson has already observed that the displacement of the micro-lenses generates aliasing and blurring. The authors use a least square estimator to calculate and compensate the disparity field among the sub-images. In the lumigraph [36], a rough knowledge of the structure of the scene is used to improve the reconstruction from a sparse set of images. Chai et al. [18] analyze the problem of the minimum sampling rate in the joint image and geometry space: using a truncated Fourier analysis they show that the number of image samples needed decreases with the number of depth levels used, providing a quantitative analysis on the number of depth levels to be used to achieve a good rendering quality. For the Stanford multi-camera array [84] the parallax estimation is part of the system calibration process: using calibration planes placed at different depth in the scene and parallel to the image plane the authors are able to find the camera positions up to an affine transformation. The drawback of their elegant solution is that the calibration is not metric and depends on the 2-plane parametrization of the light field. In a very recent work [12] Bishop shows that an improvement on classic multiview stereo methods [52] can be applied to planar light field cameras leading to substantial gain in the reconstruction of a super-resolved image.

None of the existing references tackles the problem of multiview depth estimation for more complex geometries. In [79] the authors propose a new model for catadioptric sensors, called axial-cone, which permits an efficient rendering of wide FOV images. The model is used to compute a dense depth map by a plane sweeping algorithm [24], which is regularized using graph-cuts [15]. However, the lack of an efficient forward projection in the model makes applications such as structure-from-motion quite complicated.

In this chapter we introduce a novel perspective in the process of extraction of depth information from the sampled omnidirectional light field. We discuss the estimation of dense depth maps from a position inside the sphere. The computation of a dense depth map is a

convenient solution for all image-based rendering applications, since it can be directly used to model the focal surface and render novel views from different perspectives [31][77]. The main contribution of the chapter is the formulation of a novel algorithm for dense multi-view depth estimation which innovates with respect to the state-of-the-art in the following points:

1. We use a spherical camera model to represent the imagers. In the pinhole camera model [39] the projection of a 3D point in space to the image plane depends on the distance of the point from the focal plane. In a typical stereo matching problem the distance to the scene is linearly proportional to the image disparities only if the baseline is parallel to the image plane. This is why in all stereo matching algorithms a pre-processing step is needed to align the image planes. The pinhole camera model then fits well in algorithms that extract depth maps when a two-planes parametrization of the light field is used. But it is not suitable for our spherical light field camera model. It turns out that the proposed spherical camera model is completely general and can be applied to all multiview depth estimation problems, regardless of the geometry of the surface that contains the cameras.

2. We use variational techniques. In the work on light field processing, all the algorithms are based on modified versions of plane sweeping [24] or graph-cuts [15] algorithms. While these methods are proven to be very stable, they require massive amounts of memory, which augment proportionally with the number of depth levels. On the other hand, variational techniques have proven in recent years to be an efficient solution to solve stereo problems [68, 10]. The memory requirement is low since the optimization is based on the diffusion process described in Chapter 2; they can be parallelized and distributed since they only require the exchange of local information among neighbor pixels. These properties make variational techniques hardware-friendly.

3. Our scheme handles gracefully irregular domains, since we use graph-based data structures. It is the first time that variational techniques on graphs are proposed to solve multiview depth estimation problems.

The Chapter is organized as follows: in Section 4.1 we describe the proposed model for estimating a dense depth map from one full sample of the plenoptic function $\mathcal{L}$. In Section 4.2 we show experimental results for the proposed scheme. In the last section we show how the same framework can be used to optimize the calibration parameters of the system.

4.1 Light Field Depth Estimation Algorithm

In this section we address the problem of the estimation of a dense omnidirectional depth map $\mathrm{d}(\mathbf{o}, \omega)$ in a given position $\mathbf{o} \in \mathbb{R}^3$ and a set of directions $\{\omega_i \in S^2\}$. We sample the plenoptic function with the scheme presented in Chapter 2 so we assume to have, at a given instant of time, the set of light rays $\{\mathcal{L}(\mathbf{q}_c, \omega_i), c = 1, 2, \cdots, N_c, i = 1, 2, \cdots, N_I\}$. We do not consider temporal dynamics in this chapter, but we show in Chapter 5 how to extract motion information using samples of the plenoptic function through time. We assume that the scene is Lambertian and free of occlusions. We model the air as a transparent medium, such that there is no attenuation of light intensity along a light ray. The observed light intensity coming from a point in the scene must be the same in $\mathbf{o}$ as in the point $\mathbf{q}_c$ on the surface of the sphere:

$$\mathcal{L}(\mathbf{o}, \omega_i) = \mathcal{L}(\mathbf{q}_c - \mathbf{o}, M(\mathbf{q}_c - \mathbf{o}, \mathrm{d}(\mathbf{o}, \omega_i), \omega_i)), \; \forall c = 1, 2, \cdots, N_c. \tag{4.1}$$

Let us assume $\mathbf{o} = 0$ so that we can drop $\mathbf{o}$ from the equations. In fact, Eq. (4.1) is invariant to a translation of the reference system to $\mathbf{o}$. The idea behind almost all stereo matching algorithms is to find a depth map $\mathrm{d}(\omega)$ such that the conditions in Eq. (4.1) is respected for all directions ω_i. In practice, due to the presence of noise or outliers we seek for a depth map that minimizes a energy functional of the form:

$$J(\mathrm{d}) = J_D(\mathrm{d})) + J_S(\mathrm{d}), \tag{4.2}$$

where J_D enforces the constraint on data, while J_S is an energy term that quantifies the smoothness of the solution. We choose to minimize an energy function $J_D(\mathrm{d})$ that is the sum of the squared pairwise intensity differences of the acquired plenoptic function mapped on the surface defined by $\mathrm{d}(\omega)$, *i.e.*,

$$J_D(\mathrm{d}) = \sum_{\omega_i}^{N_I} \sum_{c_j=1}^{N_c} \sum_{c_k \neq c_j}^{N_c} (\mathcal{L}(\mathbf{q}_{c_j}, M(\mathbf{q}_{c_j}, \mathrm{d}(\omega_i), (\omega_i))) - \mathcal{L}(\mathbf{q}_{c_k}, M(\mathbf{q}_{c_k}, \mathrm{d}(\omega_i), (\omega_i))))^2. \tag{4.3}$$

While the choice of squared intensity differences is quite common in the literature [71], usually it is only applied to a stereo pair and not on a full acquisition of the plenoptic function at a given instant in time. While other norms, like the sum of absolute values, are considered more robust to outliers, the use of this functional is justified because we are averaging several light rays and we can safely assume that the influence of outliers is reduced.

The number of constraints is very high in Eq. (4.3). If we do not include a smoothness term in the energy functional J, the problem of finding a minimum to the functional is however an ill-posed problem since there are large untextured areas in real scenes. The depth map of a typical scene is very smooth with sudden discontinuities, so it is usually modeled with a low total-variation (TV) norm. We have already defined the TV norm in Chapter 3 for the plenoptic graph. In fact we can apply the same definition here once we observe that the function d is defined on the set of vertices V_I and consequently inherits the same underlying connectivity structure. We therefore choose the smoothness term J_S as:

$$J_S(\mathrm{d}) = \lambda \sum_i \|\nabla_i^w \mathrm{d}(\omega_i)\|. \tag{4.4}$$

This definition of TV norm is independent of the discretization scheme, and it adapts well to our irregular sampling. An alternative definition of TV norm for perspective depth map, could be derived by the discretization of the spherical differential operator. We show in Chapter 5 that this is however a source of numerical instabilities in practical situations.

The complete minimization problem for depth estimation now reads:

$$\mathrm{d}^* = \operatorname*{argmin}_{\mathrm{d}} J_D(\mathrm{d}) + J_S(\mathrm{d}). \tag{4.5}$$

The minimization of the functional in Eq. (4.5) poses severe challenges. Since it is non convex, a common solution is to linearize the data term constrain J_D and to compensate for the model simplification with the so-called warping technique [64].

Let us now write Eq. (4.1) explicitly:

$$\mathcal{L}(\omega_i) = \mathcal{L}(\mathbf{q}_c, \frac{\mathrm{d}(\omega_i)\hat{\omega}_i - \mathbf{q}_c}{\|\mathrm{d}(\omega_i)\hat{\omega}_i - \mathbf{q}_c\|}), \forall c = 1, 2, \cdots, N_c. \tag{4.6}$$

where $\hat{\omega}_i$ is the unit vector in the 3-D space associated to ω_i. The mapping defined inside the above equation is not linear because $\omega_i \in S^2$. The actual form of the mapping makes the linearization of the functional J a non trivial problem.

We propose the following solution. Let us extend the domain of definition of $\mathcal{L}$ from $\mathbb{R}^3 \times S^2$ to $\mathbb{R}^3 \times \mathbb{R}^3$. Since we supposed the absence of occluders, we can safely write:

$$\mathcal{L}(\mathbf{q}_c, \frac{\mathrm{d}(\omega_i)\hat{\omega}_i - \mathbf{q}_c}{\|\mathrm{d}(\omega_i)\hat{\omega}_i - \mathbf{q}_c\|}) = \mathcal{L}(\mathbf{q}_c, \mathrm{d}(\omega_i)\hat{\omega}_i - \mathbf{q}_c) = \mathcal{L}(\mathbf{q}_c, \hat{\omega}_i - \mathrm{z}(\omega_i)\mathbf{q}_c), \tag{4.7}$$

where $\mathrm{z}(\omega_i) = \frac{1}{\mathrm{d}(\omega_i)}$. We linearize Eq. (4.7) using a first order Taylor expansion around $\mathrm{z} - \mathrm{z}_0$, where z_0 represents a previous estimation of the depth map:

$$\mathcal{L}(\mathbf{q}_c, \hat{\omega}_i - \mathrm{z}(\omega_i)\mathbf{q}_c) \simeq \mathcal{L}(\mathbf{q}_c, \frac{\hat{\omega}_i - \mathbf{q}_c \mathrm{z}_0(\hat{\omega}_i)}{\|\hat{\omega}_i - \mathbf{q}_c \mathrm{z}_0(\hat{\omega}_i)\|}) - \mathbf{q}_c \cdot \nabla_\omega \mathcal{L}(\mathbf{q}_c, \hat{\omega}_i - \mathbf{q}_c \mathrm{z}_0(\hat{\omega}_i))(\mathrm{z}(\hat{\omega}_i) - \mathrm{z}_0(\hat{\omega}_i)). \tag{4.8}$$

where $\mathbf{q}_c \cdot \nabla_\omega \mathcal{L}(\mathbf{q}_c, \hat{\omega}_i)$ is the usual scalar product in $\mathbb{R}^3$. We immediately observe that, by applying successive warpings, the term $\mathrm{z}(\hat{\omega}_i) - \mathrm{z}_0(\hat{\omega}_i)$ tends to zero and the linearized term is constrained back to the original domain $\mathbb{R}^3 \times S^2$, which intuitively explains why our problem is correctly posed. Using the linearized constraint J_D now reads:

$$J_D(\mathrm{z}) = \sum_{\omega_i}^{N_I} \sum_{c_j=1}^{N_c} \sum_{c_k \neq c_j}^{N_c} (\mathcal{L}(\mathbf{q}_{c_j}, \omega_i) - \mathcal{L}(\mathbf{q}_{c_k}, \omega_i) + \mathbf{q}_j \cdot \nabla_\omega \mathcal{L}(\mathbf{q}_{c_j}, \omega_i)\mathrm{z}_i - \mathbf{q}_k \cdot \nabla_\omega \mathcal{L}(\mathbf{q}_{c_k}, \omega_i)\mathrm{z}_i)^2, \tag{4.9}$$

where, for the sake of clarity in notation we use z_i to indicate $\mathrm{z}(\omega_i)$, and we drop the variable z_0. We also use the following notation:

$$y_{jk}(\omega_i) = \mathcal{L}(\mathbf{q}_{c_j}, \omega_i) - \mathcal{L}(\mathbf{q}_{c_k}, \omega_i) \tag{4.10}$$

$$a_{jk}(\omega_i) = \mathbf{q}_j \cdot \nabla_\omega \mathcal{L}(\mathbf{q}_{c_j}, \omega_i) - \mathbf{q}_k \cdot \nabla_\omega \mathcal{L}(\mathbf{q}_{c_k}, \omega_i) \tag{4.11}$$

The full linearized functional can be now written as:

$$J(\mathrm{z}) = \sum_{\omega_i}^{N_I} \sum_{c_j=1}^{N_c} \sum_{c_k \neq c_j}^{N_c} (a_{jk}(\omega_i)\mathrm{z}(\omega_i) + y_{jk}(\omega_i))^2 + \lambda \sum_i \|\nabla_i^w \mathrm{z}(\omega_i)\|. \tag{4.12}$$

The problem defined in Eq. (4.5) can be solved by minimizing Eq. (4.12) by looking for the optimal value of z. It must be observed that both d and its multiplicative inverse z have a small total variation so the functional defined in Eq. (4.12) is well defined. The linearized energy data term J_D is now convex and differentiable so we can use the forward-backward splitting method, similarly to Section 3.3. The proximal operator Prox_{J_S} needed for the backward step of the method is the same TV prox used in Section 3.3.2. The full algorithm is summarized below in `Algorithm 4.1`.

Choose $\gamma \leqslant 1/\beta$, where β is the Lipschitz constant of J_D
for $n_w = 1, .., N_w$ **do**
  $\mathrm{z}_0 = \mathrm{z}$
  $y_{jk}(\omega_i) = \mathcal{L}(\mathbf{q}_{c_j}, \omega_i - \mathrm{z}_0(\hat{\omega}_i)\mathbf{q}_{c_j})) - \mathcal{L}(\mathbf{q}_{c_k}, \omega_i - \mathrm{z}_0(\hat{\omega}_i)\mathbf{q}_{c_k}))$
  $a_{jk}(\omega_i) = \mathbf{q}_j \cdot \nabla_\omega \mathcal{L}(\mathbf{q}_{c_j}, \omega_i - \mathrm{z}_0(\hat{\omega}_i)\mathbf{q}_{c_j}) - \mathbf{q}_k \cdot \nabla_\omega \mathcal{L}(\mathbf{q}_{c_k}, \omega_i - \mathrm{z}_0(\hat{\omega}_i)\mathbf{q}_{c_k}))$
  for $n = 1, ..., N$ **do**
    $$\nabla J_D = 2 * \sum_{\omega_i}^{N_I} \sum_{c_j=1}^{N_c} \sum_{c_k \neq c_j}^{N_c} (a_{jk}(\omega_i)(\mathrm{z}_i^n - \mathrm{z}_0) + y_{jk}(\omega_i))$$
    $\mathrm{z}^{n+1} = \mathrm{Prox}_{\gamma J_S}(\mathrm{z}^n - \gamma J_S)$
  end for
end for

Algorithm 4.1: Light Field Depth Estimation (LFDE)

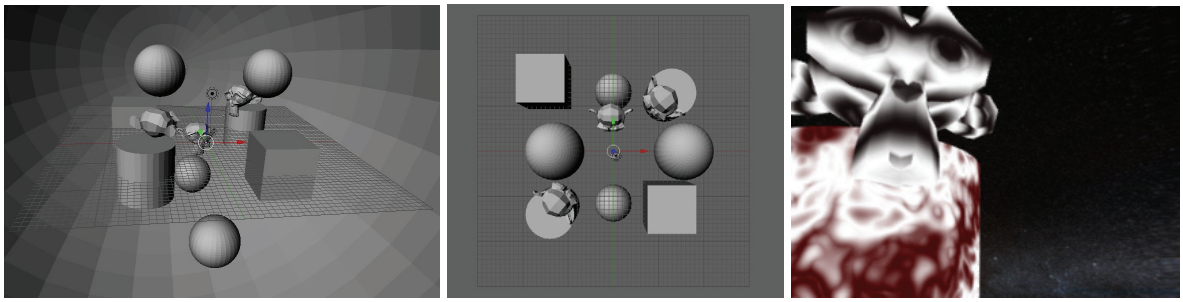
We remark that the Light Field Depth Estimation (LFDE) algorithm is completely general, because it directly uses the definition of the plenoptic function. It does not depend on a particular discretization scheme because it is based on a generic graph structure. The only assumption we make is that the scalar product $\mathbf{q}_c \cdot \nabla_\omega \mathcal{L}(\mathbf{q}_c, \hat{\omega}_i)$ can be computed. This implies the knowledge of $\mathbf{q}_c$, which can be computed in an a priori calibration phase. If a metric calibration is not needed, $\mathbf{q}_c$ can be calculated up to a multiplicative scale factor without compromising the quality of the depth estimation. The gradient terms $\nabla_\omega \mathcal{L}(\mathbf{q}_c, \hat{\omega}_i)$ can be easily calculated if $\hat{\omega}_i$ are defined on a spherical grid, otherwise one can directly calculate the scalar product by setting a least-square problem, *i.e.*, the directional gradient can be calculated in a neighborhood of $\hat{\omega}_i$ as:

$$\mathbf{q}_c \cdot \nabla_\omega \mathcal{L}(\mathbf{q}_c, \hat{\omega}_i) = \sum_{\hat{\omega}_j \sim \hat{\omega}_i} \frac{(\hat{\omega}_j - \hat{\omega}_i)(\mathcal{L}(\mathbf{q}_c, \hat{\omega}_j) - \mathcal{L}(\mathbf{q}_c, \hat{\omega}_i)) \cdot \mathbf{q}_c}{\|\hat{\omega}_j - \hat{\omega}_i\|^2} \tag{4.13}$$

Finally, the LFDE algorithm only requires local simple arithmetic computations that can be easily implemented in hardware. It is straightforward to see that the terms $y_{jk}(\hat{\omega}_i)$ and $a_{jk}(\hat{\omega}_i)$ can be computed by an exchange of local information between the plenoptic graph nodes $(\omega_i, \mathbf{q}_{c_j})$ and $(\omega_i, \mathbf{q}_{c_k})$.

4.2 Performance Assessment

To evaluate the performance of the proposed LFDE algorithm, we simulate the spherical light field camera presented in Section 2.3, with unitary radius $R = 1$, in a 3D *space scene* that presents a broad depth range varying from 4 to 100 units. We show in Figure 4.1 a geometrical representation of the Blender model and one example of a rendered view. In Figure 4.2 we show the ground truth for the *space scene* . We render the omnidirectional image and the depth map in the origin of the coordinate system, onto an equiangular grid of 480x240 for ease of visualization. To quantify the quality of the estimation we use three criteria: 1) the PSNR between the estimated depth map and the ground truth (although the PSNR is not the perfect metric to quantify the quality of a depth map, it gives an indication on how well the algorithm can find the structure of the scene); 2) the PSNR between the omnidirectional image rendered with the estimated depth and the omnidirectional image ground truth; 3) visual inspection. We also choose a simple Winner-Take-All (WTA) version of the `Algorithm 4.1` as a baseline



a) 3D View b) Top View c) Example of planar image

Figure 4.1: *The Blender model that generates the space scene*

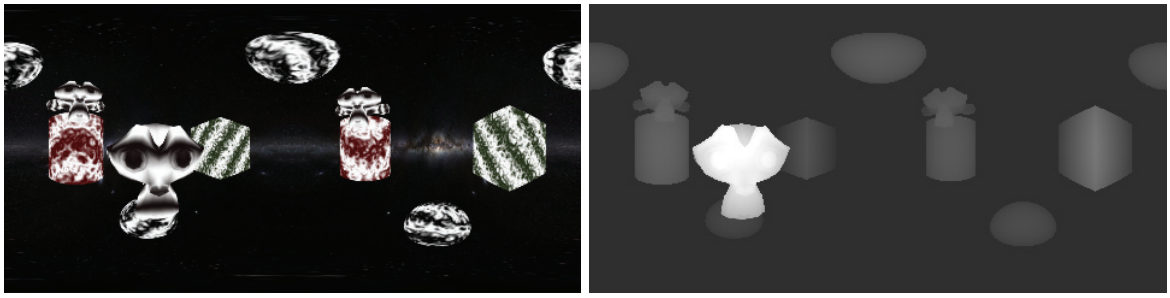
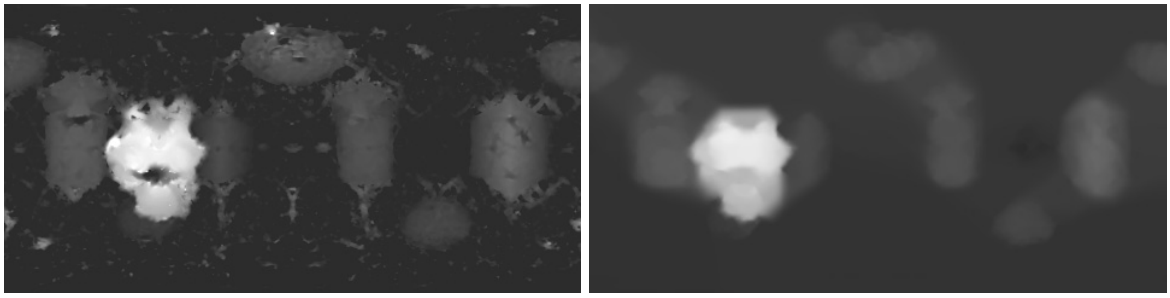


Figure 4.2: *The ground truth for the space scene . Omnidirectional image on the left, depth map on the right.*

comparison. The baseline scheme is implemented as follows. We impose that for a given ω_i, $\mathrm{z}(\omega)$ remains constant in a neighborhood of ω_i such that we can write the optimization problem as:

$$\mathrm{z}(\omega_i)^* = \underset{\mathrm{z}}{\operatorname{argmin}} \sum_{\omega \sim \omega_i} J_D(\mathrm{z}(\omega_i)), \tag{4.14}$$

where the functional $J_D(\mathrm{z})$ is described in Eq. (4.9). It is straightforward to solve the minimization above since it is a simple quadratic function of ω_i. In other words the WTA perform a local optimization and can be thought of as the non-regularized version of the light field depth estimation algorithm.



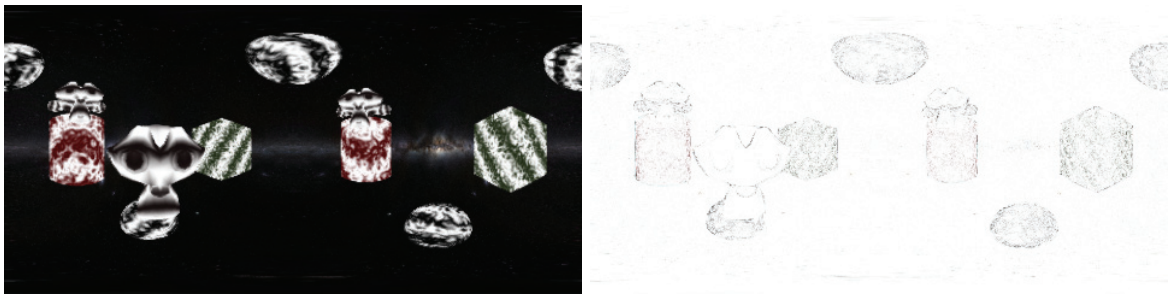
WTA (PSNR=28.6 dB LFDE (PSNR=35.4 dB

Figure 4.3: *Estimated Depth map for space scene , for WTA and LFDE algorithms.*

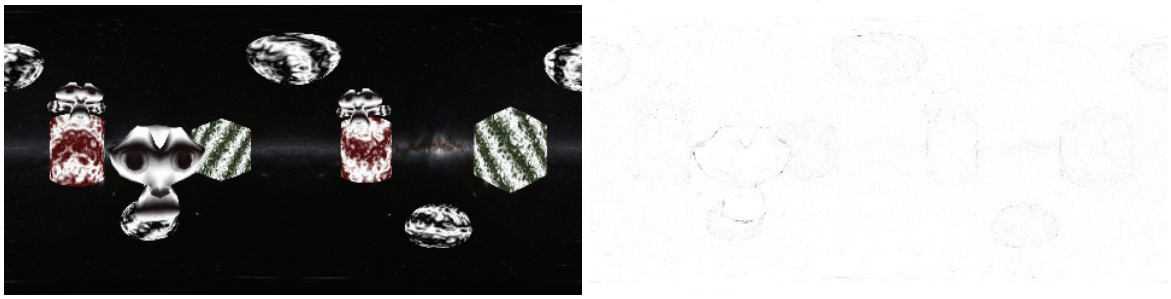
We know study the performance of the depth estimation algorithm. In Figure 4.3 we

compare both algorithms WTA and LFDE on the *space scene* . The LFDE produces smoother depth maps with higher PSNR. The depth maps found with WTA are, as expected, very clumsy in untextured areas. In Figure 4.4, we compare the algorithms in terms of rendering performance. In the comparison we include the image rendered when the system is focused at the hyperfocal distance. We can see that both algorithms perform well in terms of reduction of the residual error with respect to the ground truth. We also observe that the PSNR values are almost identical. In fact this is just due to the absence of background in this particular scene.

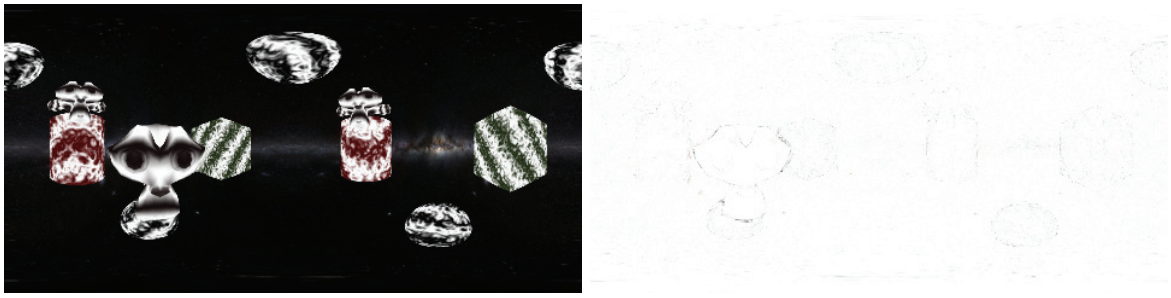
Then we run the comparison on the *room scene* , which is very textured. We show in Figure 4.5 the results on a portion of the full directional space. Since the depth range in this scene is quite small (from 10 to 14 units) compared to the resolution of the imagers, the estimated depth map is less precise compared to the other scene. We observe though that the WTA algorithm produces an incomplete depth map and induces a lot of artifacts during the rendering, while LFDE performs well on both fronts.



Hyperfocal distance (PSNR=28.5 dB)

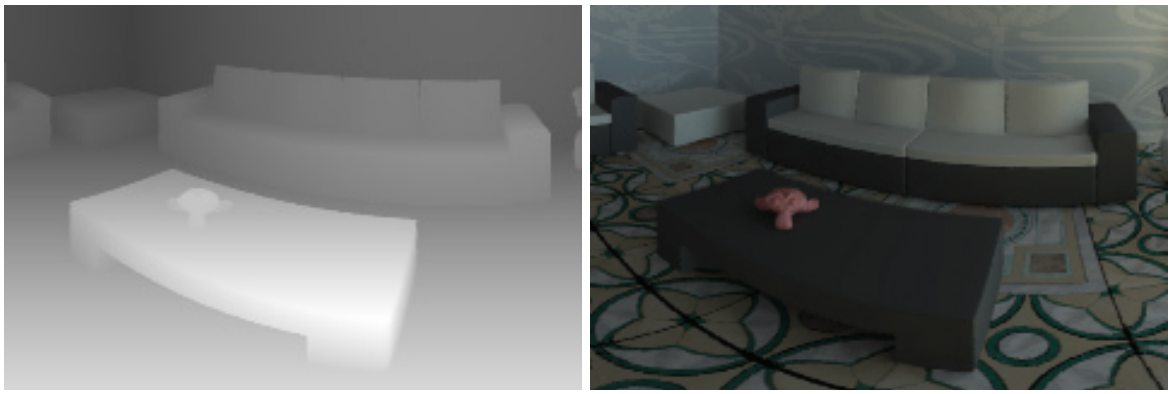


WTA (PSNR=35.4 dB)

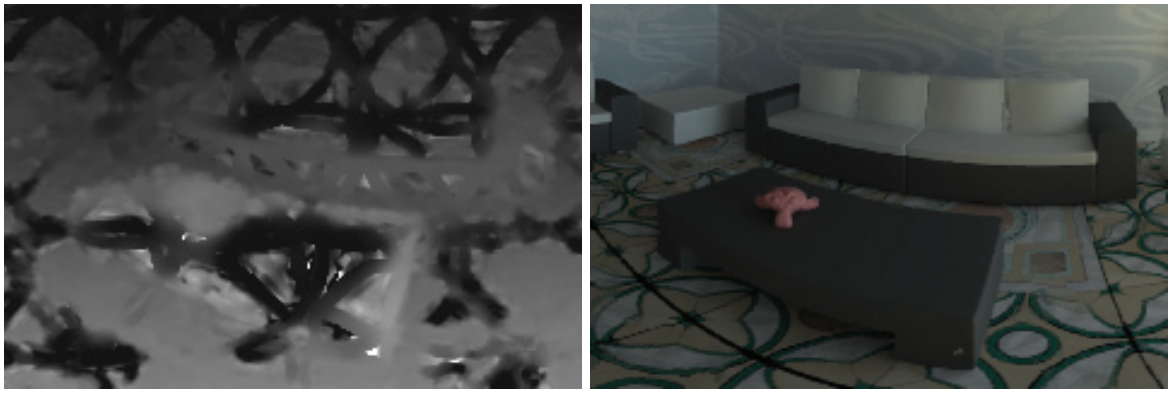


LFDE (PSNR=35.38 dB)

Figure 4.4: *Comparison of rendered images using the estimated depth maps. On the right column we show the image residual of the render images with respect to the ground truth.*



Ground truth depth | Ground truth image



WTA (PSNR=20.9 dB) | WTA (PSNR=35.8 dB)

LFDE (PSNR=23 dB) | LFDE (PSNR=38.54 dB)

Figure 4.5: *Performance of the depth map estimation for the room scene . This scene has a complex texture, so the WTA algorithm performs poorly. On the left we show the depth map while on the right we show the image rendered using the estimated depth map.*

4.3 Automatic Calibration

In Section 4.1 we implicitly assume that the system is perfectly calibrated. In this last section we show how to use the same framework described in Section 4.1 in order to optimize some of the calibration parameters of the model.

Let us consider two neighbor cameras c_j and c_k. From Eq. (4.7) we know that for a given direction ω_i:

$$\mathcal{L}(\mathbf{q}_{c_j}, \hat{\omega}_i - \mathrm{z}(\omega_i)\mathbf{q}_{c_j}) = \mathcal{L}(\mathbf{q}_{c_k}, \hat{\omega}_i - \mathrm{z}(\omega_i)\mathbf{q}_{c_k})$$

If we want to model an imperfect knowledge of the cameras orientations we can simply write:

$$\mathcal{L}(\mathbf{q}_{c_j}, \mathrm{R}_j(\hat{\omega}_i - \mathrm{z}(\omega_i)\mathbf{q}_{c_j})) = \mathcal{L}(\mathbf{q}_{c_k}, \mathrm{R}_k(\hat{\omega}_i - \mathrm{z}(\omega_i)\mathbf{q}_{c_k})), \tag{4.15}$$

where R_j and R_k are the rotation matrices that correct the imperfect camera orientations. If the rotation angles are small, a more convenient expression for the matrix rotation is obtained using the exponential map from $so(3) \rightarrow SO(3)$:

$$\mathrm{R} = I + [\hat{r}]_x \sin(\alpha) + [\hat{r}]_x^2(1 - \cos(\alpha) \simeq I + [\hat{r}]_x \sin(\alpha) \tag{4.16}$$

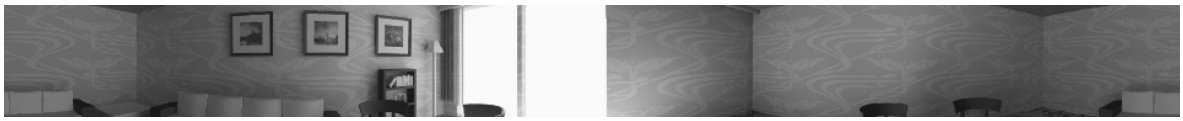
where $[\hat{r}]_x \sin(\alpha)$ is the antisymmetric matrix equivalent to the cross product, *i.e.*, $[\hat{r}]_x \sin(\alpha)\mathbf{q} = \sin(\alpha)\hat{r} \times \mathbf{q}$. We also write $r = \hat{r}\sin(\alpha)$. This approximation is justified by the use of small FOV sensors. Let us assume for now that the depth map z is known, so that we can drop it from the equations. To find an estimate of $r = [r_1 r_2 \cdots r_{camNum}]$ we consider the same energy functional defined in Eq. (4.3):

$$J_D(r) = \sum_{\omega_i}^{N_I} \sum_{c_j=1}^{N_c} \sum_{c_k \neq c_j}^{N_c} (\mathcal{L}(\mathbf{q}_{c_j}, \omega_i - [\omega_i]_x r_j) - \mathcal{L}(\mathbf{q}_{c_k}, \omega_i - [\omega_i]_x r_k))^2, \tag{4.17}$$

where we made use of the cross product property $[r]_x\omega = -[\omega]_x r$. The functional in Eq. (4.17) can be efficiently minimized using a Levenberg-Marquardt scheme [57]. The proposed approach has interesting connections with the classical bundle adjustment [82] methods used to refine the camera pose from a set of 3D correspondences. A similar idea has also been proposed by Tron [83] in the context of distributed camera pose estimation. Lovegrove propose to align a set of rotated images using a whole image alignment in [53]. Their model is much simpler then ours, since it supposes that there is no parallax among pictures. The main difference with previous works is that our algorithm directly works with the plenoptic function, rather than with a set of extracted features. This has two main advantages: i) it does not require the computation nor the matching of features, ii) it is optimized to produce visually pleasant results, since it minimizes the reprojected intensity error.

We illustrate the benefits of the above model with experiments on the syntethic *room scene* . We only consider the 20 cameras around the equator, which are the ones with enough texture information. We apply a random rotation on the camera pose, with angles chosen from a Gaussian distribution with standard deviation of 0.02. We choose a rendering angular resolution of 0.006. The automatic rotation estimation is able to reduce the average angular error, defined as the angle difference with respect to the ground truth camera orientation, from 0.016 to 0.008, reaching the limits of the rendering output resolution. The results are shown in Figure 4.7. The reduction of the angular error is also confirmed by visual inspection of the rendered images as shown in Figure 4.6.

Before optimization (PSNR=27.03 dB)



After Optimization (PSNR=31.17 dB)

Zoom on a detail

Figure 4.6: *Reconstruction before and after the automatic pose optimization tested on the room scene . In the bottom row we show a detail of the reconstruction. We observe that after optimization (right image) many artifacts disappear.*

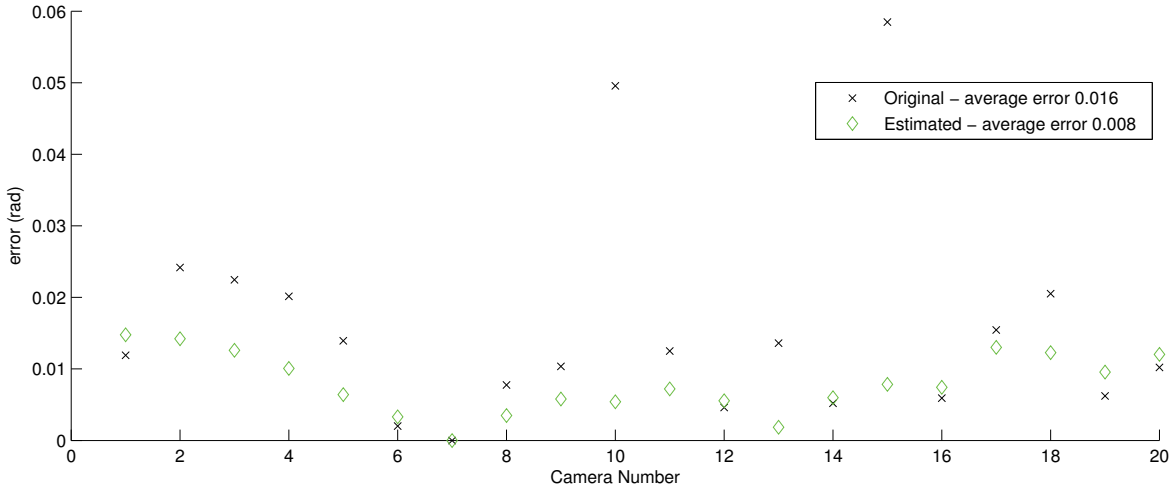


Figure 4.7: *Angular error in rad with respect to the ground truth. After the optimization the average error is reduced of a factor 3. The average error after the optimization is below the angular resolution,* i.e., $0.01rad$.

Chapter 5

Omnidirectional Dense Structure-From-Motion

Recently, omnidirectional imagers such as catadioptric cameras, have sparked tremendous interest in image processing and computer vision. These sensors are particularly attractive due to their (nearly) full field of view. The visual information coming from a sequence of omnidirectional images can be used to perform a 3D reconstruction of a scene. This type of problem is usually referred to as *Structure from Motion* (SFM) [30] in the literature. Let us imagine a monocular observer that moves in a rigid unknown world; the SFM problem consists in estimating the 3D rigid self-motion parameters, i.e., rotation and direction of translation, and the structure of the scene, *e.g.*, represented as a depth map with respect to the observer position. Structure from motion has attracted considerable attention in the research community over the years with applications such as autonomous navigation, mixed reality, or 3D video.

As show in Chapter 3 we can reconstruct an accurate omnidirectional image with single focal point. In this chapter we then introduce a novel structure from motion framework for omnidirectional image sequences. We consider that the images can be mapped on the 2-sphere, which permits to unify various models of single effective viewpoint cameras. Then we propose a correspondence-free SFM algorithm that uses only differential motion between two consecutive frames of an image sequence through brightness derivatives. Since the estimation of a dense depth map is typically an ill-posed problem, we propose a novel variational framework that solves the SFM problem on the 2-sphere when the camera motion is unknown. Variational techniques are among the most successful approaches to solve under-determined inverse problems and efficient implementations have been proposed recently so that their use becomes appealing [87]. It is possible to extend very efficient variational approaches to SFM problems, while naturally handling the geometry of omnidirectional images. We embed a discrete image in a weighted graph whose connections are given by the topology of the manifold and the geodesic distances between pixels. We then cast the depth estimation problem as a TV-L1 optimization problem, and we solve the resulting variational problem with fast graph-based optimization techniques similar to [67, 34, 90]. To the best of our knowledge, this is the first time that graph-based variational techniques are applied to obtain a dense depth map from omnidirectional image pairs.

Then we address the problem of ego-motion estimation from the depth information. The

camera motion can be reliably estimated from an omnidirectional image pair if an approximate depth map is known. We propose to compute the parameters of the 3D camera motion with the help of a low-complexity least square estimation algorithm that determines the most likely motion between omnidirectional images using the depth information. Our formulation permits to avoid the explicit computation of the optical flow field and the use of feature matching algorithms. Finally, we combine both estimation procedures to solve the SFM problem in the generic situation where the camera motion is not known a priori. This is made possible by the use of a spherical camera model which makes easy to derive a linear set of motion equations that explicitly include camera rotation. The complete ego-motion parameters can then be efficiently estimated jointly with depth. The proposed iterative algorithm alternatively estimates depth and camera ego-motion in a multi-resolution framework, providing an efficient solution to the SFM problem in omnidirectional image sequences. While ideas of alternating minimization steps have also been proposed in [38, 5]. In these works, however, the authors use planar sensors and assume to have an initial rough estimate of the depth map. In addition, they use a simple locally constant depth model. In our experiments we show that this model is an oversimplification of the real world, which does not apply to scenes with a complex structure. Experimental results with synthetic spherical images and natural images from a catadioptric sensor confirm the validity of our approach for 3D reconstruction.

The rest of the chapter is structured as follows. We first provide a brief overview of the related work in Section 5.1. Then, we describe in Section 5.2 the framework used for motion and depth estimation.The variational depth estimation problem is presented in Section 5.3, and the ego-motion estimation is discussed in Section 5.4. Section 5.5 presents the joint depth and ego-motion estimation algorithm, while Section 5.6 presents experiments of on synthetic and natural omnidirectional image sequences.

5.1 Related work

The depth and ego-motion estimation problems have been quite widely studied in the last couple of decades and we describe here the most relevant papers that present correspondence-free techniques. Correspondence-free algorithms get rid of feature computation and matching steps that might prove to be complex and sensitive to transformations between images. Most of the literature in correspondence-free depth estimation is dedicated to stereo depth estimation [71]. In the stereo depth estimation problem cameras are usually separated by a large distance in order to efficiently capture the geometry of the scene. Usually the images are registered beforehand, such that images planes are aligned. A the disparity map is found between the two image views, and the disparity is eventually translated into a depth map. In our problem, we rather assume that the displacement between two consecutive frames in the sequence is small as it generally happens in image sequences. This permits to compute the differential motion between images and to build low-complexity depth estimation through image brightness derivatives. Then, most of the research about correspondence-free depth estimation has concentrated on perspective images; the depth estimation has also been studied in the case of omnidirectional images in [6], which stays as one of the rare works that carefully considers the specific geometry of the images in the depth estimation.

On the other hand, ego-motion estimation approaches usually proceed by first estimating the image displacement field, the so-called optical flow. The optical flow field can be related to the global motion parameters by a mapping that depends on the specific imaging surface of

the camera. The mapping typically defines the space of solutions for the motion parameters, and specific techniques can eventually be used to obtain an estimate of the ego-motion [16, 40, 46, 80]. Most techniques reveal sensitivity to noisy estimation of the optical flow. The optical flow estimation is a highly ill-posed inverse problem that needs some sort of regularization in order to obtain displacement fields that are physically meaningful; a common approach is to impose a smoothness constraint on the field [41, 9]. In order to avoid the computation of the optical flow, one can use the so-called "direct approach" where image derivatives are directly related to the motion parameters. Without any assumption on the scene, the search space of the ego-motion parameters is limited by the *depth positivity constraint*. For example, the works in [42, 75] estimate the motion parameters that result into the smallest amount of negative values in the depth map. Some algorithms originally proposed for planar cameras have later been adapted to cope with the geometrical distortion introduced by omnidirectional imaging systems. For example, an omnidirectional ego-motion algorithm has been presented by Gluckman in [35], where the optical flow field is estimated in the catadioptric image plane and then back-projected onto a spherical surface. Not many, though, have been trying to take advantage from the wider field of view of the omnidirectional devices: in spherical images the focus of expansion and the focus of contraction are both present, which imply that translation motion cannot be confused with rotational one. Makadia et al. [56] use this intuition to propose an elegant algorithm which makes use of filtering operator on the sphere. The solution is proven to be robust to noise and outlier, but the computational complexity remains high and the use of spherical harmonics makes the implementation of the algorithm dependent on a specific discretization of the sphere. In our work, we rather use a gradient-descent like approach on the manifold space of the motion parameters, which permits to avoid the computation of the optical flow, while keeping the computational complexity low and ease of implementation.

5.2 Framework Description

In this section, we introduce the framework and the notation. We derive the equations that relate global motion parameters and depth map to the brightness derivatives on the sphere. Finally, we show how we embed our spherical framework on a weighted graph structure and define differential operators in this representation.

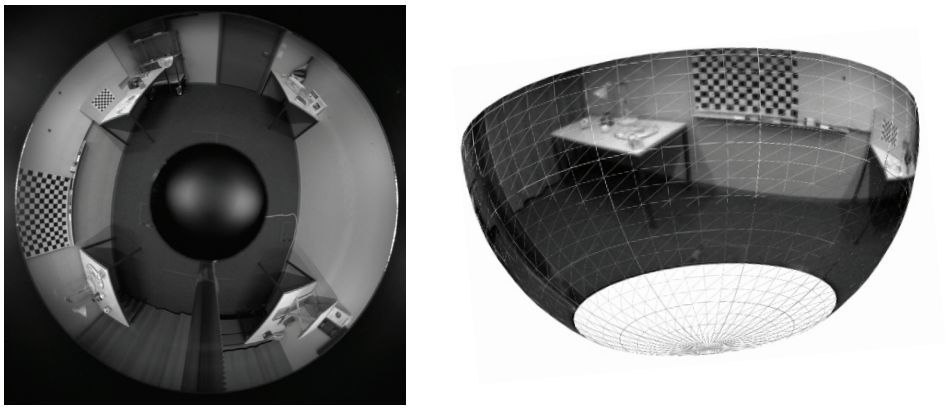


Figure 5.1: *Left: the original catadioptric image. Right: projection on the sphere*

We choose to work on the 2-sphere S^2, which is a natural spatial domain to perform processing of omnidirectional images as shown in [26] and references therein. For example, catadioptric camera systems with a single effective viewpoint permit a one-to-one mapping of the catadioptric plane onto a sphere via inverse stereographic projection [8]. The centre of that sphere is co-located with the focal point of the parabolic mirror and each direction represents a light ray incident to that point. We assume then that a pre-processing step transforms the original omnidirectional images into spherical ones as depicted in Fig. 5.1.

The starting point of our analysis is the *brightness consistency equation*, which assumes that pixel intensity values do not change during motion between successive frames. Let us denote $I(\mathrm{t}, \mathbf{y})$ an image sequence, where t is time and $\mathbf{y} = (y^1, y^2, y^3)$ describes a spatial position in 3-dimensional space. If we consider only two consecutive frames in the image sequence, we can drop the time variable t an use I_0 and I_1 to refer to the first and the second frame respectively. The brightness consistency assumption then reads: $I_0(\mathbf{y}) - I_1(\mathbf{y}+\mathbf{u}) = 0$ where $\mathbf{u}$ is the motion field between the frames. We can linearize the brightness consistency constraint around $\mathbf{y} + \mathbf{u}_0$ as:

$$I_1(\mathbf{y}+\mathbf{u}_0) + (\nabla I_1(\mathbf{y}+\mathbf{u}_0))^T(\mathbf{u}-\mathbf{u}_0) - I_0(\mathbf{y}) = 0, \tag{5.1}$$

with an obvious abuse of notation for the equality. This equation relates the motion field $\mathbf{u}$ to the (spatial and temporal) image derivatives. It is probably worth stressing that, for this simple linear model to hold, we assume that the displacement $\mathbf{u} - \mathbf{u}_0$ between the two scene views I_0 and I_1 is sufficiently small.

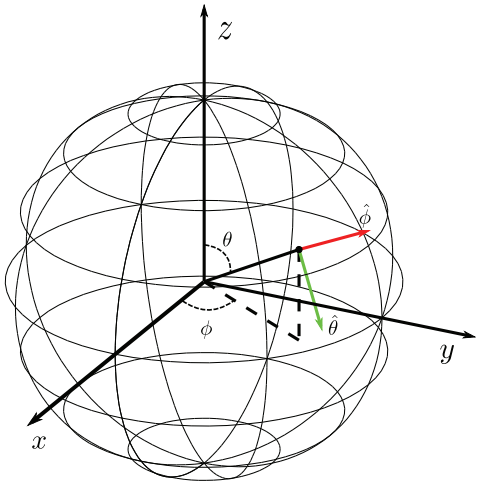


Figure 5.2: *The representation and coordinate on the 2-sphere S^2.*

When data live on S^2 we can express the gradient operator ∇ from Eq. (5.1) in spherical coordinates as :

$$\nabla I(\phi,\theta) = \frac{1}{\sin\theta}\partial_\phi I(\phi,\theta)\hat{\phi} + \partial_\theta I(\phi,\theta)\hat{\theta}, \tag{5.2}$$

where $\theta \in [0,\pi]$ is the colatitude angle, $\phi \in [0, 2\pi[$ is the azimuthal angle and $\hat{\phi}, \hat{\theta}$ are the unit vectors on the tangent plane corresponding to infinitesimal displacements in ϕ and θ respectively (see Fig. 5.2). Note also that by construction the optical flow field $\mathbf{u}$ is defined on the tangent bundle $TS = \bigcup_{\omega\in S^2} T_\omega S^2$, i.e. $\mathbf{u} : S^2 \subset \mathbb{R}^3 \to TS$.

5.2.1 Global motion and optical flow

Under the assumption that the motion is slow between frames, we have derived above a linear relationship between the motion field $\mathbf{u}$ on the spherical retina and the brightness derivatives. If the camera undergoes rigid translation $\mathbf{t}$ and rotation around the axis $\mathbf{\Omega}$, then we can derive a geometrical constraint between $\mathbf{u}$ and the parameters of the 3D motion of the camera. Let

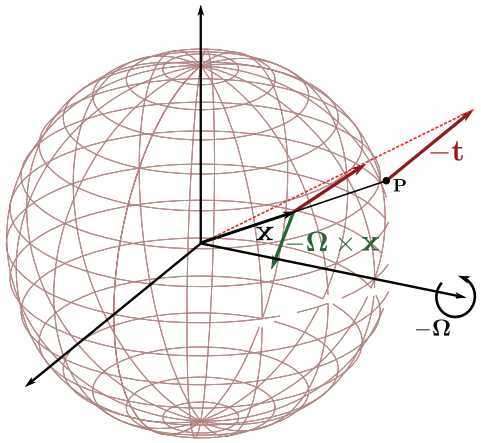


Figure 5.3: *The sphere and the motion parameters*

us consider a point $\mathbf{p}$ in the scene, with respect to a coordinate system fixed at the center of the camera. We can express $\mathbf{p}$ as: $\mathbf{p} = D(\mathbf{x})\mathbf{x}$ where $\mathbf{x}$ is the unit vector giving the direction to $\mathbf{p}$ and $\mathrm{d}(\mathbf{x})$ is the distance of the scene point from the center of the camera. During camera motion, as illustrated in Fig. 5.3, the scene point moves with respect to the camera by the quantity :

$$\delta\mathbf{p} = -\mathbf{t} - \mathbf{\Omega} \times \mathbf{x}. \tag{5.3}$$

We can now build the geometric relationship that relates the motion field $\mathbf{u}$ to the global motion parameters $\mathbf{t}$ and $\mathbf{\Omega}$. It reads

$$\mathbf{u}(\mathbf{x}) = -\frac{\mathbf{t}}{\mathrm{d}(\mathbf{x})} - \mathbf{\Omega} \times \mathbf{x} = -\mathrm{z}(\mathbf{x})\mathbf{t} - \mathbf{\Omega} \times \mathbf{x}, \tag{5.4}$$

where the function $\mathrm{z}(\mathbf{x})$ is defined as the multiplicative inverse of the distance function $\mathrm{d}(\mathbf{x})$. In the following we will refer to z as the *depth map*. In Eq. (5.4) we find all the unknowns of our SFM problem: the depth map $\mathrm{z}(\mathbf{x})$ describing the structure of the scene and the 3D motion parameters $\mathbf{t}$ and $\mathbf{\Omega}$. Due to the multiplication between $\mathrm{z}(\mathbf{x})$ and $\mathbf{t}$, both quantities can only be estimated up to a scale factor. So in the following we will consider that $\mathbf{t}$ has unitary norm.

We can finally combine Eq. (5.1) and Eq. (5.4) in a single equation:

$$I_1(\mathbf{y} + \mathbf{u}_0) + (\nabla I_1(\mathbf{y} + \mathbf{u}_0))^T(-\mathrm{z}(\mathbf{x})\mathbf{t} - \mathbf{\Omega} \times \mathbf{x} - \mathbf{u}_0) - I_0(\mathbf{y}) = 0. \tag{5.5}$$

Eq. (5.5) relates image derivatives directly to 3D motion parameters. The equation is not linear in the unknowns and it defines an under-constrained system (i.e., more unknown than equations). We will use this equation as constraint in the optimization problem proposed in the next section.

5.2.2 Discrete differential operators on the 2-Sphere

We have developed our previous equations in the continuous spatial domain, but we our images are discrete in practice. Although the 2-sphere is a simple manifold with constant curvature and a simple topology, a special attention has to be paid to the definition of the differential operators that are used in the variational framework.

We assume that the omnidirectional images recorded by the sensor are interpolated onto a spherical equiangular grid : $\{\theta_m = m\pi/M, \phi_n = n2\pi/N\}$, with $M \cdot N$ the total number of samples. This operation can be performed, for example, by mapping the omnidirectional image on the sphere and then using bilinear interpolation to extract the values at the given positions (θ_m, ϕ_n). In spherical coordinates, a simple discretization of the gradient obtained from finite differences reads:

$$\nabla_\theta f(\theta_{i,j}, \phi_{i,j}) = \frac{f(\theta_{i+1,j}, \phi_{i,j}) - f(\theta_i, \phi_j)}{\Delta\theta},$$
$$\nabla_\phi f(\theta_{i,j}, \phi_{i,j}) = \frac{1}{\sin\theta_{i,j}} \left(\frac{f(\theta_{i,j}, \phi_{i,j+1}) - f(\theta_{i,j}, \phi_{i,j})}{\Delta\phi} \right).$$

The discrete divergence, by analogy with the continuous settings, is defined by $div = -\nabla^*$ where ∇^* is the adjoint of ∇. It is then easy to verify that the divergence is given by:

$$div\mathbf{p}(\theta_{i,j}, \phi_{i,j}) = \frac{p^\phi(\theta_{i,j}, \phi_{i,j}) - p^\phi(\theta_{i,j}, \phi_{i,j-1})}{\sin\theta_{i,j}\Delta\phi} + \frac{\sin\theta_{i,j} p^\theta(\theta_{i,j}, \phi_{i,j}) - \sin\theta_{i,j} p^\theta(\theta_{i-1,j}, \phi_{i,j})}{\sin\theta_{i,j}\Delta\theta}. \tag{5.6}$$

Both Eq. (5.6) and Eq. (5.6) contain a $(\sin\theta)^{-1}$ term that induces very high values around the poles (i.e., for $\theta \simeq 0$ and $\theta \simeq \pi$) and can cause numerical instability. We therefore propose to define discrete differential operators on weighted graphs (i.e., discrete manifold) as a general way to deal with geometry in a coordinate-free fashion.

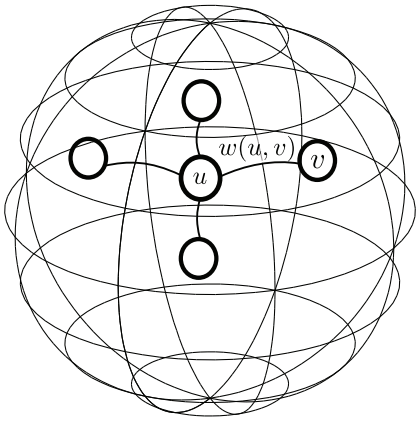


Figure 5.4: *Embedding of discrete sphere on a graph structure. The pixels u and v in the spherical image represent vertices of the graph, and the edge weight $w(u, v)$ typically captures the geodesic distance between the vertices*

We represent our discretized (spherical) imaging surface as a weighted graph, where the vertices represent image pixels and edges define connections between pixels (i.e., the topology of the surface) as represented in Figure 5.4. We refer to Section 3.1 and Section 3.2 for a

deeper definition of graphs and differential operators on graphs. We rewrite for commodity the definition of gradient and divergence on graphs:

$$(\nabla^w f)(u,v) = \sqrt{w(u,v)}f(u) - \sqrt{w(u,v)}f(v) \tag{5.7}$$

and

$$(div^w F)(u) = \sum_{u\sim v} \sqrt{w(u,v)}\,(F(v,u) - F(u,v))\,, \tag{5.8}$$

Even though both discretization methods are applicable to spherical images, the main advantages of the graph-based representation rely on the definition of differential operators directly in the discrete domain. They reveal a much more stable behavior than their counterparts from Eq. (5.6) and Eq. (5.6). On top of that, the framework provides flexibility in the choice of the discrete grid points, whose density can vary locally on the sphere.

5.3 Variational Depth Estimation

Equipped with the above formalism, we now propose a new variational framework to estimate a depth map from two consecutive frames of an omnidirectional image sequence. We assume at this point that the parameters $\mathbf{t}, \mathbf{\Omega}$ that describe the 3D motion of the camera are known. In addition, we might have an estimate of the optical flow field $\mathbf{u}_0$.

Let us consider again Eq. (5.5) that relates image derivatives to motion parameters. Since the image gradient ∇I_1 is usually sparse, Eq. (5.5) does not provide enough information to recover a dense depth map. Hence, we formulate the depth estimation problem as a regularized inverse problem using the L^1 norm to penalize deviation from the brightness constraint and the TV-norm to obtain a regular depth map possibly with sharp transitions.

We build the following error functional:

$$J(\mathrm{z}) = \int_\Omega \psi(\nabla \mathrm{z})\,\mathsf{d}\Omega + \lambda \int_\Omega |\rho(I_0, I_1, \mathrm{z})|\,\mathsf{d}\Omega, \tag{5.9}$$

and we look for the depth map z that minimizes it. In Eq. (5.9) the function ρ is the data fidelity term that describes the residual image error:

$$\rho(I_0, I_1, \mathrm{z}) = I_1(\mathbf{y} + \mathbf{u}_0) + (\nabla I_1(\mathbf{y} + \mathbf{u}_0))^T(-\mathrm{z}(\mathbf{x})\mathbf{t} - \mathbf{\Omega} \times \mathbf{x} - \mathbf{u}_0) - I_0(\mathbf{y}), \tag{5.10}$$

where we use our assumption that $\mathbf{t}, \mathbf{\Omega}$ and $\mathbf{u}_0$ are known. The regularization function ψ is given by:

$$\psi(\nabla \mathrm{z}) = |\nabla \mathrm{z}(\mathbf{x})|. \tag{5.11}$$

With such a choice of the functional J we define a TV-L1 inverse problem. Several advantages come from this choice. First the TV-L1 model is very efficient in removing noise and robust against illumination changes: it inherits these properties from the Rudin-Osher-Fatemi (ROF) model [70] and the L^1 norm fidelity term ensures robustness to outliers and also non-erosion of edges [63]. Furthermore the TV regularization is a very efficient prior to preserve sharp edges. The total variation model then suits the geometrical features of a real scene structure where the depth map is typically piecewise linear with sharp transitions on objects boundaries.

The functional in Eq. (5.9) is written in terms of continuous variables, while in practice we work with discrete images. Inspired by the continuous formulation, we now propose to

solve a similar, though purely discrete, problem. As in Chapter 3, we define the local isotropic variation of z at vertex (pixel) v by :

$$\|\nabla_v^w \mathrm{z}\| = \sqrt{\sum_{u\sim v} \left[\left(\nabla^w \mathrm{z}\right)(u,v)\right]^2}. \tag{5.12}$$

The discrete optimization problem can then be written as :

$$J(\mathrm{z}) = \sum_v \|\nabla_v^w \mathrm{z}\| + \lambda \sum_v |\rho(I_0, I_1, \mathrm{z})|. \tag{5.13}$$

The definition of ρ is the same as in Eq. (5.10), where we substitute the naive finite difference approximation of the gradient given in Eq. (5.6). Note that the discrete problem now uses two different discretizations for differential operators on S^2. The reason for this choice will be made clear below.

We now discuss the solution of the depth estimation problem in Eq. (5.13). Even though the resulting functional J is convex, it poses severe computational difficulties. Following [7], we propose a convex relaxation into a sum of two simpler sub-problems:

$$J(\mathrm{z}) = \sum_v \|\nabla_v^w \mathrm{z}\| + \frac{1}{2\theta}\sum_u (v(u) - \mathrm{z}(u))^2 + \lambda \sum_u |\rho(I_0, I_1, v)|, \tag{5.14}$$

where v is an auxiliary variable that should be as close as possible to z. If θ is small then v converges to z and the functional defined in Eq. (5.14) converges to the one defined in Eq. (5.13) as shown in [7]. The minimization must now be performed with respect to both the variables v, z. Since the functional is convex the solution can be then obtained by an iterative two-step procedure:

1. For z fixed, solve:
$$\min_v \left\{ \frac{1}{2\theta}\sum_u (v(u) - \mathrm{z}(u))^2 + \lambda |\rho(v(u))| \right\}. \tag{5.15}$$

2. For v fixed, solve:
$$\min_{\mathrm{z}} \left\{ \sum_u \|\nabla_u^w \mathrm{z}\| + \frac{1}{2\theta}\sum_u (v(u) - \mathrm{z}(u))^2 \right\}. \tag{5.16}$$

The minimization in the first step is straightforward : the problem is completely decoupled in all coordinates and the solution can be found in a point-wise manner using this thresholding scheme:

$$v = \mathrm{z} + \begin{cases} \theta\lambda\nabla I_1^T \mathbf{t} & \text{if } \rho(\mathrm{z}) < -\theta\lambda(\nabla I_1^T \mathbf{t})^2 \\ -\theta\lambda\nabla I_1^T \mathbf{t} & \text{if } \rho(\mathrm{z}) > \theta\lambda(\nabla I_1^T \mathbf{t})^2 \\ -\dfrac{\rho(\mathrm{z})}{\nabla I_1^T \mathbf{t}} & \text{if } |\rho(\mathrm{z})| \leqslant \theta\lambda(\nabla I_1^T \mathbf{t})^2. \end{cases} \tag{5.17}$$

The previous result can be easily obtained by writing the Euler-Lagrange condition for Eq. (5.15)

$$\frac{1}{\theta}(\mathrm{z} - v) + \lambda\nabla I_1^T \mathbf{t}\frac{\rho(v)}{|\rho(v)|} = 0, \tag{5.18}$$

and then analyzing the three different cases: $\rho(\mathrm{z}) > 0$, $\rho(v) < 0$ and $\rho(v) = 0$. Using the relationship $\rho(v) = \rho(\mathrm{z}) + \nabla I_1^T \mathbf{t}(v - \mathrm{z})$ we have:

- $\rho > 0$:
  $(\mathrm{z} - v) = \theta\lambda\nabla I_1^T\mathbf{t} \Rightarrow \rho(\mathrm{z}) > \nabla I_1^T\mathbf{t}(\mathrm{z} - v) = \theta\lambda(\nabla I_1^T)^2$
- $\rho < 0$:
  $(\mathrm{z} - v) = -\theta\lambda\nabla I_1^T\mathbf{t} \Rightarrow \rho(\mathrm{z}) < -\nabla I_1^T\mathbf{t}(\mathrm{z} - v) = \theta\lambda(\nabla I_1^T)^2$
- $\rho = 0$:
  $\rho(\mathrm{z}) = -\nabla I_1^T\mathbf{t}(v - \mathrm{z})$

Notice that this computation relies on evaluating the scalar product $\nabla I_1^T\mathbf{t}$, which can not be evaluated if we use a graph-based gradient, since the vector $\mathbf{t}$ is unconstrained (in particular it does not correspond necessarily to an edge of the graph). However, this part of the algorithm is not iterative and the gradient can be pre-computed, therefore avoiding severe numerical instabilities as we move closer to the poles.

The minimization in Eq. (5.16) corresponds to the total variation image denoising model, the same described in Eq. (3.30). The solution of the Eq. (5.16) when the differential operators are defined on the graph is already given in Section 3.3.2. If the differential operators are the one defined in Eq. (5.6) and Eq. (5.6) Chambolle's iterations read explicitely

$$\begin{aligned} &\mathrm{z} = v - \theta div\mathbf{p}, \\ &\mathbf{p}^{n+1} = \frac{\mathbf{p}^n + \tau\nabla(div\mathbf{p}^n - v/\theta)}{1 + \tau|\nabla(div\mathbf{p}^n - v/\theta)|}. \end{aligned} \tag{5.19}$$

where $\mathbf{p}$ represent a vector field on the sphere, *i.e.*, $\mathbf{p} \in TS$, where TS the tangent bundle already introduced in Section 5.2.

Finally, it should be noted that the algorithm is formally the same whatever discretization is chosen, i.e., the discrete operator can be given either by Eq. (5.8) or Eq. (5.6) . Experimental results however show that the graph-based operators unsurprisingly lead to the best performance.

5.4 Least Square Ego-Motion Estimation

We discuss in this section a direct approach to the estimation of the ego-motion parameters $\mathbf{t}, \mathbf{\Omega}$ from the depth map z. We propose a formulation based on least mean squares algorithm.

When we have an estimate of $\mathrm{z}(\mathbf{x})$ in Eq. (5.5), we have a set of linear constraints in the motion parameters $\mathbf{t}$,$\mathbf{\Omega}$ that can be written as :

$$\mathrm{z}(\nabla I_1)^T\mathbf{t} + (\mathbf{x} \times (\nabla I_1))^T\mathbf{\Omega} = I_0 - I_1. \tag{5.20}$$

For each direction in space $\mathbf{x}$ we can rewrite Eq. (5.20) in a matrix form:

$$A(\mathbf{x})\mathbf{b} = C(\mathbf{x}), \tag{5.21}$$

where $A(\mathbf{x}) = [(\mathrm{z}(\mathbf{x})\nabla I_1(\mathbf{x}))^T \ (\mathbf{x} \times \nabla I_1(\mathbf{x}))^T]$, and $C(\mathbf{x}) = I_0(\mathbf{x}) - I_1(\mathbf{x})$ are known matrices, while $\mathbf{b} = [\mathbf{t}; \mathbf{\Omega}]$ is the variable containing the unknown motion parameters.

We formulate the ego-motion estimation problem as follows:

$$\mathbf{b}^* = \underset{\mathbf{b}}{\operatorname{argmin}} \sum_{\mathbf{x}} (A(\mathbf{x})\mathbf{b} - C(\mathbf{x}))^2. \tag{5.22}$$

The solution to this linear least square problem is simply:

$$\mathbf{b} = \frac{\sum_{\mathbf{x}} A^T C}{\sum_{\mathbf{x}} A A^T}. \tag{5.23}$$

There are several aspects that are important for the existence and the unicity of the solution of the ego-motion estimation problem. First, the images must present enough structure. In other words, the image gradient ∇I_1 should carry enough information on the structure on the scene. In particular, since the gradient only gives information on motion that is perpendicular to image edges, the gradient itself will not help recovering motion parameters if the projection of the motion parameters on the spherical retina is everywhere parallel to the gradient direction. This situation is however highly unlikely for a real scene and a wide field of view camera.

Then, there is a possibility of confusion for certain combinations of the motion parameters. In Eq. (5.20) we compute the scalar product between the image gradient and the vector $\mathrm{z}(\mathbf{x})\mathbf{t} + \boldsymbol{\Omega} \times \mathbf{x}$, i.e., the spherical projection of 3D motion. For a small field of view, $\mathbf{x}$ does not change much and the two terms $\mathrm{z}(\mathbf{x})\mathbf{t}$ and $\boldsymbol{\Omega} \times \mathbf{x}$ could be parallel, meaning that we cannot recover them univocally. This happens for example with a rotation around vertical axis and a displacement in the perpendicular direction to both viewing direction and rotation axis. Such a confusion however disappear on a spherical retina [60].

5.5 Joint Ego-Motion and Depth Map Estimation

We have described in the previous sections the separate estimation of a dense depth map and the 3D motion parameters. The purpose of this section is to combine both estimation algorithms in a dyadic multi-resolution framework.

We embed the minimization process into a coarse-to-fine approach in order to avoid local minima during the optimization and to speed up the convergence of the algorithm. We employ a spherical gaussian pyramid decomposition as described in [81], with a scale factor of 2 between adjacent levels in order to perform the multi-resolution decomposition.

Then, we solve the depth and ego-motion estimation problems by alternating minimization steps. For each resolution level l, we compute a solution to Eq. ((5.5)) by performing two minimization steps:

1. We use the depth map estimate at the previous level $\bar{\mathrm{z}}^{l+1}(\mathbf{x})$ to initialize the depth map $\mathrm{z}_0^l(\mathbf{x})$ at the current level l. Using the least square minimization from Eq.(5.23) we can refine the estimation of the motion parameters $\mathbf{t}^l, \boldsymbol{\Omega}^l$ at level l.

2. Using the estimated motion parameters $\mathbf{t}^l, \boldsymbol{\Omega}^l$ we can find an estimate of the depth map at current level $\mathrm{z}^l(\mathbf{x})$ by solving Eq. 5.13 using the variational framework described in Section 5.3.

Since we perform a coarse-to-fine approach we only need to initialize the algorithm at the coarsest level. Let us assume that we use L levels. At the coarsest level L we make the hypothesis that a constant-depth model of the scene is sufficient to explain the apparent pixel motion between the low resolution images I_0^L and I_1^L, so we set $\mathrm{z}_0^L = K$, where K is a positive

constant different from zero which ideally should be set according to the farthest point in the scene. In practice the choice of K does not effect the performance of the algorithm as long as it is small enough. At the coarsest level, the approximation that we introduce by flattening the depth map z is well posed since all image edges are smoothed out at low resolution. An alternative for the initialization is to use an initial rough estimate of the depth map, given from external measurement. At each level l we can also obtain an estimate of the optical flow $\mathbf{u}_0^l$ as $\mathbf{u}_0^l = -\mathrm{z}_0^l(\mathbf{x})\mathbf{t}^{l+1} - \mathbf{\Omega}^{l+1} \times \mathbf{x}$, and use it to warp image I_1^l, i.e., to estimate $I_1^l(\mathbf{x} + \mathbf{u}_0^l)$. The joint depth and ego-motion estimation algorithm is summarized in Algorithm 5.1. We

1. At the coarsest level L initialize: $\mathrm{z}_0^L = K$ with $K > 0$
2. For each level $l \in [L, L-1, \ldots, 2, 1]$:
   (a) Initialize z with the solution at previous level
   $$\mathrm{z}_0^l = upsample(\mathrm{z}^{l+1}).$$
   (b) Estimate optical flow $\mathbf{u}_0^l$ as:
   $$\mathbf{u}_0^l = -\mathrm{z}_0^l(\mathbf{x})\mathbf{t}^{l+1} - \mathbf{\Omega}^{l+1} \times \mathbf{x}$$
   and use it to calculate $I_1^l(\mathbf{x} + \mathbf{u}_0^l)$.
   (c) Estimate $\mathbf{t}^l$ and $\mathbf{\Omega}^l$ using Eq.(5.23):
   $$\mathbf{b} = \frac{\sum_{\mathbf{x}} A^T C}{\sum_{\mathbf{x}} A A^T}.$$
   (d) Estimate z^l using the depth estimation algorithm described in Section 5.3 with the current estimates $\mathbf{t}^l$ and $\mathbf{\Omega}^l$.

Algorithm 5.1: Computation of $\mathrm{z}, \mathbf{t}, \mathbf{\Omega}$

conclude with some considerations regarding the complexity of the algorithm. We firstly observe that the complexity is dominated be the depth map estimation, while it does not depend on the choice of the differential operators, as long as the number of connections in the graph is sparse. Furthermore, since each operation in Eq. (5.17) and (5.19) can be performed pixel-wise, the algorithm can be efficiently implemented on graphics processing units in a similar way as described in [87]. The ego-motion estimation algorithm has low complexity since every iteration in Eq. (5.23) runs in linear time $\mathrm{O}(n)$, where n is the total number of pixels, and quickly converges to the solution

5.6 Experimental results

We analyze in this section the performance of the proposed algorithms for two sets of omnidirectional images, namely a synthetic and a natural sequence. For both sets the images are defined on an equiangular grid, so they are easily representable on a plane, as shown for

example in Fig. 5.5. In this image plane, the vertical and horizontal coordinates correspond respectively to the θ and ϕ angles. The images are represented such that the top of the image corresponds to the north pole and the bottom to the south pole.

5.6.1 Synthetic omnidirectional images

For the syntethic set we use the *room scene* shown in Figure 5.5 Since we only use 2 frames in our optimization scheme, in our experiments with the synthetic images the first frame I_0 is always the same one shown in Figure 5.5 together with the associated depth map, that we use as ground truth for the numerical evaluation of the performance of our algorithm. We generate the other frames by translating and rotating the spherical camera. The camera translation has always the same module of 0.1 units, while the dimension of the room is 24 units by 23 units. We first study the influence of the discretization scheme in the variational

Figure 5.5: *The synthetic spherical image. Middle: geographical projection. Right: depth map ground truth*

depth estimation algorithm. As discussed in Section 5.3 the TV denoising part of the depth estimation algorithm is extremely sensitive to the choice of the discrete differential operators. We show in Fig. 5.6 that the use of the differential operators from Eq. (5.6) and (5.6) lead to noisy results around the poles. We call the resulting algorithm as *TVL1-naive*. We compare the results of this implementation to those obtained by choosing the graph-based definition of the differential operators from Eq. (5.7) and (5.8). The proposed algorithm, that we call *TVL1-GrH*, clearly leads to improved performance, especially around the poles where it is much more robust than *TVL1-naive*. In Fig. 5.6 we can observe a black area in the middle of both estimated depth maps, which is not present in the ground truth image. This structure is simply due to an occlusion generated by the reflection of the window, where the brightness consistency does not hold. Then, we compare in Fig. 5.7 the results of the variational depth map estimation algorithm for four different camera motions, namely a pure translation or different combinations of rotation and translation. We compare our results to a local-constant-depth model algorithm (i.e., *LK*) similar to the one described in [54] and [38]. This approach assumes that the depth is constant for a given image patch and tries to find a least square depth estimate using the brightness consistency equation. We can observe that the TV-L1 model is much more efficient in preserving edges, so that it becomes possible to distinguish the objects in the 3D scene. The *LK* algorithm has a tendency to smooth the depth information so that objects are hardly visible.

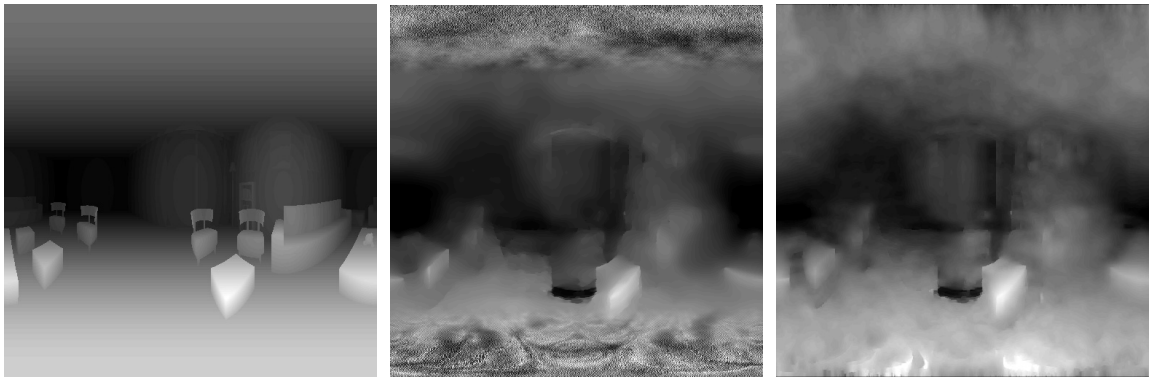


Figure 5.6: *Depth map estimation with different discrete differential operators.* Left*: ground truth.* Center*: TVL1-naive.* Right*: TVL1-GrH*

These results are confirmed in Table 5.1 in terms of mean square error of the depth map reconstruction for several synthetic sequences. It can be seen that the local-costant-depth algorithm *LK* is outperformed by the variational depth estimation algorithm with graph-based operators (*TVL1-GrH*) It is also interesting to observe the influence of the choice of the discrete differential operators. As it has been observed earlier, the discretization from Eq. (5.6) and (5.6) (*TVL1-naive*) clearly leads to the worst results, while the graph-based operators perform best.

Table 5.1: *Mean Square Error (MSE) between the estimated depth map and the ground truth*

	Seq1	Seq2	Seq3	Seq4	Seq5
LK	0.00117	0.00268	0.00158	0.00611	0.00216
TVL1-naive	0.00447	0.10319	0.10234	0.10824	0.10369
TVL1-GrH	**0.00103**	**0.00169**	**0.00167**	**0.00395**	**0.0017**

Finally, we analyze in Table 5.2 the performance of the ego-motion estimation algorithm proposed in Section 5.4. We use the same synthetic sequences as before, and the depth estimation results are used in the least mean square optimization problem for motion parameter estimation. We compare the ego-motion estimation to the true motion parameters, given in terms of translation ($\mathbf{t}$) and rotation ($\mathbf{\Omega}$) parameters. We can see that the ego-motion estimation is quite efficient for all the sequences even if the estimation algorithm is quite simple. The relative error is usually smaller than one percent.

Table 5.2: *Results for the least square motion parameters estimation*

	Seq1	Seq2	Seq3	Seq4	Seq5
true-$\mathbf{t}$	[-0.1;0;0]	[-0.1;0;0]	[-0.1;0;0]	[0;-0.1;0]	[-0.07;-0.07;0]
$\mathbf{t}$	[-0.099;0.001;-0.004]	[-0.099;0;-0.004]	[-0.099;0.002;-0.005]	[0.;-0.099;-0.006]	[-0.069;-0.07;-0.009]
true-$\mathbf{\Omega}$	[0;0;0]	[0;0;0.0175]	[0.0175;0;0]	[0;0;0.0175]	[0.0175;0;0]
$\mathbf{\Omega}$	[0;-0.001;0]	[0;-0.002;0.016]	[0.0177;-0.0025;0]	[0;0;0.0182]	[0.0181;0;0]

5.6.2 Natural omnidirectional images

These images have been captured by a catadrioptric system positioned in the middle of a room. We then move the camera on the ground plane and rotate it along the vertical

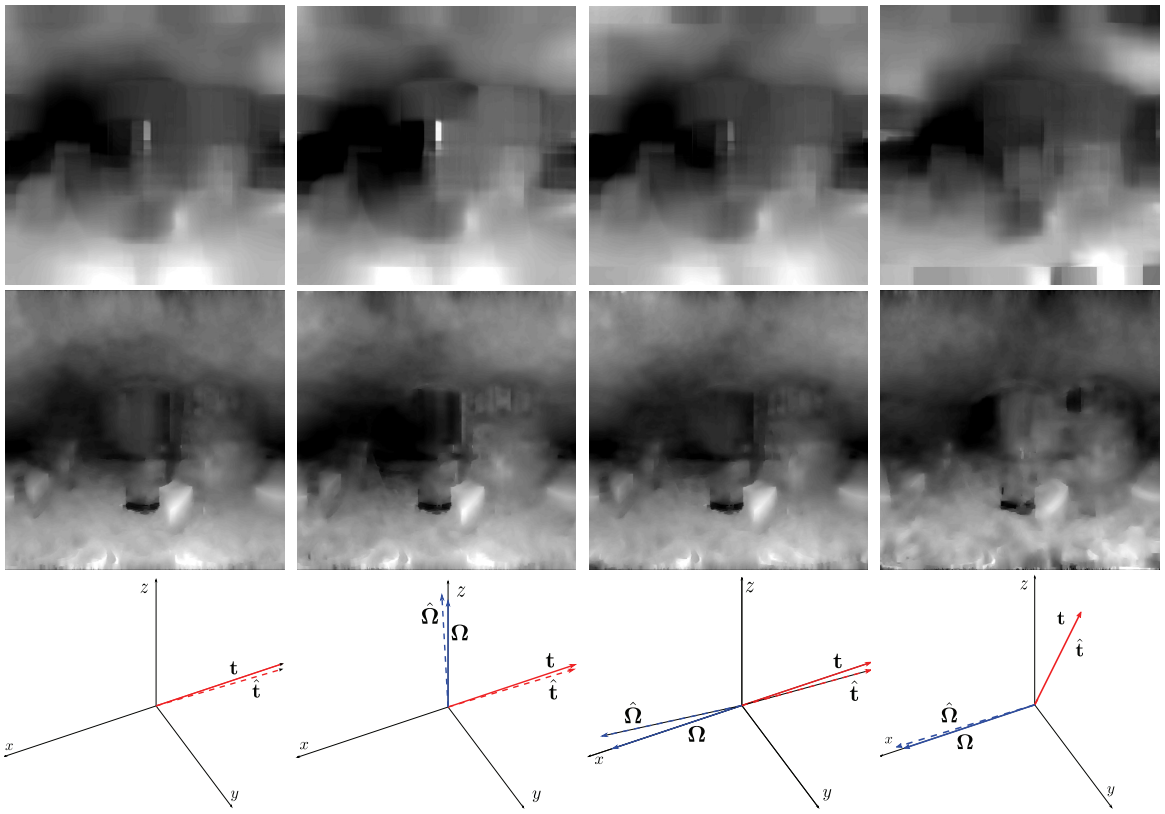


Figure 5.7: *LK (top) vs TVL1-naive (middle) for four different camera motions. On the bottom we show also* $\mathbf{t}$ *in red and* $\mathbf{\Omega}$ *in blue; the estimated motion vectors are represented with a dashed line*

axis. The resulting images are shown in Fig. 5.8, where we also illustrate the result of the projection of the captured images on the sphere. We have also measured the depth map in this environment with help of a laser scanner, and we use these measures for visual evaluation of the depth map estimation algorithm.

We first analyze the performance of our depth estimation algorithm for natural spherical images, and we compare the estimated depth map to the depth information measured by the laser scanner. We show in Fig. 5.9 that the estimated depth map is quite accurate when compared to the LK algorithm, since the proposed algorithm is able to detect and delineate clearly the objects in the scene. It confirms the efficiency of the variational framework proposed in this paper.

Finally, we show that our depth estimation provides accurate information about the scene content by using this information for image reconstruction. We use one of the images of the natural image sequence as a reference image, and we predict the next image using the depth information. We compute the difference between the second image and respectively the reference image, and the approximation of the second image by motion compensation. We can observe in Fig. 5.10 that the estimated depth map leads to efficient image reconstruction, as the motion compensated image provides a much better approximation of the second image than the reference image. The depth information permits to reduce drastically the energy of the prediction error, especially around the main edges in the sequence. It outlines the

potential of our depth estimation algorithm for efficient image or 3D reconstruction.

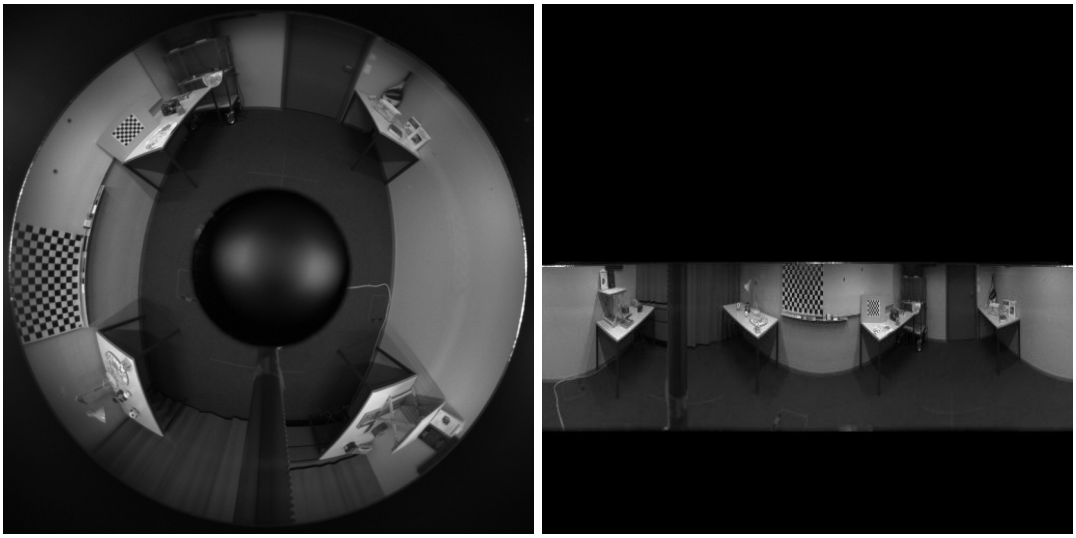


Figure 5.8: *Natural omnidirectional images from a room.* Top*: Catadioptric image sequence.* Bottom*: Projection of the catadioptric images on a spherical surface*

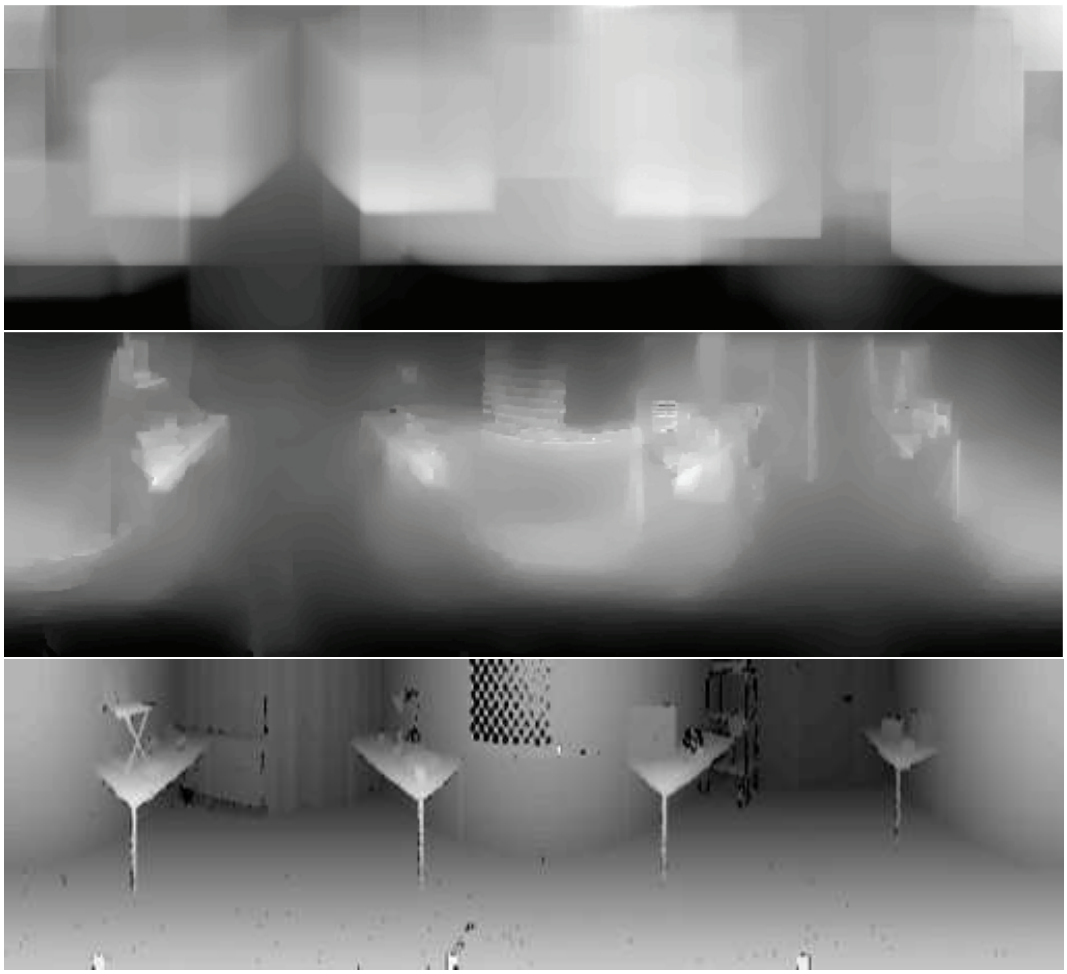


Figure 5.9: *Visual comparison of the estimated depth map on natural images. (*Top*): LK. (*Middle*): the proposed TVL1-GrH.* Bottom*: depth map from a laser scanner.*

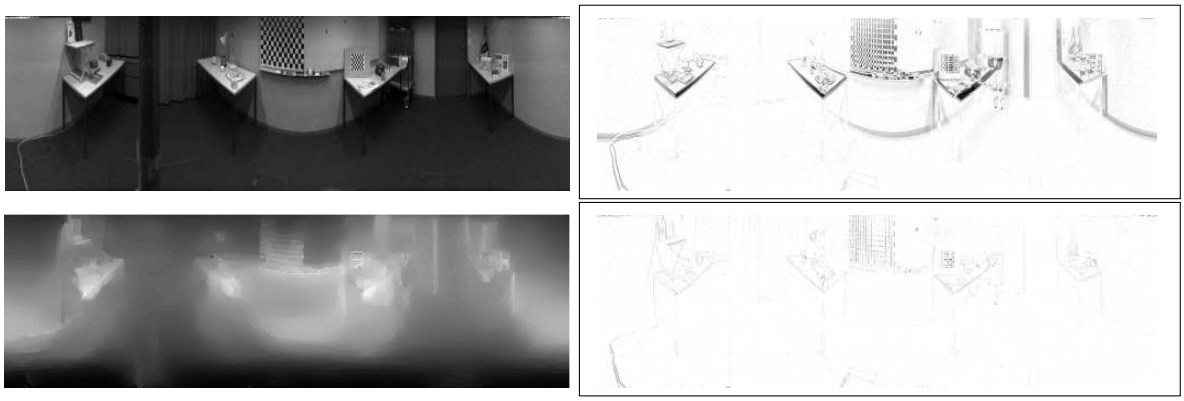


Figure 5.10: *Analysis of the estimated depth map.* Top - left*: First image of the catadrioptic sequence.* Top - right*: Image difference* $I_0 - I_1$ *.* Bottom - left*: Estimated depth map.* Bottom - right*: Image difference after motion compensation.*

Chapter 6

Application: The Panoptic Camera

The very efficient visual system of flying insects has provided early inspiration for the presented hardware vision system. The common fly has two faceted eyes which provide an omnidirectional, and which ease some computer vision tasks like ego-motion and depth estimation. Each one of the fly faceted eye is an omnidirectional vision system composed of several thousands of rudimentary image sensors called ommatidias [88].

Mimicking the faceted eye's concept, in this last Chapter of the dissertation we present a physical implementation of the Spherical Light Field Camera Model presented in Section 2.3. We the omnidirectional camera by layering miniature CMOS image sensors over the surface of an hemispherical structure. We name it *Panoptic*, after the hundred-eyes giant populating many stories in the greek mythology [1]. The camera has two distinguishable features. First it is an omnidirectional camera, *i.e.*, it is able to record light information from any direction around its center. Second, each CMOS camera has a distinct focal plane; hence the whole system implements a 4D light field sampler.

Early attempts in fabricating omnidirectional vision (with a single focal point) are based on regular sensors capturing the image formed on a parabolic mirror [33]. Conversely, non-omnidirectional cameras still recording plenoptic (multi focal) information have been developed for almost 10 years using a lenticular array placed above a sensor plane [2]. An alternate solution has been proposed in [23], where a number of commercial cameras are placed in orbital plane, enabling the post processing reconstruction of a panoramic image used as an omnidirectional vision turret for robotic applications. The FlyCam [32] is one of the first attempt to construct a panorama imaging device from off-the-shelf hardware components. Recently, two attempts in miniaturizing the omnidirectional vision system have been made, specially using microfabrication technologies into mimicking the insect compound eye [48], [45].

Solutions which have been proposed so far to realize omnidirectional vision suffer from various flaws which harm their practicality and effectiveness. Most of the proposed systems involve bulky or heterogeneous hardware, in the form of computers and/or laser-based distance measurement systems prohibiting actual portability, three-dimensional mirrors which are very delicate to manipulate, may cause local image distortion due to a complex and difficult to guarantee fabrication process as well as misalignment with the imager. Alternatively, the

[1]The Panoptic camera is a joint work with the Microelectronic System Laboratory at EPFL, headed by prof. Y. Leblebici. The project has been supported by the Swiss NSF under grant number 200021-125651.

attempts to realize micrometer size omnidirectional vision mainly focus on the vision system, where the data communication and image processing is not considered.

Building on the theory developed in the previous Chapters we describe here a solution solving these issues is proposed by applying a system-level consideration of omnidirectional vision acquisition and subsequent image processing.

The major design effort that is applied in the development of the Panoptic camera relates to its inspiration from biology, and is derived into the following four working hypotheses, namely, i) integration of the vision acquisition and processing; the unique data acquisition system consists of identical image sensors, all of which are integrated into a compact system whose major mechanical limits is dictated by the size of the sensors and processing electronics and their interconnectivity; moreover, targeted applications only need data capture from the aforementioned image sensors, *i.e.*, excluding the usage of any additional sensor such as distance sensor, etc. ii) scalability of the system; various incarnations of the camera are envisioned, and the design must be scalable by construction; iii) individual cameras with low (or limited) resolution; a Panoptic camera consisting of a large number of image sensors, each with low resolution is the favored design, in contrast with a solution consisting of few high-resolution image sensors; iv) real-time operation is a necessity in the image capture stage as well as in an embedded early image processing stage.

6.1 The Panoptic Camera Configuration

The physical realization of the omnidirectional image sensor consists in the layering of CMOS imagers on the surface of a hemisphere such that each each imager has its optical axis aligned with the surface normal. While we consider only an hemisphere for practical reasons, all the considerations about the design may apply to the full sphere.

6.1.1 Hemispheric Arrangement

The locations of CMOS imagers on the surface of a hemisphere respect the sampling scheme proposed in Chapter 2. Since we deal with real sensors, in the coverage method we have to account for the physical dimension of the camera package. Since the proposed sampling generates positions which are approximately equally distributed on the spherical surface, the general coverage method for one hemisphere is to assign each camera a circular face with constant area.

In order to define the spherical coordinate of each sensor location, the hemispherical surface of a unit sphere is divided into $N_{\text{flo}} + 1$ latitude floors. All circular faces located on a floor have the same latitude angle. The top most floor located on the North pole of the hemisphere only contains one circular face. The latitude angle θ_n of the n^{th} floor ($0 \leq n \leq N_{\text{flo}}$) is obtained from:

$$\theta_n = 2n\gamma_0, \quad \gamma_0 = \tfrac{\pi}{2(2N_{\text{flo}}+1)}, \tag{6.1}$$

where γ_0 is the radius of the circular face on the unit sphere such that $\theta_{N_{\text{flo}}} = \frac{\pi}{2} - \gamma_0$. Scaling this sphere allows to match γ_0 with the true (half) width of each CMOS imager.

The centers $(\theta_n, \phi_{n,j})$ of the circular faces located on each latitude floor are evenly positioned according to

$$\phi_{n,j} \;=\; j\Delta\phi_n, \quad \Delta\phi_n = \tfrac{2\pi}{N_n}, \tag{6.2}$$

with $0 \leq j < N_n$ and N_n determined such that $\Delta\phi_n$ is greater than the longitudinal angle occupied by one face. According to the sine formula for spherical trigonometry , this last angle is given by $2\arcsin\left(\frac{\sin\gamma_0}{\sin\theta_n}\right)$, and for $n > 0$,

$$N_n = \left\lfloor \pi \;/\; \arcsin\left(\tfrac{\sin\gamma_0}{\sin\theta_n}\right) \right\rfloor. \tag{6.3}$$

Hence, the total number of centers which equals the total number of cameras in the Panoptic device, is given by $N_{\text{cam}} = \sum_{n=0}^{N_{\text{flo}}} N_n$. For instance, for $N_{\text{flo}} = 0, 1, 2$ and 3, $N_{\text{cam}} = 1, 6, 15$ and 29, respectively.

In the following, the face centers are labeled with a single index $0 \leq i < N_{\text{cam}}$, so that each center is associated to the spherical coordinates $\mathbf{q}_i = (\theta_i, \phi_i)$, with $i = 0$ assigned to the North pole, and the mapping $i = i(n, j) = n\, N_{n-1} + j$ for $0 < n \leq N_{\text{flo}}$ and $0 \leq j < N_n$.

In the following, the face centers are labeled with a single index $0 \leq i < N_{\text{cam}}$, so that each center is associated to the spherical coordinates $\mathbf{q}_i = (\theta_i, \phi_i)$, with $i = 0$ assigned to the North pole, and the mapping $i = i(n, j) = n\, N_{n-1} + j$ for $0 < n \leq N_{\text{flo}}$ and $0 \leq j < N_n$.

As an example, Figure 6.1 depicts the hemispherical structure with seven floors ($N_{\text{flo}} = 6$). The 7-floor hemispherical structure contains $N_{\text{cam}} = 104$ circular faces.

In parallel with the spherical coordinates of the camera centers, their corresponding expression in the 3-D coordinate system $(\vec{x}, \vec{y}, \vec{z})$ centered on the hemisphere center is utilized, where $\vec{z}$ is identified with the vertical direction of the device, that is, $\vec{z}$ points toward the hemisphere North pole.

In this case, the 3-D coordinate $\vec{\mathbf{q}}_i$ (distinguished by a vectorial notation) of the i^{th} camera center $\mathbf{q}_i = (\theta_i, \phi_i)$ is given by

$$\vec{c}_i = R\,(\sin\theta_i \cos\phi_i\, \vec{x} + \sin\theta_i \sin\phi_i\, \vec{y} + \cos\theta_i\, \vec{z}),$$

where R stands for the radius of the Panoptic hemisphere.

6.1.2 Camera Orientations

Camera positions on the Panoptic device are identified with their respective camera focal points, that is, the i^{th} camera is centered on $\mathbf{q}_i = (\theta_i, \phi_i)$.

In addition to its location, each camera c_i is also characterized by three vectors: the "target" direction $\vec{t}_i$ pointing in the camera line of sight (focus direction), the "up" direction $\vec{u}_i$ providing the vertical direction in the pixel representation of the camera, and a vector $\vec{v}_i$ orthogonal to the two first, that is, the horizontal direction in the pixel domain. The vectors $\vec{u}_i$ and $\vec{v}_i$ vector form an orthogonal referential for the pixel coordinates of each camera.

Given the positioning scheme defined in Section 6.1.1, each camera c_i (for $1 \leq i < N_{\text{cam}}$) is oriented so that, first, the target direction is normal to the sphere surface, that is, $\vec{t}_i = \vec{\mathbf{q}}_i$, and second, the vectors $\vec{u}$ and $\vec{v}$ are aligned, respectively with the tangential vectors $\vec{e}_{\phi,i} = (\vec{z} \wedge \vec{c}_i)/\sin\theta_i$ and $\vec{e}_{\theta,i} = \vec{\mathbf{q}}_i \wedge \vec{e}_\phi$ to the sphere at $\mathbf{q}_i$ (with $\wedge$ the common vectorial product between two vectors). For the North pole camera $\mathbf{q}_0$, $\{\vec{t}_0, \vec{u}_0, \vec{v}_0\} = \{\vec{z}, \vec{x}, \vec{y}\}$ is selected.

Explicitly, considering the aforementioned placement of the cameras on the hemisphere structure of the Panoptic device, the three unit vectors of the i^{th} camera are obtained from

$$\begin{pmatrix} \vec{t}_i \\ \vec{u}_i \\ \vec{v}_i \end{pmatrix} = \begin{pmatrix} \sin\theta_i\cos\phi_i & \sin\theta_i\sin\phi_i & \cos\theta_i \\ -\cos\theta_i\sin\phi_i & -\cos\theta_i\cos\phi_i & \sin\theta_i \\ -\sin\phi_i & \cos\phi_i & 0 \end{pmatrix} \begin{pmatrix} \vec{x} \\ \vec{y} \\ \vec{z} \end{pmatrix}. \tag{6.4}$$

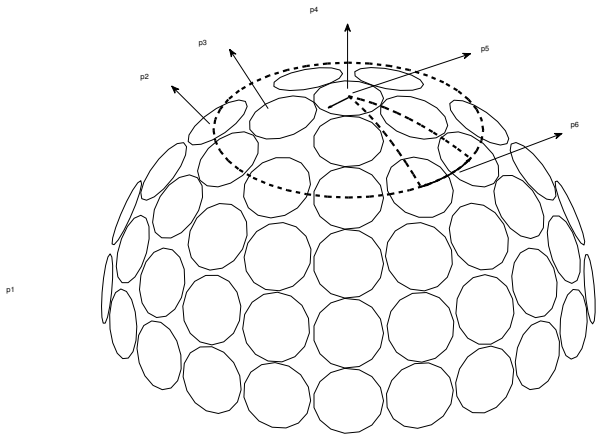


Figure 6.1: *Hemispherical structure with seven floors*

6.1.3 Intrinsic Camera Parameters

Additional important parameters characterize the intrinsic camera properties, in addition to the location and to the orientations of each camera in the Panoptic device. Since the Panoptic device is made of a collection of identical imagers, these parameters are assumed to be identical for each camera.

The main intrinsic parameters are the focal length, denoted by $f_L > 0$, the field-of-view (FOV) $\alpha > 0$, defined as the angle formed with the optical axis of the camera, and the resolution of the camera, that is, the size $n_h \times n_v$ of the pixel grid.

According to a pinhole camera model [39], the focal length controls the mapping between light ray direction and pixel coordinates, while the FOV determines the limit angle around its optical axis beyond which the camera is unable to record light.

6.2 Omnidirectional Vision Construction

In the following we detail the algorithmic steps that we used for the implementation of the light field interpolation described in Chapter 2 in the proposed hardware architecture.

In the reconstruction process, the elementary operation is to find the value of the light field $\mathcal{L}(\mathbf{x}, \omega)$ for a fixed position $\mathbf{x} = \bar{\mathbf{x}}$ and for every ω in the discretized sphere $\mathcal{S}_d$. The discretization of the sphere should follow the proposed scheme in Section 2.2, but another discretization might be chosen, *e.g.*, the equirectangular grid described in Section **??** to ease the display of the omnidirectional reconstruction. The direction of each light ray to interpolate is identified by the unit vector $\omega = (\theta_\omega, \phi_\omega)$ in spherical coordinates. For the sake of simplicity, it is assumed that $\bar{\mathbf{x}}$ is localized in the hemisphere center, *i.e.*, $\bar{\mathbf{x}} = 0$, but the same developments can be generalized to any other observation point. In the following this shorthand is used $\mathcal{L}(\bar{\mathbf{x}}, \omega) = \mathcal{L}(\omega)$.

The construction of the virtual omnidirectional view $\mathcal{L}(\omega) \in \mathbb{R}$ is performed in two algorithmic steps.

First Algorithmic Step Given the direction $\omega \in \mathcal{S}_d$, we have to determine the cameras having ω in their FOV. This is done by finding all the camera index $0 \leq i < N_{\text{cam}}$ such that:

$$\omega_{t_i} \ = \hat{\omega} \cdot \vec{t_i} \ > \cos \tfrac{\alpha}{2}, \tag{6.5}$$

where α is the camera FOV. The angle between $\hat{\omega}$ and $\vec{t_i}$ is controlled to be smaller than $\frac{\alpha}{2}$. After finding the contributing cameras, the next step consists in projecting the direction $\hat{\omega}$ on the camera plane to extract the pixel coordinates.

Using the pinhole camera model [39], the contributing two dimensional position (x_{u_i}, x_{v_i}) on the i^{th} camera image plane (which is identified by coordinate unit vectors $\vec{u}_i$ and $\vec{v}_i$) is expressed as:

$$(x_{u_i}, x_{v_i}) \ = \ -(\hat{\omega} \cdot \vec{u}_i, \hat{\omega} \cdot \vec{v}_i) \, \tfrac{f_L}{\omega_{t_i}}, \tag{6.6}$$

where f_L represents the camera focal length in (6.6).

The position (x_{u_i}, x_{v_i}) on the image frame of each contributing camera is likely not to coincide with an exact pixel location of a camera image frame. The light intensity of the contributing position can be estimated by the light intensity of the nearest actual pixel location to the contributing position. An alternate method consists of interpolating the value using, *e.g.*, bilinear interpolation.

As a final result, the first algorithmic step estimates the values $\mathcal{L}(\mathbf{q}_i, \omega)$ for each contributing camera i satisfying (6.5).

Second Algorithmic Step After finding the intensity $\mathcal{L}(\mathbf{q}_i, \omega)$ of all the contributing cameras i in direction ω, the second algorithmic step to compute $\mathcal{L}(\omega)$ is the one described in Eq. (2.22):

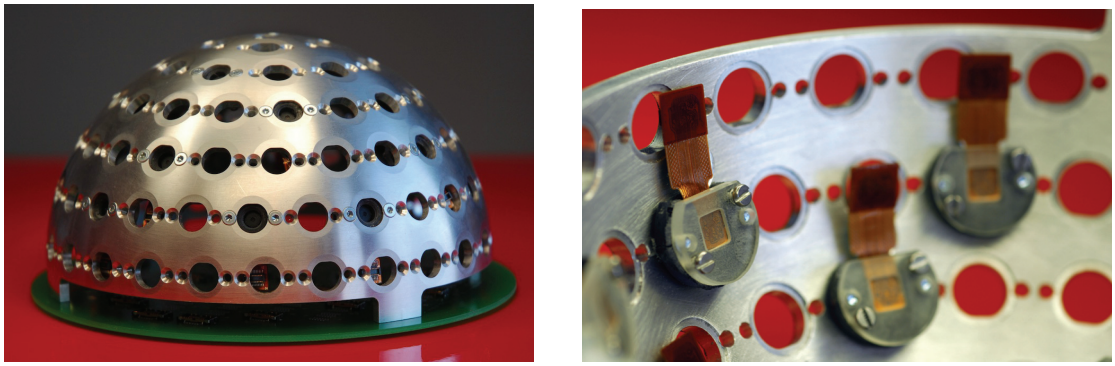
$$\mathcal{L}(\omega) = \frac{\sum_i g(\|\mathbf{q}_i - R\hat{\omega}\|)\mathcal{L}(\mathbf{q}_i, \omega)}{\sum_i g(\|\mathbf{q}_i - R\hat{\omega}\|)}. \tag{6.7}$$

6.3 Physical Realization

A custom Panoptic camera prototype was built using a classical digital machining of an aluminum structure, and polyvinyl chloride (PVC) camera holders [4]. The location of the cameras is based on the circular positions of the hemisphere structure shown in Figure 6.1. The fabricated Panoptic camera is shown in Figure 6.2. The diameter of the hemisphere is $2\,R = 129$mm. The fabricated hemisphere structure is placed over a circular printed circuit board which provides access to embedded imagers through flexible wire connections.

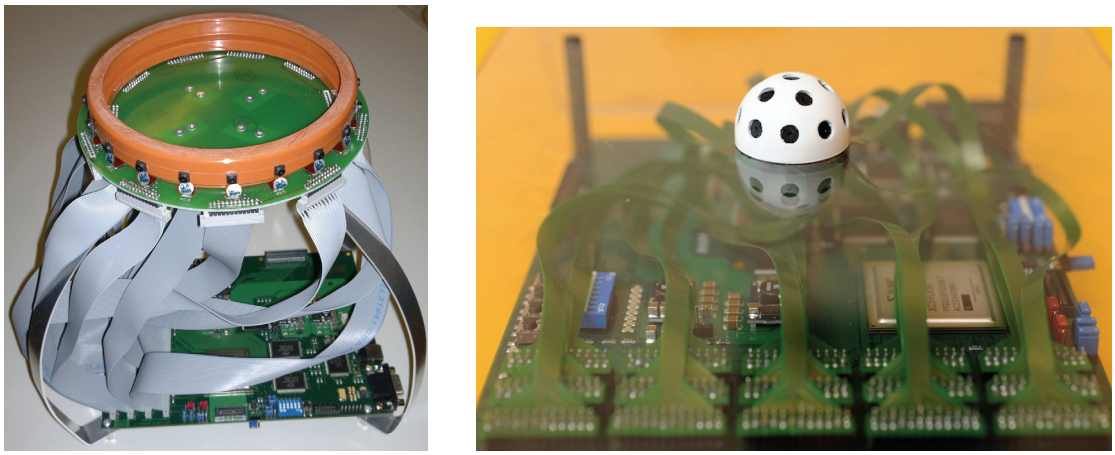
The camera module utilized in the built Panoptic prototype is PixelPlus PO4010N single chip CIF 368×304 pixels (with an effective resolution of 352×288) camera. The nominal diagonal, vertical and horizontal angle-of-view are equal to 72.3, 66 and 68 degrees, respectively. Hence, $\alpha = 66\,\pi/180$ is assumed. The effective focal length is $f_L = 1.27$mm.

Other custom prototypes were built in addition to the one in Figure 6.2. The prototype in Figure 6.3(a), that we call the Panoptic Ring camera obeys to a simplified model, consisting of 20 camera modules, arranged around the equator. Most of the experiments in Section 6.6 are performed with this prototype, since at the moment of the writing the prototype in Figure 6.2 has only 30 populated camera positions over the 104 available, which make it not adequate to perform certain task, such as depth map estimation or synthetic aperture photography.



(a) (b)

Figure 6.2: *(a) Side view, (b) and internal view of the fabricated Panoptic camera with individual imagers*



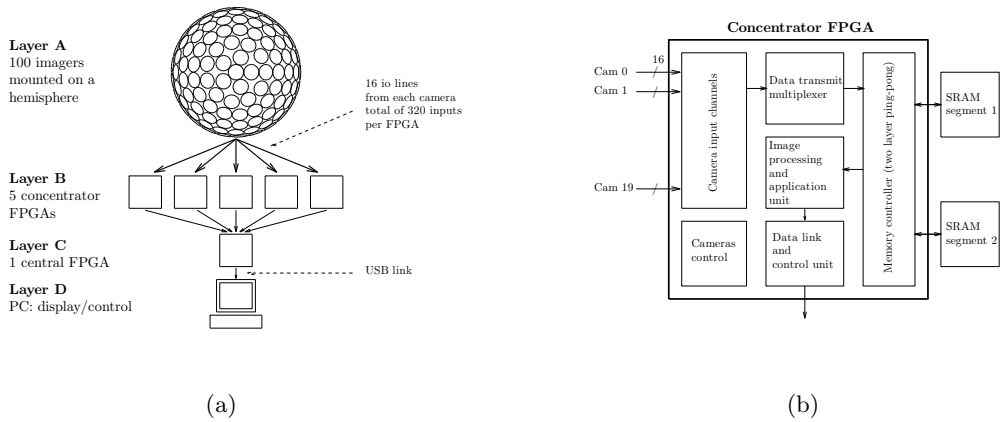
(a) (b)

Figure 6.3: *Two additional prototypes (a) The Panoptic Ring camera, (b) The small Panoptic camera*

In Figure 6.3(b), we show a a third prototype composed by 15 imagers connected to a single FPGA. It has a radius of $3cm$ (the same dimension ping-pong ball), and it has been used to show the first example of real-time reconstruction of an omnidirectional HD image at 25 frames per seconds on the FPGA [3].

6.4 FPGA Development Platform

This section describes some of the hardware implementation details: while we do that for completeness, the content is quite technical and might be skipped without getting off the track. We refer to [3, 4] for a more detailed description. The Panoptic system is designed with the aim of having its own custom ASIC imagers with integrated intra and inter imager signal processing features and integrated signal processing ASIC cores dedicated for omnidirectional imaging and its applications. An hardware emulation platform has been designed



(a) (b)

Figure 6.4: *(a) Architecture of the full hardware system and, (b) architecture of a concentrator FPGA*

and developed based on Field Programmable Gate Array (FPGA) for the practice of implementing, containing the Panoptic device and its applications in a real-time environment, and qualifying it for an ASIC solution.

A FPGA based system which supports a Panoptic camera with up to 100 imagers generating 16 bit common intermediate format (*i.e.*, CIF 352×288 pixels) images at 25 frame per second rate is devised. This system receives an aggregate bit rate of 3.8 Gb/s.

Prior to the development of the system, a careful feasibility study has been carried out, focusing on the system level required hardware specifications in terms of image acquisition rate, data transmission bandwidths, image processing rate, memory bandwidth and capacity, required level of architectural parallelism, FPGA pin count, connectivity, which conducted to the development of a layered system architecture, shown in Fig. 6.4(a).

The system consists of four layers, i) layer A: 100 imagers with programmable resolution, up to CIF, ii) layer B: five concentrator FPGAs, handling local image processing over 20 imagers in parallel, each, iii) layer C: one central FPGA which processes the final omnidirectional image construction, based on data transmitted in parallel from the concentrators, iv) layer D: a PC is in charge of the applicative layer consisting of displaying the operation results transmitted from the central FPGA. The PC is not a mandatory block in the system which is autonomous; it is only used to display results in the prototype implementation. In the final application embedding the Panoptic camera, real time display capability or data communication capabilities shall be provided.

6.4.1 Concentrator FPGA

An FPGA board has been designed utilizing Xilinx Virtex5 XC5VLX50-1FF1153C as the concentrator FPGA module. Each concentrator FPGA board supports up to 20 imagers with 16 input/output lines, each. The concentrator FPGA board contains two zero bus turn around (ZBT) SRAMs with minimum capacity to hold twenty 16-bit color images with CIF frame size, and an operating bandwidth of 166 MHz. High-speed LVDS connections are provided for the concentrator FPGA board as a mean for data and control signal communication with the central FPGA module.

The architecture of the concentrator FPGA system is depicted in Fig. 6.4(b). The concentrator FPGA consists of five blocks. The arrow lines depicted in Fig. 6.4(b) demonstrate the image data flow inside the concentrator FPGA. Image data originating from the cameras enters the concentrator FPGA via the camera channel input block. The data transmit multiplexer block multiplexes the 20 input camera channels and passes the timed multiplexed data to the memory controller block. The memory controller block stores the incoming image frame data from the 20 cameras inside one of the SRAMs; at the same time it also retrieves the previously stored images from the other SRAM and hands it over to the image processing and application unit block. The SRAMs swap their role (*i.e.*, one being written and one being read) each time a new image frame data is fully received. The image processing and application unit block is in charge of anticipated signal processing. In addition, some basic functionalities such as real-time single-channel image capture, simultaneous capture of twenty images, single-channel video display, etc are also considered for this block. The image processing and application unit block hands over its processed image data to the data link and control unit block. The data link and control unit block is in charge of transmitting the processed image data to the central FPGA module and servicing the control data requests received from the central FPGA module. To support the programmability feature of the cameras, a camera control block is also considered. The central FPGA can access this block through the data link and control unit block.

The concentrator FPGA functionality is categorized into two major tasks, regarding the captured image data. One is related to the multiplexing of the camera input channels and the other to the image processing application. Each of these operations imposes a minimum performance limit to the concentrator FPGA. The maximum of the two is considered as the minimum performance limit of the concentrator FPGA.

The concentrator FPGA must multiplex the incoming image data from 20 cameras. The cameras output their image data on a per-line basis, assuming the synchronization of all the 20 cameras connected to the concentrator FPGA. The concentrator FPGA first captures the incoming line from all the 20 cameras. While receiving the next line from the cameras, the concentrator FPGA also transmits the multiplexed version of the received previous line to one of the SRAMs. Thus the amount of time taken by the concentrator FPGA to transmit the multiplexed version of the received image data lines must be equal or less than the amount of time it takes for a single camera (assuming all the cameras to be the same) to transmit one line of image data. In mathematical form, this is expressed as:

$$N_{\text{cam}} \times \frac{n_h}{F_{\text{fpga}}} \leq \frac{C_w}{F_{\text{cam}}}. \tag{6.8}$$

In (6.8), I_w represents the frame width of the image, F_{fpga} the concentrator FPGA clock frequency, N_{cam} the number of cameras interfaced to the concentrator FPGA, C_w the cameras frame width and F_{cam} the rate at which the cameras transmit their pixel data to the outside world. The first minimum required performance of the concentrator FPGA is obtained by solving the inequality:

$$N_{\text{cam}} \times \frac{n_h}{C_w} \times F_{\text{cam}} \leq F_{\text{fpga}}. \tag{6.9}$$

Another performance criterion reflects the amount of time a concentrator FPGA spends to conduct an image processing application. Irrespective of the type of the application, the real-time feature of the system requires that the image processing time be less than or equal to the amount of time a single camera spends to generate one full frame. The amount of time

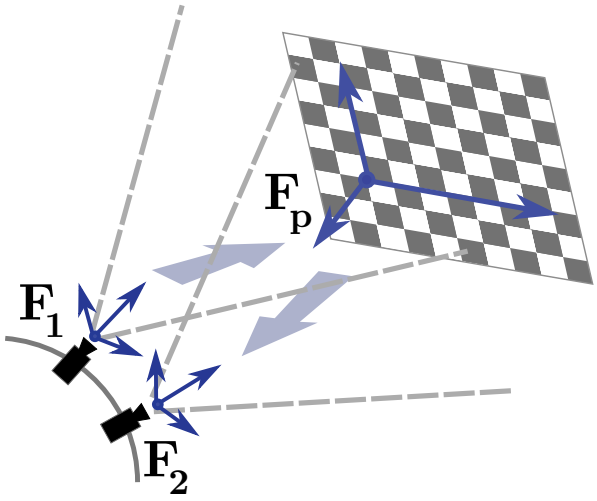


Figure 6.5: *Extrinsic parameter calibration principles*

needed for a typical camera to generate one full frame is obtained from the frame rate. Hence the second performance requirement is obtained from

$$T_{pc} \leq \frac{1}{f_{\text{ps}}}, \tag{6.10}$$

where f_{ps} is the camera frame per second rate, and T_{pc} is the image processing application process time. The value of T_{pc} is dependent on the concentrator FPGA operating clock frequency F_{fpga} and the architecture designed to conduct the image processing.

6.4.2 Central FPGA

The main task devoted to the central FPGA consists of receiving data processed by the concentrator FPGAs, apply the final image processing stage, and transfer the final results to a PC through a USB link for displaying. The central FPGA board has been developed based on the concentrator board architecture, thus forming a modular system. The performance requirement of the central FPGA module depends on the rate of the processed data which it receives from the concentrator FPGAs and the maximum local processing time (which is essentially expressed as (6.10)) needed to conduct the final image processing stage.

6.5 Calibration

The reconstruction method described in Section 6.2, and the subsequent hardware realizations described in the next sections assume a perfect knowledge of intrinsic camera parameters such a FOV, lens distortion, focal length, intensity dynamics, as well as extrinsic camera parameters including camera localizations and orientations on the surface of the hemisphere.

The fabricated Panoptic cameras shown in Figure 6.2 intrinsically minimizes errors of extrinsic parameters as a benefit of the digital machining of a rigid aluminum structure. The use of identical CMOS cameras in all Panoptic facets target the same goal. Nevertheless, a good estimation of the discrepancy between the theoretical and the actual camera intrinsic

and extrinsic configuration is mandatory. Similar consideration are valid for the Panoptic camera in Figure 6.3(a). Although the positions on the plastic ring is known approximately, a good calibration is needed to avoid artifacts in the output images. In the following we will describe a calibrations method, which is an adaptation of state-of-the-art algorithms in computer vision.

6.5.1 Intrinsic Calibration

Intrinsic camera parameters characterize the mapping between a 3-D point in the scene and the observed 2-D position on the camera plane and are split into two classes. The first class is dedicated to the linear *homography*, that is, the 3×4 *camera* matrix mapping of 3-D points in (homogeneous) coordinates into 2-D (homogeneous) pixel coordinates [39]. The second class models the non-linear mapping effects such as the lens distortion. The reader is referred to [39] for more details of the theoretical estimation of these parameters.

For practical purposes, these parameters are extracted using the "Camera Calibration Toolbox for Matlab" from [13]. For each camera intrinsic calibration, this toolbox uses the vision of one flat checkerboard pattern of known size presented under different orientations.

6.5.2 Extrinsic Calibration

The toolbox [13] is also used to accurately determine the true extrinsic parameters of each camera c_i on the surface of the Panoptic sphere, which corresponds to the estimation of the camera center $\mathbf{q}_i$ and of the three vectors $(\vec{t}_i, \vec{u}_i, \vec{v}_i)$, *i.e.*, the optical axis direction $\vec{t}_i$ and an orthonormal base on the image plane $(\vec{u}_i, \vec{v}_i)$ as described in Section 6.1.2.

The position of one camera relatively to one other can be determined, provided that both observe the same checkerboard pattern and that their intrinsic parameters have previously been calibrated. The intrinsic calibration of the camera matrix provides knowledge of the mapping between the coordinate system of one camera, *i.e.*, the one determined by the camera focal point (origin) $\mathbf{q}$ and the vectors set $\{\vec{t}, \vec{u}, \vec{v}\}$, and a coordinate system defined on the checkerboard plane. As depicted in Figure 6.5, it is for instance possible to jump from the coordinate system of one camera F_1 to that of the checkerboard plane F_p, and similarly from F_2 to F_p. Therefore, using the inverse mapping from F_p to F_2, every point or vector expressed in the coordinate system of F_1 can be described in F_2. *A fortiori*, this representation in F_2 is therefore also available for the vectors $\{\vec{\mathbf{q}}_1, \vec{t}_1, \vec{u}_1, \vec{v}_1\}$.

The full extrinsic calibration of the Panoptic device consists of, i) arbitrarily considering one camera coordinate system as the fundamental system (e.g., the North pole camera $\mathbf{q}_0$), and ii) estimating the change of coordinate system between neighboring cameras thanks to the simultaneous observation of checkerboard planes between pair of neighboring cameras.

In the end of the process, working by overlapping camera neighborhood, the coordinates of the four vectors $\{\vec{\mathbf{q}}_i, \vec{t}_i, \vec{u}_i, \vec{v}_i\}$ for each camera c_i are expressed in the common North pole frame $\{\vec{\mathbf{q}}_0, \vec{t}_0, \vec{u}_0, \vec{v}_0\}$.

6.6 Experimental Results

In this section we show few examples of the results obtained used the proposed signal processing framework applied to the images coming from the Panoptic camera an the Panoptic Ring camera. We briefly recall the experimental setup:

1. Panoptic camera: 30 CMOS imagers around an hemisphere with an equivalent angular resolution of $\Delta\theta_{\mathbf{x}} = 0.78$rad. Each imager has a resolution of 352x288 for an equivalent angular resolution $\Delta\theta\omega = 0.0026$rad. The radius of the hemisphere is $R = 6.5$cm. The hyperfocal distance of the system is $\mathrm{d}_h = \frac{R\Delta\theta_{\mathbf{x}}}{4\Delta\theta_\omega} = 9.6$m

2. Panoptic Ring camera: 20 CMOS imagers around the equator with an spatial angular resolution of $\Delta\phi_{\mathbf{x}} = 0.314$rad. Each imager has a resolution of 352x288 for an equivalent angular resolution $\Delta\theta\omega = 0.0026$rad. The radius of the hemisphere is $R = 7$cm. The hyperfocal distance of the system is $\mathrm{d}_h = \frac{R\Delta\theta_{\mathbf{x}}}{4\Delta\theta_\omega} = 4$m

The hyperfocal distance is an indicator of the operating range of the system. We use the same imagers for both prototypes, with identical angular resolution $\Delta\theta_\omega$. Also the radius of both systems are quite similar, they have a difference of 0.5 cm. The Panoptic Ring camera has a small hyperfocal distance which indicates that it operates for shorter distances. This is why we only use the Panoptic Ring during our tests on the Light Field Depth Estimation algorithm.

During the experiments we render the images using an equirectangular grid.

Figure 6.6: *Example of images acquired from the Panoptic CMOS imagers.*

6.6.1 Omnidirectional Imaging

The first application we demonstrate is the reconstruction of one omnidirectional image with single viewpoint. We choose the center the hemisphere $\mathbf{x} = 0$ as focal of the systems. The systems are both focused at the hyperfocal distance. The omnidirectional images are rendered with the same angular resolution $\Delta\theta_\omega$ of the CMOS imagers. We use a narrow aperture, *i.e.*, we use a narrow kernel for the interpolation.

In Figure 6.7 we show an omnidirectional reconstruction using the Panoptic camera. The images coming from the CMOS imagers are shown in Figure 6.6. The omnidirectional image is interpolated on the upper hemisphere, *i.e.*, $\theta_{\mathbf{x}} \in]0, \pi/2]$. The Panoptic Camera is calibrated using the procedure described in Section 6.5. In Figure 6.9 we show another scenario and we illustrate the benefits of using the automatic calibration algorithm described in Section 4.3 to refine the calibration parameters. Since we optimize the pairwise light intensities, the automatic algorithm refines the parameters in order to have a visually pleasant result as evident from the zoom on a detail.

In Figure 6.8 we show some frames from a video sequence acquired with the Panoptic Ring camera. The panorama has a vertical FOV around the equator of about 0.3 rad, *i.e.*, $\theta_{\mathbf{x}} \in [\pi/2 - 0.3, \pi/2 + 0.3]$.

Figure 6.7: *Omnidirectional reconstruction (hemispherical) of the Rolex Learning Center*

6.6.2 Automatic Camera Calibration and Depth Estimation

We test the algorithms described in Chapter 4 on the ring camera. This is a fundamental test to validate the camera model used in the thesis. We show that automatic calibration algorithm and the Light Field Depth Estimation algorithm work without previous calibration of the system. The Panoptic Ring camera calibration parameters are set to their nominal values: for the external calibration we use the nominal camera positions and orientation found during the manual placing of the cameras on ring. For the internal calibration of the cameras we use the nominal values coming from the constructor. In Figure 6.10 we show the results of the automatic calibration procedure. During the optimization we used a multiresolution scheme, iterating the optimization at increasing rendering resolutions $\Delta\theta_\omega/(2^s), s = 2, 1, 0$, to avoid the minimization to be stuck on local minima. The system is focused at the hyperfocal distance. In the reconstructions after the optimization there are no visual artifacts.

The optimized calibration parameters are then used to calculate a dense depth map of the scene. The results are given in Figure 6.12. The Light Field Depth Estimation (LFDE) algorithm offers a much cleaner depth map when compared to a simple WTA approach, which justify the use of the total-variation regularization. In the same figure we show the rendered images using the estimated depth maps: the depth map estimated with WTA produces geometric artifacts.

Finally in Figure 6.12 we used the estimated depth map to project the rendered image in the 3D space. We generated the mesh in a very naive way and we do not do any kind of post-processing, so the projection look a bit clumsy, but it gives a clear indication that the estimated depth map can give useful information on the structure of the scene.

6.6.3 Manifold Reconstruction: 3D-Stereo Panorama

The last experiment we propose is the generation of a manifold mosaic [66], *i.e.*, the generation of an image without a unique focal point. We choose to render a panoramic 3D-stereo image as described in [65]. The results are shown in Figure 6.13 as an Anaglyph 3D.

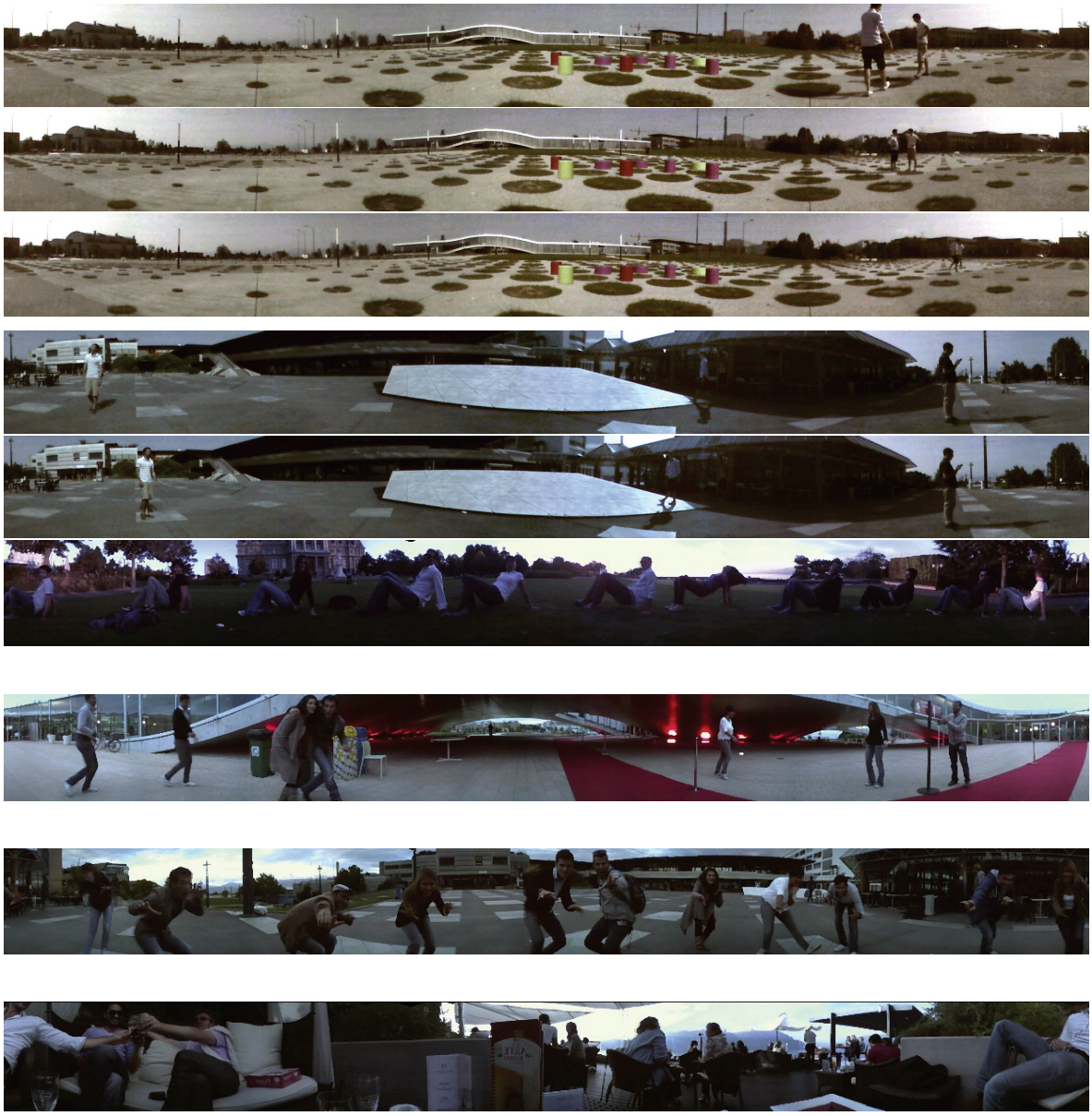


Figure 6.8: *Example of panorama image sequences taken with the Panoptic Ring camera.*

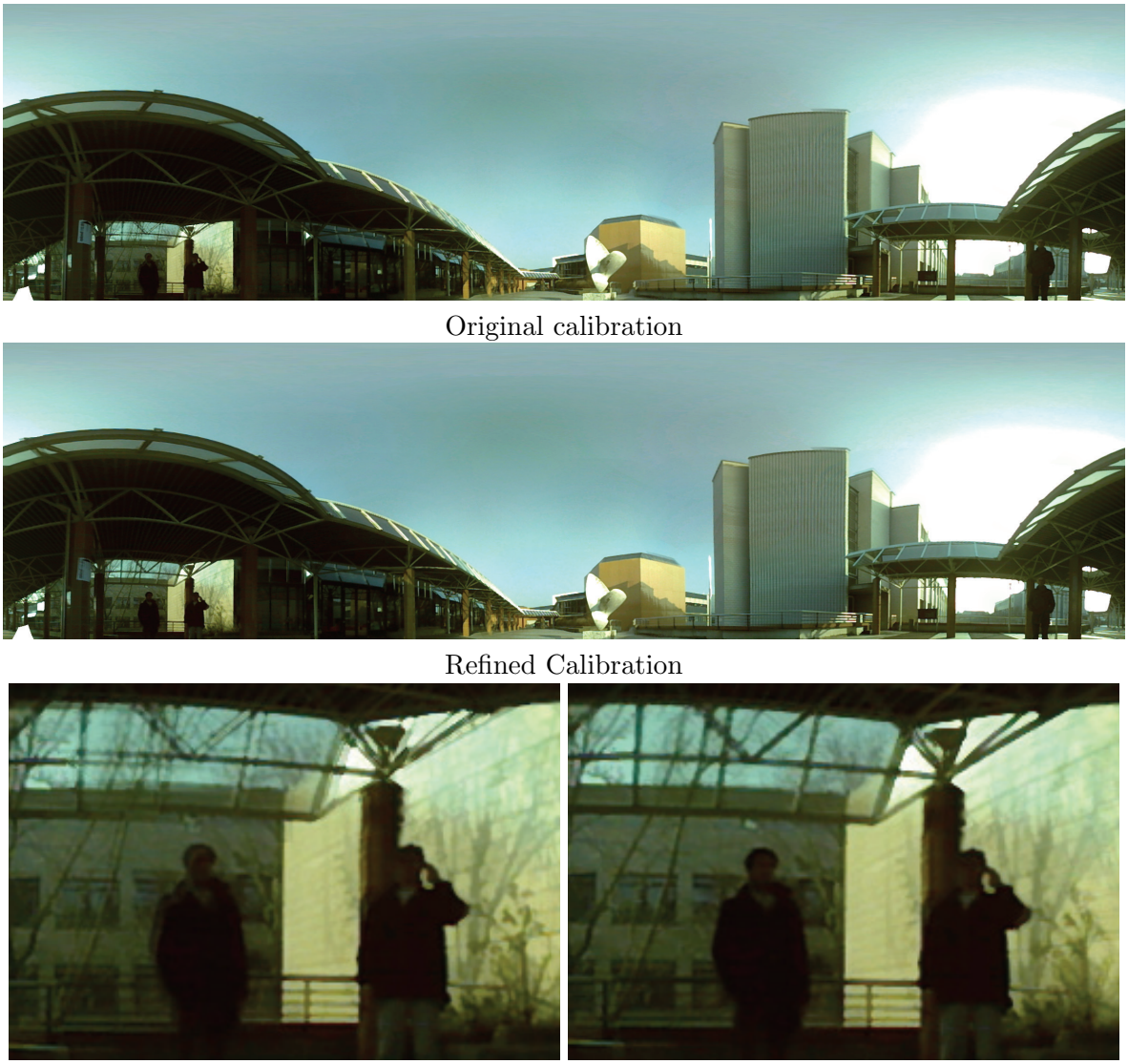


Figure 6.9: *Refinement of the Panoptic camera calibration using the automatic calibration algorithm. Bottom: zoom on a detail. On the right the result after the calibration refinement.*

Figure 6.10: *Effect of using the automatic calibration algorithm on the Panoptic Ring camera.*

Figure 6.11: *Dense depth estimation using the Panoptic Ring camera. The WTA estimation is noisy on untextured area and causes geometric artifacts in the rendered image.*

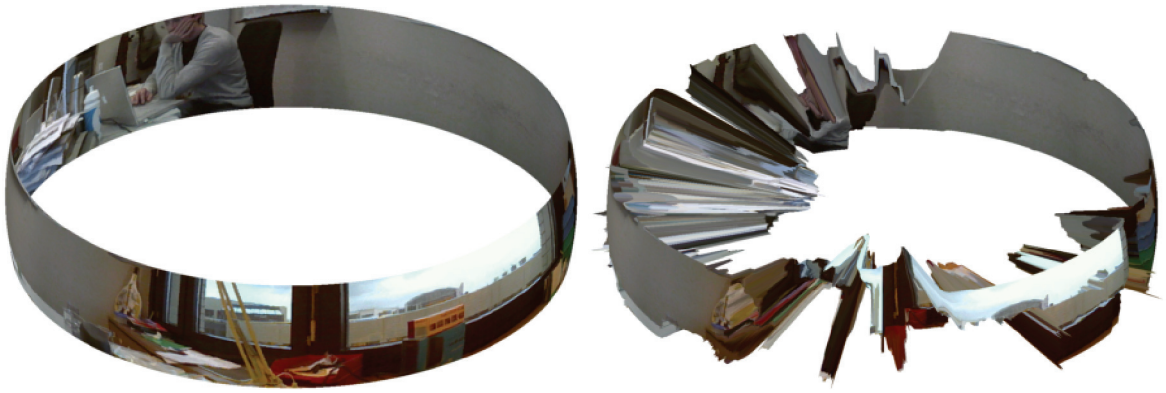


Figure 6.12: *3D representation of the panoramic image using the estimated depth map.*

Figure 6.13: *Stereo panorama using the Panoptic Ring camera.*

Chapter 7

Conclusions

In this thesis we design a new imaging device, which is able to capture the omnidirectional light field. We propose a model called Spherical Light Field Camera, which captures the full structure of light rays intersecting a sphere of finite volume in the 3D space. The camera model comprises several perspective imagers layered around a sphere and oriented along the surface normal. The vision system of some flying insects is a valuable source of inspiration for the design of such a camera. The visual system of the common fly is composed indeed of thousands of sensors, called ommatidia, placed on a spherical surface. We show that there is a mathematical foundation for such a choice. Such a configuration minimizes the number of parameters required to describe the omnidirectional light field.

The first achievement of the thesis consists in the analysis of the number of samples necessary to acquire the omnidirectional light field. This solves a very practical problem: how many cameras should we position around the sphere? What resolution should the imagers have? In the real world these questions determine the feasibility of a physical system for omnidirectional imaging.

The light field captured by the cameras can be readily used to reconstruct an omnidirectional image of the scene. We can further see the full system as a digital spherical photographic lens. Classic concepts in photography like depth-of-field or focus, can thus be transposed to the formation of omnidirectional photos.

The second achievement of the thesis is the formalization of the concept of omnidirectional light field photography. We propose a new set of graph-based tools, based on emerging techniques in signal processing that make use of graph structures. We then embed the light field into a graph: the light rays are the nodes of the graph, the correlation among light rays are represented through the nodes of the graph and the process of image formation is a diffusion process where neighboring nodes only exchange information. One of the most important contributions here is the demonstration that classical linear filtering techniques used in light field photography can be implemented through graph-based techniques. This result is important for two main reasons:

1. It makes the difficult problem of filtering on manifold, *e.g.*, the sphere, very simple and easy to implement in custom hardware.

2. It opens a full set of new features like the implementation of non-linear photographic lenses.

The effectiveness of the proposed techniques are confirmed by experimental results. We propose an example of a non-linear lens, based on the minimization of the total-variation of the system, which shows an increased robustness to noise.

Our camera model has several applications, like in autonomous navigation, where an omnidirectional vision is of primary importance for motion stabilization and obstacle avoidance. In such applications, it is also of fundamental importance to extract additional information about the structure of the scene.

The third achievement of the thesis is a new framework for the extraction of dense depth maps directly from plenoptic derivatives. The proposed algorithm uses all the available information at the same time and works directly with light intensity measurement. It is shown to work with synthetic and real images. We also show that the estimated depth map does not contain geometrical artifacts in the reconstruction of images or the generation of synthetic views. While most existing techniques could not be directly applied to our camera model, it turns out that the proposed framework can be used in all existing multi-camera systems. On top of that, our variational solution is designed to have low memory requirements, which is one of the fundamental limits of current computer vision techniques. The same framework can be easily extended to the estimation of camera motion and to the geometric calibration of multi-camera systems, which are very important practical problems in camera networks.

Finally, we do not only propose a new camera model, we also design a real camera prototype called the *Panoptic Camera*. We implement it using miniature CMOS imagers layered around a spherical support. The cameras are connected through a network of FPGAs. The FPGAs collect the video streams from the cameras and process them in real time. The physical realization of the system is made possible by the distributed nature of the proposed algorithms; the tremendous throughput coming from the imagers could not be handled using a centralized architecture.

Given the complexity of the problem addressed in this thesis, many theoretical and practical aspects are still open. From the theoretical point of view, better bounds on the number of cameras needed for a correct acquisition could be found. In this thesis we used the assumption that the perspective imagers have a narrow field of view, since this is what is available with the current technology. The possibility of using larger FOV imagers would necessitate a smaller number of sensors, which in turn would ease the task of the miniaturization of the complete system.

Also we did not explicitly include a model of color formation inside the real imaging sensor. We implicitly assumed that we are given different color channels and that we could process them separately. This is in fact an oversimplification if we want to produce high quality images with a real system. Another related aspect is the dynamic range of the acquired photos. Most sensors quantize the acquired light using 8 bits. A very interesting problem that needs to be addressed is the following: what is the optimal exposure pattern for the imagers to retain a dynamic light range that is as large as possible?

Another practical open problem is the fully automatic calibration of the system. Although our calibration algorithm shows promising results, we think that the performance could be largely improved.

Undoubtedly there is still some work to be done before pushing a Spherical Light Field Camera out of a research laboratory. But I look forward to the moment where I will proudly hold my personal Panoptic in my hands.

Bibliography

[1] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20, 1991.

[2] E. H. Adelson and J. Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):99–106, February 1992.

[3] Hossein Afshari, Laurent Jacques, Luigi Bagnato, Alexandre Schmid, P. Vandergheynst, and Y. Leblebici. Hardware implementation of an omnidirectional camerawith real-time 3D imaging capability. In *Proceedings of the IEEE 3DTV Conference*, pages 1–4. IEEE, May 2011.

[4] Hossein Afshari, Laurent Jacques, Luigi Bagnato, Alexandre Schmid, Pierre Vandergheynst, and Yusuf Leblebici. The PANOPTIC camera: A plenoptic sensor with Real-Time omnidirectional capability. *Journal of Signal Processing Systems*, pages 1–24, March 2012.

[5] A.K. Agrawal and R. Chellappa. Robust ego-motion estimation and 3d model refinement using depth based parallax model. In *Proceedings of the IEEE International Conference on Image Processing*, volume 4, pages 2483 – 2486 Vol. 4, October 2004.

[6] Zafer Arican and Pascal Frossard. Dense disparity estimation from omnidirectional images. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 399–404. IEEE, September 2007.

[7] JF Aujol, G Gilboa, T Chan, and S Osher. Structure-texture image decomposition - modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, 67(1):111–136, Jan 2006.

[8] Simon Baker and Shree K. Nayar. A theory of Single-Viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2):175–196, November 1999.

[9] SS Beauchemin and JL Barron. The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3):433–466, 1995.

[10] Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, December 2008.

[11] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, December 2006.

[12] Tom E. Bishop and Paolo Favaro. The light field camera: Extended depth of field, aliasing, and superresolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):972–986, May 2012.

[13] J.Y. Bouguet. Camera calibration toolbox for matlab.

[14] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, volume 3. 2011.

[15] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.

[16] AR Bruss and BKP Horn. Passive navigation. *Computer Vision Graph*, 21(1):3–20, Jan 1983.

[17] Emilio Camahort, Apostolos Lerios, and Don Fussell. Uniformly sampled light fields. Technical report, Austin, TX, USA, 1998.

[18] Jin X. Chai, Xin Tong, Shing C. Chan, and Heung Y. Shum. Plenoptic sampling. In *Proceedings of ACM SIGGRAPH*, pages 307–318, New York, NY, USA, 2000.

[19] Antonin Chambolle and Anil Kokaram. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, January 2004.

[20] Antonin Chambolle and Thomas Pock. A First-Order Primal-Dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, May 2011.

[21] Tony F. Chan and Jianhong Shen. Nontexture inpainting by Curvature-Driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, December 2001.

[22] Fan R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, February 1997.

[23] Christopher M. Cianci, Xavier Raemy, Jim Pugh, and Alcherio Martinoli. *Communication in a Swarm of Miniature Robots: The e-Puck as an Educational Tool for Swarm Robotics*, volume 4433 of *Lecture Notes in Computer Science*, chapter 7, pages 103–115. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[24] R. T. Collins. A space-sweep approach to true multi-image matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 358–363, 1996.

[25] Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In Heinz H. Bauschke, Regina S. Burachik, Patrick L. Combettes, Veit Elser, D. Russell Luke, and Henry Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, volume 49 of *Springer Optimization and Its Applications*, chapter 10, pages 185–212. Springer New York, New York, NY, 2011.

[26] Kostas Daniilidis, Ameesh Makadia, and Thomas Bulow. Image processing in catadioptric planes: Spatiotemporal derivatives and optical flow computation. *Proceedings of the IEEE Workshop on Omnidirectional Vision*, 0:3–10, 2002.

[27] M. N. Do, D. Marchand-Maillet, and M. Vetterli. On the bandwidth of the plenoptic function. *IEEE Transactions on Image Processing*, 21(2):708–717, February 2012.

[28] J. R. Driscoll and D. M. Healy. Computing fourier transforms and convolutions on the 2-Sphere. *Advances in Applied Mathematics*, 15(2):202–250, June 1994.

[29] A. Elmoataz, O. Lezoray, and S. Bougleux. Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. *IEEE Transactions on Image Processing*, 17(7):1047–1060, July 2008.

[30] Olivier Faugeras, Quang-Tuan Luong, and Theodore Papadopoulo. *The Geometry of Multiple Images*. MIT Press, 2001.

[31] C. Fehn. 3D-TV Using Depth-Image-Based Rendering (DIBR). In *Proceedings of Picture Coding Symposium*, San Francisco, CA, USA, December 2004.

[32] J. Foote and D. Kimber. FlyCam: practical panoramic video and automatic camera control. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 1419–1422, 2000.

[33] Christopher Geyer and Kostas Daniilidis. Catadioptric projective geometry. *International Journal of Computer Vision*, 45(3):223–243, December 2001.

[34] Guy Gilboa and Stanley Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, January 2009.

[35] J Gluckman and S.K Nayar. Ego-motion and omnidirectional cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 999–1005, jan 1998.

[36] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of ACM SIGGRAPH*, pages 43–54, New York, NY, USA, 1996.

[37] David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, March 2011.

[38] K.J Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. *Proceedings of the IEEE Workshop on Visual Motion*, pages 156–162, 1991.

[39] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, April 2004.

[40] DJ Heeger and AD Jepson. Subspace methods for recovering rigid motion .1. algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, Jan 1992.

[41] BKP Horn and BG Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[42] BKP Horn and EJ Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1):51–76, Jan 1988.

[43] Pelican Imaging. http://www.pelicanimaging.com/.

[44] Aaron Isaksen, Leonard McMillan, and Steven J. Gortler. Dynamically reparameterized light fields. In *Proceedings of ACM SIGGRAPH*, pages 297–306, New York, NY, USA, 2000.

[45] Ki-Hun Jeong, Jaeyoun Kim, and Luke P. Lee. Biologically inspired artificial compound eyes. *Science*, 312(5773):557–561, April 2006.

[46] A Jepson and D Heeger. A fast subspace algorithm for recovering rigid motion. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 124 – 131, oct 1991.

[47] Rongjie Lai and Tony F. Chan. A framework for intrinsic image processing on surfaces. *Computer Vision and Image Understanding*, July 2011.

[48] Luke P. Lee and Robert Szema. Inspirations from biological optics for advanced photonic systems. *Science*, 310(5751):1148–1150, November 2005.

[49] Leonardo and I. Richte. *The Notebooks of Leonardo Da Vinci*. Oxford University Press, 1980.

[50] M. Levoy. Light fields and computational imaging. *Computer*, 39(8):46–55, August 2006.

[51] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of ACM SIGGRAPH*, pages 31–42, New York, NY, USA, 1996.

[52] Chia K. Liang, Tai H. Lin, Bing Y. Wong, Chi Liu, and Homer H. Chen. Programmable aperture photography: multiplexed light field acquisition. In *Proceedings of ACM SIGGRAPH*, New York, NY, USA, 2008.

[53] Steven Lovegrove and Andrew J. Davison. Real-time spherical mosaicing using whole image alignment. In *Proceedings of the European Conference on Computer Vision*, ECCV'10, pages 73–86, Berlin, Heidelberg, 2010. Springer-Verlag.

[54] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 2 of *IJCAI'81*, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[55] Lytro. http://www.lytro.com/.

[56] Ameesh Makadia, Christopher Geyer, and Kostas Daniilidis. Correspondence-free structure from motion. *International Journal of Computer Vision*, 75(3):311–327, February 2007.

[57] Donald W. Marquardt. An algorithm for Least-Squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 1963.

[58] Jason D. McEwen, Gilles Puy, Jean-Philippe Thiran, Pierre Vandergheynst, Dimitri Van De Ville, and Yves Wiaux. Sampling theorems and compressive sensing on the sphere. *Proceedings of SPIE*, 8138(1):81381F–81381F–9, September 2011.

[59] S. K. Nayar. Catadioptric omnidirectional camera. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, June 1997.

[60] J Neumann, C Fermuller, and Y Aloimonos. Eyes from eyes: new cameras for structure from motion. *Proceedings of the IEEE Workshop on Omnidirectional Vision*, pages 19–26, 2002.

[61] Jan Neumann and Cornelia Fermüller. Plenoptic video geometry. *The Visual Computer*, 19(6):395–404, October 2003.

[62] Ren Ng. Fourier slice photography. In *Proceedings of ACM SIGGRAPH*, pages 735–744, New York, NY, USA, 2005.

[63] M. Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20:99–120, 2004.

[64] Nils Papenberg, Andrés Bruhn, Thomas Brox, Stephan Didas, and Joachim Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, April 2006.

[65] S. Peleg, M. Ben-Ezra, and Y. Pritch. Omnistereo: panoramic stereo imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):279–290, March 2001.

[66] Shmuel Peleg, Benny Rousso, Alex R. Acha, and Assaf Zomet. Mosaicing on adaptive manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1144–1154, 2000.

[67] Gabriel Peyré, Sébastien Bougleux, and Laurent Cohen. Non-local regularization of inverse problems computer vision. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Proceedings of the European Conference on Computer Vision*, volume 5304 of *Lecture Notes in Computer Science*, chapter 5, pages 57–68. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2008.

[68] Thomas Pock, Thomas Schoenemann, Gottfried Graber, Horst Bischof, and Daniel Cremers. A convex formulation of continuous multi-label problems. In *Proceedings of the European Conference on Computer Vision*, ECCV '08, pages 792–805, Berlin, Heidelberg, 2008. Springer-Verlag.

[69] Barrett Richard, Berry Michael, Chan Tony F., Demmel James, Donato June, Dongarra Jack, Eijkhout Victor, Pozo Roldan, Romine Charles, and van der Vorst Henk. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, 1994.

[70] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Proceedings of the International Conference of the Center for Nonlinear Studies on Experimental Mathematics*, 60(1-4):259–268, November 1992.

[71] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense Two-Frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, April 2002.

[72] Heung Y. Shum and Li W. He. Rendering with concentric mosaics. In *Proceedings of ACM SIGGRAPH*, pages 299–306, New York, NY, USA, 1999.

[73] David I. Shuman, Pierre Vandergheynst, and Pascal Frossard. Chebyshev polynomial approximation for distributed signal processing. In *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems and Workshops*, pages 1–8, June 2011.

[74] David I. Shuman, Pierre Vandergheynst, and Pascal Frossard. Distributed signal processing via chebyshev polynomial approximation. *arXiv*, November 2011.

[75] D Sinclair, A Blake, and D Murray. Robust estimation of egomotion from normal flow. *International Journal of Computer Vision*, 13(1):57–69, Jan 1994.

[76] A. Singer. From graph to manifold laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128–134, July 2006.

[77] Aljoscha Smolic, Peter Kauff, Sebastian Knorr, Alexander Hornung, Matthias Kunter, Marcus Muller, and Manuel Lang. Three-Dimensional video postproduction and processing. *Proceedings of the IEEE*, 99(4):607–625, April 2011.

[78] R. Swarninathan and S. K. Nayar. Non-metric calibration of wide-angle lenses and polycameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–419, 1999.

[79] Yuichi Taguchi, Amit Agrawal, Ashok Veeraraghavan, Srikumar Ramalingam, and Ramesh Raskar. Axial-cones: modeling spherical catadioptric cameras for wide-angle light field rendering. In *Proceedings of ACM SIGGRAPH Asia*, New York, NY, USA, 2010.

[80] T.Y. Tian, C. Tomasi, and D.J. Heeger. Comparison of approaches to egomotion computation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 315 –320, June 1996.

[81] I. Tosic, I. Bogdanova, P. Frossard, and P. Vandergheynst. Multiresolution motion estimation for omnidirectional images. In *Proceedings of EUSIPCO*, 2005.

[82] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment — a modern synthesis vision algorithms: Theory and practice. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, chapter 21, pages 153–177. Springer Berlin / Heidelberg, Berlin, Heidelberg, April 2000.

[83] Roberto Tron and Rene Vidal. Distributed image-based 3-D localization of camera sensor networks. In *Proceedings of the IEEE Conference on Decision and Control*, pages 901–908, December 2009.

[84] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy. Using plane + parallax for calibrating dense camera arrays. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2–9, 2004.

[85] Andreas Weishaupt, Luigi Bagnato, and Pierre Vandergheynst. Fast structure from motion for planar image sequences. In *Proceedings of Eusipco*, 2010.

[86] Yinyu Ye. Interior-Point polynomial algorithms in convex programming (y. nesterov and a. nemirovskii). *SIAM Review*, 36(4):682+, 1994.

[87] C Zach, T Pock, and H Bischof. A duality based approach for realtime tv-l1 optical flow. In Fred Hamprecht, Christoph Schnörr, and Bernd Jähne, editors, *Pattern Recognition*, volume 4713 of *Lecture Notes in Computer Science*, pages 214–223. Springer Berlin / Heidelberg, 2007.

[88] R. Zbikowski. Fly like a fly [micro-air vehicle]. *IEEE Spectrum*, 42(11):46–51, November 2005.

[89] Cha Zhang and Tsuhan Chen. Spectral analysis for sampling image-based rendering data. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(11):1038–1050, November 2003.

[90] Dengyong Zhou and Bernhard Schölkopf. A regularization framework for learning from graph data. *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*, Jan 2004.

[91] Dengyong Zhou and Bernhard Schölkopf. Regularization on discrete spaces. In Walter G. Kropatsch, Robert Sablatnig, and Allan Hanbury, editors, *Pattern Recognition*, volume 3663 of *Lecture Notes in Computer Science*, chapter 45, pages 361–368. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2005.

• Luigi Bagnato

Address	EPFL STI IEL, ELE 236 (Bâtiment ELE) Station 11 CH-1015 Lausanne, Switzerland
Telephone	+41 21 693 26 57
e-mail	l.bagnato@gmail.com
web	http://lts2www.epfl.ch/bagnato/
Nationality	Italian

Objective My long term goal is to become an executive in a fast growing high-tech company.

Education and training

2007 - 2012 **Doctoral School in *Electrical Engineering***
Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

2003 - 2006 **Master of Science in *Telecommunications and Computer Engineering***
University of Perugia, Perugia, Italy
110/110 cum laude

2000 - 2003 **Bachelor of Science in Information Technology Engineering**
University of Perugia, Perugia, Italy
110/110 cum laude

Work experience

2007 - present **Ecole Polytechnique Fédérale de Lausanne, Switzerland**
Research Assistant at LTS2 and LTS4 laboratory:
Research on Spherical Vision, 3D reconstruction, Light Field Processing. Algorithm development for Panoptic, the first Spherical Light Field camera.
Teaching Assistant:
Advanced Signal Processing (Master level)
Signals and Systems (Bachelor level)

2007 **University of Milano-Bicocca, Milano, Italy**
External collaborator at Department of Informatics, Systems and Communication (DISCo):
Research and development of face detection and recognition software integrated in Olivetti's commercial products.

Languages

Italian	Mother Tongue
English	fluent
French	fluent
Spanish	Intermediate
German	Basic

Language Certification Trinity College Exam of Spoken English 'Grade 12'

Computer Skills

Programming Languages	Java,C++,C,Matlab, Python, C#, VHDL, SQL, Javascript, PHP
Web Technology	HTML, PHP, Javascript, WebGL
O.S.	Windows, Linux, Mac OSX
Software	Spice, Electric, Blender, Adobe Suite

Personal Publications

[1] Hossein Afshari, Laurent Jacques, Luigi Bagnato, Alexandre Schmid, P. Vandergheynst, and Y. Leblebici. Hardware implementation of an omnidirectional camerawith real-time 3D imaging capability. In *Proceedings of the IEEE 3DTV Conference*, pages 1–4, May 2011.

[2] Hossein Afshari, Laurent Jacques, Luigi Bagnato, Alexandre Schmid, Pierre Vandergheynst, and Yusuf Leblebici. The PANOPTIC Camera: A Plenoptic Sensor with Real-Time Omnidirectional Capability. *Journal of Signal Processing Systems*, pages 1–24, March 2012.

[3] Alexandre Alahi, Luigi Bagnato, Damien Matti, and Pierre Vandergheynst. Foreground Silhouettes Extraction robust to Sudden Changes of background Appearance. In *Proceedings of the IEEE International Conference on Image Processing*, September 2012.

[4] Luigi Bagnato, Yannick Boursier, Pascal Frossard, and Pierre Vandergheynst. Plenoptic based super-resolution for omnidirectional image sequences. In *Proceedings of the IEEE International Conference on Image Processing*, pages 2829–2832. IEEE, September 2010.

[5] Luigi Bagnato, Pascal Frossard, and Pierre Vandergheynst. Optical flow and depth from motion for omnidirectional images using a TV-L1 variational framework on graphs. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1469–1472. IEEE, November 2009.

[6] Luigi Bagnato, Pascal Frossard, and Pierre Vandergheynst. A Variational Framework for Structure from Motion in Omnidirectional Image Sequences. *Journal of Mathematical Imaging and Vision*, 41(3):182–193, November 2011.

[7] Luigi Bagnato, Pascal Frossard, and Pierre Vandergheynst. Plenoptic Spherical Sampling. In *Proceedings of the IEEE International Conference on Image Processing*, September 2012.

[8] Luigi Bagnato, Matteo Sorci, Gianluca Antonini, Giuseppe Baruffa, Andrea Maier, Peter Leathwood, and Jean-Philippe Thiran. Robust Infants Face Tracking Using Active Appearance Models: A Mixed-State CONDENSATION Approach Advances in Visual Computing. In *Advances in Visual Computing*, volume 4841 of *Lecture Notes in Computer Science*, chapter 2, pages 13–23. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2007.

[9] Laurent Jacques, Eugenio De Vito, Luigi Bagnato, and Pierre Vandergheynst. Shape from Texture for Omnidirectional Images. In *Proceedings of EUSIPCO*, 2008.

[10] Andreas Weishaupt, Luigi Bagnato, and Pierre Vandergheynst. Fast Structure from Motion for Planar Image Sequences. In *Proceedings of Eusipco*, 2010.