Coherent multi-dimensional segmentation of multiview images using a variational framework and applications to image based rendering

by
Jesse Berent

A Thesis submitted in fulfilment of requirements for the degree of
Doctor of Philosophy of Imperial College London

Communications and Signal Processing Group
Electrical and Electronic Engineering Department
Imperial College London
2008

Abstract

Image Based Rendering (IBR) and in particular light field rendering has attracted a lot of attention for interpolating new viewpoints from a set of multiview images. New images of a scene are interpolated directly from nearby available ones, thus enabling a photorealistic rendering. Sampling theory for light fields has shown that exact geometric information in the scene is often unnecessary for rendering new views. Indeed, the band of the function is approximately limited and new views can be rendered using classical interpolation methods. However, IBR using undersampled light fields suffers from aliasing effects and is difficult particularly when the scene has large depth variations and occlusions. In order to deal with these cases, we study two approaches:

New sampling schemes have recently emerged that are able to perfectly reconstruct certain classes of parametric signals that are not bandlimited but characterized by a finite number of parameters. In this context, we derive novel sampling schemes for piecewise sinusoidal and polynomial signals. In particular, we show that a piecewise sinusoidal signal with arbitrarily high frequencies can be exactly recovered given certain conditions. These results are applied to parametric multiview data that are not bandlimited.

We also focus on the problem of extracting regions (or layers) in multiview images that can be individually rendered free of aliasing. The problem is posed in a multi-dimensional variational framework using region competition. In extension to previous methods, layers are considered as multi-dimensional hypervolumes. Therefore the segmentation is done jointly over all the images and coherence is imposed throughout the data. However, instead of propagating active hypersurfaces, we derive a semi-parametric methodology that takes into account the constraints imposed by the camera setup and the occlusion ordering. The resulting framework is a global multi-dimensional region competition that is consistent in all the images and efficiently handles occlusions. We show the validity of the approach with captured light fields. Other special effects such as augmented reality and disocclusion of hidden objects are also demonstrated.

Acknowledgment

Contrary to popular belief, the life of a PhD student is not as easy as it seems. It is an experience with ups, downs and arounds. Thankfully, the overall endeavor is inspiring, challenging and fun too. In my case, there are several people that have provided invaluable guidance and support.

First and foremost, I would like to express my deepest gratitude towards my supervisor Dr. Pier Luigi Dragotti. He has always known how to point me in the right direction while being positive about my ideas as well. I thank him for his willingness to share his wisdom and knowledge with me. He also given me all the opportunities a grad student would dream of and all this with the usual huge smile. I would also like to point out that despite him having had two children while I was working on this thesis, he has never forgotten about me or any of his students. On the whole, I am very grateful for his supervision and motivational speeches that never failed to boost confidence and get things done.

Throughout the research presented in this thesis I have had the opportunity to interact with many interesting people, be it at conferences, while giving talks or visiting academics. In particular, I would like to thank Mike Brookes, Dr. Luciano Sbaiz and Dr. Thierry Blu for their comments and suggestions. I would also like to thank Prof. Martin Vetterli for giving me the opportunity to spend a few months in his lab and to discuss my work with him. Several students have also contributed to this work including Justin Wong and Yizhou 'Eagle' Wang.

Moving on to an altogether different matter, I would have the thank the Imperial College Builders Arms club: Dr. Nicolas Gehrig, Dr. Fotis Talantzis, Dr. Nikolay Gaubitch, Dr. Loic Baboulaz, Dr. Jon Gudnason, Beth Jelfs (Thank you Beth, at least

we had one lady amongst us) and Mark Thomas. It is always nice to see we are real nerds and have to discuss Fourier transforms, Expectation-Maximization, partial differential equations, and all this at the pub. Perhaps discussing the rate of innovation of pints would have been more appropriate.

Back home, my swiss friends Ronnie Yarisal, Xavier Righetti and Thierry Bertossa as well as my brothers Leo and Max Berent made my holidays a nice break. My parents have also been very supportive and enthusiastic about my studies and enabled me to pursue my education in the best conditions. Thank you.

Last but definitely not least, I would like to express my deepest gratitude to my lovely wife Carin who has put up with me for the past few years and never failed to remind me that there are other things in life than plenoptic functions (I beg to differ), level sets, piecewise sinusoidal signals and so on.

Contents

List of Figures

List of Tables

Statement of Originality

I declare that the content embodied in this thesis is the outcome of my own research work under the guidance of my thesis advisor Dr P. L. Dragotti. Any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline. The material of this thesis has not been submitted for any degree at any other academic or professional institution.

Jesse Berent

Abbreviations

IBR:	Image Based Rendering
EPI:	Epipolar Plane Image
VR:	Virtual Reality
ICT:	Image Cube Trajectory
LDI:	Layered Depth Image
PDE:	Partial Differential Equation
SNR:	Signal-to-Noise Ratio
CAD:	Computer-Aided Design
FRI:	Finite Rate of Innovation

Chapter 1

Introduction

1.1 Motivation

Today's visual media systems provide a convincing experience. However, the mainstream capture, transmission and display technologies remain two-dimensional. A natural and very popular extension is to provide a three-dimensional experience. Many new technologies are being developed for such purposes. Cameras, as well as memory and processing power, are constantly improving while getting cheaper. These facts are making it increasingly popular to develop systems capable of providing the user with a three-dimensional feel of the scene. Research is being undertaken in different aspects of the problem (see [1] for a recent overview). For instance, we have seen the development of 3D displays [46] and stereoscopic systems that provide the user with the possibility to see the scene in three-dimensions. Other methods focus on immersive technologies that give the user the freedom to change the viewpoint for instance with a mouse or a joystick. Commercial applications of these systems include environment browsing such as museums, tourist attractions, hotel lobbies, sports events and so on. These applications are exciting and bring us one step further towards the ultimate 'being there' experience. Before we get there, there are some important issues which need to be solved. Obviously, such a capturing system requires images from multiple viewpoints. The resulting data flow puts great strain on the resources from data handling and processing to storage and transmission. It is therefore of prime importance to take advantage of the inherent redundancy that results when many cameras are looking at the same scene.

In this work, we focus on the freeviewpoint aspect of the problem. That is, to provide the user with certain degree of freedom of movement in the point from which the scene is viewed. There are essentially two different ways to deal with this problem. Traditional 3D graphics rendering systems create views of scenes using object models, textures and light sources. A model of the 3D world, usually in the form of a mesh, is available or estimated and new views are rendered by projecting the objects on an arbitrary viewpoint. This method is a *model-based* approach. While it can provide great viewing freedom, it is in general difficult to cope with cluttered natural scenes where a full geometric model is not always easy to obtain. It requires either to use range finding and scanning equipment or to resort to computer vision methods to estimate geometry. Despite recent progress in scene modeling from multiview images [63], it is still difficult to build 3D models of complicated environments.

Image based rendering (IBR) has appeared as an alternative to traditional graphics. This approach entails capturing the scene by taking many images from different viewing points and switching from one view to the other. Compared to a full geometric representation, images are easier to capture and are photorealistic by definition. New viewpoints are obtained simply by interpolating intensity values from nearby available images or light rays. In this case, the scene is represented not by the objects that constitute it but by the collection of light rays captured by the cameras. This approach is therefore *image-based* and can provide a very convincing rendering of real-world scenes without a full geometric model. There are, however, some challenges involved in image based rendering as well. Clearly, a smooth rendering requires taking an enormous amount of images which is difficult to capture, store and process. Interpolation enables one to use fewer images but is difficult in cluttered scenes due to occlusions, disocclusions and large depth variations. In these cases, artifact-free renderings require an excessively large number of images. Therefore, there is a need to devise an automatic algorithm that, by exploiting the inherent properties of multiview data, is able to interpolate viewpoints even in the presence of

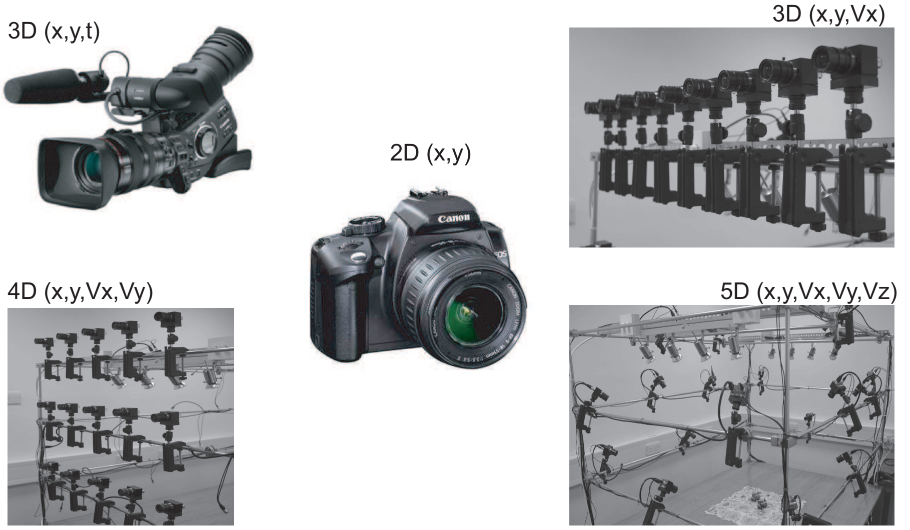


Figure 1.1: Capturing the plenoptic function. From the still image camera to the video camera or multiview imaging systems, all the sensing devices illustrated sample the plenoptic function with a varying number of degrees of freedom.

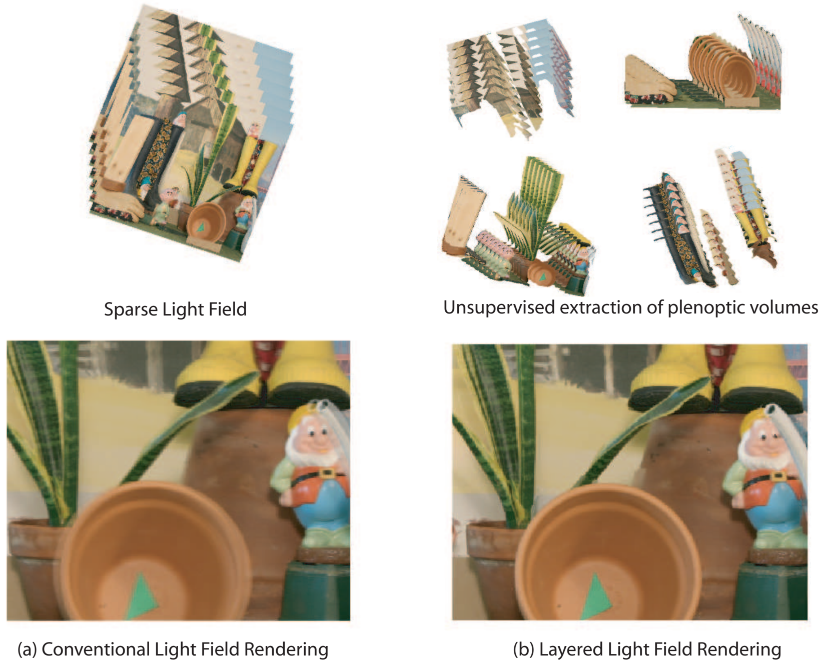
occlusions and large depth variations.

1.2 The plenoptic function and its sampling

At the heart of image based rendering is the characterization of visual information. The data acquired by multiple cameras from multiple viewpoints can be parameterized with a single function called the *plenoptic function*. It was first introduced by Adelson and Bergen [2] in an attempt to describe what one sees from an arbitrary viewpoint in space. Such a function requires seven dimensions in order to characterize all the free parameters. Indeed, three are needed for the position of the viewpoint (v_x, v_y, v_z), two for the viewing angle (θ_x, θ_y), one for the wavelength λ and finally one for the time t. In most cases, assumptions can be made to reduce the number of dimensions and different parameterizations have been proposed (e.g. [19, 36, 50, 55, 60, 84]). Indeed, there are many different ways to capture the plenoptic function and most of the popular sensing setups, some of which are illustrated in Figure 1.1, do not necessarily sample all the dimensions. From its introduction by Levoy and Hanrahan in the mid nineties, the four-dimensional light field parameterization [50] has benefited from a huge popularity thanks to the highly structured nature of the data. In this case, the plenoptic function is sampled with a uniformly distributed two-dimensional camera array. Several of such arrays have already been developed [19, 79, 81, 86] demonstrating the technical feasibility. The problem of freeviewpoint imaging, in this context, is the problem of interpolating light fields.

As the problem of image based rendering is in essence a sampling and interpolation problem, the spectral properties of light fields have been extensively studied [18, 83, 84]. In these papers, it is shown in various ways that the plenoptic function is approximately bandlimited. Moreover, it is shown that the width of the band depends mainly on the depth variations in the scene. Such an assumption means that sampling and interpolation in a traditional Shannon sense is possible. However, there are many reasons why the band is in fact not limited. Indeed, scenes are often made of different objects each of which has different textures. The boundaries of these objects are discontinuities which cause the spectrum to be unlimited. Moreover, it can be shown that even in the absence of occlusions and bandlimited textures, the resulting plenoptic function is not necessarily bandlimited [26]. These spectral based methods therefore have limitations since they are adapted to scenes with small depth variations and no occlusions or require very densely sampled data in order to reduce aliasing.

New sampling schemes have recently been developed that are able to cope with non-bandlimited signals. In particular, sampling schemes have recently emerged that are capable of sampling and perfectly reconstructing signals that follow piecewise models [28, 29, 75]. Using annihilating filter theory and signal moments, these schemes are able to cope with parametric signals such as Dirac impulses and piecewise polynomial signals in the one-dimensional [28, 29, 75] and multi-dimensional [68] cases. Some of these results have been applied to the sampling and interpolation of parametric plenoptic functions [22, 34]. However, the classes of signals that are recoverable remain limited. In this context, we will show in Chapter 6 that more general parametric signals such as piecewise sinusoidal and combinations of piecewise sinusoids and polynomials can be exactly recovered using similar principles. This leads to more general classes of parametric plenoptic functions that can be perfectly reconstructed from their sampled versions.



Sparse Light Field

Unsupervised extraction of plenoptic volumes

(a) Conventional Light Field Rendering

(b) Layered Light Field Rendering

Figure 1.2: (a) Conventional light field rendering. (b) Light field interpolation with a layered representation.

1.3 Interpolation by segmentation of multiview images

In conventional light field rendering, the data is very densely sampled (e.g. hundreds of images) and classical linear interpolation is enough to render views with little or no aliasing. However, more geometric information is needed when the data is sparsely sampled and occlusions occur (see for example the blurring in Figure 1.2(a)). Some IBR methods use complex geometry for rendering single objects [14, 36]. However, these models are sometimes difficult to obtain in scenes containing numerous objects and occlusions. As shown in [18, 83, 84], the main culprits causing artifacts in the interpolated images are depth variations and occlusions. In this light, Shum et al. [69] showed that very good quality renderings can be achieved by segmenting the light fields into approximately planar regions. In particular, the authors decompose the light field into coherent layers (also known as IBR objects [33]) that are represented with a collection of corresponding layers in each of the input images. This approximate decomposition enables one to interpolate viewpoints without aliasing (see Figure 1.2(b)). Indeed, these layers capture the coherence of the plenoptic function and make occlusion events explicit. Their extraction is therefore a very useful step in numerous multiview imaging applications including not only image based rendering [18,84], but also object-based compression [33] and disparity compensated and shape adaptive wavelet coding [21]. Other applications include scene interpretation and understanding [45]. All these applications make it very attractive to develop methods that are able to extract such regions.

It is worth mentioning that layered representations were first introduced by Wang and Adelson [78] for video coding purposes. Many dense stereo reconstruction algorithms such as [3, 65] and later [51, 80, 89] are also based on layered representations. These methods use two or more frames and differ from [24, 33, 69] and our approach in that they construct a model-based representation of the scene in a reference view. That is, the scene is modeled by a collection of layers each of which is characterized by its planar model, texture and spatial support. New views are obtained by warping the layers and their textures. Rendering new images with such a representation suffers from several disadvantages. For example, the warping of layers onto the reference view and then onto the novel view involves two resamplings which can reduce the quality of the reconstructed image.

The segmentation of light fields into continuous and approximately planar regions is in general a hard and ill-posed problem. In [69] and extensions such as [33], the segmentation is achieved using a semi-manual approach which enables a very good accuracy. However, this method requires human intervention and is a time consuming approach (up to a few hours [69]). In unsupervised methods, the segmentation is usually obtained by initializing a set of regions and using an iterative method that converges towards the desired partitioning. Some layer extraction methods include k-means clustering [78] and linear subspace approaches [44]. Other common methodologies use graph-based methods such as Graph Cuts [13, 51, 80]. Alternative approaches such as active contours [43] are based on the computation of the gradient of an energy functional and use a steepest descent methodology to converge towards a minimum. These methods have several advantages in particular dimensional scalability which is an attractive trait when dealing with the high dimensional plenoptic function.

The segmentation of light fields is somewhat related to video segmentation. Indeed, in both cases, the problem is to segment the plenoptic function albeit with different degrees of freedom. Some methods are based on a two-frame analysis [40, 41, 53]. More recently, authors have suggested the use of a larger number of frames in order to impose coherence throughout the video data. That is, to treat the segmentation problem as a three-dimensional one. For example, Mitiche et al. [56] tackled motion detection in videos with a three-dimensional active contour. Later, Ristivojevic and Konrad [58] considered the extraction of the tunnels carved out by layers in videos which enables one to take into account long term effects such as occlusions. A recent survey of these methods was presented in [45].

Similar ideas have emerged in multiview imaging, in particular, by studying the well-structured epipolar plane image (EPI) [12]. In that paper, Bolles et al. looked at the trajectories followed by points throughout many views in order to estimate depth in the scene. This analysis was further extended by Feldmann et al. [31]. However, both these methods generate a sparse or incomplete depth map. The problem of dense segmentation was studied by Criminisi et al. under the name of EPI-tube extraction [24] where collections of trajectories are gathered in order to generate a dense layer segmentation. The analysis is performed in a two-dimensional fashion by analyzing slices of the data. In all these methods it is emphasized that considering all the available images in a single function (i.e. the plenoptic function) allows for a more robust segmentation and generates a representation which is consistent in all the images. In this light, the key to our approach is to take advantage of the inherent structure and redundancy in the data. Such a method therefore requires to perform a dense four-dimensional segmentation of light fields, a problem that has remained largely unexplored.

1.4 Original contributions

As far as the author is aware, the following aspects of the thesis are believed to be original contributions:

- **Plenoptic hypervolumes** Layers in a scene carve out object tunnels [58] in videos and EPI-tubes [24] in multi-baseline stereo images. In extension, they carve out multi-dimensional hypervolumes in the plenoptic function. For example, a four-dimensional region is carved out in a light field. Following these concepts, we introduce plenoptic hypervolumes (a.k.a. plenoptic manifolds [6]) in Section 2.4. These are an image-based representation of a scene that decompose light fields into coherent regions. Following this concept, we look into methods to extract the whole volumes or hypervolumes carved out by approximately planar layers in light fields. While some similar methods for three-dimensional video analysis and EPI volumes have been developed (e.g. [24, 58]), no full coherent three or four-dimensional unsupervised segmentation scheme has been explored.

- **Semi-parametric variational framework for segmenting light fields** Variational frameworks including the classical active contour and level-set methods extend naturally to any number of dimensions. A first approach to extract plenoptic hypervolumes in light fields is to use a classical four-dimensional active hypersurface. However, the structure of a light field is such that points in space are mapped onto particular four-dimensional trajectories. Moreover occlusions occur in a well-defined manner. Therefore, in Sections 4.4 and 4.5, we impose these two constraints and modify the classical evolution equations. This leads to a new semi-parametric evolution approach. In Chapter 5, we show the validity of the proposed approach by analysing many different natural light fields. Moreover, we show that even though the plenoptic hypervolumes are extracted using simplified depth models, the objects still show their original shapes in the reconstructed images and aliasing-free rendering is achieved.

- **New sampling schemes for parametric non-bandlimited signals** Recent sampling schemes have shown that it is possible to sample and perfectly reconstruct signals that are not bandlimited but characterized by a finite number of degrees of freedom per unit of time. Moreover, it is possible to do so with physically realizable finite support sampling kernels. Current methods provide answers for Diracs and piecewise polynomial signals. In Sections 6.5 and 6.6, we extend these results to oscillating functions and in particular piecewise sinusoidal signals and combinations of piecewise sinusoidal and polynomial signals. Such signals are notoriously difficult to reconstruct since they are concentrated both in time and frequency. However, we show that under certain conditions on the sampling kernels, they can be sampled and perfectly reconstructed. Interestingly, the method described in these sections is able to perfectly reconstruct piecewise sinusoidal signals with arbitrarily high frequencies and arbitrarily close discontinuities given certain conditions on the number of sines and discontinuities. We show that these new schemes can be applied to perfectly reconstruct certain classes of parametric plenoptic functions.

The work presented in this thesis has led to the following publications:

- J. Berent, P. L. Dragotti, and T. Blu, "Sampling piecewise sinusoidal signals with finite rate of innovation methods," to be submitted, IEEE Transactions on Signal Processing, 2008.
- J. Berent and P.L. Dragotti, "Plenoptic manifolds: exploiting structure and coherence in multiview images," IEEE Signal Processing Magazine, vol 24, no. 7, pp. 34-44, November 2007.
- J. Berent and P.L. Dragotti, "Unsupervised extraction of coherent regions for image based rendering," Proceedings of British Machine Vision Conference (BMVC'07), Warwick, UK, September 10-13, 2007.
- J. Berent and P. L. Dragotti, "Segmentation of epipolar-plane image volumes with occlusion and disocclusion competition," Proceedings of IEEE International Workshop on Multimedia Signal Processing (MMSP'06), Victoria, Canada, October 3-6, 2006.

- J. Berent and P. L. Dragotti, "Perfect reconstruction schemes for sampling piecewise sinusoidal signals," Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'06), Toulouse, France, May 14-19, 2006, pp. 377-380.
- J. Berent and P. L. Dragotti, "Efficient representation and segmentation of multiview images," BMVA symposium on 3D Video-Analysis, Display and Applications, London, UK, February 6, 2008.

1.5 Thesis outline

The thesis is structured as follows:

In Chapter 2, we study the structure of the plenoptic function and discuss its sampling and interpolation. More precisely, we start by using the pinhole camera model and discuss how points are mapped onto the plenoptic domain. In doing so, we start by three-dimensional plenoptic functions such as videos and EPI-volumes. We then follow by describing in more detail the structure of the four-dimensional light fields. We then discuss the sampling and interpolation of the plenoptic function and light fields in particular. This analysis is done both in the spatial and spectral domains. Finally, we discuss model and image-based layered representations and introduce the concept of plenoptic hypervolumes.

In Chapter 3, we describe image segmentation using the deformable model (or active contour) method. In particular, we describe the variational formulations that are based on boundary or region information. For both cases, we show how to derive the steepest descent of the error function in order to converge towards the desired segmentation. Common descriptors for dissimilarity and similarity segmentation schemes are also presented. Finally, we discuss the issues of implementing the partial differential equations that govern the evolution of active contours and present the popular level-set method.

Chapter 4 applies the active contour methodology to light fields in four dimensions. However, instead of doing so in a straightforward manner, we study how the structure of light fields can be imposed onto the evolution of the deformable models. In particular, this chapter shows how to constrain the shape of the evolving hypersurface such that the structure is consistent with that of light fields. The resulting framework is a semi-parametric approach. Initializations of the method are also discussed and the algorithm is summarized.

Chapter 5 shows the validity of the proposed approach with captured natural light fields. We describe all the parameters of the scheme and quantify how they were set for the illustrated experiments. A variety of segmentation results along with interpolated viewpoints are shown. Finally, the chapter also shows the usefulness of the plenoptic hypervolume extraction scheme for other applications such as occlusion removal and augmented reality.

Chapter 6 deals with the problem of sampling and interpolation of the plenoptic function in a more theoretical manner. That is, new sampling schemes based on finite rate of innovation have recently been developed that are able to sample and perfectly reconstruct signals that are not bandlimited but characterized by a finite number of parameters. The chapter starts by introducing current finite rate of innovations methods that use compact support sampling kernels. We then propose two new sampling schemes that are able to perfectly reconstruct piecewise sinusoidal signals. Some interesting extensions are also shown.

Finally, Chapter 7 concludes the thesis with a summary of the achievements and a presentation of possible directions for future research.

Chapter 2

Introduction to the plenoptic function and its interpolation

2.1 Introduction

The plenoptic function [2] depends on seven variables namely the viewing position (v_x, v_y, v_z), the viewing direction (θ_x, θ_y), the wavelength λ and the time t if dynamic scenes are considered. It is therefore written as $I = I_7(\theta_x, \theta_y, \lambda, t, v_x, v_y, v_z)$. In practice, the plenoptic function is usually represented with the Cartesian coordinates used in numerous computer vision and image processing algorithms. It therefore becomes

$$I = I_7(x, y, \lambda, t, v_x, v_y, v_z), \tag{2.1}$$

where x and y are analogous to the coordinates on the image plane. These parameters are illustrated in Figure 2.1. When a camera captures an image of a scene, it is effectively taking a sample of the plenoptic function.

It is far from trivial to deal with all the dimensions of the plenoptic function. Indeed, the seven dimensions make it difficult to derive mathematical properties. They also lead to a huge number of images that need to be captured in order to sample all the dimensions. In an attempt to simplify the problem, most of the work on IBR makes assumptions to reduce the dimensionality. These assumptions include dropping the wavelength, con-

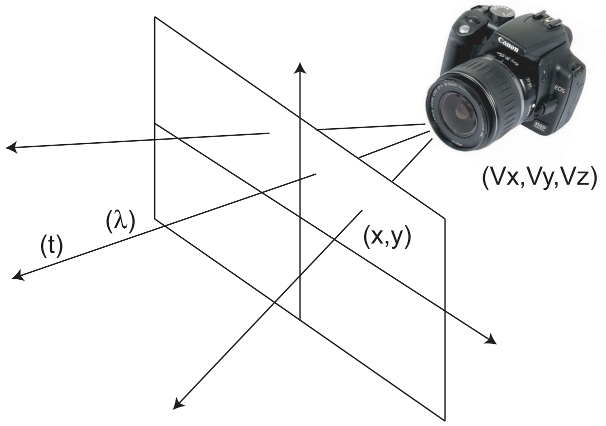


Figure 2.1: The plenoptic function. The intensity impinging on a point in space (v_x, v_y, v_z) depends on the viewing direction (x, y), the time t and the wavelength λ.

sidering static scenes or constraining the camera locations (or viewing space). The surface plenoptic function [84] introduced by Zhang and Chen assumes that air is transparent and therefore the intensity along a light ray remains constant unless it is occluded. This enables them to drop one dimension. McMillan and Bishop introduced plenoptic modeling [55] where the wavelength is omitted and static scenes are considered. This reduced their parameterization to five dimensions. Coupling the assumptions of both methods leads to the four-dimensional light field parameterization [50] introduced by Levoy and Hanrahan. Further restricting the camera locations to a line results in the three-dimensional epipolar plane image (EPI) volume [12]. Finally, image based rendering using a fixed camera center is known as image mosaicing or panoramas of which Quicktime VR [23] is a good example.

Perhaps the most widespread and practical representation of the plenoptic function is the light field [50] also known as the lumigraph [36] or the ray space [32]. In the seminal work of Levoy and Hanrahan [50], the interpolation of the plenoptic function is done using a large number of images and no geometric information about the scene. The interpolated images are obtained by classical linear interpolation of the nearby available ones. This enables a photorealistic rendering of complicated environments without modeling objects. The drawback is the necessity of huge amounts of data which can be impractical. This raises several interesting questions such as: How many images or viewpoints are required

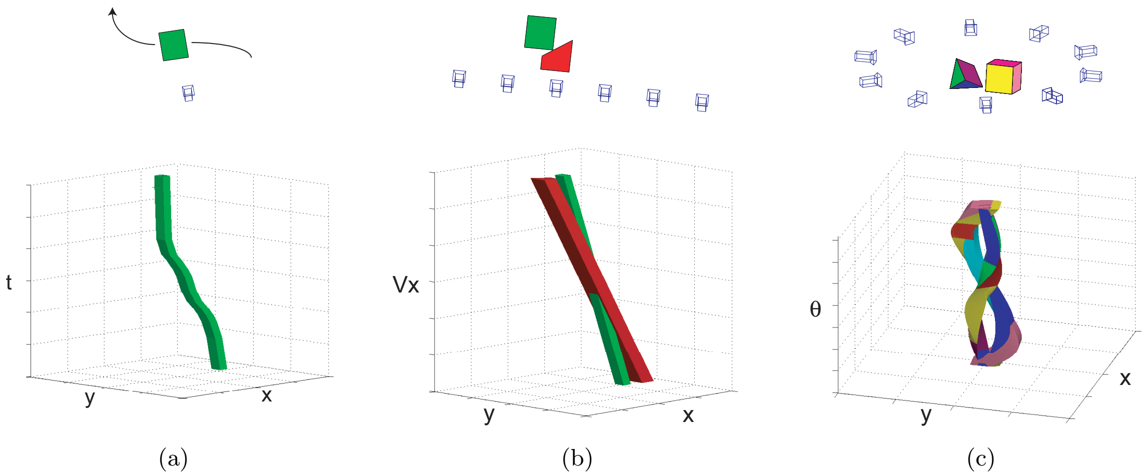


Figure 2.2: Figure 2.2(a) shows the trajectories carved out by a flat object in the space-time volume. Figure 2.2(b) illustrates the trajectories carved out by two flat objects in the case of a linear camera array where v_x denotes the position of the cameras along a line. Figure 2.2(c) shows the trajectories generated by two objects in the case of a circular camera array where θ denotes the angle of the camera position around the circle. Note that the structure in Figure 2.2(a) depends on the movement of the objects which means it is not necessarily predefined. In both the other cases (linear and circular still-image camera arrays), the structure is constrained by the camera setup and occlusion events can be predicted.

for a good quality rendering? And can we use some approximate geometrical information to enhance the rendering quality? Many pioneering works have provided answers to these questions some of which are the object of study in this chapter.

Several authors have proposed comprehensive surveys on the plenoptic function and IBR methods (see for instance [61, 85] and more recently [48]). In this chapter, we look into the structure of the plenoptic function in Section 2.2 and Section 2.3 discusses its sampling and interpolation in the spatial and spectral domains. Section 2.4 discusses geometric representations with a particular emphasis on the differences between model-based and image-based layered representations. In doing so this section introduces the concept of plenoptic hypervolumes. Finally, Section 2.5 presents a summary of the chapter and highlights some of the important points relevant to the remainder of the thesis.

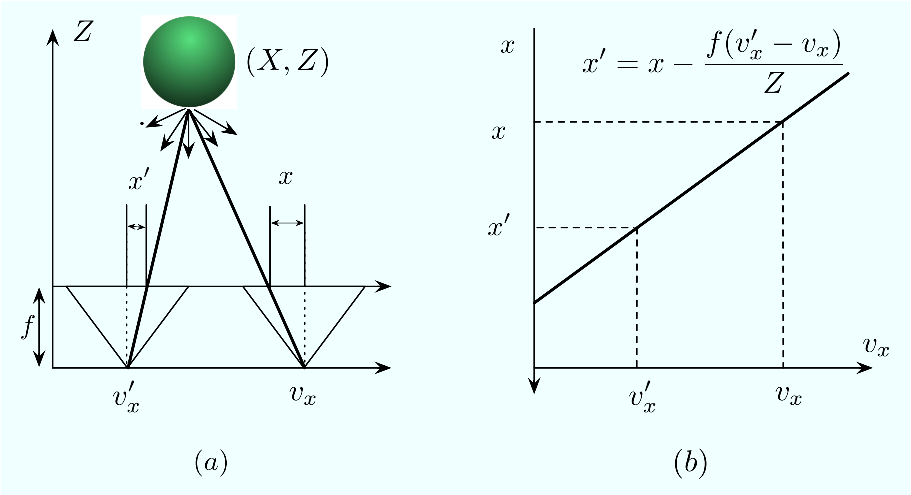


Figure 2.3: (a) Given the depth Z at which the light ray in (x', v'_x) intersects with the object, it is possible to find the corresponding light ray in (x, v_x). (b) Writing x as a function of v_x leads to the equation of a line with slope inversely proportional to the depth. This (x, v_x) space is the epipolar plane image.

2.2 Plenoptic structures

The vast majority of works on image based rendering assume the pinhole camera model.[1] The model says that points in the world coordinates $\vec{X} = (X, Y, Z)$ are mapped onto the image plane (x, y) in the point where the line connecting $\vec{X}$ and the camera center intersects with the image plane [38]. The focal length f measures the distance separating the camera center and the image plane. Using similar triangles, it can be shown that the mapping is given by

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} fX/Z \\ fY/Z \end{pmatrix},$$

where we assume that the principal point is located at the origin. Consider now the case of the video camera. The point in space $\vec{X}$ is free to move in time and its mapping onto

[1] There are some exceptions such as the work of Kubota et al. [47].

the video data becomes

$$\begin{pmatrix} X(t) \\ Y(t) \\ Z(t) \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ t \end{pmatrix} = \begin{pmatrix} fX(t)/Z(t) \\ fY(t)/Z(t) \\ t \end{pmatrix},$$

which is the parameterization of a trajectory in the 3D plenoptic domain. Note that the intensity along this trajectory remains fairly constant if the radiance of the point does not change in time. We can therefore write

$$I(x', y', t') = I(x' - p_x(t), y' - p_y(t), t),$$

where $p_x(t)$ and $p_y(t)$ represent the motion of the point from time t to time t'. Assuming the scene is made of moving objects, neighboring points in space will generate similar neighboring trajectories in the video data (see Figure 2.2(a)). Hence, apart from the object boundaries, the information captured varies mainly in a smooth fashion. Note that in the general case, the trajectories do not have much structure. Indeed, there is no real prior constraining the shape of the trajectory unless some assumptions are made on the movement and the rigidity of the objects. Nevertheless, in natural videos, assuming a certain degree of smoothness and temporal coherence is usually a valid assumption [45].

Let us now give one degree of freedom to the position of the camera (e.g. in v_x) instead of the time dimension. In this case, the plenoptic function reduces to the epipolar plane image (EPI) volume [12]. It can be acquired either by translating a camera along a rail or by a linear camera array. According to the pinhole camera model, points in real-world coordinates are mapped onto the EPI volume as a function of v_x according to

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ v_x \end{pmatrix} = \begin{pmatrix} fX/Z - fv_x/Z \\ fY/Z \\ v_x \end{pmatrix},$$

where we notice that a point in space generates a line. Given the depth Z at which the light ray (x, y, v_x) intersects with the object in (X, Y, Z), it is possible to find the corresponding ray in (x', y', v'_x) which intersects the same 3D point (see Figure 2.3(a)). Again using similar triangles, it is straightforward to derive the relations

$$\begin{aligned} x' &= x - \frac{f(v'_x - v_x)}{Z} \\ y' &= y, \end{aligned}$$

which shows that points are shifted by distance depending on the depth Z and the viewpoint change $(v'_x - v_x)$. This shift is generally referred to as the disparity. Assuming Lambertian[2] surfaces and no occlusions, the radiance emanating from the point (X, Y, Z) is viewed from any location v_x with the same intensity. We can therefore write

$$I(x', y', v'_x) = I(x' - \frac{f(v_x - v'_x)}{Z}, y, v_x),$$

which means that the intensity along the line in the EPI remains constant. Furthermore, the slope of the line is inversely proportional to the depth of the point Z. Therefore, the data in this parameterization, as opposed to the video, has a very particular structure which is noticeable in Figure 2.2(b). The occurrence of occlusions, for example, is predictable since a line with a larger slope will always occlude a line with a smaller slope. This property follows naturally from the fact that points closer to the image plane will occlude points that are further away. The example illustrated in Figure 2.4 portrays this property with natural images. Note that the concept of EPI analysis is not necessarily restricted to the case of cameras placed along a line and has been extended by Feldmann et. al [31] with the Image Cube Trajectories (ICT). They show that other one-dimensional camera setups such as the circular case illustrated in Figure 2.2(c) generate particular trajectories in the plenoptic domain and occlusion compatible orders can be defined.

The light field is a four-dimensional parameterization of the plenoptic function $I(x, y, v_x, v_y)$ in which the viewpoints are limited to a bounding box. Light rays are most commonly parameterized by their intersection with two planes namely the image plane (x, y) and the camera plane (v_x, v_y) as illustrated in Figures 2.5 and 2.6. In its traditional form, this data is captured by a planar camera array although other setups are possible

[2] A Lambertian surface is such that the light is reflected with the same intensity in all directions.

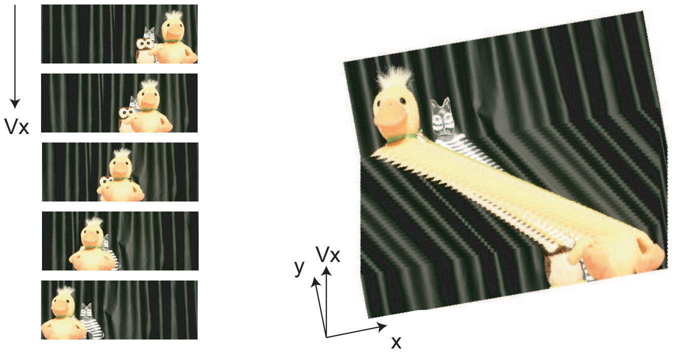


Figure 2.4: The Epipolar Plane Image (EPI) volume. Cameras are constrained to a line resulting in a 3D plenoptic function where x and y are the image coordinates and v_x denotes the position of the camera. Points in space are projected onto lines where the slope of the line is inversely proportional to the depth of the point. The volume illustrated is sliced in order to show the particular structure of the data.

including unstructured arrays [14]. Note that a three-dimensional (x, y, v_x) slice of the light field is equivalent to the epipolar plane image volume shown in Figure 2.4. This case is also sometimes referred to as the simplified light field [19]. In the 4D case, the camera is free to move on a plane. A point in space is therefore mapped onto

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = \begin{pmatrix} X/Z - fv_x/Z \\ Y/Z - fv_y/Z \\ v_x \\ v_y \end{pmatrix}, \tag{2.2}$$

which is a four-dimensional line as a function of v_x and v_y. Given the depth Z at which the light ray (x, y, v_x, v_y) intersects with the object in $\vec{X}$, it is possible to find the corresponding ray (x', y', v'_x, v'_y) which intersects the same 3D point. By extension of the EPI volume, we have the relations

$$\begin{aligned} x' &= x - \frac{f(v'_x - v_x)}{Z} \\ y' &= y - \frac{f(v'_y - v_y)}{Z}, \end{aligned}$$

which shows that points are shifted by distance depending on the depth Z and the viewpoint change $(v'_x - v_x, v'_y - v_y)$. Again, assuming Lambertian surfaces and no occlusions,

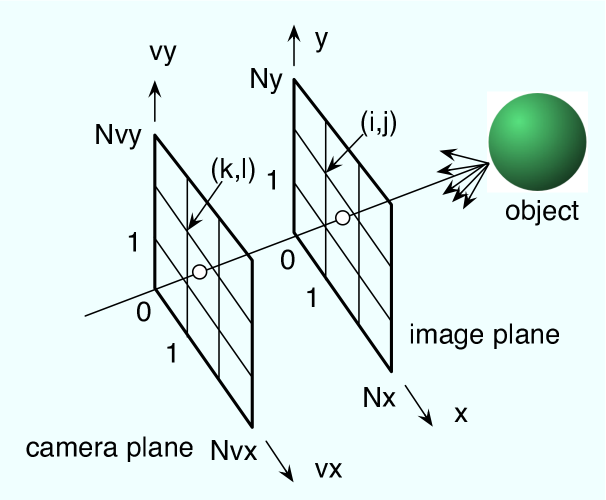


Figure 2.5: One parameterization of the light field. Each light ray is uniquely parameterized with four dimensions namely its intersection with the camera plane (v_x, v_y) and the image plane (x, y). The discretization of these two planes are indexed by (k, l) and (i, j) for the camera and image planes respectively.

the radiance emanating from the point $\vec{X}$ is viewed from any location (v_x, v_y) with the same intensity. We can therefore write

$$I(x', y', v_x', v_y') = I(x' - \frac{f(v_x - v_x')}{Z}, y' - \frac{f(v_y - v_y')}{Z}, v_x, v_y), \tag{2.3}$$

which means that the intensity along the line in the light field remains constant. The light field therefore also has a very particular structure which is a four-dimensional extension of the EPI volume.

2.3 Sampling and interpolation of the plenoptic function

In practice, the plenoptic function is captured by a finite number of cameras having a finite resolution and a finite number of frames if the time dimension is captured. Usually, the sampling periods for the image and the time dimensions are much smaller than the camera location periods. Therefore, we will focus on the sampling and interpolation in these dimensions and in particular using the light field representation. This will be the case for the remainder of this chapter and indeed the remainder of the thesis.

Using a planar camera array, we have access to the sampled version of the light field

$$I[i,j,k,l] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(\vec{x})\varphi(x - iT_x, y - jT_y, v_x - kT_{v_x}, v_y - lT_{v_y})d\vec{x},$$

where $\vec{x} = (x, y, v_x, v_y)$, φ is the sampling kernel, $\{i, j, k, l\} \in \mathbb{Z}$ are the sample points and $\{T_x, T_y, T_{v_x}, T_{v_y}\} \in \mathbb{R}^+$ are the sampling periods for the x, y, v_x and v_y dimensions respectively. The rendering of a new view is obtained by interpolating the sampled light field $I[i, j, k, l]$. The interpolated value $\tilde{I}(\vec{x})$ is computed using a classical interpolation framework in four dimensions. We therefore have

$$\tilde{I}(\vec{x}) = \sum_{i=0}^{N_x}\sum_{j=0}^{N_y}\sum_{k=0}^{N_{v_x}}\sum_{l=0}^{N_{v_y}} I[i,j,k,l]\psi(x - iT_x, y - jT_y, v_x - kT_{v_x}, v_y - lT_{v_y}), \tag{2.4}$$

where $\psi(\vec{x})$ is the interpolation kernel (or basis function). For example, one might choose a basis function that is one on the nearest sample point to $\vec{x}$ and zero elsewhere. Such an interpolation will generate a piecewise constant light field effectively acting as a nearest neighbor method. Alternatively one might choose a quadrilinear basis function that is one on the sample point and linearly drops down to zero at all neighboring points. The interpolated value is thus computed from the 16 neighboring light rays. Such a rendering is simple, computationally efficient and independent of scene complexity. However a huge number of images is necessary for rendering viewpoints free of artifacts. Indeed, it is common in light field rendering to use hundreds and even thousands of images [50]. The use of an undersampled light field will result in blurring and ghosting effects. These effects, of which an example is illustrated in Figure 2.7(a), are caused by correspondence mismatches due to the larger viewpoint changes in between sample images. They can also be interpreted as a form of aliasing due to undersampled data. In the lumigraph [36], these effects are reduced by reconstructing a rough geometry of the scene. This geometry then drives the choice of the interpolation kernel which will be designed to correct for depth. In the following sections, we discuss these issues in the spatial and spectral domains.

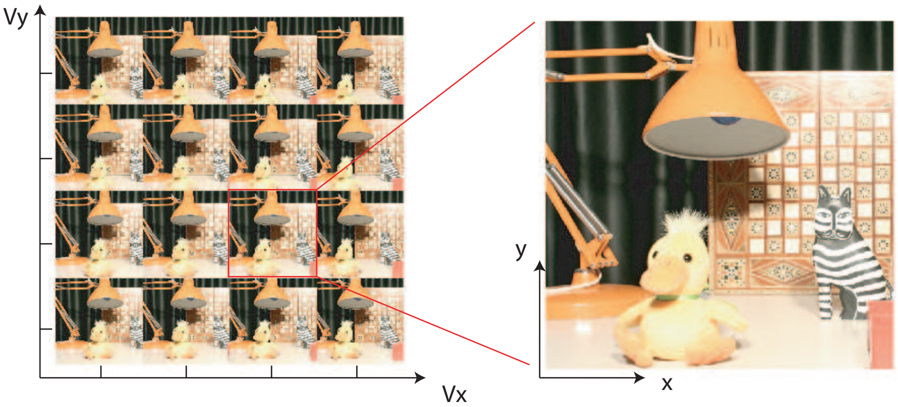


Figure 2.6: A sparse (4x4) light field image array. Light rays are parameterized with the image coordinates (x, y) and the camera location (v_x, v_y).

2.3.1 Spatial analysis

In Section 2.2, we emphasized that the light field has a particular structure. It is made of lines with slopes depending on the geometry of the objects in the scene. This geometry, if available, can be used to adapt the interpolation kernel according to the depth [18,36,39]. The new interpolation kernel $\psi'(x, y, v_x, v_y)$ is obtained by computing the corresponding points in neighboring images with (2.3) and using the previous interpolation in (2.4) which gives

$$\begin{aligned} &\psi'(x - iT_x, y - jT_y, v_x - kT_{v_x}, v_y - lT_{v_y}) = \\ &\quad \psi(x - iT_x - \frac{f(kT_{v_x} - v_x)}{Z}, y - jT_y - \frac{f(lT_{v_y} - v_y)}{Z}, v_x - kT_{v_x}, v_y - lT_{v_y}). \end{aligned} \tag{2.5}$$

Such a method is equivalent to interpolating along a certain direction in the EPI. Figure 2.8 illustrates the sample points used in classical linear interpolation with squares and the depth corrected sample points are shown with triangles. Note that the depth corrected basis function ψ' reduces to ψ (i.e. no depth correction) when posing $Z = \infty$.

Clearly, the use of the depth corrected interpolation kernel requires a continuous geometric model of the scene. For this purpose, one may reconstruct a relatively accurate proxy for a single object as is the case for instance in [14,36]. Algorithms to estimate this

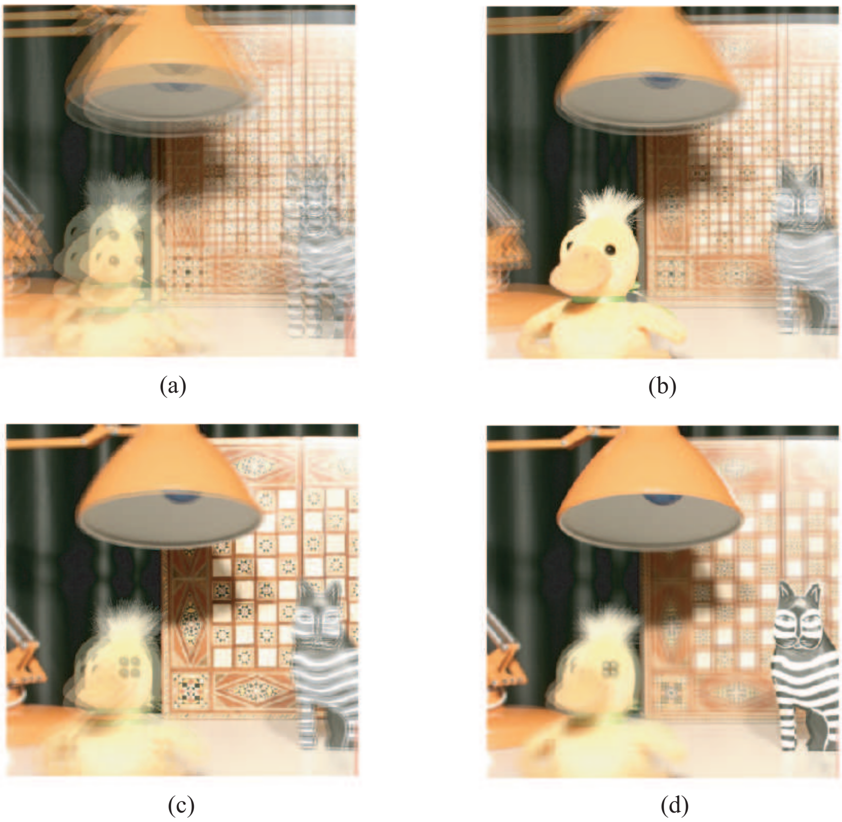


Figure 2.7: Ghosting effects caused by undersampled light field rendering. Figure 2.7 (a) illustrates the result using light field rendering with no depth information. Figures 2.7 (b-c) illustrate the effect of moving the rendering depth from the minimum to the maximum depths respectively. Figure 2.7 (d) shows the result obtained using the optimal rendering depth in [18].

geometry include space carving [49] and variational methods [30]. However, this estimation is a difficult task in cluttered scenes due to the presence of occlusions. In many works such as [18, 39], a single plane is used as a proxy. Figures 2.7(b-d) illustrate interpolated images using planar geometric proxies at different depths. From these images, it is clear that the objects situated at a distance close to the rendering depth appear in focus. The data is interpolated from sample values that originate from the same 3D points. However, objects that are far from the rendering depth appear blurred and in double images. This effect is due to the fact that the interpolation is done with points that do not correspond to the real objects as illustrated in Figure 2.9. Another phenomenon that produces similar problems is the occlusion which is also illustrated in the same figure. The effect of moving

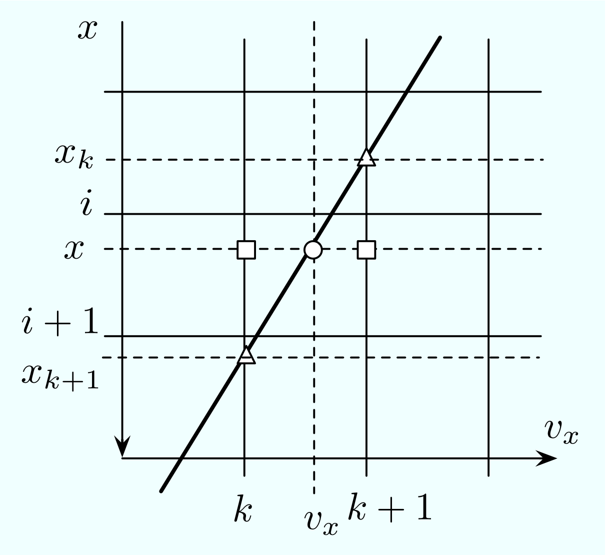


Figure 2.8: Depth corrected linear interpolation. Knowing the depth Z of the point to interpolate in image v_x, it is possible to find the corresponding light rays in the available sample images in k and $k+1$. The squares show the points used for classical linear interpolation and the triangles show the points used by the depth corrected interpolation kernel.

the rendering depth along with the use of different interpolation filters has been studied in detail in [18,39].

There are several conclusions to be drawn from the discussion above. Objects situated far from the camera plane require less depth correction since the shift, proportional to Z^{-1}, tends to zero. Second, a very highly sampled camera plane also requires little or no depth correction since the terms $kT_{v_x} - v_x$ and $lT_{v_y} - v_y$ tend to zero. Inversely, large depth variations and sparsely sampled camera arrays require some form of depth correction. Finally, the knowledge of occlusions is useful to avoid interpolating points from intensities that belong to two different objects in the scene.

2.3.2 Spectral analysis

In the previous section, we looked into the spatial properties of light fields in order to interpolate new view points. The sampling and interpolation of light fields can also be studied using classical sampling theory. That is, by computing the spectral support of the function and determining the sampling frequency necessary for an aliasing-free interpola-

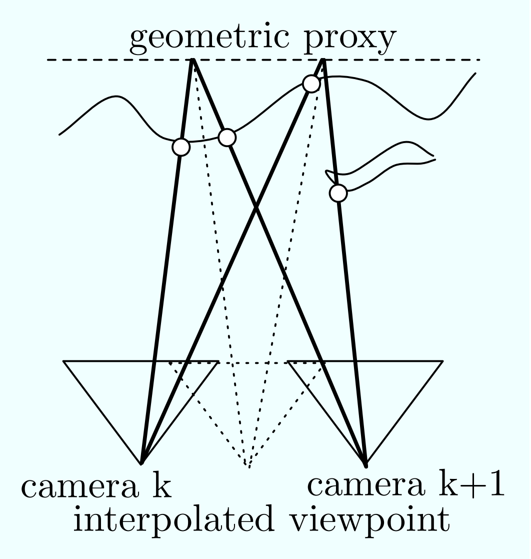


Figure 2.9: View interpolation with a single depth plane as geometric proxy. Points that are far from the rendering depth will appear blurred and in double images since the interpolation is done with different points in the scene. Occlusions have a similar effect.

tion. Clearly, the spectrum of the light field data will depend on the sampling rate of the cameras and scene properties such as the reflectance of surfaces and occlusions. However, several pioneering works such as Chai et al. [18] and Zhang and Chen [84] enable to provide approximated answers.

The four-dimensional Fourier transform of a light field is given by

$$\hat{I}(\omega_x, \omega_y, \omega_{v_x}, \omega_{v_y}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\vec{x}) e^{-j(\omega_x x + \omega_y y + \omega_{v_x} v_x + \omega_{v_y} v_y)} dx dy dv_x dv_y,$$

where

$$I(x, y, v_x, v_y) = I(x + \frac{f v_x}{Z}, y + \frac{f v_y}{Z}, 0, 0),$$

since we assume that the scene is Lambertian and there are no occlusions. The image $I(x, y, 0, 0)$ is a reference image located in $v_x = v_y = 0$. As pointed out in [18], this computation is very complicated if we take into consideration the general case. However, some properties can be deduced from approximations. For instance, assume the scene has a constant depth $Z = Z_0$. The Fourier transform of this light field is thus given by

$$\begin{aligned}\hat{I}(\omega_x, \omega_y, \omega_{v_x}, \omega_{v_y}) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x, y, 0, 0)e^{-j(\omega_x x+\omega_y y)}dxdy \\ &\quad \int_{-\infty}^{\infty} e^{-j(\frac{f}{Z_0}\omega_x+\omega_{v_x})v_x}dv_x \int_{-\infty}^{\infty} e^{-j(\frac{f}{Z_0}\omega_y+\omega_{v_y})v_y}dv_y \\ &= 4\pi^2 \hat{I}_2(\omega_x, \omega_y)\delta(\frac{f}{Z_0}\omega_x + \omega_{v_x})\delta(\frac{f}{Z_0}\omega_y + \omega_{v_y}),\end{aligned}$$

where $\hat{I}_2(\omega_x, \omega_y)$ is the 2D Fourier transform of $I(x, y, 0, 0)$ and $\delta(x)$ is the Dirac distribution. This analysis enables the authors in [18] to draw several interesting conclusions. For simplicity, we consider only the projection of $\hat{I}(\omega_x, \omega_y, \omega_{v_x}, \omega_{v_y})$ onto the (ω_x, ω_{v_x}) plane and denote it by $\hat{I}(\omega_x, \omega_{v_x})$. The spectrum of the constant depth light field is supported on a line defined by $\frac{f}{Z_0}\omega_x + \omega_{v_x} = 0$. The spectral support of a scene with depth varying between Z_{min} and Z_{max} is therefore approximately bound by the two lines $\frac{f}{Z_{min}}\omega_x + \omega_{v_x} = 0$ and $\frac{f}{Z_{max}}\omega_x + \omega_{v_x} = 0$. These bounds are illustrated in Figure 2.10(a). Let the sampling kernel be a Dirac function such that the samples are given by

$$I[i, j, k, l] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(\vec{x})\delta(\frac{x}{T_x} - i)\delta(\frac{y}{T_y} - j)\delta(\frac{v_x}{T_{v_x}} - k)\delta(\frac{v_y}{T_{v_y}} - l)d\vec{x}.$$

The Fourier transform of the sampled light field becomes

$$\hat{I}(\omega_x, \omega_y, \omega_{v_x}, \omega_{v_y}) = \sum_{m\in\mathbb{Z}}\sum_{n\in\mathbb{Z}}\sum_{r\in\mathbb{Z}}\sum_{q\in\mathbb{Z}} \hat{I}(\omega_x - \frac{m}{T_x}, \omega_y - \frac{n}{T_y}, \omega_{v_x} - \frac{r}{T_{v_x}}, \omega_{v_y} - \frac{q}{T_{v_y}}),$$

which is a sum of shifted versions of the original Fourier transform $\hat{I}$ as illustrated in Figure 2.10(b). Aliasing occurs when the replicated versions of the spectrum overlap. Such an analysis suggests that the minimal overlap is obtained by using an interpolation filter that is adapted to the rendering depth [18]:

$$\frac{1}{Z_{opt}} = \frac{1}{2}(\frac{1}{Z_{min}} + \frac{1}{Z_{max}}). \tag{2.6}$$

Therefore some geometrical information about the scene (i.e. Z_{min} and Z_{max}) is beneficial. It also becomes clear that the area of the support in the frequency domain depends on the difference between the closest and the furthest points in the scene. A scene with

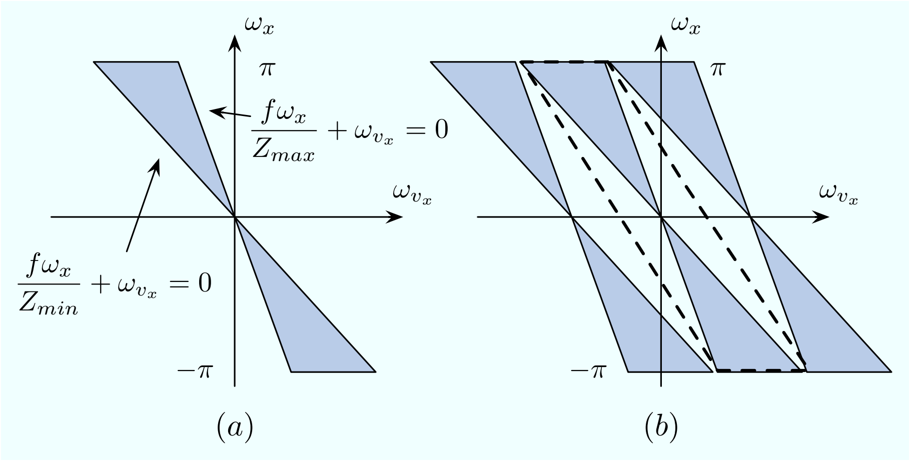

ω_x
π
$\frac{f\omega_x}{Z_{max}} + \omega_{v_x} = 0$
ω_{v_x}
$\frac{f\omega_x}{Z_{min}} + \omega_{v_x} = 0$
$-\pi$
(a)
ω_x
π
ω_{v_x}
$-\pi$
(b)

Figure 2.10: (a) The support of the spectrum of the 2D light field is approximately bound by the minimum and maximum depths in the scene. (b) The spectrum of the sampled light field contains replicated versions of the original one. The optimal reconstruction filter is skewed to the disparity $0.5(1/Z_{min} + 1/Z_{max})$.

large depth variations will therefore result in more aliasing. Moreover, it is possible to decompose the scene into a collection of approximately constant depth regions. Each region will have a small depth variation and can be individually rendered free of aliasing. Naturally this is an approximation since the decomposition of the scene into constant depth regions inherently implies a windowing which translates to the convolution with infinite support sinc functions in the frequency domain. Nevertheless a large part of the signal's energy lies in a bandlimited window. A formal analysis of the tradeoff between geometry in terms of the number of bits for depth and sampling density can be found in [18]. Finally, occlusions also imply a windowing. The spectrum of the light field will therefore be spread and more aliasing will occur.

2.4 Layered model and image-based representations

Both the spatial and spectral analysis presented above support the fact that some geometric information about the scene is beneficial for taking advantage of the coherence in plenoptic data and in particular for view interpolation. In this section, we review different ways to represent the geometry of a scene.

2.4.1 Model-based layers

In layered representations, the scene is characterized by multiple layers corresponding to different objects or parts of objects in a scene. Such a representation first appeared for videos in [78] where the data is represented as a collection of layers undergoing a parametric motion. Each layer is represented with its spatial support, its motion model (e.g. affine) and its texture. In [3], a similar representation is used for decomposing multiview data into a set of planar patches (or sprites). A per pixel depth residual is added to each layer in order to compensate for surfaces that are not planar. The different layers are extracted by warping images to the reference image according to a planar model. The layer is then segmented using a matching functional (squared error for instance). This layered representation is described on a single reference image. Such a representation suffers from the fact that rendering an image from a virtual viewpoint will most likely expose previously occluded areas. The rendered viewpoint will therefore have disocclusion artifacts.

The layered depth image (LDI) [65] representation allows for a multi-valued depth map with texture information for each depth value. It is obtained by merging information from multiple images to a single center of projection (see Figure 2.11(a)). This representation, along with the previous layer representation, suffers from the fact that texture needs to be resampled when it is warped onto reference images. Moreover, synthesized views are obtained by reprojecting the texture onto the desired viewpoint which involves a second resampling. Such manipulations can lead to blurring effects due to the resampling process. The LDI cube [52] partially solves the problem by effectively using multiple LDIs to represent the scene from different viewing positions. The LDI closest to the virtual viewing point is used which will minimize resampling problems.

2.4.2 Image-based layers and plenoptic hypervolumes

As we saw in the Section 2.2, points in space are mapped onto trajectories in the plenoptic function. Layers that are made of neighboring points in space are therefore mapped onto volumes (or more generally hypervolumes) that are made of neighboring trajectories. This

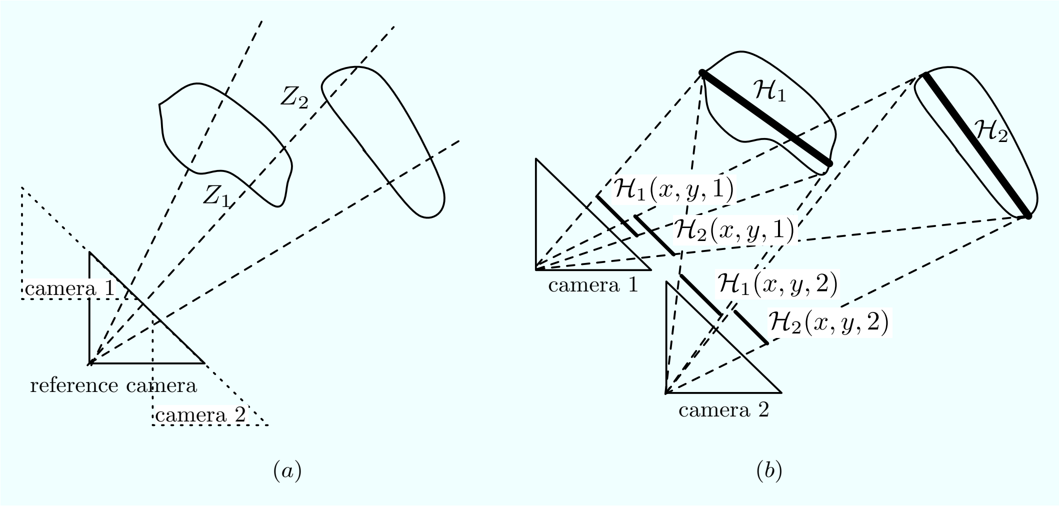


Figure 2.11: The layered depth image (LDI) in (a) represents the scene with a multi-valued depth map in a reference image. The plenoptic hypervolumes $\mathcal{H}_n$ in (b) are represented with a collection of corresponding regions $\mathcal{H}_n(x, y, k)$ in each of the input images k. The geometry of each plenoptic hypervolume is modeled with a simple plane.

collection of trajectories generates a multi-dimensional hypervolume $\mathcal{H}$ which we will call *plenoptic hypervolume* (also known as *plenoptic manifold* [6]). This concept can be seen as the generalization of the object tunnels [58] in videos and the EPI-tubes [24] in EPI volumes.

Definition 1. *A plenoptic hypervolume is defined by the region carved out by a layer in the plenoptic function.*

In contrast to the LDI, the plenoptic hypervolume representation does not use reference images. Contrary to the LDI that builds a geometric model of the scene, a plenoptic hypervolume segments the plenoptic function itself. Thanks to this, it uses simple geometric proxies such as planes and there is no need for an accurate per pixel depth. Note that the concept of plenoptic hypervolume shares many ideas with the coherent layers in [69] and the IBR objects in [33]. However, it is conceptually different in that it is a continuous representation of the regions carved out by layers in the plenoptic function. As illustrated in Figure 2.11(b), the intersection of the plenoptic hypervolume $\mathcal{H}$ with an image in k is the layer $\mathcal{H}(x, y, k)$ on that image.

There are two important elements to retain from the structure of the plenoptic function. First, the multi-dimensional plenoptic trajectories are constrained by the camera setup. This is illustrated by the way points in space are mapped onto the plenoptic domain. In the following, we will refer to this prior as the *geometry constraint*. Second, there is a well-defined occlusion ordering. Points at different depths generate different trajectories and will intersect in the event of an occlusion. The study of these trajectories determines which point will occlude the other. This prior will be referred to as the *occlusion constraint*. There are several benefits in considering the extraction of the whole hypervolume carved out by objects instead of building a model-based layered representation. In particular, recombining the plenoptic hypervolumes reconstructs exactly the original data as illustrated in Figure 2.12. The procedure enables a global vision of the problem and operates on the entire available data. That is, all the images are taken into account simultaneously and the segmentation is consistent throughout all the views which increases robustness.

In [69], the contours of layers are semi-manually extracted on one image and propagated to the other views (i.e. the two other dimensions v_x and v_y) using a user-defined depth map. By performing the segmentation in this manner, the coherence of the layers is enforced in all the views. Some authors have tackled the problem of unsupervised coherent region extraction. These methods are usually based on the analysis of the epipolar plane image. The depth is estimated by computing the variance along lines in the EPI. Occlusions are dealt with by using the fact that lines with larger slopes will occlude the lines with smaller slopes. In [24], horizontal slices of the EPI volume are analyzed in order to gather lines with similar slopes. Although the authors convert this segmentation to a LDI representation by using a single reference image for texture, this method effectively extracts plenoptic hypervolumes. As opposed to the method in [24] which analyses the data slice by slice, the method that we presented in [6,7] which we describe in more details in Chapter 4 is based on a four-dimensional framework. It therefore exploits coherence in four dimensions, that is, the whole stack of images is analyzed in a global manner.

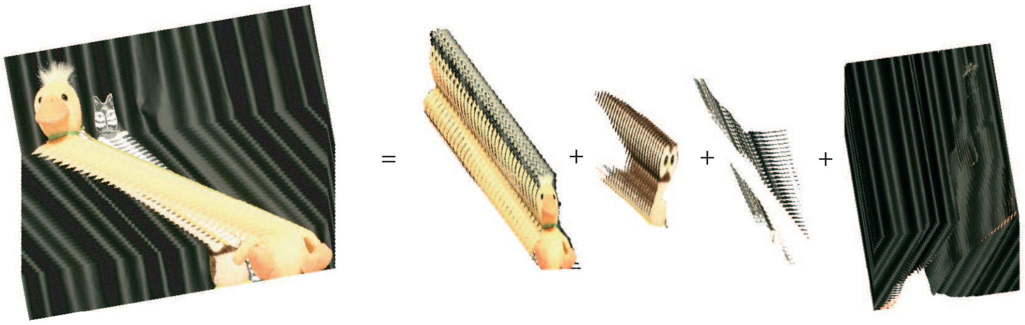


Figure 2.12: Decomposition of the plenoptic function (EPI volume in this example) into four plenoptic hypervolumes. When added together, the four extracted regions reconstruct exactly the original data.

2.5 Summary

Interpolating the plenoptic function (i.e. image based rendering) is an increasingly popular method for rendering photorealistic images from virtual viewpoints. The data is not represented by the 3D objects that constitute the scene but by the collection of light rays captured by the cameras (i.e. the plenoptic function). These light rays can be nicely parameterized in the form of the four-dimensional light field. Using this parameterization, new views are synthesized by classical interpolation (i.e. usually linear interpolation). Depending on the scene, the amount of images necessary for an aliasing free rendering is extremely large. In this chapter, we showed through spatial and spectral analysis that the artifacts caused by undersampled light fields are mainly due to large depth variations and occlusions. It is therefore beneficial to use some geometric information to drive the choice of the interpolation kernels. Obtaining an accurate 3D model of complicated and cluttered scenes is a difficult problem. However, the plenoptic function provides a nice framework for studying the data in a global manner and imposing a coherent analysis. Using this representation, we suggested that in extension to the object tunnels in videos and EPI-tubes in multi-baseline stereo data, objects carve multi-dimensional hypervolumes in the plenoptic function that we called plenoptic hypervolumes. Just like in the three-dimensional cases, the hypervolumes contain highly regular information since they are constructed with images of the same objects. There is therefore clearly potential for robust analysis and efficient representation.

The important aspects of the plenoptic function that emerged in this chapter are highlighted in the following points:

- The plenoptic function is a high dimensional function (seven dimensions in general, four dimensions in the case of the light field).
- The structure of the plenoptic function depends on the camera setup sampling it.
- Based on photoconsistency, the plenoptic function has a high degree of regularity and coherence.
- Layers in the scene carve out plenoptic hypervolumes in the plenoptic domain.

Chapter 3

Variational methods for multi-dimensional data segmentation

3.1 Introduction

The goal of segmentation is to partition the data into regions that are of particular interest. When dealing with images, these regions are usually different objects or parts of objects in a scene that need to be separated from a background. In automatic segmentation schemes, the regions of interest are differentiated from each other by some mathematical property. This is usually based on boundary information such as sharp intensity changes or on region information such as texture, spatial homogeneity and motion or disparity. One popular way to obtain this segmentation is to minimize an appropriate cost functional. In variational methods, this energy minimization is solved using partial differential equations (PDEs) to evolve a deformable model in the direction of the negative energy gradient, thus attracting it towards the region to segment.

These methods became very popular since the active contours (a.k.a. snakes) pioneered by Kass et al. [43]. They have been applied successfully in a number of image and video segmentation schemes [16,17,20,53,58], motion detection [41,42,56] and also multiview scene modeling [30,35] to name but a few. The popularity of the method comes from several attractive properties which are discussed in this chapter. One trait of relevance in our context is the capability of the framework to extend naturally to any number of dimensions. Several works such as [56, 58] have applied the method in three dimensions for space-time video segmentation. An application of the four-dimensional case has been studied in [35] for coherent space-time scene modeling from multiview images.

In this chapter, we start by formalizing the setup in Section 3.2. We then derive the velocity vector fields that follow from the two main cues for extracting regions of interest. First, one may use a dissimilarity model and look for the edges of the regions to segment. These methods are called boundary-based since they only take into account boundary information. Section 3.3 describes how to derive the velocity vector field in this case and portrays some of the dissimilarity measures used in recent segmentation schemes. Second, one may use a similarity measure effectively looking for points with similar statistics or motion in the case of videos. The derivation of the velocity vector in this context is described in Section 3.4 along with the description of some of the common similarity measures used. Section 3.5 discusses implementation issues and describes the popular level-set method. Finally, a summary of the chapter is presented in Section 3.6.

3.2 The setup

Let $\mathcal{D} \subset \mathbb{R}^m$ be a multi-dimensional domain. For instance, an image gives $m = 2$, a video $m = 3$ and a light field $m = 4$. The problem of segmentation can be seen as finding the optimal partitions $\{\mathcal{H}_1, \ldots, \mathcal{H}_N\}$ of $\mathcal{D}$ depending on a certain segmentation criteria. For clarity, we consider here the simpler case where $N = 2$. That is, the domain $\mathcal{D}$ is separated in two parts namely $\mathcal{H}$ the region to segment and $\overline{\mathcal{H}}$ its complement. Clearly, $\mathcal{H} \cup \overline{\mathcal{H}} = \mathcal{D}$. The boundary of the two regions denoted $\partial\mathcal{H}$ is a closed curve which we also write as $\vec{\Gamma}$ (see Figure 3.1). All the derivations in this chapter are valid in the general case where $N > 2$ as will become clear in Chapter 4.

The problem of segmentation can be posed as an energy minimization. That is, an

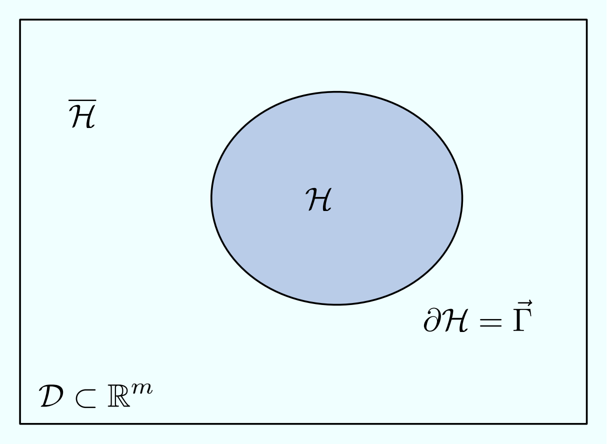

$\overline{\mathcal{H}}$
$\mathcal{H}$
$\partial\mathcal{H} = \vec{\Gamma}$
$\mathcal{D} \subset \mathbb{R}^m$

Figure 3.1: The domain $\mathcal{D}$ is separated into the inside of the region to segment $\mathcal{H}$ and its complement $\overline{\mathcal{H}}$. The interface between the two regions is the curve $\partial\mathcal{H} = \vec{\Gamma}$.

energy functional is designed such that it is minimal when the partitioning $\{\mathcal{H}, \overline{\mathcal{H}}\}$ coincides with the sought after segmentation. The idea is to start with an initial estimate and to introduce a dynamical scheme where the region $\mathcal{H}$ is made dependent on an evolution parameter τ such that $\partial\mathcal{H} = \vec{\Gamma}$ becomes $\partial\mathcal{H}(\tau) = \vec{\Gamma}(\tau)$. It is then possible to compute the derivative of the energy functional with respect to τ and deduce the steepest descent in order to evolve $\vec{\Gamma}(\tau)$ in such a way that the energy converges towards a minimum albeit local. This evolving boundary is known as an active contour [43]. In practice, the evolving interface is modeled by a parametric m-dimensional curve $\vec{\Gamma}(\vec{\sigma}, \tau) \subset \mathbb{R}^m$ where $\vec{\sigma} \in \mathbb{R}^{m-1}$. The partial differential equation (PDE) defining the deformations of the active contour is given by [16, 43]

$$\frac{\partial\vec{\Gamma}(\vec{\sigma}, \tau)}{\partial\tau} = \vec{v}_\Gamma(\vec{\sigma}, \tau) = F(\vec{\sigma}, \tau)\vec{n}_\Gamma(\vec{\sigma}, \tau) \qquad \text{with } \vec{\Gamma}(\vec{\sigma}, 0) = \vec{\Gamma}_0(\vec{\sigma}), \tag{3.1}$$

where $\vec{v}_\Gamma(\vec{\sigma}, \tau)$ is the speed function (or velocity vector field), $\vec{n}_\Gamma$ is the outward normal vector and $\vec{\Gamma}_0$ is the starting point defined by the initialization of the algorithm or the user. Figure 3.2 illustrates this setup and shows the example of an evolving interface in three dimensions. The problem is to define a velocity vector field $\vec{v}_\Gamma$ that will attract the active contour towards the desired region to segment based on the energy functional. The steady state of the PDE in (3.1) should therefore be obtained when the deformable model

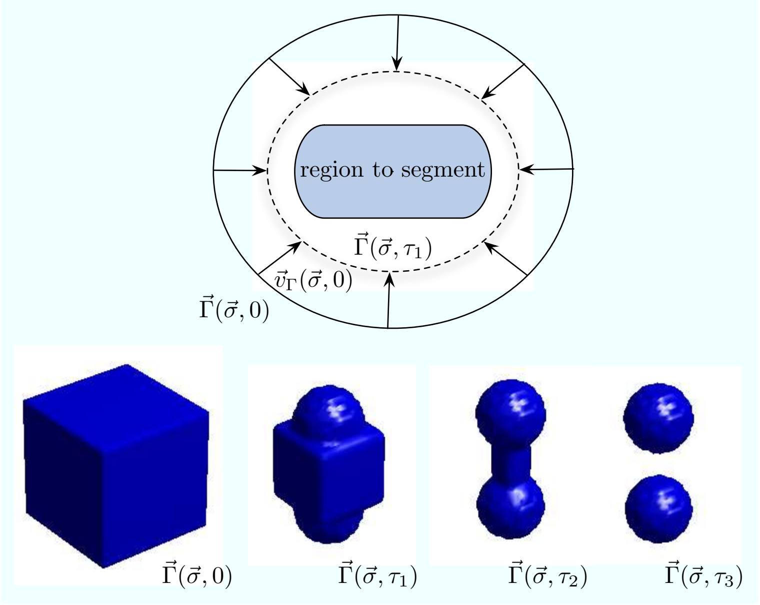

region to segment
$\vec{\Gamma}(\vec{\sigma},\tau_1)$
$\vec{v}_\Gamma(\vec{\sigma},0)$
$\vec{\Gamma}(\vec{\sigma},0)$
$\vec{\Gamma}(\vec{\sigma},0)$
$\vec{\Gamma}(\vec{\sigma},\tau_1)$
$\vec{\Gamma}(\vec{\sigma},\tau_2)$
$\vec{\Gamma}(\vec{\sigma},\tau_3)$

Figure 3.2: Evolution of the active contour $\vec{\Gamma}(\vec{\sigma},\tau)$ towards the boundary of the region to segment. The speed function $\vec{v}_\Gamma(\vec{\sigma},\tau)$ defines the deformation of the contour. The second row illustrates the evolution of an active contour in three dimensions (or active surface) towards objects to segment.

has reached the contour of the object. That is, the speed function $\vec{v}_\Gamma$ is designed such that the curve $\vec{\Gamma}$ converges towards the region to segment when $\tau \to \infty$.

In practice, there are two main ways to define the energy functionals that will govern the evolution of the active contours. Indeed, the energy may be defined on the boundary of the region leading to a boundary-based method. Alternatively, the energy may be defined on the whole region leading to a region-based method. Note that these two methods are not mutually exclusive. In the next sections, we describe the derivations of the speed functions and present the common segmentation criteria for images and videos.

3.3 Boundary-based segmentation

The first class of variational methods for segmentation are based on boundary information. Therefore the energy is a function of the boundary $\partial\mathcal{H}$ of $\mathcal{H}$ only. That is, we have

$$E(\tau) = \int_{\partial\mathcal{H}(\tau)} d_b(\vec{x})d\vec{\sigma}, \tag{3.2}$$

where $d_b(\vec{x}) : \mathbb{R}^m \rightarrow \mathbb{R}$ is a given potential function also known as a descriptor. This energy is at the basis of the original 'snakes' method [43]. It is also known as the geodesic active contour method [16] because the underlying energy functional can be seen as the length of the contour weighted by a potential function. The descriptor is designed such that the $E(\tau)$ is minimal when the desired region of interest has been found. Therefore, one has to solve the minimization problem:

$$\underset{\vec{\Gamma}(\tau)}{\operatorname{argmin}} \int_{\partial\mathcal{H}(\tau)} d_b(\vec{x}) d\vec{\sigma}, \tag{3.3}$$

which may be found using a classical steepest descent method. Many authors including [16] [42] have shown that the derivative of the functional with respect to τ is given by

$$\frac{dE(\tau)}{d\tau} = \int_{\vec{\Gamma}} [-\vec{\nabla} d_b(\vec{x}) \cdot \vec{n}_\Gamma + d_b(\vec{x})\kappa](\vec{v}_\Gamma \cdot \vec{n}_\Gamma) d\vec{\sigma},$$

where $\vec{v}_\Gamma$, $\vec{n}_\Gamma$ and κ are the speed, the outward unit normal and the mean curvature of $\vec{\Gamma}$ respectively, $\vec{\nabla}$ is the gradient operator and $\cdot$ denotes the scalar product. The steepest descent is therefore obtained with the evolution equation

$$\vec{v}_\Gamma = [-d_b(\vec{x})\kappa + (\vec{\nabla} d_b(\vec{x}) \cdot \vec{n}_\Gamma)]\vec{n}_\Gamma, \tag{3.4}$$

where in this equation $\vec{x} = \vec{\Gamma}(\vec{\sigma}, \tau)$. Hence by comparing (3.4) with (3.1), we now have the velocity for the energy minimizing active contour. Note that the case where $d_b(\vec{x})$ is a positive constant μ leads an evolution minimizing only the length of the contour. The velocity in this case becomes $\vec{v}_\Gamma = -\mu\kappa\vec{n}_\Gamma$ and is also known as the mean curvature flow.

3.3.1 Descriptors for boundary-based methods

The early active contour methods that have been applied to image segmentation were based on the assumption that different objects in an image $I(\vec{x})$ generate intensity discontinuities and have smooth contours. The flow driving the evolution of the curve is therefore designed to 'lock' the contour on strong intensity gradients (effectively acting as an edge detector) while maintaining a certain smoothness. In general, this descriptor can be written as:

$$d_b(\vec{x}) \quad = \quad g(|\vec{\nabla} I(\vec{x})|),$$

where $\vec{\nabla} I(\vec{x})$ is the image gradient. The function $g : [0, \infty[\to \mathbb{R}^+$ is a strictly decreasing function such as that $g(r) \to 0$ when $r \to \infty$. For example, one might use

$$g(r) = \frac{1}{1 + r^p},$$

where $p \geq 1$. The evolution of the contour in (3.4) will therefore be inhibited in regions with large intensity gradients. In the evolution equation (3.4), the term $g(|\vec{\nabla} I(\vec{x})|)\kappa$ tends to smooth the contour by reducing its curvature unless $g(|\vec{\nabla} I(\vec{x})|)$ is close to 0 which means a strong edge. The term $(\vec{\nabla} g(|\vec{\nabla} I(\vec{x})|) \cdot \vec{n}_\Gamma)$ tends to evolve the contour towards an intensity edge as long as $\vec{\nabla} g(|\vec{\nabla} I(\vec{x})|)$ is not orthogonal to $\vec{n}_\Gamma$.

The geodesic active contour formulation described above has also been used for motion tracking in videos. In particular, Mitiche et al. [56] used a three-dimensional active surface to extract moving objects from image sequences $I(x, y, t)$. In this work, the authors proposed the descriptors

$$d_b(\vec{x}) \quad = \quad g(|\delta I(\vec{x})|),$$

with

$$\delta I(\vec{x}) = \frac{\frac{\partial I}{\partial t}}{(\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2)^{\frac{1}{2}}},$$

which is the normal component of the optical velocity. The active surface is therefore designed to lock onto motion discontinuities.

3.4 Statistical region-based segmentation

While for some applications, purely boundary based functionals are sufficient, it is usually beneficial to use more information. Indeed, the regions to segment may have different properties such as texture or spatial and motion homogeneity that cannot be included in a boundary-based energy functional. Moreover, due to their boundary-based nature, these types of functionals are not adapted to objects that have diffuse boundaries. Finally, as we saw in the previous section, they often rely on image gradients which makes them susceptible to noise. All these issues can be dealt with by using region-based methods as was nicely illustrated by Chan and Vese [20].

It is worth mentioning here that the region-based active contour method shares some similarities with the watershed algorithm [8, 77]. In this method, images are considered as a topographical surface made of hills and valleys. The watershed algorithms are designed to segment these topographical surfaces in different basins separated by the watershed lines. In order to perform this task, the idea is to fill the valleys in a similar way that water fills a basin. The main drawback of these methods are their sensitivity to noise, often resulting in oversegmentation [37]. Moreover, there is no smoothness constraint in these methods making it difficult, for instance, to track regions over multiple images [37]. Both these issues can be dealt with using the region-based active contour methods.

Zhu and Yuille [88] were the first authors to coin the term 'region competition' and introduced a generalized framework based on probabilities. In this formulation, the shortest curve is sought that optimally separates the domain $\mathcal{D}$ into $\mathcal{H}$ and $\overline{\mathcal{H}}$ given probability functions that the point $\vec{x}$ in the domain belongs to $\mathcal{H}$ or to $\overline{\mathcal{H}}$. Following their work, an optimal partition $\{\mathcal{H}, \overline{\mathcal{H}}\}$ of the image domain can be computed by minimizing the energy

$$E(\tau) = \int_{\mathcal{H}(\tau)} d_{in}(\vec{x}, \vec{p}_{in}) d\vec{x} + \int_{\overline{\mathcal{H}}(\tau)} d_{out}(\vec{x}, \vec{p}_{out}) d\vec{x} + \int_{\partial\mathcal{H}(\tau)} \mu d\vec{\sigma}, \tag{3.5}$$

where $\{d_{in}(\vec{x}, \vec{p}_{in}), d_{out}(\vec{x}, \vec{p}_{out})\} : \mathbb{R}^m \rightarrow \mathbb{R}$ are volume potentials that measure the coherence of a point $\vec{x}$ with a model defined by the parameters $\vec{p}_{in}$ and $\vec{p}_{out}$. For instance, these parameters may be the mean and the variance of a normal density function. The last term is a regularization term smoothing the contour.

Since minimizing (3.5) jointly for $\{\mathcal{H}, \overline{\mathcal{H}}\}$ and the parameters $\{\vec{p}_{in}, \vec{p}_{out}\}$ is very complicated, the problem is decomposed into two iterated steps. First, given an initial estimate of the regions, one can solve the minimization for the parameters. That is, we are looking for

$$\underset{\{\vec{p}_{in},\vec{p}_{out}\}}{\operatorname{argmin}} \int_{\mathcal{H}} d_{in}(\vec{x},\vec{p}_{in})d\vec{x} + \int_{\overline{\mathcal{H}}} d_{out}(\vec{x},\vec{p}_{out})d\vec{x},$$

where the last term in (3.5) is omitted since it does not depend on the parameters $\vec{p}_{in}$ and $\vec{p}_{out}$. This is a standard problem and we will not delve on it here. More interesting is the second step that consists in estimating the regions $\{\mathcal{H},\overline{\mathcal{H}}\}$ given the parameters $\{\vec{p}_{in},\vec{p}_{out}\}$. In this case, we are effectively solving

$$\underset{\vec{\Gamma}(\tau)}{\operatorname{argmin}} \int_{\mathcal{H}(\tau)} d_{in}(\vec{x},\vec{p}_{in})d\vec{x} + \int_{\overline{\mathcal{H}}(\tau)} d_{out}(\vec{x},\vec{p}_{out})d\vec{x} + \int_{\partial\mathcal{H}(\tau)} \mu d\vec{\sigma},$$

which means looking for the shortest curve that best separates a region that follows the probability model d_{in} from the one that follows the probability model d_{out}. It can be shown that the derivative of the functional (3.5) with respect to τ is given by [41, 70]

$$\frac{dE(\tau)}{d\tau} = \int_{\vec{\Gamma}} [d_{in}(\vec{x},\vec{p}_{in}) - d_{out}(\vec{x},\vec{p}_{out}) + \mu\kappa](\vec{v}_{\Gamma}\cdot\vec{n}_{\Gamma})d\vec{\sigma}. \tag{3.6}$$

Hence the evolution equation associated with the steepest descent of the energy becomes

$$\vec{v}_{\Gamma} = [d_{out}(\vec{x},\vec{p}_{out}) - d_{in}(\vec{x},\vec{p}_{out}) - \mu\kappa]\vec{n}_{\Gamma}, \tag{3.7}$$

where again by comparing (3.7) with (3.1), we now have a speed function for the active contour. Note that the competition formulation is now clear. Indeed, discarding the curvature term $\mu\kappa$, a point $\vec{x}$ belonging to the inside of the region to segment will have a small d_{in} and large d_{out} resulting in a positive force. The point will therefore be incorporated. Inversely, a point belonging to the outside of the region will have a small d_{out} and a large d_{in} resulting in a negative speed. The point will therefore be rejected. This competition is illustrated in Figure 3.3. The flow in (3.7) is the one used in many variational image and video segmentation algorithms including [41, 53, 58] and [20].

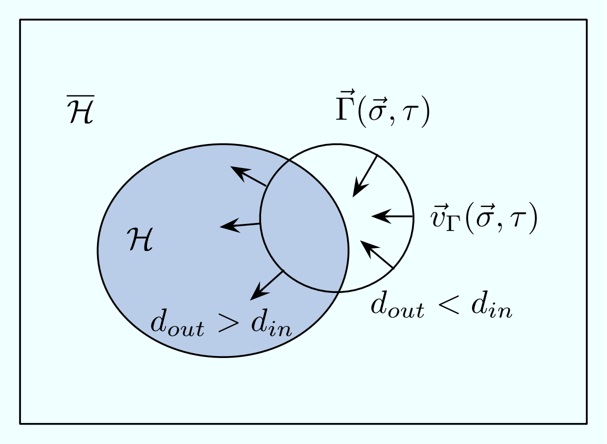


Figure 3.3: Region competition determining the evolution of the active contour towards the region to segment. A point belonging to the inside of the region to segment will have a small d_{in} and large d_{out} resulting in a positive speed $\vec{v}_\Gamma$. The point will therefore be incorporated. Inversely, a point belonging to the outside of the region will have a small d_{out} and a large d_{in} resulting in a negative speed. The point will therefore be rejected.

3.4.1 Descriptors for region-based methods

Region-based variational frameworks have attracted a lot attention and many authors have proposed different descriptors for different segmentation problems. In this section, we describe some of the descriptors proposed for intensity and motion-based segmentation.

The works based on probabilities such as [88] treat the segmentation problem as a Bayesian inference. Maximizing the a posteriori probability is equivalent to minimizing the negative logarithm which leads to the descriptors:

$$\begin{aligned} d_{in}(\vec{x},\vec{p}_{in}) &= -\log P(I(\vec{x})|\vec{p}_{in}) \\ d_{out}(\vec{x},\vec{p}_{out}) &= -\log P(I(\vec{x})|\vec{p}_{out}), \end{aligned}$$

where P is the probability density and $\vec{p}_{in}$ and $\vec{p}_{out}$ are the parameters of the density function. For instance, one might choose the Gaussian distribution:

$$P(I(\vec{x})|(\eta_n,\rho_n)) = \frac{1}{\sqrt{2\pi}\rho_n} e^{-\frac{(I(\vec{x})-\eta_n)^2}{2\rho_n^2}},$$

where η_n and ρ_n are the mean and the variance of the region $\mathcal{H}_n$ respectively. A particular case of this functional is the Chan and Vese model or 'active contours without edges' [20]. Indeed, posing $2\rho_{in}^2 = 2\rho_{out}^2 = 1$ gives the functionals

$$\begin{aligned} d_{in}(\vec{x}, \eta_{in}) &= [I(\vec{x}) - \eta_{in}]^2 \\ d_{out}(\vec{x}, \eta_{out}) &= [I(\vec{x}) - \eta_{out}]^2, \end{aligned}$$

which are the ones used in their paper. The active contour is therefore made to separate regions with different mean values. While these descriptors are based on the statistics of the intensity in the regions to segment, the availability of multiple images of a scene taken at different times (i.e. videos) or from different viewpoints (i.e. light fields) enables one to use other cues. For instance, regions may be segmented according to a particular motion or depth model. This usually assumes that the radiance of a point on an object does not change in time and that the surfaces of the objects in the scene are Lambertian. Numerous works using active contours have used these cues for segmenting moving images, several of which are described here.

Jehan-Besson et al. [41] used region-based active contours for motion detection. In that paper, the authors assume a known static background $B(\vec{x})$ and used the descriptors

$$\begin{aligned} d_{in}(\vec{x}) &= \zeta \\ d_{out}(\vec{x}) &= [B(\vec{x}) - I(\vec{x})]^2, \end{aligned}$$

where ζ is a positive constant. The goal is therefore to find regions in the frame $I(\vec{x})$ that have moved. Clearly, the ζ parameter acts as a threshold. Indeed, we notice from the evolution equation (3.7) that regions where the intensity changes such that $[B(\vec{x}) - I(\vec{x})]^2$ is larger than ζ will be included in the active contour. Inversely, the point is rejected when the squared intensity changes are smaller than ζ. These descriptors, though quite simple, have been applied successfully for extracting moving objects from static backgrounds. They also have the advantage that motion parameters need not be estimated. However, in many cases, the background is changing as well (i.e. a moving camera). In order to account for this case, the descriptors need to take motion into account. In particular, Mansouri and Konrad [53] proposed the descriptors

$$\begin{aligned} d_{in}(\vec{x}, p_{in}) &= [I(p_{in}(\vec{x}), t_{i+1}) - I(\vec{x}, t_i)]^2 \\ d_{out}(\vec{x}, p_{out}) &= [I(p_{out}(\vec{x}), t_{i+1}) - I(\vec{x}, t_i)]^2, \end{aligned}$$

where $\{p_{in}(\vec{x}), p_{out}(\vec{x})\} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ are motion transformations (e.g. affine). These descriptors effectively lead to an energy minimization based on the least squares error for a point and the motion model for each region.

The two sets of descriptors described above used two-dimensional active contours and take into consideration only two consecutive frames. Several works (e.g. [56,58]) have studied the use of a large amount of frames in order to impose temporal coherence in the segmentation. In particular, Ristivojevic and Konrad [58] extended the two-frame method in [53] by using the variance along a motion trajectory throughout a large set of frames. This leads to the descriptors:

$$\begin{aligned} d_{in}(\vec{x}, p_{in}) &= \frac{1}{N_f} \sum_{i=1}^{N_f} [I(p_{in}(\vec{x}, t_i)) - m_{in}(\vec{x})]^2 \\ d_{out}(\vec{x}, p_{out}) &= \frac{1}{N_f} \sum_{i=1}^{N_f} [I(p_{out}(\vec{x}, t_i)) - m_{out}(\vec{x})]^2, \end{aligned}$$

with

$$\begin{aligned} m_{in}(\vec{x}, p_{in}) &= \frac{1}{N_f} \sum_{i=1}^{N_f} I(p_{in}(\vec{x}, t_i)) \\ m_{out}(\vec{x}, p_{out}) &= \frac{1}{N_f} \sum_{i=1}^{N_f} I(p_{out}(\vec{x}, t_i)), \end{aligned}$$

where N_f is the number of frames under consideration and $\{p_{in}(\vec{x}, t_i), p_{out}(\vec{x}, t_i)\} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a motion trajectory.

3.5 Implementing the evolution of active contours

A natural way to implement the evolution equation in (3.1) consists in using an explicit representation by discretizing the curve Γ with a set of connected control points. The displacement for each control point is computed according to the speed $\vec{v}_\Gamma$. While this approach is natural and computationally efficient, it has some drawbacks. First, the control points may evolve in such a way that they are closer and closer together or further and further apart which leads to numerical instabilities in the computation of derivatives. Therefore one needs to introduce a reparameterizing scheme in order to retain stability. Constantly resampling the contour, for example, is one way of achieving this. Second, the speed function may cause the curve to be separated in two regions or inversely, to merge with other curves (i.e. topological changes). Therefore the evolution scheme must also introduce a numerical test to enable the merging and splitting of contours. These methods rely on somewhat ad-hoc tests and the procedure becomes even more problematic as the number of dimensions increases.

Other methods compute the PDE in (3.1) using an implicit discretization of the contour. One very popular representation is the level-set method which we describe in the next sections. The method is presented here in the two-dimensional case since the extension to the multi-dimensional case is straightforward thereafter.

3.5.1 The level-set method

The level-set method [64] addresses stability and topology issues of active contours by implicitly representing a curve $\vec{\Gamma}(s,\tau) = [x(s,\tau), y(s,\tau)] \subset \mathbb{R}^2$ as the zero level of a higher dimensional surface $z = \phi(x,y,\tau) \subset \mathbb{R}^3$ and evolving the surface as opposed to the curve itself. In order to derive an evolution equation for the surface that will solve the original PDE in (3.1), the level-set function ϕ must satisfy two conditions. First, $\phi(\vec{\Gamma}(s,\tau),\tau) = 0$ needs to hold for all s. In other terms, the partial derivative of $\phi(\vec{\Gamma}(s,\tau),\tau)$ with respect to s must always be zero. Applying the chain rule, we have

$$\frac{d\phi}{ds} = \frac{\partial \phi}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial \phi}{\partial y}\frac{\partial y}{\partial s} = 0 \qquad \Leftrightarrow \qquad \frac{\vec{\nabla}\phi}{|\vec{\nabla}\phi|} = -\vec{n}_\Gamma,$$

where $\vec{\nabla}$ is the gradient operator and $\vec{n}_\Gamma$ is the outward normal to the curve $\vec{\Gamma}$. Second, $\phi(\vec{\Gamma}(s,\tau),\tau) = 0$ needs to hold for all iterations τ. Therefore, the partial derivative of $\phi(\vec{\Gamma}(s,\tau),\tau)$ with respect to τ must also be zero. Again applying the chain rule gives

$$\frac{d\phi}{d\tau} = \frac{\partial \phi}{\partial x}\frac{\partial x}{\partial \tau} + \frac{\partial \phi}{\partial y}\frac{\partial y}{\partial \tau} + \frac{\partial \phi}{\partial \tau} = 0 \qquad \Leftrightarrow \qquad \frac{\partial \phi}{\partial \tau} + \vec{\nabla}\phi(\vec{\Gamma}(s,\tau),\tau)\cdot\frac{\partial \vec{\Gamma}}{\partial \tau} = 0.$$

Combining the two conditions along with the definition of the speed of the original curve $\frac{\partial \vec{\Gamma}}{\partial \tau} = F\vec{n}_\Gamma$ enables one to write the level-set equation

$$\frac{\partial \phi(x,y,\tau)}{\partial \tau} = F(x,y)|\vec{\nabla}\phi(x,y,\tau)|. \tag{3.8}$$

As a result of the two conditions, the solution to (3.1) will be given by the zero level of ϕ in $\tau \to \infty$. For instance, the evolution for the classical region-based active contour (3.7) becomes

$$\frac{\partial \phi(x,y,\tau)}{\partial \tau} = [d_{out}(x,y) - d_{in}(x,y) - \mu\kappa_\phi(x,y)]|\vec{\nabla}\phi(x,y,\tau)|,$$

where κ_ϕ is the curvature of the level-set of ϕ given by

$$\kappa_\phi = -\vec{\nabla}\cdot(\frac{\vec{\nabla}\phi}{|\vec{\nabla}\phi|}).$$

The level set surface ϕ is free to expand, shrink, rise and fall in order to generate the deformations of the original curve and topological changes are naturally handled (see Figure 3.4). Moreover, since this evolution equation is defined over the whole domain, there is no need to parameterize the curve with individual points. The numerical computations are performed using finite differences on a fixed cartesian grid, thus solving the stability issue.

In practice, the level-set function $\phi(x,y,\tau)$ is usually defined as the signed distance function from the curve $\vec{\Gamma}$. That is, for each point (x,y) and $\forall\tau \geq 0$, the function ϕ

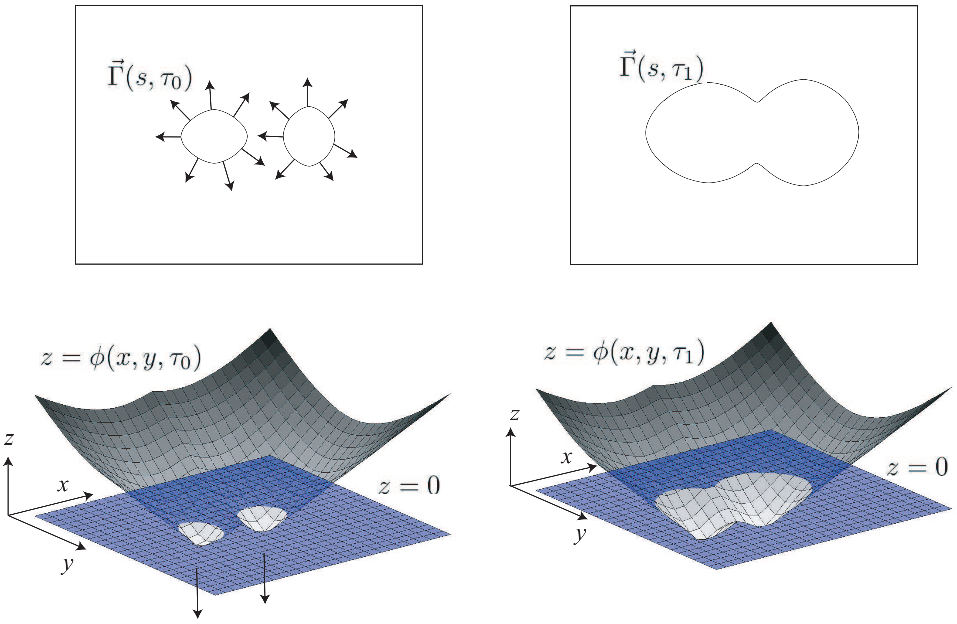


Figure 3.4: The level-set method. The evolving curve $\vec{\Gamma}$ is implicitly represented as the zero level of a higher dimensional function ϕ. Numerical computations for the curve evolution are performed on a fixed cartesian grid and topological changes are naturally handled.

represents the signed distance of (x, y) from the contour $\vec{\Gamma}$. The negative values of ϕ are defined as inside the curve and the positive values are outside (see Figure 3.4).

3.5.2 Reinitialization

Computing the level-set function ϕ using the evolution equation (3.8) may cause the evolving interface to get stuck in local minima or lead to a very large amount of evolution. This is due to the fact that the gradient of the function ϕ may tend to infinity as the number of iterations is increasing. Inversely, the gradient may tend to zero which inhibits the evolution of the boundary. This problem is usually solved by reinitializing ϕ to a signed distance function such that $|\nabla\phi| = 1$. This is performed by using the evolution equation

$$\frac{\partial\phi(x, y, \tau)}{\partial\tau} = sign(\phi(x, y, \tau))(1 - |\nabla\phi(x, y, \tau)|), \tag{3.9}$$

where $sign(x)$ is a function that gives the sign of x. By alternating this evolution equation with the one in (3.8), it is ensured that the level-set function will remain regular.

Other more sophisticated methods have also been proposed that modify the partial differential equation. The speed function F is computed only on points belonging to the contour $\vec{\Gamma}$. The speed is then extended to the whole domain in such a manner that the PDE remains stable and the reinitialization does not need to be performed.

3.5.3 Fast methods

The advantages of the level-set method clearly come at a the cost of computational complexity since the gradients and the amount of evolution need to be computed for all the levels of ϕ. Some solutions to reduce the number of computations have been developed such as the narrowband method [64]. This implementation enables one to reduce the complexity of the level-set method from $O(N^2)$ to $O(kN)$ where N is the size of the grid on which the level-set function is evaluated. The basic idea is to perform the computations not on the whole image domain but on a narrow band in the vicinity of the zero level of the level-set function ϕ. The narrow band is updated when the evolving front reaches its borders.

Other methods have been proposed that reduce the number of computations but provide only approximate solutions [67]. The solutions of the evolution equations are not found using PDEs but through an optimality condition for the final contour based on a speed test. Only simple decisions are made such as the insertion or the deletion of points in the contour. These methods enable a considerable reduction of computations and real-time video segmentations have been reported in [45].

3.6 Summary

Variational frameworks using active contours have had a big impact on segmentation methods in the computer vision and image processing communities. Throughout the past few decades, these methods have enabled many authors to solve successfully image and video segmentation problems.

In the first part of the chapter, we showed that the segmentation problem can be posed as an energy minimization problem. This energy minimization can be solved by initializing a set of regions bounded by active contours and evolving them in the direction of steepest descent. In order to do this, the derivatives of the energy functionals had to be computed. While doing so, we made the distinction between boundary-based and region-based methods. For both cases, we described some of the criteria used in image and video segmentation. In the last part of the chapter, we discussed the issues of implementing the PDEs that govern the evolution of the active contours. We then presented the level-set method which straightforwardly enables a stable implementation and has the ability to handle topological changes.

The key points to retain are highlighted in the following:

- Active contour methods are straightforward to extend to multi-dimensional signals.
- The methods are flexible in terms of the criteria (i.e. the descriptors) for segmentation.
- The level-set method enables a stable implementation and topological changes are naturally handled.

Chapter 4

A semi-parametric approach for extracting plenoptic hypervolumes in light fields

4.1 Introduction

The data in light fields is characterized with a particular structure and a high degree of regularity. This regularity is nicely captured with plenoptic hypervolumes that are characterized by the region carved out by layers in each of the images. It is therefore beneficial to extract these regions for numerous applications involving multiview images. However, obtaining an approximate and continuous decomposition of the light filed is a challenging task. In [69], the segmentation is performed in a supervised fashion where the contours of layers are semi-manually defined on a key frame. These contours are then propagated to all the views using a user defined planar depth map. Despite progress in unsupervised stereo methods [59] and layer extraction schemes (e.g. [3,51,78,89]), it is still difficult to obtain a coherent segmentation. Most of these methods focus on the extraction of layers rather than the coherent volumes or hypervolumes carved out in multiview data. Moreover, not all these methods are scalable to higher dimensional plenoptic functions and treat all the images equally. Several authors have tackled the problem using the particular structure of the EPI images [24,31]. However, these methods remain local or provide only sparse depth maps.

In this chapter, we address the problem of deriving an unsupervised multi-dimensional analysis to extract coherent regions in structured multiview data. Our method is based on the fact that layers in the scene carve out EPI-tubes [24] in EPI volumes and more generally hypervolumes in light fields. Unlike the methods in [24,31], our approach imposes coherence in all dimensions in order to extract these volumes or hypervolumes and generates a regular and approximate decomposition of the scene. The method presented is a novel multi-dimensional variational framework that explicitly takes into account the structure of the data. That is, the contours of the hypervolumes in the image dimensions (x, y) are extracted using the level-set method [64] which is a non-parametric approach. However, the dependencies between viewpoints (v_x, v_y) and occlusions are constrained by the camera setup. They can therefore be parameterized. The resulting framework is a semi-parametric region competition that is global and hence uses all the available data jointly. This in turn insures that the segmentation is coherent across all the images and occlusions are efficiently and naturally handled.

The chapter is organized as follows: In Section 4.2 we pose the problem in a more formal way. Section 4.3 derives a light field segmentation scheme based on the classical active contour method. Section 4.4 discusses the plenoptic constraints and how they affect the segmentation framework. In Section 4.5, we present a novel variational framework based on the shape constraints of the regions carved out by plenoptic hypervolumes in light fields. Section 4.6 deals with initializations of the algorithm. Finally, Section 4.7 gives an overview of the segmentation scheme and we conclude in Section 4.8. Note that experimental results are presented and discussed in the following chapter.

4.2 Problem formulation

Let $I(x, y, v_x, v_y)$ be a light field where (x, y) are the image coordinates and (v_x, v_y) define the position of the camera on a plane. Our goal is to partition the four-dimensional image

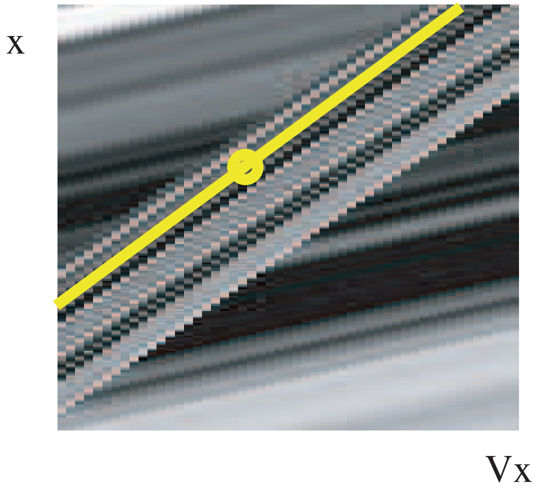


Figure 4.1: The descriptor is given by the squared difference between the intensity $I(\vec{x})$ and the mean of the intensity along the line the point belongs to.

domain $\mathcal{D}$ into a set of plenoptic hypervolumes $\{\mathcal{H}_1, \dots, \mathcal{H}_N\}$ such that each region $\mathcal{H}_n$ is consistent with an approximate depth model. In essence, the problem of extracting the plenoptic hypervolumes consists in finding the boundaries of the regions $\partial\mathcal{H}_n = \vec{\Gamma}_n \subset \mathbb{R}^4$ that delimit the contour of the layers on all the views. Note that due to occlusions, the full hypervolumes $\mathcal{H}_n$ are not always available. Therefore we denote with $\mathcal{H}_n^{\perp}$ the available hypervolume. This notation will become clearer in Section 4.4.2.

The first step in this segmentation process is to design a measure $d_n(\vec{x}, p_n)$ that measures the consistency of a point $\vec{x} = (x, y, v_x, v_y)$ with a particular depth model p_n. Recall from Chapter 2 that points in space $\vec{X} = (X, Y, Z)$ are mapped onto the light field according to:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = \begin{pmatrix} X/Z - v_x/Z \\ Y/Z - v_y/Z \\ v_x \\ v_y \end{pmatrix}, \tag{4.1}$$

where we assume for simplicity that the focal length of the cameras is unity. A point in space is therefore mapped onto a four-dimensional trajectory in the light field hypervolume.

Call $x_n = X/Z$ and $y_n = Y/Z$ the projection of the point $\vec{X}$ onto the image in $v_x = v_y = 0$ and define $p_n = 1/Z$. We can therefore assume that

$$I(x_n - v_x p_n, y_n - v_y p_n, v_x, v_y) = I(x_n - v'_x p_n, y_n - v'_y p_n, v'_x, v'_y),$$
$$\forall\{(x_n - v_x p_n, y_n - v_y p_n, v_x, v_y), (x_n - v'_x p_n, y_n - v'_y p_n, v'_x, v'_y)\} \in \mathcal{H}_n^{\perp},$$

where we are considering Lambertian surfaces. The effect of specularities in the EPI setup has been studied in [24, 71] although we will not take these effects into account here. It is worth mentioning that there are methods to remove specular highlights using EPI analysis [24]. Since we do not take into account reflections, we may define the consistency measure $d_n(\vec{x}, p_n)$ as the normalized squared difference between $I(\vec{x})$ and the mean of the intensities along the plenoptic trajectory defined by the slope p_n (see Figure 4.1). We therefore have

$$d_n(\vec{x}, p_n) = [I(\vec{x}) - m_n(\vec{x}, p_n)]^2 \tag{4.2}$$

with

$$m_n(\vec{x}, p_n) = \frac{\int\int I_n(x_n - v_x p_n, y_n - v_y p_n, v_x, v_y) dv_x dv_y}{\int\int O_n(x_n - v_x p_n, y_n - v_y p_n, v_x, v_y) dv_x dv_y}$$

where the functions $I_n(\vec{x})$ and $O_n(\vec{x})$ are introduced in order to define the integration bounds. That is,

$$I_n(\vec{x}) = \begin{cases} I(\vec{x}), & \vec{x} \in \mathcal{H}_n^{\perp} \\ 0, & \vec{x} \notin \mathcal{H}_n^{\perp} \end{cases}$$

and $O_n(\vec{x})$ is the binary function

$$O_n(\vec{x}) = \begin{cases} 1, & \vec{x} \in \mathcal{H}_n^{\perp} \\ 0, & \vec{x} \notin \mathcal{H}_n^{\perp}. \end{cases}$$

We are therefore looking for a partitioning $\{\mathcal{H}_1^{\perp}, \ldots, \mathcal{H}_N^{\perp}\}$ of the light field such that

$$E_{tot}(\vec{\Gamma}_1, \ldots, \vec{\Gamma}_N) = \sum_{n=1}^{N} E_n(\vec{\Gamma}_n) = \sum_{n=1}^{N} \int_{\mathcal{H}_n^{\perp}} d_n(\vec{x}, p_n) d\vec{x} \tag{4.3}$$

is minimal. Therefore the minimization problem we are seeking to solve is the following:

$$\underset{\{\vec{\Gamma}_1,\ldots,\vec{\Gamma}_N,p_1,\ldots,p_N\}}{\operatorname{argmin}} \sum_{n=1}^{N} \int_{\mathcal{H}_n^\perp} d_n(\vec{x},p_n)d\vec{x}. \tag{4.4}$$

This is a classical region-based optimization problem that may be solved using a variational framework and active contours.

4.3 A 4D variational approach based on active contours

In this section, we derive a region-based variational method to extract plenoptic hypervolumes based on four-dimensional active contours. That is, we represent the light field with N plenoptic hypervolumes $\mathcal{H}_n^\perp$ each of which is bounded by the hypersurface $\vec{\Gamma}_n$. A straightforward approach is to apply the variational method described in Chapter 3. Note that for the moment, we will assume that the depth models p_n are known. Their estimation will be treated further on in Section 4.5.2. The regions $\mathcal{H}_n^\perp$ are made dependent on an evolution parameter τ such that the derivative can be computed in order to evolve the hypervolumes in a steepest descent fashion. One way to minimize the E_{tot} in (4.3) consists in iteratively evolving each hypervolume $\mathcal{H}_n^\perp$. Take for example the case where there are two layers and a background. The hypervolumes are $\mathcal{H}_1^\perp$, $\mathcal{H}_2^\perp$ and $\mathcal{H}_3^\perp = \mathcal{D} \setminus \{\mathcal{H}_1^\perp \cup \mathcal{H}_2^\perp\}$ where $\mathcal{D}$ is the four-dimensional light field domain. The energy functional in (4.3) may therefore be written as

$$\begin{aligned} E_{tot}^1(\tau) &= \int_{\mathcal{H}_1^\perp(\tau)} d_1(\vec{x},p_1)d\vec{x} + \underbrace{\int_{\mathcal{H}_2^\perp(\tau)} d_2(\vec{x},p_2)d\vec{x} + \int_{\mathcal{H}_3^\perp(\tau)} d_3(\vec{x},p_3)d\vec{x}}_{\int_{\overline{\mathcal{H}_1^\perp}(\tau)} d_1^{out}(\vec{x},p_1^{out})d\vec{x}} \\ &= E_{in}^1(\tau) + E_{out}^1(\tau), \end{aligned}$$

which is equivalent to the standard two-region case described in Chapter 3. Similarly, for any region $\mathcal{H}_n^\perp$, the energy is written as

$$E_{tot}^n(\tau) = \int_{\mathcal{H}_n^\perp(\tau)} d_n(\vec{x},p_n)d\vec{x} + \int_{\overline{\mathcal{H}_n^\perp}(\tau)} d_n^{out}(\vec{x},p_n^{out})d\vec{x}, \tag{4.5}$$

where all the other regions are gathered in $\overline{\mathcal{H}_n^{\perp}}(\tau)$ and $d_n^{out}(\vec{x}, p_n^{out}) = d_i(\vec{x}, p_i)$ when $\vec{x} \in \mathcal{H}_i^{\perp}$ for all $i \neq n$. Region-based active contour methods have shown that the gradient of the energy is:

$$\frac{dE_{tot}^n(\tau)}{d\tau} = \int_{\vec{\Gamma}_n} [d_n(\vec{x}, p_n) - d_n^{out}(\vec{x}, p_n^{out})](\vec{v}_{\Gamma_n} \cdot \vec{n}_{\Gamma_n}) d\vec{\sigma}, \tag{4.6}$$

where $\vec{v}_{\Gamma_n} = \frac{\partial \vec{\Gamma}_n}{\partial \tau}$ is the velocity, $\vec{n}_{\Gamma_n}$ is the outward unit normal vector, $d\vec{\sigma}$ is a differential hypersurface element and $\cdot$ denotes the scalar product. The steepest descent of the energy therefore yields the following partial differential equation:

$$\vec{v}_{\Gamma_n} = [d_n^{out}(\vec{x}, p_n^{out}) - d_n(\vec{x}, p_n)]\vec{n}_{\Gamma_n}. \tag{4.7}$$

This flow will drive the evolution of the hypersurfaces $\vec{\Gamma}_n$ in an unconstrained fashion as illustrated in Figure 4.2(a). While the equation in (4.7) is valid in the general case, it does not take into account the geometry and occlusion constraints that are inherent to the plenoptic function and in particular light fields (see Chapter 2). As we will see in Section 5.3, this method fails to capture regions that are strongly occluded and does not necessarily generate a coherent segmentation. Recall that points in space are mapped onto lines in the plenoptic domain and the slope of the lines are inversely proportional to the depth of the points. The next sections show how the evolution of the hypersurfaces can be modified in order to take into account these constraints.

4.4 Imposing the plenoptic constraints

In this section, we study the effects of the plenoptic constraints, namely the geometry and occlusion constraints, and how to impose them in the energy minimization framework.

4.4.1 Geometry constraints

Let $\vec{\gamma}_n(s, \tau) = [x_n(s, \tau), y_n(s, \tau)]$ be the 2D contour defined by the intersection of the hypersurface $\vec{\Gamma}_n$ and the image plane in $v_x = v_y = 0$. That is, it represents the contour

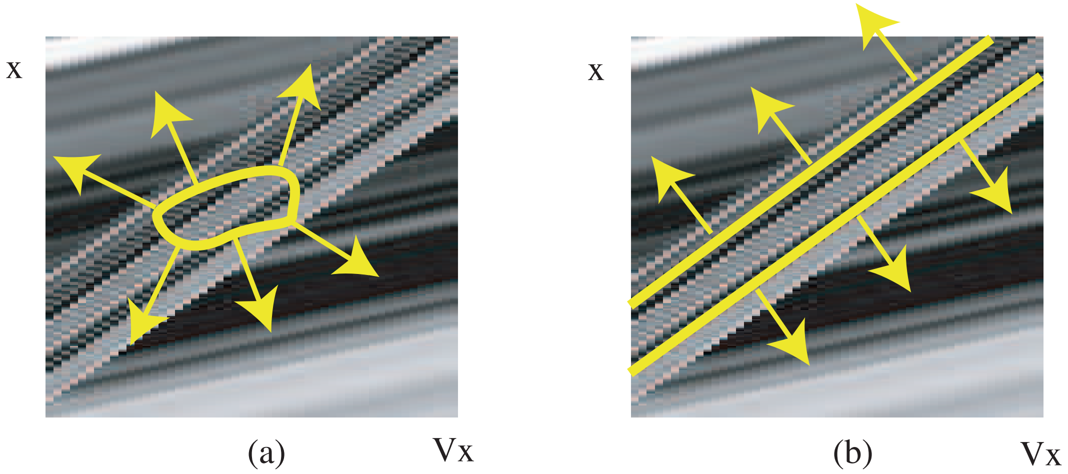


Figure 4.2: Unconstrained and constrained surface evolutions under the EPI parameterization. In Figure 4.2(a) no particular shape constraints are applied to the active surface $\vec{\Gamma}$. In Figure 4.2(b) the surface is parameterized using the two-dimensional contour $\vec{\gamma}$ in $v_x = 0$ and is constrained to evolve in a particular manner coherent with the structure of the EPI.

of the layer on a single image. In general, the intersection of the hypervolume with an image in $(v_x, v_y) = constant$ is the layer on that image. According to (4.1), the boundary plenoptic hypervolume under these assumptions can be parameterized by

$$\vec{\Gamma}_n(s, v_x, v_y, \tau) = \begin{pmatrix} x_n(s,\tau) - v_x p_n(s,\tau) \\ y_n(s,\tau) - v_y p_n(s,\tau) \\ v_x \\ v_y \end{pmatrix} \tag{4.8}$$

and is completely determined by the curve $\vec{\gamma}_n(s,\tau)$ if we assume that $p_n(s,\tau)$ is known. It is therefore possible to propagate the position and the shape of $\vec{\gamma}_n(s,\tau)$ on all the images. That is, the shape variations in the hypersurface $\vec{\Gamma}_n$ are completely determined by the shape variations of the curve $\vec{\gamma}_n$ in the two-dimensional subspace. Moreover, an explicit derivation of the normal and velocity vectors shows that the projection of $\vec{v}_{\Gamma_n} \cdot \vec{n}_{\Gamma_n}$ onto the subspace is related to $\vec{v}_{\gamma_n} \cdot \vec{n}_{\gamma_n}$ with

$$\vec{v}_{\Gamma_n} \cdot \vec{n}_{\Gamma_n} = \chi_n(s, v_x, v_y, \tau)(\vec{v}_{\gamma_n} \cdot \vec{n}_{\gamma_n}), \tag{4.9}$$

where $\chi_n(s, v_x, v_y, \tau)$ is the weighting function depending on the p_n and the camera setup. We prove in Appendix A the following proposition:

Proposition 1. *Assume a four-dimensional hypersurface parameterized by*

$$\vec{\Gamma}(s, v_x, v_y, \tau) = \begin{pmatrix} x(s,\tau) - v_x p(s,\tau) \\ y(s,\tau) - v_y p(s,\tau) \\ v_x \\ v_y \end{pmatrix},$$

where $\vec{\gamma}(s,\tau) = [x(s,\tau), y(s,\tau)]$ is the contour of the surface in $v_x = v_y = 0$. The τ is the evolution parameter and $\vec{v}_\Gamma = \partial\vec{\Gamma}/\partial\tau$ and $\vec{v}_\gamma = \partial\vec{\gamma}/\partial\tau$ are the velocities. Then the normal speed $\vec{v}_\Gamma \cdot \vec{n}_\Gamma$ of $\vec{\Gamma}$ projected onto the subspace in $v_x = v_y = 0$ is related to the normal speed $\vec{v}_\gamma \cdot \vec{n}_\gamma$ of $\vec{\gamma}$ by the relation $\vec{v}_\Gamma \cdot \vec{n}_\Gamma = \chi(s, v_x, v_y, \tau)(\vec{v}_\gamma \cdot \vec{n}_\gamma)$ with

$$\chi(s, v_x, v_y, \tau) = \frac{(1 + \frac{\partial p}{\partial x} v_x + \frac{\partial p}{\partial y} v_y)\sqrt{(\frac{\partial x}{\partial s})^2 + (\frac{\partial y}{\partial s})^2}}{\sqrt{(\frac{\partial x}{\partial s} + (\frac{\partial p}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial s})v_x)^2 + (\frac{\partial y}{\partial s} + (\frac{\partial p}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial s})v_y)^2}}.$$

In order to understand the consequences of this constraint, let us look at particular cases using a simplified light field with $v_y = 0$ (i.e. the three-dimensional EPI volume). Assume we impose that the layers are fronto-parallel such as the one illustrated in Figure 4.3(a). In this case the $p_n(s,\tau)$ is constant since all the lines in the EPI are parallel and we have $\chi_n(\vec{\sigma}) = 1$ (see Figures 4.3(b-c)). The intuition behind this property is easily grasped. Given the fact that the layer is fronto-parallel, its contour will simply be a translated version of itself on all the images.[1] Moreover, every point will contribute with the same weight since χ is unity. Assume now that the layer is slanted as illustrated in Figure 4.3(d). In this case, we have $p_n(s,\tau) = a_1 x(s,\tau) + a_0$ and therefore $\frac{\partial p}{\partial x} = a_1$ and $\frac{\partial p}{\partial y} = 0$. An example of an EPI with this depth model is illustrated in Figure 4.3(e). The weighting function becomes:

$$\chi(s, v_x, \tau) = \frac{(1 + a_1 v_x)\sqrt{(\frac{\partial x}{\partial s})^2 + (\frac{\partial y}{\partial s})^2}}{\sqrt{(\frac{\partial x}{\partial s}(1 + a_1 v_x))^2 + (\frac{\partial y}{\partial s})^2}},$$

[1] Recall that we assume the pinhole camera model.

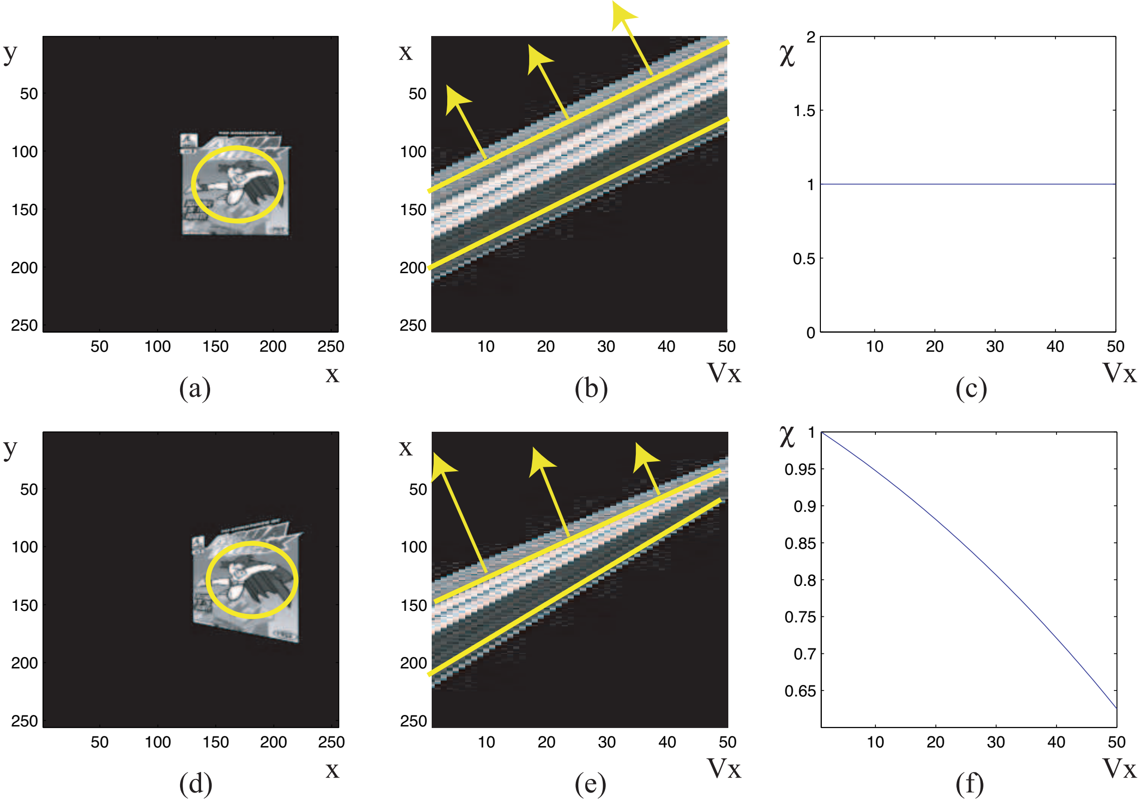


Figure 4.3: Normal speed correspondences in the fronto-parallel and slanted plane cases. Figures 4.3(a-c) illustrate the first image, the EPI and the χ respectively for the fronto-parallel case. Figures 4.3(d-f) illustrate the slanted plane case.

of which an example is plotted in Figure 4.3(f). Note that the $\chi(s, v_x, \tau)$ falls down to zero in the point where all the lines in the EPI intersect (i.e. $v_x = -1/a_1$). This corresponds to the degenerate case where the normal of the slanted plane is orthogonal with the line connecting the plane and the camera center in v_x. Therefore the plane is effectively not visible from this viewpoint which physically justifies the $\chi = 0$. Note also that more weight (i.e. a bigger χ) is attributed to the cameras in the viewpoints v_x where the texture of the slanted plane is better sampled. This again makes physical sense.

4.4.2 Occlusion constraints

In the previous section, we have seen how to constrain the shape of the plenoptic hypervolume depending on the geometry of the layer and the camera setup. The second main factor to consider is occlusion and the occlusion ordering. Let $\mathcal{H}_n$ be the plenoptic hyper-

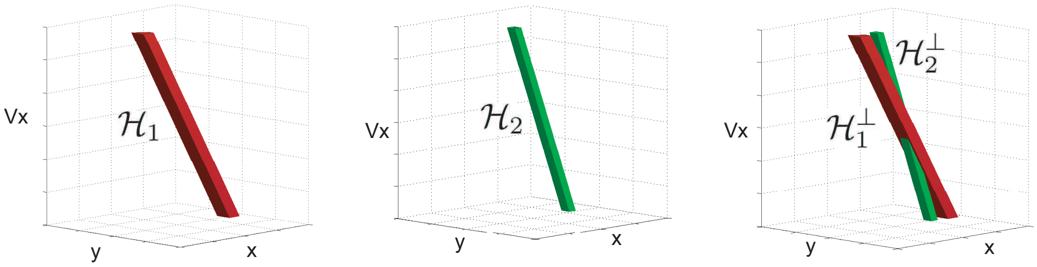


Figure 4.4: The occlusion constraint with two 3D plenoptic hypervolumes under the EPI parameterization. When put together, plenoptic hypervolumes $\mathcal{H}_1$ and $\mathcal{H}_2$ become $\mathcal{H}_1^\perp$ and $\mathcal{H}_2^\perp$. The occlusion constraint says that $\mathcal{H}_1^\perp = \mathcal{H}_1$ and $\mathcal{H}_2^\perp = \mathcal{H}_2 \cap \overline{\mathcal{H}_1^\perp}$.

volume as if it was not occluded and $\mathcal{H}_n^\perp$ denote the hypervolume with the occluded areas removed as illustrates in Figure 4.4. Assuming the camera centers lie on a line or a plane (as in the light field parameterization), the occlusion ordering stays constant throughout the views. Therefore, if the $\mathcal{H}_n$s are ordered from front ($n = 1$) to back ($n = N$), the occlusion constraint [5,7] can be written as

$$\mathcal{H}_n^\perp = \mathcal{H}_n \cap \sum_{i=1}^{n-1} \overline{\mathcal{H}_i^\perp}, \tag{4.10}$$

where $\cdot^\perp$ denotes that the plenoptic hypervolume has been geometrically orthogonalized such that the occluding hypervolume carve through the background ones (see Figure 4.4). A commonly used approach to deal with occlusions in EPI analysis is to start by extracting the frontmost regions (or lines) and removing them from further consideration [24,31]. That is, when extracting each $\mathcal{H}_n$, the $\sum_{i=1}^{n-1} \overline{\mathcal{H}_i^\perp}$ is known. While this approach is straightforward, it has some drawbacks. First, the extraction of occluded objects will depend on how well the occluding objects were extracted. Second, it does not enable a proper competition formulation since the background regions are not being estimated at the time of the extraction of the foreground ones.

4.5 A constrained region competition method

From Sections 4.4.1 and 4.4.2, we know that the shapes of the plenoptic hypervolumes in the light field are constrained. In this section, we show how the evolution equation for the deformable model can be modified in order to take into account these constraints. In particular, we show that the problem can be solved using a two-dimensional active contour method instead of a four-dimensional one.

4.5.1 Estimation of the contours given the depth models

In this step of the energy minimization, we fix the parameters p_n and seek to solve

$$\underset{\{\vec{\Gamma}_n\}}{\operatorname{argmin}} \int_{\mathcal{H}_n^{\perp}(\tau)} d_n(\vec{x}, p_n) d\vec{x} + \int_{\overline{\mathcal{H}_n^{\perp}(\tau)}} d_n^{out}(\vec{x}, p_n^{out}) d\vec{x},$$

subject to the constraints in (4.8) and (4.10). Consider the following manipulation to the gradient of the energy functional in (4.6):

$$\begin{aligned}
\frac{dE_n(\tau)}{d\tau} &= \int_{\vec{\Gamma}_n} [d_n(\vec{x}, p_n) - d_n^{out}(\vec{x}, p_n^{out})](\vec{v}_{\Gamma_n} \cdot \vec{n}_{\Gamma_n}) d\vec{\sigma} \\
&= \int_{\vec{\Gamma}_n} [d_n(\vec{x}, p_n) - d_n^{out}(\vec{x}, p_n^{out})] \chi_n(s, v_x, v_y)(\vec{v}_{\gamma_n} \cdot \vec{n}_{\gamma_n}) d\vec{\sigma} \\
&= \int_{\vec{\gamma}_n} (\vec{v}_{\gamma_n} \cdot \vec{n}_{\gamma_n}) \underbrace{\int\int O_n(\vec{x})[d_n(\vec{x}, p_n) - d_n^{out}(\vec{x}, p_n^{out})] \chi_n(s, v_x, v_y) dv_x dv_y}_{D_n(s, p_n) - D_n^{out}(s, p_n^{out})} ds,
\end{aligned}$$

where we have used (4.9) and the fact that $(\vec{v}_{\gamma_n} \cdot \vec{n}_{\gamma_n})$ does not depend on (v_x, v_y). The gradient of the energy can therefore be rewritten as

$$\frac{dE_{tot}(\tau)}{d\tau} = \int_{\vec{\gamma}_n} [D_n(s, p_n) - D_n^{out}(s, p_n^{out})](\vec{v}_{\gamma_n} \cdot \vec{n}_{\gamma_n}) ds,$$

where the $D_n(s, p_n)$ and $D_n^{out}(s, p_n^{out})$ are the original descriptors weighted by χ and integrated over the lines delimiting the hypersurface $\vec{\Gamma}_n$. This leads to a new evolution equation for the 2D contour:

$$\vec{v}_{\gamma_n} = [D_n^{out}(s, p_n^{out}) - D_n(s, p_n)]\vec{n}_{\gamma_n}.$$

In practice, we use the evolution equation

$$\vec{v}_{\gamma_n} = [D_n^{out}(s, p_n^{out}) - D_n(s, p_n) - \mu\kappa_n(s)]\vec{n}_{\gamma_n}, \tag{4.11}$$

where a smoothness term proportional to the curvature $\kappa_n(s)$ of $\vec{\gamma}_n$ is added in order insure regular curves and reject outliers. The μ is a positive constant weighting factor determining the influence of the regularization term. There are several advantages to using evolution equation (4.11) as opposed to (4.7). First, it constrains the shape of the hypervolume according to the camera setup. Second, it is implemented as an active contour in two dimensions instead of an active hypersurface in four dimensions which reduces the computational complexity.

We now have a constrained evolution that takes into account the geometry constraint. The second constraint to take into account are occlusions. Due to the occlusion ordering in (4.10), a foreground region evolving changes the background regions. That is, the evolution of $\mathcal{H}_n^{\perp}$ changes all the $\mathcal{H}_i^{\perp}$ where i goes from $n+1$ to N. Hence these occluded regions will contribute to the $\frac{dE_{tot}^n(\tau)}{d\tau}$ which leads to a competition. However, the other hypervolumes (i.e. $\mathcal{H}_i^{\perp}$ where i goes from 1 to $n-1$) are not affected by the shape changes in $\mathcal{H}_n^{\perp}$ and thus do not contribute to the derivative with respect to τ. Hence they will not compete. Take for example the case where there are two layers and a background, and we are evolving the first hypervolume. The regions are:

$$\begin{aligned}
\mathcal{H}_1^{\perp}(\tau) &= \mathcal{H}_1(\tau) \\
\mathcal{H}_2^{\perp}(\tau) &= \mathcal{H}_2 \cap \overline{\mathcal{H}_1^{\perp}}(\tau) \\
\mathcal{H}_3^{\perp}(\tau) &= \mathcal{D} \cap (\overline{\mathcal{H}_1^{\perp}}(\tau) \cap \overline{\mathcal{H}_2^{\perp}}(\tau)),
\end{aligned}$$

where we notice that all three regions are evolving (i.e. they depend on τ). Now assume we fix $\mathcal{H}_1^{\perp}$ and evolve $\mathcal{H}_2^{\perp}$. The occlusion constraint says that

$$\begin{aligned}
\mathcal{H}_1^{\perp} &= \mathcal{H}_1 \\
\mathcal{H}_2^{\perp}(\tau) &= \mathcal{H}_2(\tau) \cap \overline{\mathcal{H}_1^{\perp}} \\
\mathcal{H}_3^{\perp}(\tau) &= \mathcal{D} \cap (\overline{\mathcal{H}_1^{\perp}} \cap \overline{\mathcal{H}_2^{\perp}}(\tau)),
\end{aligned}$$

which shows that $\mathcal{H}_1^{\perp}$ does not depend on τ and hence will not contribute to the derivative of $E_{tot}^2(\tau)$. However the background $\mathcal{H}_3^{\perp}(\tau)$ is changing and is therefore in competition with $\mathcal{H}_2^{\perp}(\tau)$. This condition can be translated into the energy minimization by posing

$$d_n^{out}(\vec{x}, p_n^{out}) = \begin{cases} d_i(\vec{x}, p_i) & \forall \vec{x} \in \mathcal{H}_i^{\perp} \quad \text{and } i > n \\ 0 & \forall \vec{x} \in \mathcal{H}_i^{\perp} \quad \text{and } i < n. \end{cases}$$

This constraint together with (4.10) and (4.8) insure that the extracted plenoptic hypervolumes have a shape that is consistent with structure of the plenoptic function.

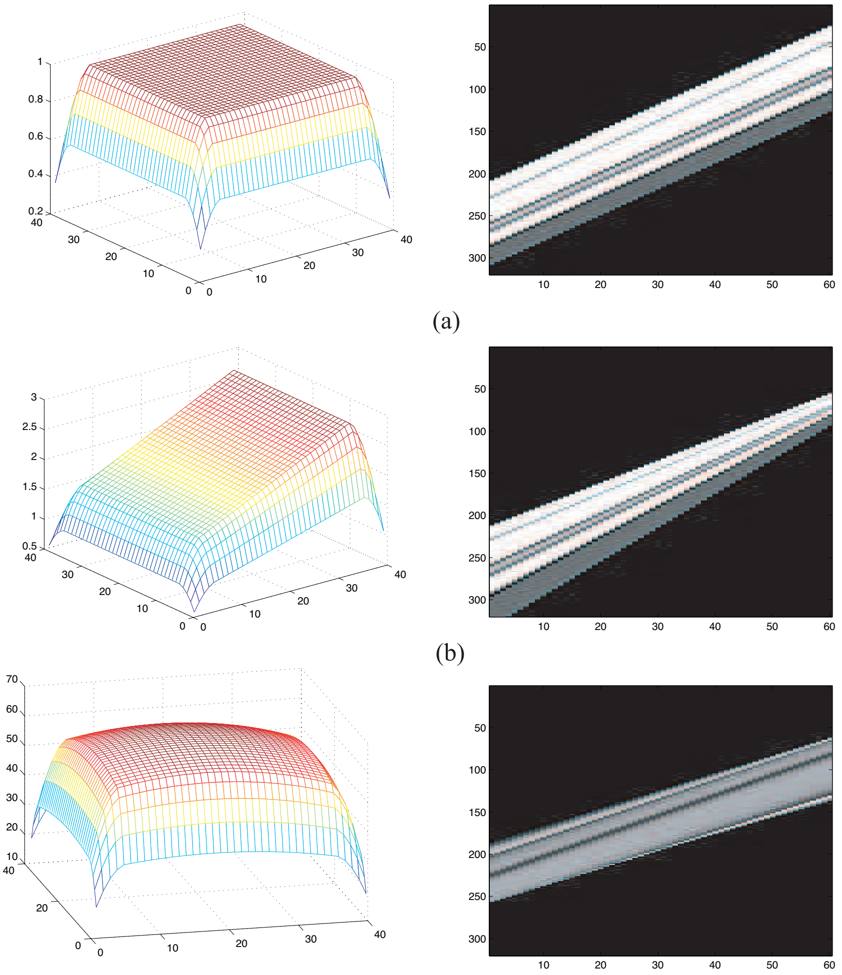
While it is preferable in most cases to use a competition-based active contour, it is relevant to mention here that the approach presented in this section is compatible with a threshold-based method. Indeed, choosing $D_n^{out}(s, p_n^{out}) = \zeta$ where ζ is a positive constant parameter leads to the evolution equation

$$\vec{v}_{\gamma_n} = [\zeta - D_n(s, p_n) - \mu\kappa(s)]\vec{n}_{\gamma_n}. \tag{4.12}$$

This evolution requires less computation since the competition term does not need to be computed. The resulting active contour will incorporate points where $D_n(s, p_n)$ is smaller than the preset threshold ζ. Points where $D_n(s, p_n)$ is larger than ζ will be rejected.

4.5.2 Estimation of the depth models given the contours

In the previous section, we derived a constrained curve evolution adapted to the structure of the light field. The counterpart of the curve evolution in the minimization of (4.3) is



(a)

(b)

(c)

Figure 4.5: Depth modeling under the EPI parameterization. The depth map for each layer is modeled with a linear combination of bicubic splines. This enables one to model a wide variety of smooth depth maps including (a) constant depth, (b) slanted plane and (c) smooth surfaces. The left column illustrates the depth map and the right column illustrates an example of an EPI with the respective depth map.

the estimation of the depth parameters p_n. In this step, one has to solve

$$\underset{\{p_1,\dots,p_N\}}{\operatorname{argmin}} \sum_{n=1}^{N} \int_{\mathcal{H}_n^{\perp}} d_n(\vec{x}, p_n) d\vec{x} \tag{4.13}$$

which is a standard least squares problem. Inspired by the layer extraction method in [51], we model the depth map for the layer n as a linear combination of bicubic splines $\beta(x, y)$:

$$p_n(x, y) = \sum_{i,j} P_n(i, j)\beta(x - T_x i, y - T_y j), \tag{4.14}$$

where i and j are on a uniformly sampled grid and (T_x, T_y) define the grid size. The weights $P_n(i, j)$ are determined using non-linear optimization methods such as the ones in Matlab's optimization toolbox. There are several advantages to this particular depth model. First, a variety of smooth depth maps can be modeled such as the one illustrated in Figure 4.5(c). Second, only a limited amount of weights on control points need to be estimated depending on the lattice size. Finally, the depth map can be forced to model a simplified geometry if an accurate depth reconstruction is not necessary. For instance, strictly fronto-parallel regions can be extracted by forcing all the weights to be the same for a given layer as in Figure 4.5(a). Slanted planes also belong to the family of depth maps that can be modeled using splines as shown in Figure 4.5(b).

4.6 Initializing the algorithm

Initialization plays an important role in active contour methods. There are two reasons why this is the case. First, the evolution equations driving the deformable models are based on partial differential equations. The steady state solution (i.e. in $\tau = \infty$) is therefore dependent on the initial condition and the active contours might get 'stuck' in local minima. Second, the initialization decides how many hypervolumes are used to represent the light field data unless a specific layer merging and splitting step is added in the algorithm. In our case, we take the number of hypervolumes as an input to the algorithm. This enables the scheme to provide the user with a tradeoff in terms of computational complexity and accuracy of segmentation (i.e. number of depth layers used).

In order to start the segmentation scheme, the algorithm needs initial estimates for contours $\vec{\gamma}_n$ of the plenoptic hypervolumes and each of their depth models p_n. Three main steps are involved in this estimation: First, estimate a dense or sparse depth map for each or a subset of the images in the light field. One may use any of the following methods: (a) a known depth map obtained for instance with range finding equipment, (b) two-view or N-view stereo algorithms with a single-valued or multiple-valued depth map as output [59] or (c) the slopes of the lines in the EPI of the light field [12, 31]. Second, classify points in the light field that follow a particular depth model. In our implementation, regions in the images that have a similar depth are merged into a single plenoptic hypervolume. The number of bins in which to merge the regions with similar depth is defined by the number of layers to represent the light field. Third, initialize the level set functions ϕ_n for each in the evolving curves $\vec{\gamma}_n$ by projecting all the light field points onto the image in $v_x = v_y = 0$. Note that since the level-set method is topology independent, it is not necessary for the initial ϕ_n to have the right topology.

4.7 Overall optimization

We perform the overall energy minimization after initialization in three iterative steps: First, estimate the depth parameters in each individual hypervolume $\mathcal{H}_n^{\perp}$ using classical non-linear optimization methods and the depth model in (4.14). Second, update the occluded regions using the occlusion constraint in (4.10). Third, evolve each boundary $\vec{\gamma}_n$ individually using the evolution equation in (4.11) and the level-set method. In this step, one may use one or a few iterations of the evolution depending on how often the disparity model needs to be adjusted. The algorithm is stopped when there is no significant decrease in the total energy or after a predetermined number of iterations. Table 4.1 illustrates how the shape and occlusion constraints are applied in the segmentation scheme by giving an overview of the algorithm.

4.8 Summary and key results

In this chapter, we posed the problem of extracting coherent layers in a multi-dimensional variational framework. We used the classical active contour methods in order to derive an evolution equation for the deformable models. However, instead of applying these methods

```
Step 1: Initialize a set of plenoptic hypervolumes $\mathcal{H}_n^{\perp}$ characterized by their 2D contours
        $\vec{\gamma}_n$ and depth parameters $p_n$
Step 2: Estimate depth parameters $p_n$ given the contours $\vec{\gamma}_n$
Step 3: Update occlusion ordering and update hypervolumes with $\mathcal{H}_n^{\perp} = \mathcal{H}_n \cap \sum_{i=1}^{n-1} \overline{\mathcal{H}_i^{\perp}}$
Step 4: For each $\vec{\gamma}_n$

            Fix the other $\vec{\gamma}_i$ for $i \neq n$

            Compute speed function with competition terms
                $D_n(s, p_n) = \int\int O_n(\vec{x}) d_n(\vec{x}, p_n) \chi(s, v_x, v_y) dv_x dv_y$
            and
                $D_n^{out}(s, p_n^{out}) = \int\int O_n(\vec{x}) d_n^{out}(\vec{x}, p_n^{out}) \chi(s, v_x, v_y) dv_x dv_y$
            with
                $d_n^{out}(\vec{x}, p_n^{out}) = \begin{cases} d_i(\vec{x}, p_i) & \forall \vec{x} \in \mathcal{H}_i^{\perp} \text{ and } i > n \\ 0 & \forall \vec{x} \in \mathcal{H}_i^{\perp} \text{ and } i < n \end{cases}$

            Evolve contour $\vec{\gamma}_n$ with evolution equation
                $\vec{v}_{\gamma_n} = [D_n^{out}(s, p_n^{out}) - D_n(s, p_n) - \mu \kappa_n(s)] \vec{n}_{\gamma_n}$

Step 5: Go to Step 2 or stop when there is no significant decrease in energy
```

Table 4.1: Overview of the plenoptic hypervolume extraction algorithm.

in a straightforward manner, we proposed several modifications that take into account the inherent nature of the plenoptic function. That is, points in space are mapped onto particular trajectories in the plenoptic domain (i.e. lines in light fields). This constraint can be imposed on the evolution equation by parameterizing the hypersurfaces in a particular way. Second, occlusions occur in a specific order. This leads to an occlusion ordering for the plenoptic hypervolumes in which foreground regions compete with the regions they occlude. The imposition of these constraints results in a novel multi-dimensional scheme. Since the formulation is global, coherence and consistency is enforced on all the four dimensions. Moreover, occlusions are naturally handled and all the images are treated equally and jointly.

The important aspects of the method presented in this chapter are highlighted in the following points:

- The shape of the plenoptic hypervolumes are constrained by the camera setup and occlusions.

- These constraints can be enforced on the evolution of the active hypersurfaces using a semi-parametric approach.
- The resulting framework is global and efficiently handles occlusions.

Chapter 5

Experimental results and image based rendering applications

5.1 Introduction

There are many applications that stand to benefit from the extraction of plenoptic hypervolumes. Throughout the previous chapters, we have put particular emphasis on image based rendering and the problem of interpolating new viewpoints for freeviewpoint television and immersive technologies. However, the segmentation of the plenoptic function is a useful step in other applications as well. In particular, occlusion removal and augmented reality where objects are removed from or inserted in the scene. In this chapter, we present results for all these applications.

Capturing light fields requires camera arrays. While it may be possible to create light field data synthetically, a particular effort was made to use only natural images. To this effect, a variety of data sets are analyzed with the plenoptic hypervolume extraction algorithm described in Chapter 4. Most of the image sequences are simplified light fields (i.e. EPI volumes). These are easier to capture and more practical to show the results while still portraying the concepts well. The data sets include some taken from standardized multiview image sequences. We also captured several multiview images using a camera mounted on a rail. With these results, we show that the algorithm is practical and is able

Name	Number of cameras	Image size	Origin
Cones	(1 by 9)	(375 by 450)	Middlebury stereo
Dwarves	(1 by 7)	(555 by 695)	Middlebury stereo
EE lobby	(1 by 5)	(800 by 800)	Self-acquired
Tank	(1 by 15)	(250 by 250)	Self-acquired
Animal family	(1 by 32)	(235 by 625)	Self-acquired
Desk	(4 by 4)	(500 by 500)	Self-acquired

Table 5.1: Overview of the light fields.

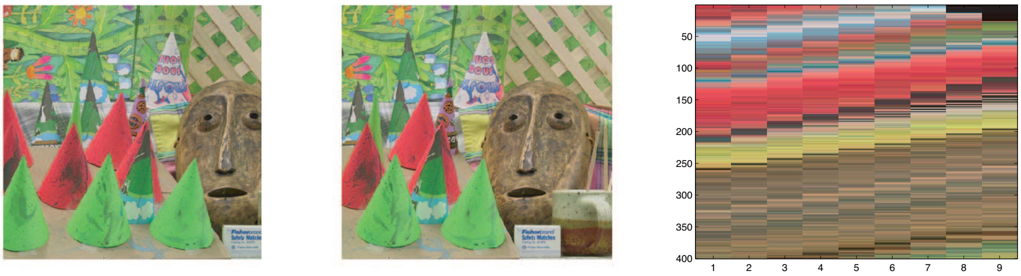
to cope with images taken in a controlled laboratory environment as well as real-world environments. Some scenes were chosen to demonstrate the quality of the interpolated images using the proposed approach. Others were chosen to show the robustness of the algorithm with respect to occlusions. Note that, due to the absence of a ground truth, we will present mainly qualitative results however some quantitative measures may be applied by performing leave-one-out tests for view interpolation applications.

The chapter is organized as follows: In Section 5.2 we present an overview of the data sets and their origin. Section 5.3 illustrates light field segmentation results for different data sets and numbers of layers chosen to represent the light field. Section 5.4 presents a simple view interpolation algorithm based on the extraction of plenoptic hypervolumes and shows some rendered images. Section 5.5 illustrates layer and scene manipulation results. Finally, we conclude in Section 5.6.

5.2 Data sets

A number of light fields have been analyzed. In this section, we describe where and how these data sets have been acquired. The *Cones* and the *Dwarves* images were obtained from the Middlebury stereo vision website[1] and are depicted in Figures 5.1 and 5.2 respectively. These data sets are calibrated multi-baseline stereo images that were captured in a controlled environment. The *EE lobby* and *Tank* data sets illustrated in Figures 5.3 and 5.4 were captured by translating a camera using a manually controlled rail. All the equipment used to captured these light fields is available off-the-shelf. Note that the lobby

[1] http://vision.middlebury.edu/stereo/



(a) (b) (c)

Figure 5.1: *Cones* data set. The first and the last images are depicted in (a) and (b) respectively. The EPI at slice $y = 200$ is shown in (c).

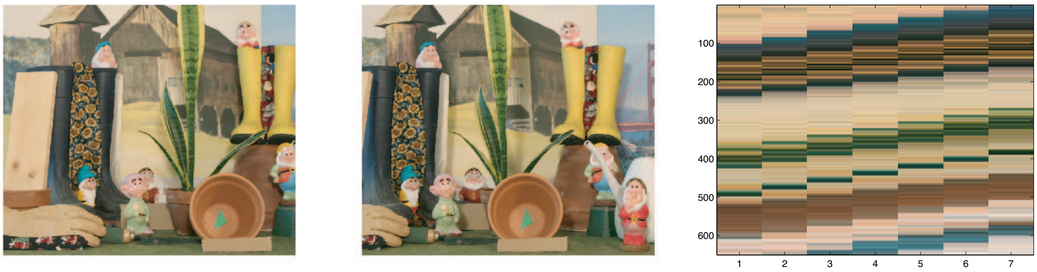
sequence is a natural environment where lighting is not controlled. Finally, the *Animal family* and *Desk* data sets depicted in Figures 5.5 and 2.6 were captured using a computer controlled robotic gantry.[2] We used a Nikon D50 camera to capture the images. In all the acquisitions, we did not use the flash and switched off the auto-focus function. The zoom was also kept constant and the white balance was fixed such that the gain remained constant throughout the capturing process. Finally, the images were captured in jpeg format with a resolution of 1504 x 1000. Note that the images were cropped. The camera was calibrated using a calibration checkerboard and a freely available Matlab calibration software[3] which is loosely based on [87]. Note that the extrinsic parameters are not computed in our case since the viewpoints are assumed to be uniformly sampled and placed along a line or on a plane. The calibration was performed primarily to correct the distortions caused by the lens. Table 5.1 shows an overview of the light fields along with their image sizes, number of images and origin. All the light fields have a dynamic range from 0 to 255.

5.3 Segmentation results

In this section, we present light field segmentation results. All the image sequences were converted to grayscale for processing. For all the data sets, the layers were initialized

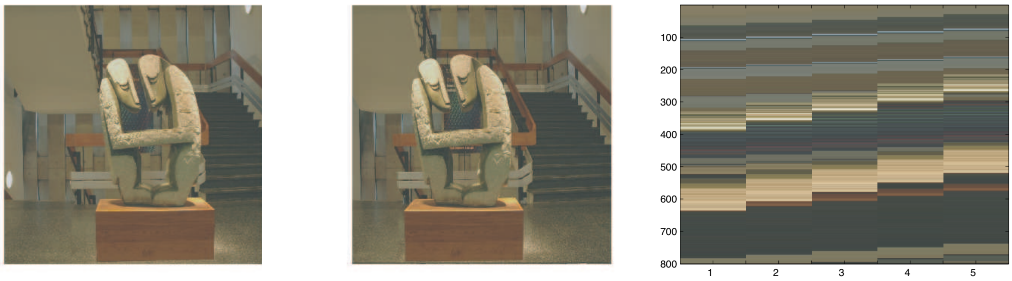
[2] The author would like to acknowledge the Audiovisual Communications Laboratory at the Swiss Federal Institute of Technology (EPFL) for providing the equipment to capture these data sets.

[3] http://www.vision.caltech.edu/bouguetj/



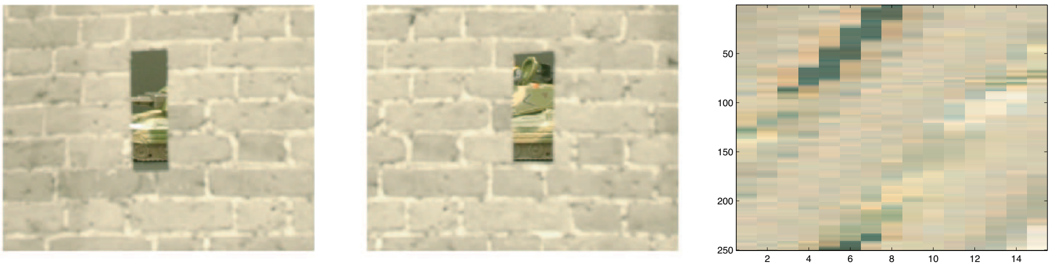
(a) (b) (c)

Figure 5.2: *Dwarves* data set. The first and the last images are depicted in (a) and (b) respectively. The EPI at slice $y = 350$ is shown in (c).



(a) (b) (c)

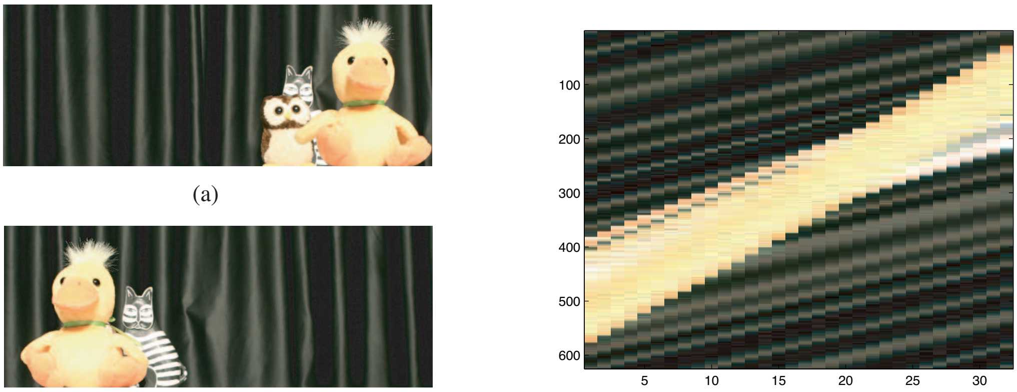
Figure 5.3: *EE lobby* data set. The first and the last images are depicted in (a) and (b) respectively. The EPI at slice $y = 350$ is shown in (c).



(a) (b) (c)

Figure 5.4: *Tank* data set. The first and the last images are depicted in (a) and (b) respectively. The EPI at slice $y = 120$ is shown in (c).

using a state-of-the-art two-frame stereo algorithm [57]. The evolution of the curves was implemented using the level-set method. Finally, all the depth maps for the plenoptic hypervolumes were forced to be constant. This is quite a limiting assumption. However, it is a valid one as our primary goal is interpolation and fronto-parallel regions in light



(a)

(b)

(c)

Figure 5.5: *Animal family* data set. The first and the last images are depicted in (a) and (b) respectively. The EPI at slice $y = 160$ is shown in (c).

Parameter	Value
μ	0.02
number of layers	variable
depth model	constant
total level set iterations	200
reinitialization to signed distance function every	50
estimation of depth parameters every	100

Table 5.2: Parameter values used in all the segmentation results.

fields can be rendered free of aliasing. Note that the algorithm is straightforwardly capable of dealing with more sophisticated depth maps, however, this would increase the segmentation time since more parameters need to be estimated. Table 5.2 summarizes the parameters of the segmentation scheme and quantifies how they were set for the results presented in this chapter. The regularization factor μ in (4.11) was set to 0.02 and all the speed functions for the evolving curves were normalized such that they are bounded by $[-1, 1]$. In practice, we found that this value was sufficient to ensure that the contours remain regular without smoothing the boundaries of the layers. In more noisy images, a larger value of μ might be chosen. The depth maps were estimated every 100 iterations of the level-set evolution and the level-set functions were reinitialized to a signed distance function every 50 iterations. This reinitialization was performed by running 100 iterations of the evolution equation in (3.9). All these parameters were chosen to provide reasonable results without running an excessive number of iterations. Note that in general the number of iterations for the level-set method to converge depends on the images.

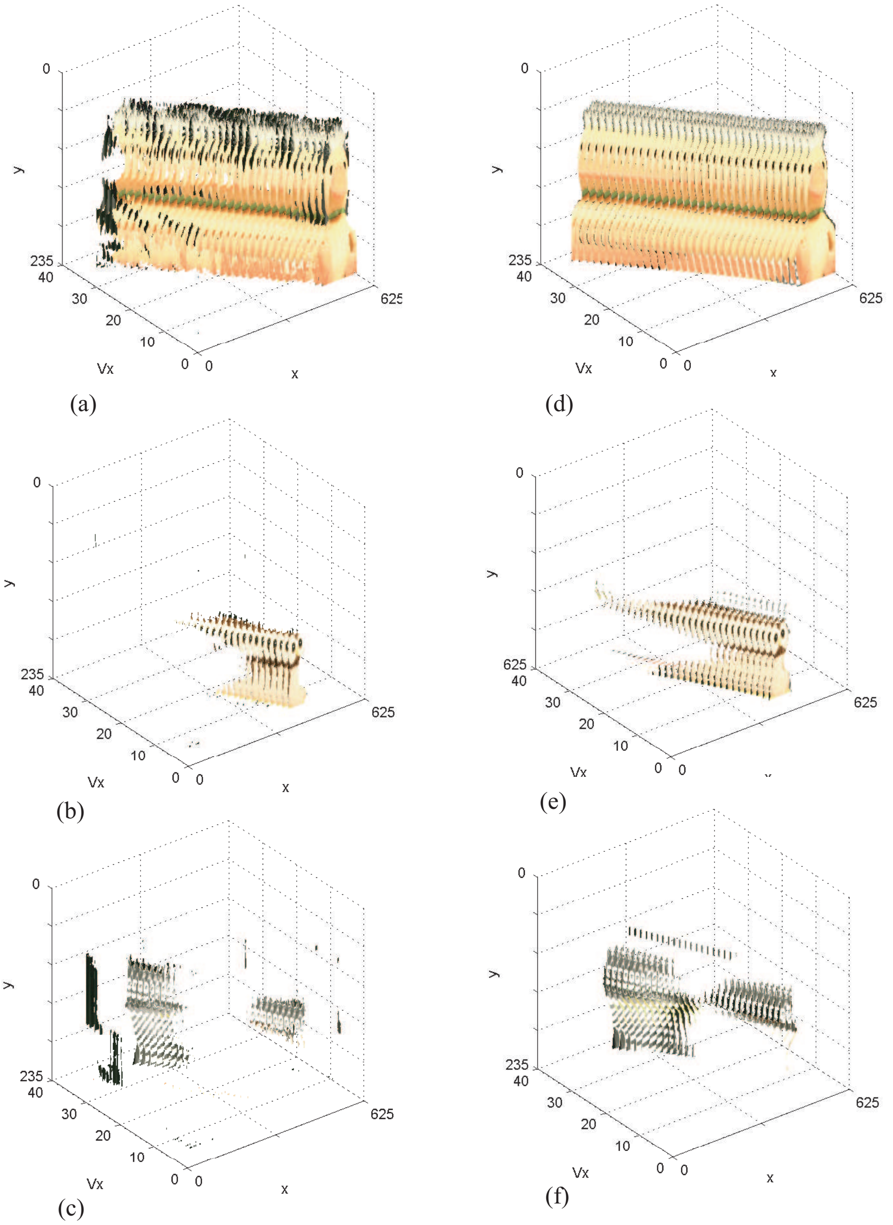


Figure 5.6: Comparison between the constrained and unconstrained volume evolutions on the *Animal family* light field. For both cases the initialization is identical. (a-c) Extracted volumes in the unconstrained case. (d-f) Extracted volumes in the constrained case.

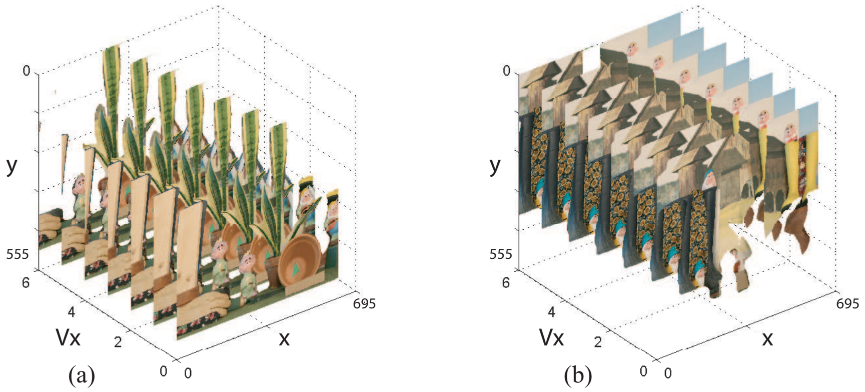


Figure 5.7: Two-layer representation of the *Dwarves* light field.

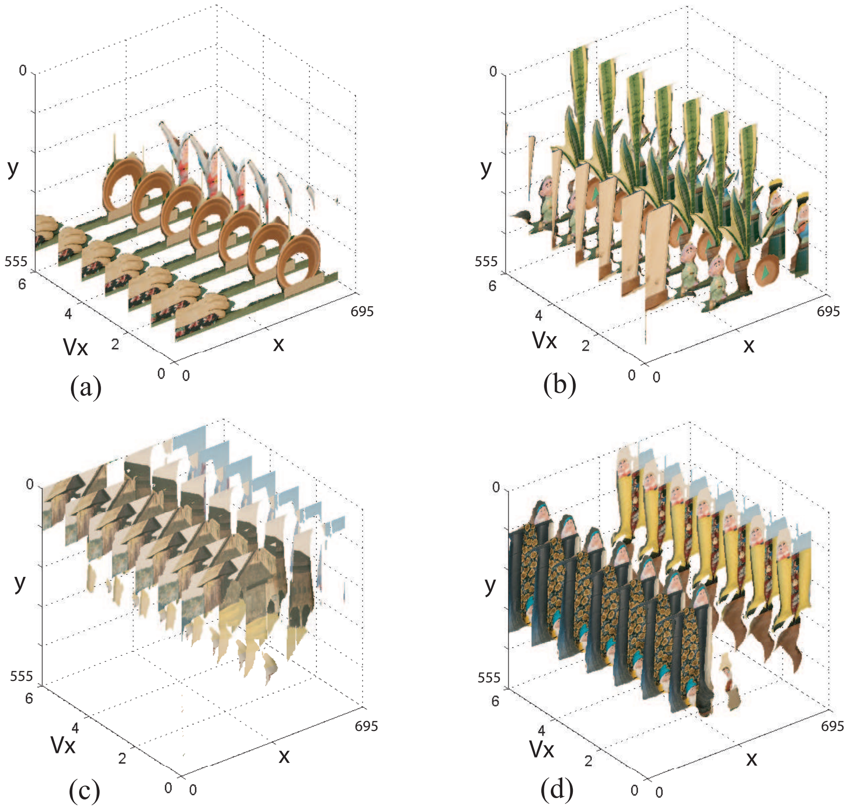


Figure 5.8: 12-layer representation of the *Dwarves* light field. (a) Layers 1-3, (b) layers 4-6, (c) layers 7-9 and (d) layers 10-12.

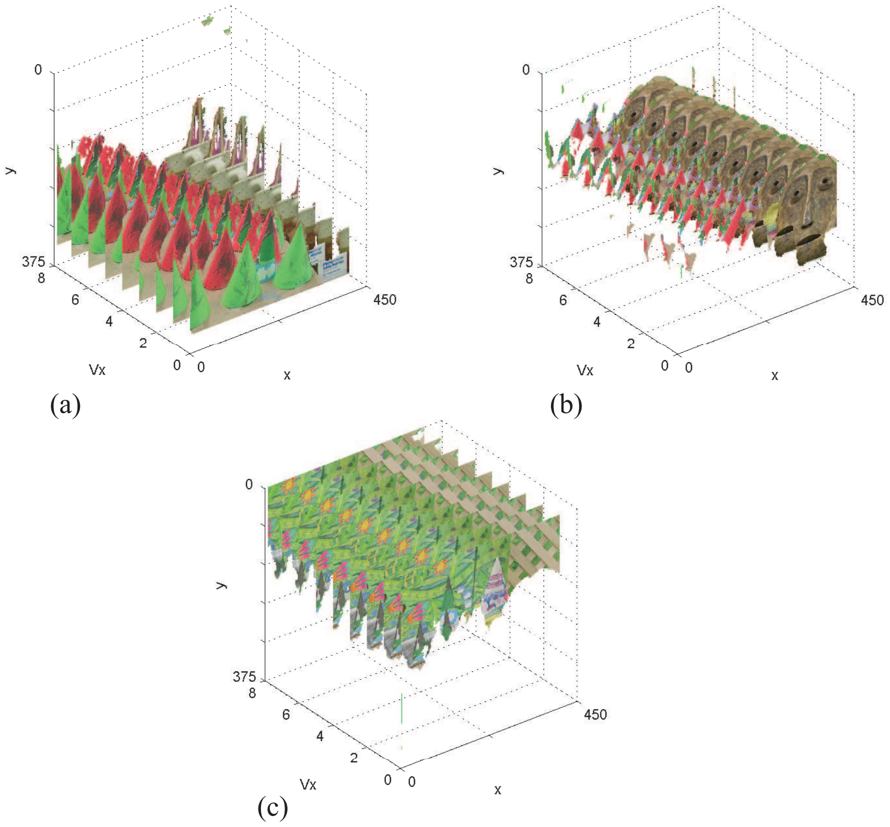
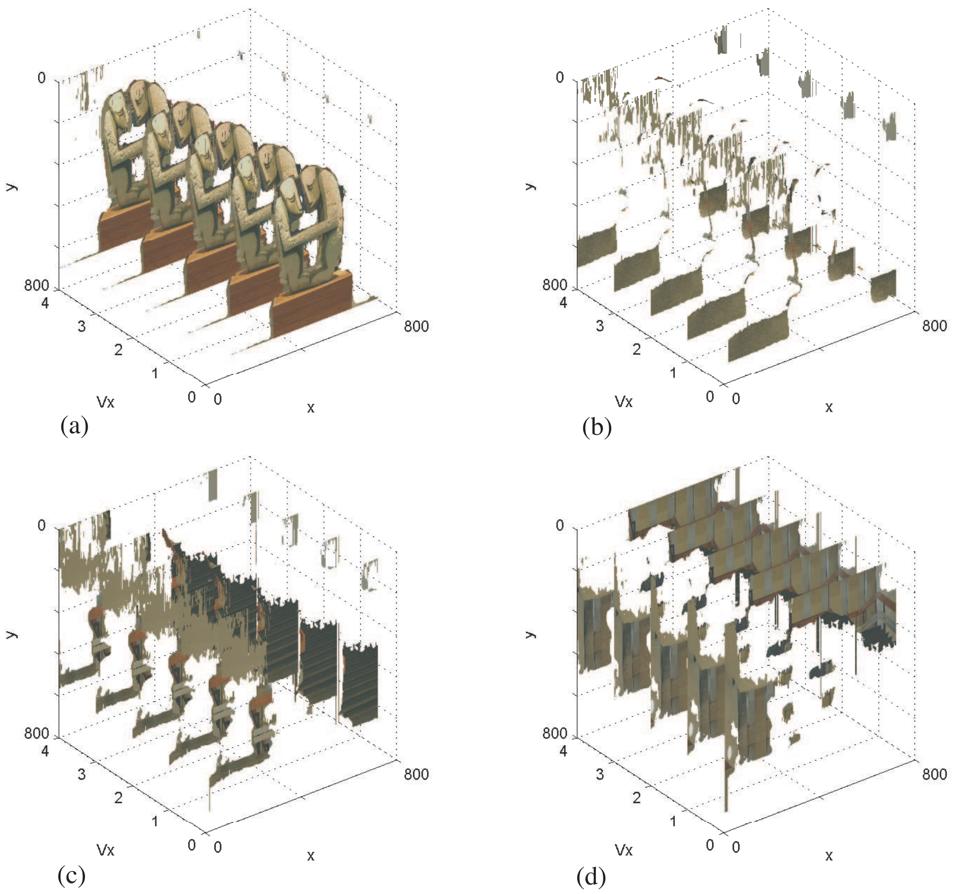


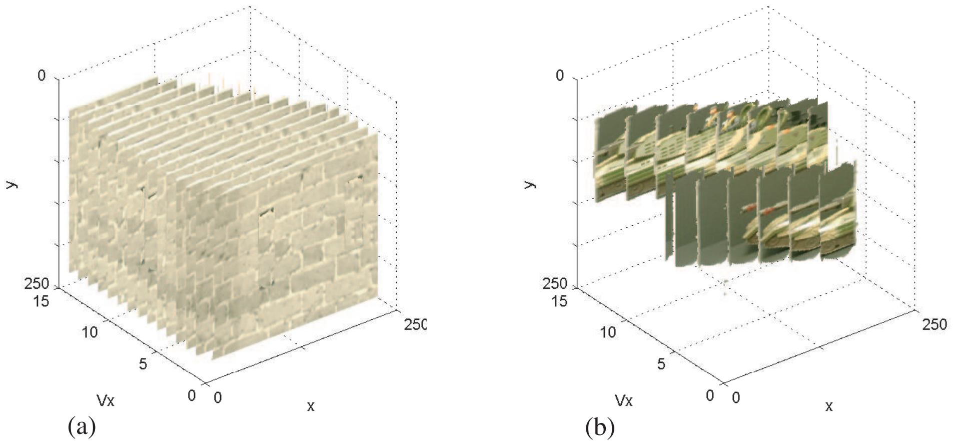
Figure 5.9: 10-layer representation of the *Cones* light field. (a) Layers 1-4, (b) layers 4-6, (c) layers 7-10.

The *Animal family* light field depicted in Figure 5.5 was chosen to show the resilience of the algorithm to occlusions. The scene is made of three layers and a background with images in which some of the layers are completely occluded. This is in particular the case for the ‘owl’ layer and the ‘cat’ layer which is visible, then totally occluded for a large number of frames and reappears. For this light field, we compare the results of the unconstrained evolution in Section 4.3 with the constrained evolution presented in Section 4.5. For both cases, the initialization and the number of iterations was identical. From Figures 5.6(a-c), it is clear that the extracted volumes correspond rather well to the scene. However, there are some errors in particular on areas that are barely visible such as the ‘owl’ layer. The extracted regions and in particular the foreground layer in Figure 5.6(a), are not consistent throughout the views. In the constrained case depicted in Figures 5.6(d-f), the extracted volumes are coherent throughout all the views and oc-



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

Figure 5.10: 4-layer representation of the *EE lobby* light field.



(a) (b)

Figure 5.11: Two-layer representation of the *Tank* light field.

clusions are better captured. This comes from the fact that all the images are treated jointly and the plenoptic constraints are applied. The *Dwarves* and *Cones* light fields have many objects with textured and textureless regions as well as occlusions. Segmentation results for the *Dwarves* data set are illustrated in Figures 5.7 and 5.8 for the two-layer and 12-layer representations respectively. Segmentation results for the *Cones* light field are shown in Figure 5.9. In general, the volumes are consistent with the scene despite the occlusions and disocclusions. The volumes also do not show the discontinuities across scan lines that may occur when individual EPIs (i.e. slices of the volume) are analyzed such as in [24]. All these aspects come from the fact that the data is analyzed in a global manner and coherence is enforced in all the dimensions. We note, however, that some errors occur in large textureless regions due to the absence of a colour-based term in the energy minimization. These errors will not significantly hinder the rendering quality since the interpolation will be applied to constant intensity regions. Segmentations of the *EE lobby* and the *Tank* light fields are illustrated in Figures 5.10 and 5.11 respectively.

Table 5.3 shows the segmentation times for some of the light fields. The segmentation algorithm was programmed in a combination of Matlab and C using the mex compiler. In this context, the segmentation scheme requires processing times ranging roughly from 100 to 1200 seconds on a 2.8 GHz Pentium-IV PC. The main factors influencing these times are the images sizes, the number of images and the number of layers chosen to represent the light fields. These times result from the fact the level-set method in its original form has a high computational complexity. On top of that, no particular attempt was made to use more efficient implementations. It is worth mentioning that faster methods exist and improvements of several orders of magnitude could be expected [64]. Nevertheless, these segmentation times remain reasonable.

5.4 Light field interpolation

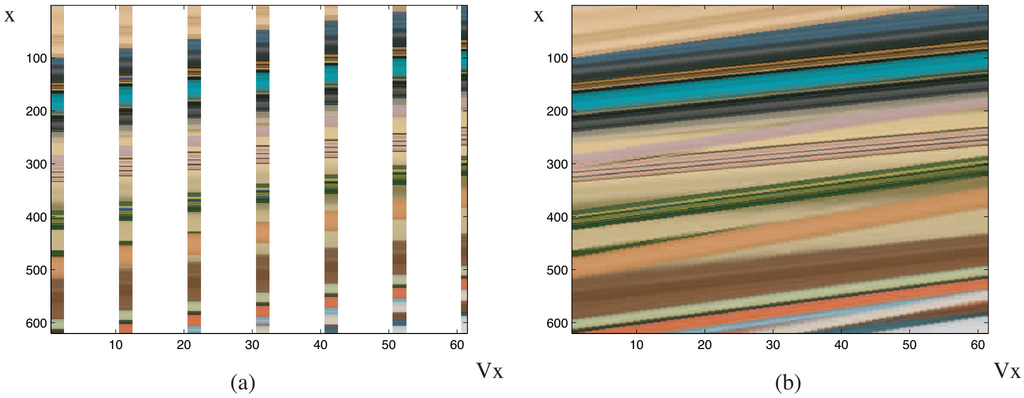
In this section, we illustrate light field rendering results. All the interpolated images are obtained through depth corrected linear interpolation as described in Section 2.3 using the approximate depth estimated by the plenoptic hypervolume extraction algorithm. That

Data set	Num. of layers	Segmentation time	Rendering time (10 frames)	Rendering time per frame
Cones	10	438	3.5	0.35
Dwarves	2	141	5.9	0.59
"	5	401	5.9	0.59
"	12	1101	6.7	0.67
Animal family	4	540	6.2	0.62
Desk	5	1181	8.4	0.84
EE lobby	4	583	8.3	0.83

Table 5.3: Running times (in seconds) for different light fields.

is, the intensity to interpolate $\tilde{I}(x, y, v_x, v_y)$ is linearly interpolated from the available samples and the depth corrected kernel in (2.5). Particular attention is taken not to take into account occluded pixels. That is, sample points $I[i, j, k, l]$ that do not belong to the same layer (i.e. occluded regions) are disregarded in the interpolation. In all our results, we found that a binary alpha map for each of the layers produced good rendering results. It may be worthwhile for some light fields to use a more sophisticated coherence matting approach [69] to blend the layers. All the results are compared to conventional light field rendering [18, 50]. That is, we used a single plane placed at the optimal depth in (2.6). Rendering times for the light fields are given in Table 5.3. Note that here again no particular effort was made to optimize the code for speed. Real-time light field rendering using a similar layered representation has been reported [69].

The interpolation of the *Dwarves* light field is illustrated in Figures 5.12 and 5.13. In Figure 5.12, we show the sampled EPI for a slice of the light field and its interpolated version which has a ten-fold increase of viewpoints v_x. Note that the interpolated EPI in Figure 5.12(b) has a structure which is consistent with the structure expected in an EPI. Indeed, both the geometry and the occlusion constraints of Chapter 2 are satisfied. That is, the EPI is made of a collection of lines and the ones with larger slopes occlude the ones with smaller slopes. Interpolated images are illustrated in Figure 5.13 along with zoomed regions corresponding to a 150 by 170 region of interest. We show the interpolation results using the optimal constant depth algorithm [18] presented in Chapter 2 and our layer-based scheme where the number of layers is defined by the user in the initialization



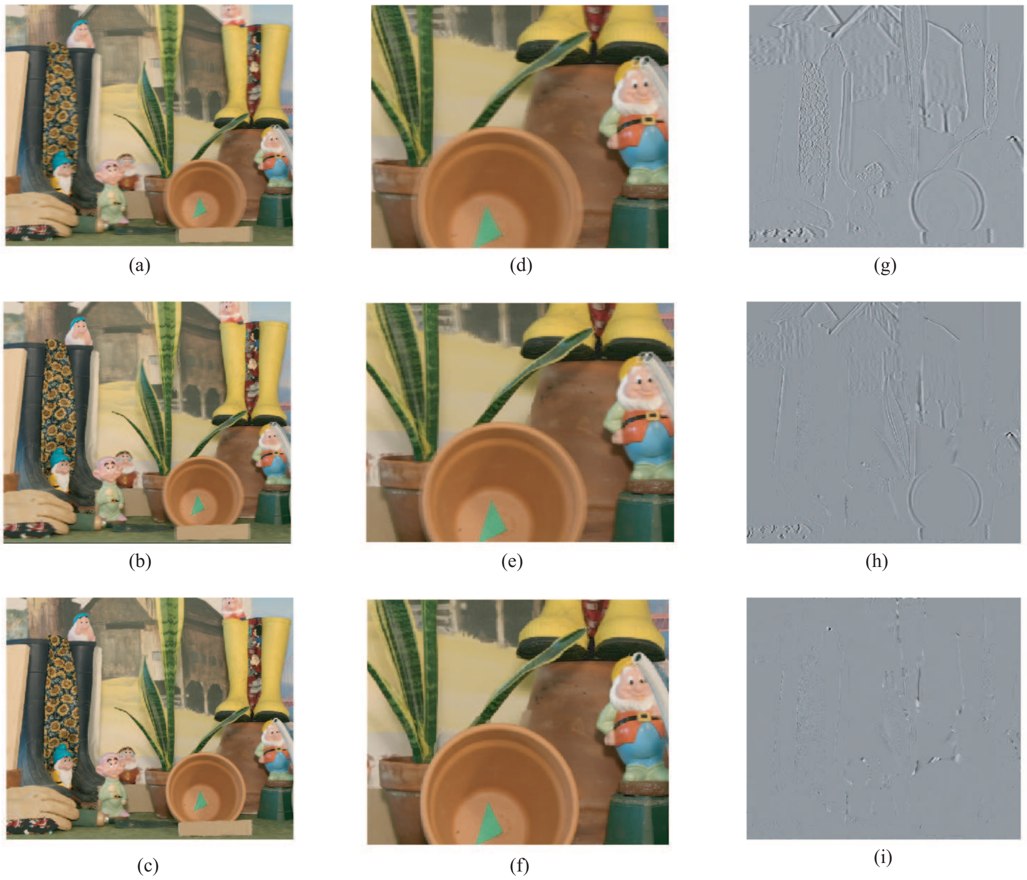
(a)

(b)

Figure 5.12: Interpolation of the *Dwarves* light field. In this example, (a) shows the available sampled EPI at slice $y = 375$ and (b) illustrates the interpolated EPI obtained with the 12-layer representation. Note that the interpolated EPI has a structure which is consistent with the structure of the plenoptic function. That is, it is made of a collection of lines and the ones with larger slopes occlude the ones with smaller slopes.

of the segmentation. On the whole, we notice that aliasing is visible in Figures 5.13(a) and 5.13(b). This blurring and ghosting is especially visible on the contour of the pot in the foreground and the nozzle of the watering can on the right hand side of the image. These effects are greatly reduced when using more layers as in Figure 5.13(c). For this light field, we performed a leave-one-out test by removing one of the original images and rendering it using the remaining ones. The difference between the ground truth image and the interpolated ones is shown in Figures 5.13(g-i). From these images, we notice that errors tend to occur on object boundaries and highly textured regions. However, the more layers we use, the better the result. This is visible in Figure 5.13(i) were it is clear that the errors are in general corrected with some more localized artifacts due to the segmentation accuracy. Table 5.4 gives signal-to-noise ratios between the reconstructed and ground truth images for different numbers of the layers chosen to represent the light field. The SNRs using the unconstrained segmentation are also given. In general, we note that the constrained evolution to extract the coherent regions produces a significant reduction in the reconstruction error. We also note that in general, an increase in the number layers to represent the light field produces the better results.

The interpolation of the *Cones* light field is shown in Figure 5.14. Here again,



(a) (d) (g)

(b) (e) (h)

(c) (f) (i)

Figure 5.13: Interpolation of the *Dwarves* light field. (a) Interpolated viewpoint using conventional light field rendering (i.e. a single constant depth plane at the optimal depth). (b) Interpolated viewpoint using a two-layer representation. (c) Interpolated viewpoint using a 12-layer representation. Images (d-f) illustrate a 150 by 170 pixel region of interest and (g-i) show the difference with the ground truth image.

we notice that the interpolated viewpoint in Figure 5.14(a) obtained using conventional light field rendering is blurred. This is because the sample images are too far apart and the resulting light field is undersampled. The interpolated viewpoint using the 10-layer representation illustrated in Figure 5.14(b) yields a good quality rendering. Differences from the ground truth image are illustrated in Figures 5.14(e) and 5.14(f). More rendering results of the *EE lobby* and *Desk* light fields are illustrated in Figures 5.15 and 5.16 respectively. Note that these scenes contain some shadows and light specular effects. Nevertheless, the rendered images are of good quality. Note also that despite the fronto-parallel depth model and small number of layers, the light fields are photorealistically

Number of layers	unconstr. evo.	constr. evo.
1	16.51	16.51
2	17.49	20.26
5	20.22	22.41
12	23.18	27.08

Table 5.4: Leave-one-out test on the *Dwarves* light field. The table gives the SNR (in dB) between the rendered image and the ground truth image for unconstrained and constrained layer extraction schemes as well as different numbers of layers.

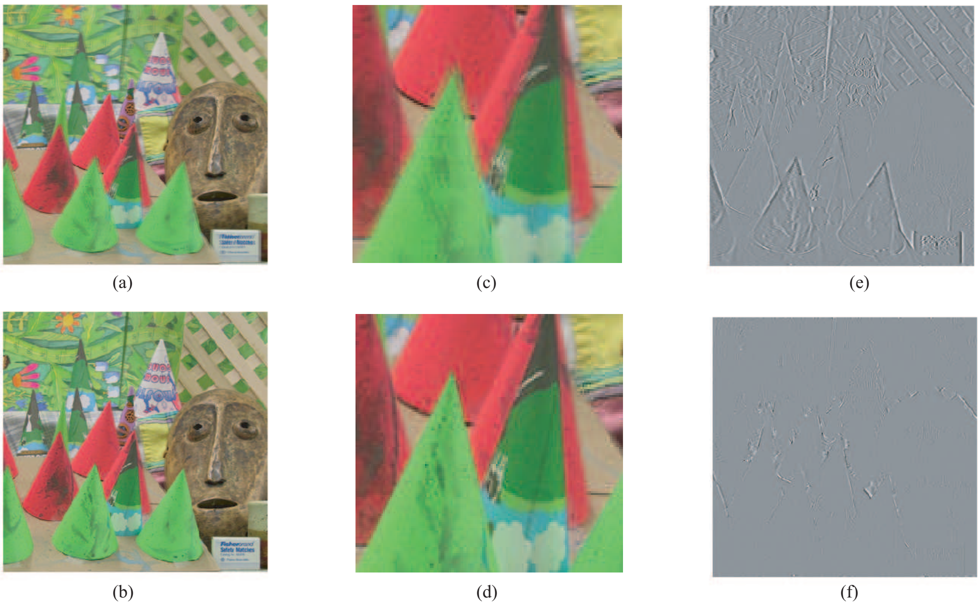
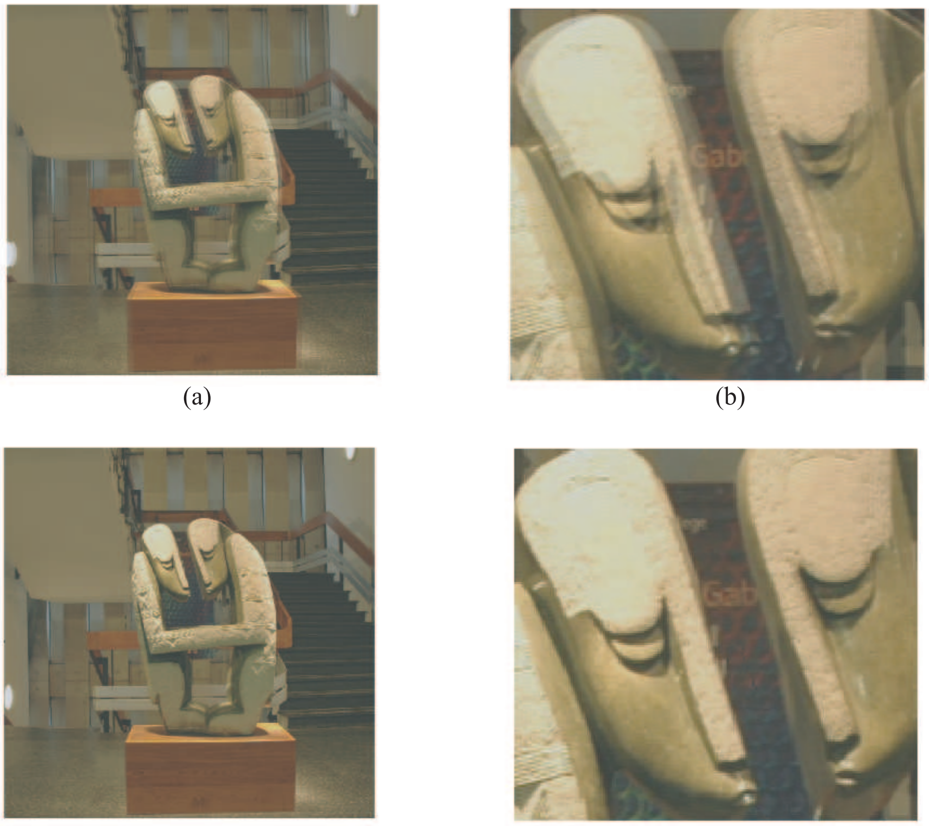


Figure 5.14: Interpolation of the *Cones* light field. (a) Interpolated viewpoint using conventional light field rendering (i.e. a single constant depth plane at the optimal depth). (b) Interpolated viewpoint using a 10-layer representation. Images (c-d) illustrate a 100 by 120 pixel region of interest and (e-f) show the difference with the ground truth image.

rendered.

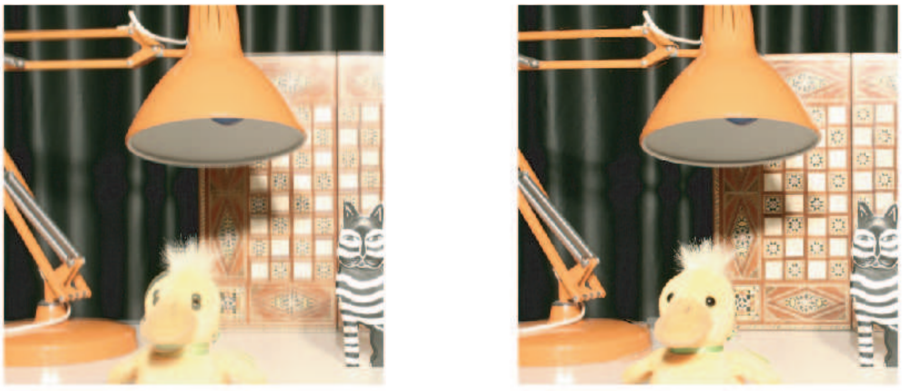
Finally, we tested the proposed method against a state-of-the art stereo algorithm [57] and the EPI analysis method [24]. The interpolation method was identical for all the methods. Table 5.5 shows the SNRs of the interpolated images with the ground truth images. The numbers in bold represent the best method for the given light field. In these cases, we note that the proposed method outperforms the other algorithms for all the light fields expect the *Cones* data set. Examples of rendered images using the



(a) (b)

(c) (d)

Figure 5.15: Interpolation of the *EE lobby* light field. (a-b) Interpolated viewpoint using conventional light field rendering. (c-d) Interpolated viewpoint using a 4-layer representation.



(a) (b)

Figure 5.16: Interpolation of the *Desk* light field. (a) Interpolated viewpoint using conventional light field rendering. (b) Interpolated viewpoint using a 5-layer representation.

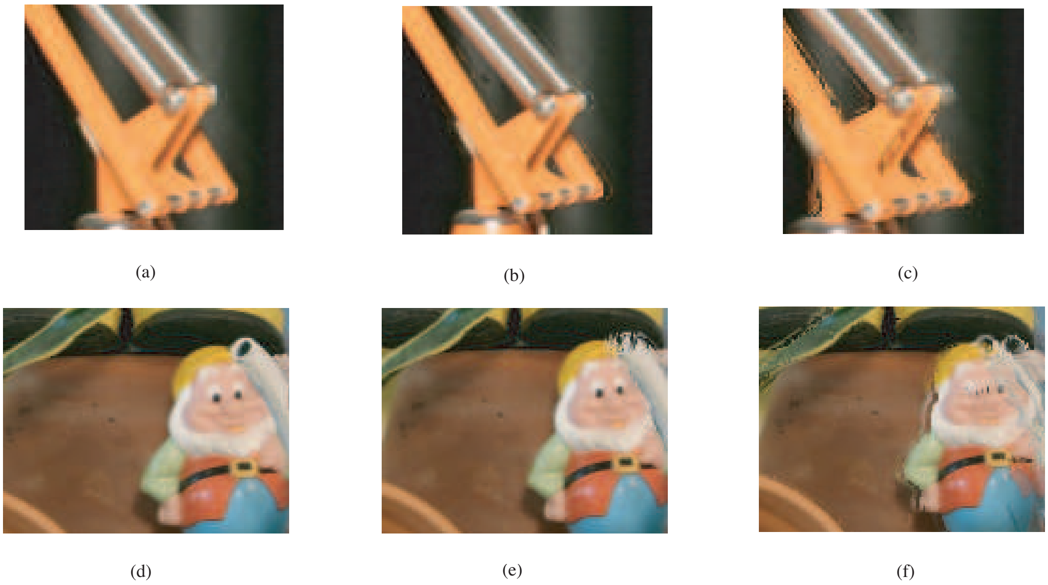
Data set	proposed method	stereo [57]	EPI analysis [24]
Cones	25.43	**29.91**	21.77
Dwarves	**27.08**	26.24	19.56
EE lobby	**17.65**	17.21	11.69
Animal family	**23.04**	17.47	19.02
Desk	**25.57**	25.46	19.79

Table 5.5: Leave-one-out test. The table gives the SNR (in dB) between the rendered image and the ground truth image for the proposed method as well as a stereo method and the EPI-tube extraction algorithm. The result in bold show the best method for the given light field.

different algorithms are illustrated in Figure 5.17. From these examples, we notice that object boarders are better defined and smoother renderings are achieved in the case of the coherent extraction of plenoptic volumes.

5.5 Extrapolation of occluded regions and augmented reality

In the previous section, we showed with experimental results that the decomposition of light fields into plenoptic hypervolumes with approximately constant depth enables one to render views from novel viewpoints. The extraction of plenoptic hypervolumes is also a useful step in many other applications including the extrapolation of occluded areas. The variational framework used takes into account all the images which makes it particularly resilient to occlusions. Similarly to layer-based representations [65, 78], the segmentation of light fields enables one to manipulate the data by recombining them in different ways. Note that a layer-based representation (i.e alpha map, texture and plane or motion parameters) is not used as such. Rather, the new scenes are rendered using the full volumes. This enables one to overcome some of the artifacts caused when over simplified depth models are used. The first example depicted in Figure 5.18 illustrates how the available volume $\mathcal{H}_n^\perp$ as shown in Figure 5.18(a) can be extrapolated to obtain the full region $\mathcal{H}_n$ shown in Figure 5.18(b). Linear extrapolation is performed using the available lines and their intensities. That is, each EPI line is simply extended along its slope. The second



(a) (b) (c)

(d) (e) (f)

Figure 5.17: Comparison of rendered views using different depth estimation and segmentation algorithms. (a-c) Interpolated viewpoints of the *Desk* light field for the proposed method, the stereo algorithm in [57] and the EPI tube extraction method in [24] respectively. (d-f) Illustrate interpolated images for the *Dwarves* light field in the same order.

example is the *Tank* light field which is depicted in Figure 5.19. It has a foreground layer (i.e. the wall) which strongly occludes the background layer (i.e. the tank). We compared our algorithm with two other occlusion removal algorithms. First, we applied the synthetic aperture method in [39, 79] which produced the image in Figure 5.19(d). Note that there are no holes and the full tank is visible. However, it is blurred since the synthetic aperture is essentially averaging over the lines in the EPIs. The foreground layer is thus contaminating the background one. The second result depicted in Figure 5.19(e) was obtained by applying the stereo algorithm in [57] on all consecutive images. Regions with the disparity corresponding to the wall layer are removed. Here, the absence of a model for occlusions throughout multiple views and the lack of consistency drastically reduced the quality of the disoccluded image. Finally, Figure 5.19(f) shows the result using the plenoptic hypervolume segmentation algorithm.

Another useful application consists in reconstructing the scene by combining existing plenoptic volumes with external ones. These can be captured by other camera arrays

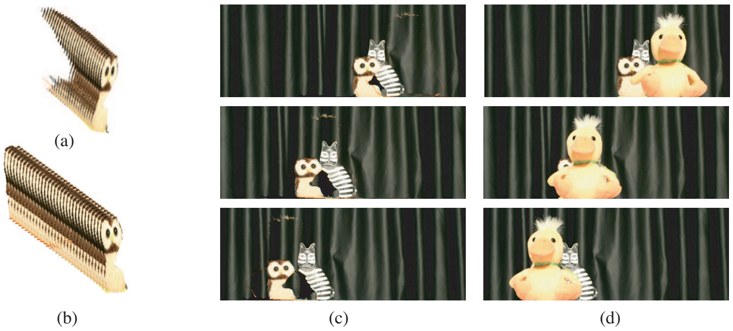


Figure 5.18: Image-based object removal. The extracted volume illustrated in (a) is extrapolated along its EPI lines in order to reconstruct the occluded regions as shown in (b). This enables one to reconstruct the scene, for example, by removing the region carved out by the duck and using the extrapolated background volumes (c). Note that there are some holes since some regions are never visible in the entire stack of images. The images in (d) show the original data for comparison.

or synthetically generated in order to perform augmented reality. As an example, we use a CAD software to generate the synthetic images of a teapot shown in Figure 5.20(a). The visible parts of the volume are determined using equation (4.10) which leads to the orthogonalized volume depicted in Figure 5.20(b). When combined with the original data, all the occlusions are naturally handled. Some of the rendered images are illustrated in Figure 5.20(c). Although the regions were extracted using fronto-parallel models, the objects still show their original shapes in the reconstructed images. This is noticeable, for instance, in the duck's beak and its shadow that are accurately rendered throughout the views despite the fact the 'duck' layer was extracted using a constant depth model.

5.6 Summary and key results

In this chapter, we presented experimental results for the extraction of plenoptic hypervolumes in light fields. Many different data sets captured with various multiview imaging systems and in controlled and uncontrolled environments have been analyzed. The segmentation of these light fields is in general accurate and occlusions are correctly captured. Moreover, the segmentation process is done in a reasonable time.

Image based rendering results were demonstrated and compared with conventional

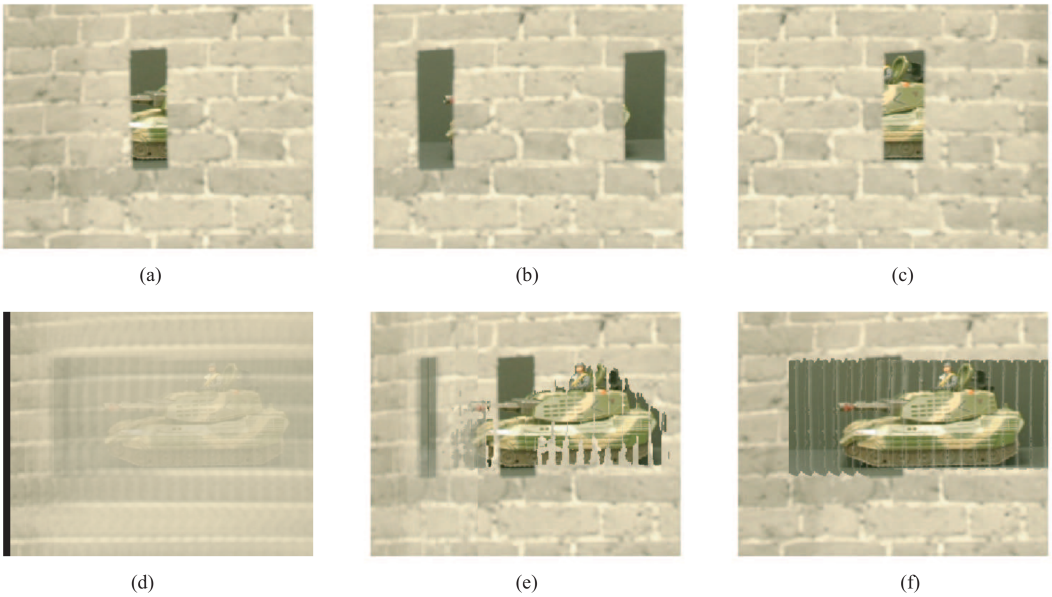


Figure 5.19: Image-based disocclusion of the *Tank* light field. (a-c) Three of the original 15 input images. (d) Disocclusion using synthetic aperture. (e) Disocclusion using a state-of-the-art stereo algorithm. (f) Disocclusion using the plenoptic layer extraction algorithm.

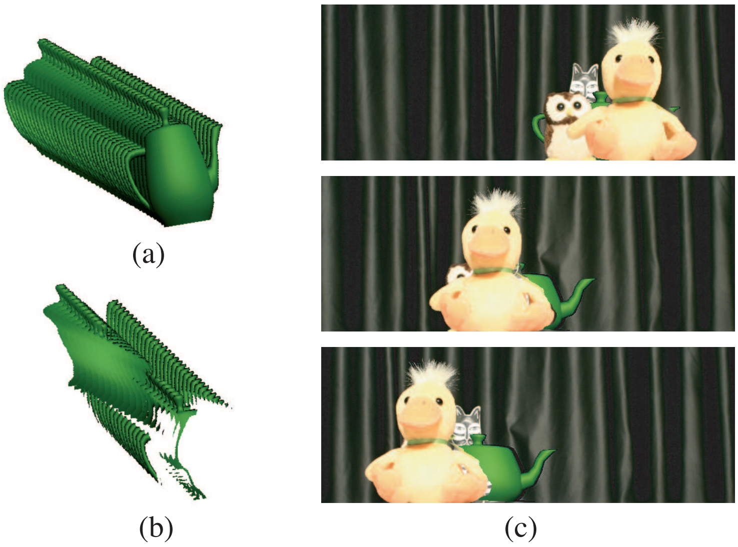


Figure 5.20: Image based object insertion. The synthetically generated volume carved out by a teapot (a) is geometrically orthogonalized with the existing extracted volumes (b). It is then straightforward to recombine the regions to recreate a scene where the teapot is inserted (c).

light field rendering. Since all the data sets have many objects and occlusions, the images are not sufficiently densely sampled and ghosting effects occur. In using plenoptic hypervolumes, we showed that the artifacts caused by undersampled light fields are significantly reduced while keeping the natural aspect of the images. Finally, we showed other applications that benefit from the segmentation of the plenoptic data. In particular, we illustrated some object removal and insertion results with a particular emphasis on objects that are strongly occluded. Moreover, despite the simplified depth model used (i.e. piecewise fronto-parallel), the reconstructed images still show their original geometry and lighting changes. This comes from the fact that we use the full plenoptic volumes in the reconstruction process.

The encouraging results reported in this chapter may be used for instance to create walkthrough environments such as museums and tourist attractions. One may also use these view interpolation schemes for marketing purposes such as showcasing hotel lobbies. Finally, occlusion removal may be used for military applications as well as removing unwanted objects such as people from a scene.

Chapter 6

Sampling piecewise sinusoidal signals with an application to the plenoptic function

6.1 Introduction

In this chapter, we study new methods to perfectly reconstruct certain classes of plenoptic functions from their sampled versions.

The classical sampling theorem states that any bandlimited function $x(t)$ such that $X(\omega) = 0 \; \forall \; |\omega| > \omega_{max}$ can be exactly recovered from its samples given that the rate $2\pi/T$ is greater or equal to twice the highest frequency component [66]. The continuous-time (or space) signal is recovered with $x(t) = \sum_{k\in\mathbb{Z}} y[k]\text{sinc}(t/T - k)$ where $\text{sinc}(t) = \sin(\pi t)/\pi t$ and $y[k] = x(kT)$. Consider a scene made of several planar objects with constant depth and periodic bandlimited textures such as the one in Figure 6.1. This case is the example taken in many works such as [83, 84] studying the spectral properties of the plenoptic function. Then the projections of this scene under the light field parameterization are piecewise sinusoidal.[1] The resulting spectrum is therefore not bandlimited as it is made of the convolution between Diracs and sinc functions and the classical reconstruction formula

[1] We assume Lambertian surfaces.

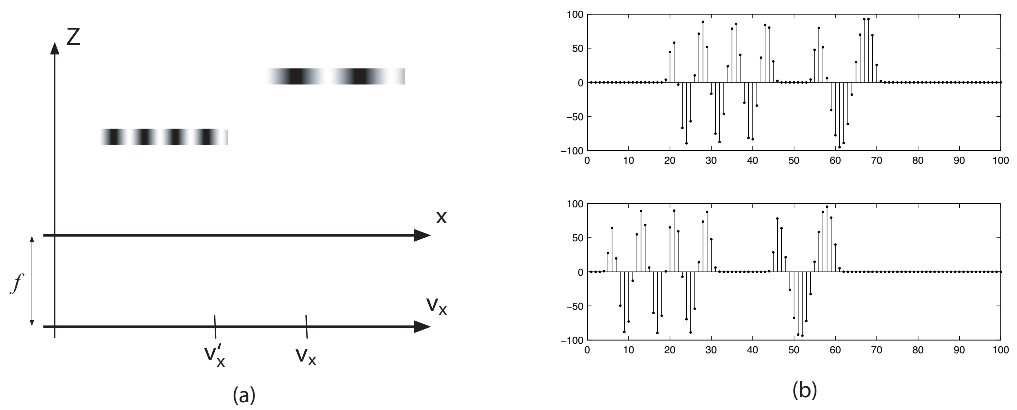


Figure 6.1: A two-layer scene with sinusoids pasted as texture. (a) Two sensors capture the scene from locations v_x and v'_x such that there are no occlusions. (b) The samples observed from the sensors located in v_x and v'_x. We recover the perfectly the two views using the piecewise sinusoidal reconstruction scheme. The full plenoptic function can be recovered thereafter using standard back projection.

cannot be used. From a Shannon point of view, the discontinuities are seen as infinite innovation processes and therefore require an infinite number of samples. Hence, events concentrated in time or space are not precisely measurable and aliasing occurs in the reconstructed data.

Recently, there has been a new trend in sampling theory that is not based solely on the Fourier representation. Rather they are based on sparsity either in terms of their representation in a basis [15, 27] or in terms of their parametric representation [29, 75]. In particular, a sampling scheme has recently been developed by Vetterli et al. [75] where it is made possible to sample and perfectly reconstruct signals that are not bandlimited but are completely determined by a finite number of parameters. Such signals are said to have a Finite Rate of Innovation (FRI). For instance, the authors derive a method to recover some classes of FRI signals such as streams of Diracs, differentiated Diracs and piecewise polynomials using sinc or Gaussian kernels. Later, in [28, 29], it was shown that these signals can also be recovered using more realistic compact support sampling kernels such as those satisfying the Strang-Fix conditions [73], exponential splines [74] and functions with a rational Fourier transform. The reconstruction process for these schemes is based on the annihilating filter, a tool widely used in spectral estimation [72], error correction coding [9] and interpolation [76]. These results provide an answer for precise time (or space) localization (i.e. Diracs and polynomial signals) but in some sense lack frequency localization capabilities.

In this chapter, we extend FRI theory to oscillating functions. In particular, we investigate the case where the continuous-time signal is piecewise sinusoidal therefore it contains both time and frequency components. We derive two methods to retrieve exactly continuous-time piecewise sinusoidal signals from their sampled versions. Hence, we will show that although the scene in Figure 6.1 is not bandlimited, it can be perfectly reconstructed from a finite number of cameras having finite resolution. The problem of computing the depth of the planes is then reduced to the standard back projection problem. If the planes are different and the viewpoints are such that there are no occlusions, then two sensors are sufficient to perfectly recover the textures and locations of the planes. Furthermore, all the intermediate viewpoints $[v'_x, v_x]$ can be exactly interpolated with perspective projections.

The chapter is organized as follows: Section 6.2 describes the sampling setup and the signals of interest. Sections 6.3 and 6.4 discuss the sampling kernels which can be used in our scheme and recall some of the aspects of annihilating filter theory. Using these kernels, Section 6.5 derives a global method for retrieving the parameters of a general piecewise sinusoidal signal. Section 6.6 discusses local reconstruction methods that have a lower complexity. In Section 6.7, we briefly discuss some extensions of the algorithm, namely adding piecewise polynomials to piecewise sine waves. Section 6.8 illustrates examples and we conclude in Section 6.9.

6.2 Sampling setup and signals of interest

The acquisition device, be it a camera or another sensing device, can be modeled with a smoothing kernel $\varphi(t)$ and a uniform sampling period $T > 0$. Following this setup, the observed discrete-time signal is given by

$$y[k] = \int_{-\infty}^{\infty} \varphi(t - kT)x(t)dt = \langle \varphi(t - kT), x(t) \rangle \tag{6.1}$$

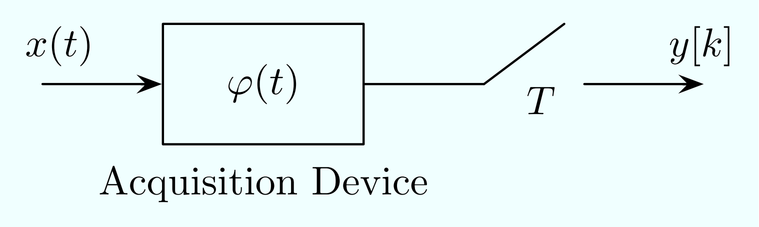


Figure 6.2: Sampling setup. The continuous-time signal $x(t)$ is filtered by the acquisition device and sampled with period T. The observed samples are $y[k] = \langle \varphi(t-kT), x(t) \rangle$.

with $k \in \mathbb{Z}$ as shown in Figure 6.2. The fundamental problem of sampling is to recover the original continuous-time waveform $x(t)$ using the set of samples $y[k]$. The signals $x(t)$ under study in this context are piecewise sinusoidal. More precisely, we consider signals of the type:

$$x(t) = \sum_{d=1}^{D} \sum_{n=1}^{N} A_{d,n} \cos(\omega_{d,n} t + \Phi_{d,n}) \xi_{d,d+1}(t), \tag{6.2}$$

where $\xi_{d,d+1}(t) = u(t-t_d) - u(t-t_{d+1})$, $u(t)$ is the Heaviside step function and $t_{d+1} > t_d$ for all d; and study their reconstruction from the samples $y[k]$ given in (6.1). Such signals are notoriously difficult to reconstruct since they are concentrated both in time and frequency. For this reason, the schemes in [28, 29, 75] as well as the Shannon type schemes would not enable an exact recovery. However such signals have a finite rate of innovation and it is possible to retrieve the parameters $\omega_{d,n}$, $A_{d,n}$ and $\Phi_{d,n}$ of the sinusoids along with the exact locations t_d given certain conditions on the sampling kernel $\varphi(t)$.

6.3 Sampling kernels

Many sampling schemes such as the classical Shannon reconstruction [66] and some of the original FRI schemes [75] rely on the ideal low-pass filter (i.e. the sinc function). This filter is not realizable in practice since it is of infinite support. It is therefore an attractive aspect to develop sampling schemes where the kernels are physically valid and realizable. It was recently shown that FRI sampling schemes may be used with sampling kernels that are of compact support [28, 29]. In this section, we present these kernels.

6.3.1 Polynomial reproducing kernels

A polynomial reproducing kernel $\varphi(t)$ is a function that together with its shifted version is able to reproduce polynomials. That is, we have

$$\sum_{k\in\mathbb{Z}} c_{m,k}\varphi(t-k) = t^m \qquad m = 0, 1, \ldots, M$$

given the right choice of weights $c_{m,k}$. Perhaps the most basic and intuitive such kernels are the classical B-splines that are of compact support. The B-spline of order zero is a function $\beta_0(t)$ with Fourier transform $\hat{\beta}_0(\omega) = \frac{1-e^{-j\omega}}{j\omega}$. The higher order B-splines of degree N are obtained through N successive convolutions of $\beta_0(t)$ and they are able to reproduce polynomials of degrees zero to N.

6.3.2 Exponential reproducing kernels

An exponential reproducing kernel $\varphi(t)$ is a function that together with its shifted version is able to reproduce exponentials. That is, we have

$$\sum_{k\in\mathbb{Z}} c_{m,k}\varphi(t-k) = e^{\alpha_m t} \qquad \text{with } \alpha_m = \alpha_0 + m\lambda \text{ and } m = 0, 1, \ldots, M \tag{6.3}$$

given the right choice of weights $c_{m,k}$. One important family of such kernels are the exponential splines (E-splines) that appeared in early works such as [25, 54, 62, 82] and were further studied in [74]. These functions are extensions of the classical B-splines in that they are built with exponential segments instead of polynomial ones. The first order E-spline is a function $\beta_{\alpha_n}(t)$ with Fourier transform $\hat{\beta}_{\alpha_n}(\omega) = \frac{1-e^{\alpha_n - j\omega}}{j\omega-\alpha_n}$. A series of interesting properties are derived in [74]. In particular, it is shown that an E-spline has compact support. The E-splines of order N are constructed by N successive convolutions of first order ones:

$$\hat{\beta}_{\vec{\alpha}}(\omega) = \prod_{n=1}^{N} \frac{1-e^{\alpha_n - j\omega}}{j\omega - \alpha_n} \tag{6.4}$$

where $\vec{\alpha} = (\alpha_1, \ldots, \alpha_N)$. The higher order spline is also of compact support and it can reproduce any exponential in the subspace spanned by $\{e^{\alpha_1 t}, \ldots, e^{\alpha_N t}\}$ [74]. Furthermore, since the exponential reproduction property is preserved through convolution [74], we have that any kernel of the form $\varphi(t) * \beta_{\vec{\alpha}}(t)$ is also able to reproduce the same exponentials.

6.4 Annihilating filters and differential operators

The annihilating filter is at the heart of finite rate of innovation sampling schemes. In this section, we recall the annihilating filter method and show how the filters in the case of exponential signals are related to the exponential splines [74]. We also show how an exponential signal may be converted into a stream of differentiated Diracs using an appropriate differential operator.

6.4.1 The annihilating filter method

Assume that a discrete-time signal $s[k]$ is made of weighted exponentials such that $s[k] = \sum_{n=1}^{N} a_n u_n^k$ with $u_n \in \mathbb{C}$ and assume we wish to retrieve the exponentials u_n and the weights a_n of $s[k]$. The filter $h[k]$ with z-transform

$$H_{\vec{u}}(z) = \prod_{n=1}^{N} (1 - u_n z^{-1}) \tag{6.5}$$

and $\vec{u} = (u_1, \ldots, u_N)$ is called annihilating filter of $s[k]$ since $(h * s)[k] = 0 \ \forall k \in \mathbb{Z}$. We can therefore construct the following system of equations

$$\begin{pmatrix} \vdots & \vdots & \ldots & \vdots \\ s[0] & s[-1] & \ldots & s[-N] \\ s[1] & s[0] & \ldots & s[-N+1] \\ \vdots & \vdots & \ddots & \vdots \\ s[N] & s[N-1] & \ldots & s[0] \\ \vdots & \vdots & \ldots & \vdots \end{pmatrix} \begin{pmatrix} h[0] \\ h[1] \\ \vdots \\ h[N] \end{pmatrix} = \vec{0}.$$

Notice that $N+1$ equations are sufficient to determine $\vec{h}$ therefore we write the system in matrix form as

$$\mathbf{S}\vec{h} = \vec{0}, \tag{6.6}$$

where $\mathbf{S}$ is the appropriate $N+1$ by $N+1$ submatrix involving $2N+1$ samples of $s[k]$. If s admits an annihilating filter, we have $Rank(\mathbf{S}) = N$ hence the matrix is rank deficient. The zeros of the filter $H_{\vec{u}}(z)$ uniquely define the u_ns since they are distinct and any filter $\vec{h}$ satisfying the Toeplitz system in (6.6) has u_n as its roots. Given the u_ns, the weights a_n are obtained by solving a system of equations using N consecutive samples of $s[k]$. These form a classic Vandermonde system which also has unique solution given that the u_ns are distinct. A straightforward extension of the above annihilating filter is that a signal $s[k] = \sum_{n=1}^{N} \sum_{r=0}^{R-1} k^r u_n^k$ is annihilated by the filter

$$H_{\vec{u}}(z) = \prod_{n=1}^{N} (1 - u_n z^{-1})^R \tag{6.7}$$

which has multiple roots of order R. For a more detailed discussion of the annihilating filter method we refer to [72].

Let us return to the sinusoidal case. Clearly, a filter of type $H_{\vec{u}}(z)$ will also annihilate a discrete sinusoidal signal $y[k] = \sum_{n=1}^{N} A_n \cos(\omega_n k + \Phi_n)$ since it can be written in the form of a linear combination of complex exponentials. In this case, the filter is obtained by posing $\vec{u} = e^{\vec{\alpha}}$ and

$$\vec{\alpha} = (j\omega_1, \ldots, j\omega_N, -j\omega_1, \ldots, -j\omega_N). \tag{6.8}$$

We simplify the notation by expressing $H_{e^{\vec{\alpha}}}(z)$ as $H_{\vec{\alpha}}(z)$. By comparing (6.5) with (6.4) and using $z = e^{j\omega}$, we see that the annihilating filter for a linear combination of exponentials can be expressed with an E-spline as

$$H_{\vec{\alpha}}(e^{j\omega}) = \hat{\beta}_{\vec{\alpha}}(\omega) \prod_{n=1}^{N} (j\omega - \alpha_n) \tag{6.9}$$

where the second term is a differential operator which is discussed in the following section.

6.4.2 Differential operators

Let $L\{x(t)\}$ be a differential operator of order N:

$$L\{x(t)\} \quad = \quad \frac{d^N x(t)}{dt^N} + a_{N-1}\frac{d^{N-1}x(t)}{dt^{N-1}} + \ldots + a_1 x(t)$$

with constant coefficients $a_n \in \mathbb{C}$. This operator can also be defined by the roots of its characteristic polynomial

$$L(s) \quad = \quad s^N + a_{N-1}s^{N-1} + \ldots + a_1 = \prod_{n=1}^{N}(s - \alpha_n).$$

Using the same notation as in [74], we express the operator as $L_{\vec{\alpha}}$ where $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N)$. Posing $s = j\omega$, we have in the frequency domain

$$L_{\vec{\alpha}}(j\omega) = \prod_{n=1}^{N}(j\omega - \alpha_n).$$

The null space of the operator, denoted $\mathcal{N}_{\vec{\alpha}}$, contains all the solutions to the differential equation $L_{\vec{\alpha}}\{x(t)\} = 0$. It is shown in [74] that $\mathcal{N}_{\vec{\alpha}} = \text{span}\{e^{\alpha_1 t}, \ldots, e^{\alpha_N t}\}$. It is therefore said that the operator has exponential annihilation properties. Moreover, the operator has sinusoidal annihilation properties when $\vec{\alpha}$ is defined as in (6.8). This follows naturally from the fact that sinusoids are linear combinations of complex exponentials. Therefore, given the right $\vec{\alpha}$, the operator $L_{\vec{\alpha}}$ will produce a zero output for the corresponding sinusoidal input. It is also relevant to mention here that the Green function $g_{\alpha_n}(t)$ of the operator L_{α_n} is a function such that $L_{\alpha_n}\{g_{\alpha_n}(t)\} = \delta(t)$ where $\delta(t)$ is the Dirac distribution. In this case, the Green function is given by $g_{\alpha_n}(t) = e^{\alpha_n t}u(t)$ [74] where $u(t)$ is the Heaviside step function. Consequently, we have that $L_{\alpha_n}\{e^{\alpha_n t}u(t - t_n)\} = \delta(t - t_n)$. Finally, using the linearity property of the derivative, it is straightforward to show that

$$L_{\vec{\alpha}}\{\sum_{n=1}^{N} e^{\alpha_n t}u(t - t_n)\} = \sum_{n=1}^{N}\sum_{r=0}^{N-1} a_{n,r}\delta^{(r)}(t - t_n), \tag{6.10}$$

where $\delta^{(r)}(t)$ is a differentiated Dirac of order r. Hence, the appropriate differential operator applied to a piecewise exponential signal will produce a stream of differentiated Diracs in the discontinuities.

6.5 Reconstruction of piecewise sinusoidal signals using a global approach

All the necessary tools to sample piecewise sinusoidal signals have now been laid down. For mathematical convenience, we write the continuous-time signal as:

$$x(t) = \sum_{d=1}^{D} \sum_{n=1}^{2N} A_{d,n} e^{j(\omega_{d,n} t + \Phi_{d,n})} \xi_{d,d+1}(t), \tag{6.11}$$

which is made of D pieces containing a maximum of N sinusoids each. Assume now that this signal is sampled with a kernel $\varphi(t)$ which is able to reproduce exponentials $e^{\alpha_m t}$ with $\alpha_m = \alpha_0 + \lambda m$ and $m = 0, 1, \ldots, M$. Following previous FRI methods [29], weighting the samples with the appropriate coefficients $c_{m,k}$ gives

$$\begin{aligned} \tau[m] &= \sum_{k \in \mathbb{Z}} c_{m,k} y[k] = \langle \sum_{k \in \mathbb{Z}} c_{m,k} \varphi(t-k), x(t) \rangle \\ &= \int_{-\infty}^{\infty} e^{\alpha_m t} x(t) dt \end{aligned} \tag{6.12}$$

where we have used (6.3). Note that $\tau[m]$ is an exponential moment of the original continuous-time waveform $x(t)$. Plugging (6.11) into (6.12) gives the moments

$$\tau[m] = \sum_{d=1}^{D} \sum_{n=1}^{2N} \tilde{A}_{d,n} \frac{\left[e^{t_{d+1}(j\omega_{d,n} + \alpha_m)} - e^{t_d (j\omega_{d,n} + \alpha_m)} \right]}{j\omega_{d,n} + \alpha_m} \tag{6.13}$$

where $\tilde{A}_{d,n} = A_{d,n} e^{j\Phi_{d,n}}$. These moments are a sufficient representation of the piecewise sinusoidal signal since the frequencies of the sinusoids and the exact locations of the discontinuities can be found using the annihilating filter method. Let us define the polynomial $Q(\alpha_m) = \prod_{d=1}^{D} \prod_{n=1}^{2N} (j\omega_{d,n} + \alpha_m)$ of degree $2DN$. Multiplying both sides of (6.13), we find the expression

$$Q(\alpha_m)\tau[m] = \sum_{d=1}^{D}\sum_{n=1}^{2N}\tilde{A}_{d,n}P_{d,n}(\alpha_m)[e^{t_{d+1}(j\omega_{d,n}+\alpha_m)} - e^{t_d(j\omega_{d,n}+\alpha_m)}] \qquad (6.14)$$

where $P_{d,n}(\alpha_m)$ is a polynomial of maximum degree $2DN - 1$. Recall that we impose $\alpha_m = \alpha_0 + \lambda m$ which means that (6.14) is equivalent to $\sum_{d=1}^{D+1}\sum_{r=0}^{2ND-1} b_{r,d}m^r e^{\lambda t_d m}$ where $b_{r,d}$ are weights. Therefore a filter of the type:

$$H(z) = \prod_{d=1}^{D+1}(1 - e^{\lambda t_d}z^{-1})^{2DN} = \sum_{k=0}^{K} h[k]z^{-k}$$

with $K = (D+1)2DN = 2D^2N + 2DN$ will annihilate (6.14) as shown in (6.7). It follows that

$$\sum_{k=0}^{K} h[k]Q(\alpha_{n-k})\tau[n-k] = 0 \qquad (6.15)$$

with $n = K, K+1, \ldots, M$. Recall that Q is a polynomial in α_m and it can therefore be written as

$$Q(\alpha_m) = \sum_{l=0}^{L} r[l]\alpha_m^l$$

where $L = 2DN$. Using this notation, the system in (6.15) becomes

$$\sum_{k=0}^{K}\sum_{l=0}^{L} h[k]r[l](\alpha_{n-k})^l\tau[n-k] = 0$$

for $n = K, \ldots, M$. For clarity, we write the system in matrix form which gives

$$\begin{pmatrix} \tau[K] & \ldots & (\alpha_K)^L\tau[K] & \ldots & \tau[0] & \ldots & (\alpha_0)^L\tau[0] \\ \tau[K+1] & \ldots & (\alpha_{K+1})^L\tau[K+1] & \ldots & \tau[1] & \ldots & (\alpha_1)^L\tau[1] \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \tau[K+U] & \ldots & (\alpha_{K+U})^L\tau[K+U] & \ldots & \tau[U] & \ldots & (\alpha_U)^L\tau[U] \end{pmatrix} \begin{pmatrix} h[0]r[0] \\ \vdots \\ h[0]r[L] \\ \vdots \\ h[K]r[0] \\ \vdots \\ h[K]r[L] \end{pmatrix} = \vec{0},$$

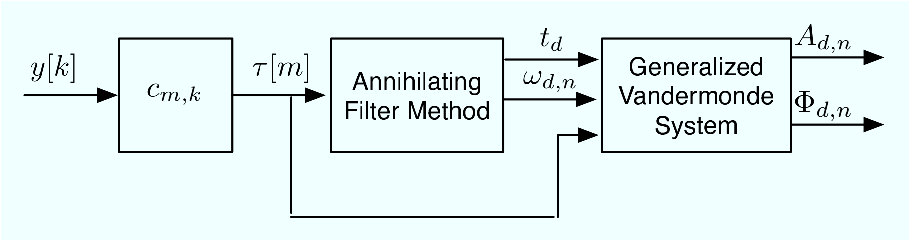


Figure 6.3: Overview of the algorithm for the global recovery of a piecewise sinusoidal signal.

where $U = M - K \geq (K+1)(L+1) - 1$. Solving the system with $h[0] = 1$ enables one to find the $r[l]$s. Subsequently, we find the $h[k]$s. The roots of the filter $H(z)$ and the polynomial $Q(\alpha_m)$ give the locations of the switching points and the frequencies of the sine waves respectively. The number of exponential moments $\tau[m]$ required to build a system with enough equations to find the parameters of the piecewise sinusoidal signal is $M = K + U = 4D^3N^2 + 4D^2N^2 + 4D^2N + 6DN$.

At this point, we have determined all the frequencies of the sinusoids and the locations of the discontinuities. However, the polynomial $Q(\alpha_m)$ does not enable to distinguish which frequencies are present in which pieces. This information, along with the amplitudes and phases of the sinusoids are found by building a generalized Vandermonde system

$$\tau[m] = \sum_{i=1}^{D+1} \sum_{d=1}^{D} \sum_{n=1}^{2N} \tilde{A}_{d,n} \frac{e^{t_i(j\omega_{d,n}+\alpha_m)}}{j\omega_{d,n} + \alpha_m} \tag{6.16}$$

which requires $2ND(D+1)$ moments $\tau[m]$ and enables to determine the $\tilde{A}_{d,n}$s. This system provides a unique solution since the exponents are distinct. The full algorithm is depicted in Figure 6.3. We summarize the above derivation with the following theorem:

Theorem 1. *Assume a sampling kernel $\varphi(t)$ that can reproduce exponentials $e^{\alpha_0+\lambda m}$ with $m = 0, 1, \ldots, M$. A piecewise sinusoidal signal with D pieces having a maximum of N sinusoids is uniquely determined by the samples $y[k] = \langle \varphi(t-kT), x(t) \rangle$ if $M \geq 4D^3N^2 + 4D^2N^2 + 4D^2N + 6DN$.*

For example, a sine wave truncated in t_1 and t_2 (i.e. $D = 1$ and $N = 1$) requires to compute the exponential moments up to order 18. Note that the method is based on the

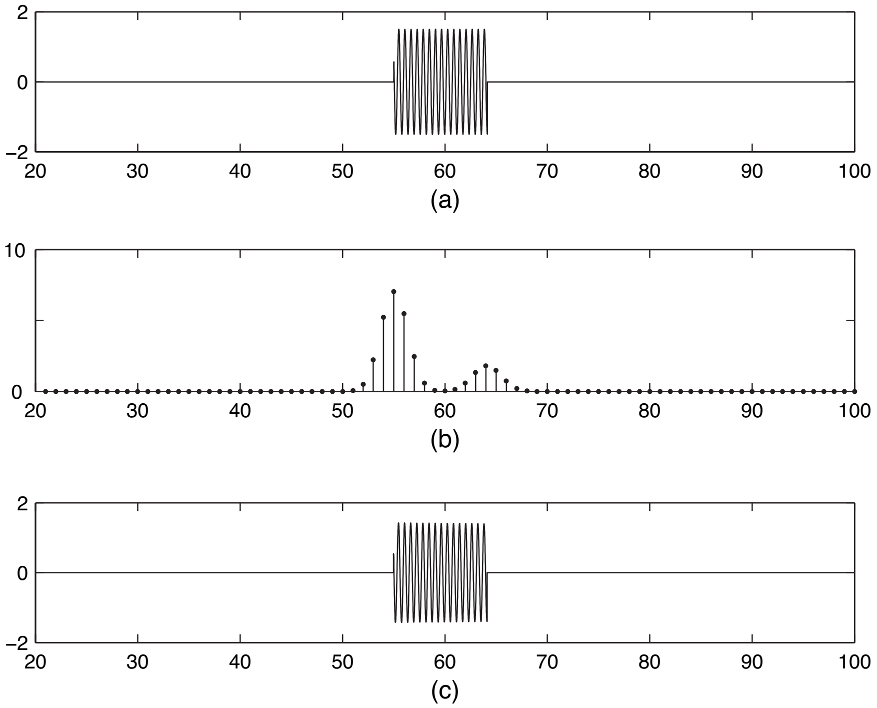


Figure 6.4: Sampling a truncated sine wave. (a) The original continuous-time waveform. (b) The amplitude of the observed samples. (c) The reconstructed signal. Note that the signal is not bandlimited and the frequency of the sine wave itself is higher than the Nyquist rate for the given sampling period.

rate of innovation of the signal only. That is, there are no constraints for example on the frequencies of the sine waves. In particular, we are not limited by the Nyquist frequency. It also means that the locations of the discontinues t_1 and t_2 may be arbitrarily close. Figure 6.4 illustrates the sampling and perfect reconstruction of a truncated sine wave.

6.6 Reconstruction of piecewise sinusoidal signals using a local approach

In the previous section, we saw that is it possible to retrieve the parameters of a sampled piecewise sinusoidal signal given that the sampling kernel is able to reproduce exponentials of a certain degree. This degree however increases very rapidly with the number of sinusoids and pieces. In this section, we show that the complexity can be reduced by making assumptions on the signal. In the first case, we assume that the frequencies of the sine waves are known and we retrieve the exact locations of the discontinuities. In the second case, we assume that the discontinuities are sufficiently far apart such that a classical spectral estimation method can be run in each piece in order to estimate the frequencies independently of the discontinuities.[2] We emphasize the fact that all the reconstruction steps in this section are local therefore they are extendable to infinite length piecewise sinusoidal signals.

6.6.1 Local reconstruction with known frequencies

Consider a piecewise sinusoidal signal $x(t)$ as defined in (6.11) and assume the frequencies $\omega_{d,n}$ are known at the reconstruction. This can be the case, for instance, when information is transmitted using the switching points (or the discontinuities) and we wish to retrieve these locations exactly. The samples $y[k]$ are again given by (6.1). Since the frequencies of the sine waves are known, we can construct the annihilating filter

$$H_{\vec{\alpha}}(z) = \prod_{d=1}^{D}\prod_{n=1}^{N}(1 - e^{j\omega_{d,n}}z^{-1})(1 - e^{-j\omega_{d,n}}z^{-1}) \qquad (6.17)$$

with coefficients $h_{\vec{\alpha}}[k]$. Assume now that we apply this filter to the samples $y[k]$. The expression for the annihilated signal $y_{ann}[k]$ gives

$$\begin{aligned} y_{ann}[k] &= h_{\vec{\alpha}}[k] * \langle \varphi(t-k), x(t)\rangle \\ &\overset{(a)}{=} \frac{1}{2\pi}\langle e^{-j\omega k}\hat{\varphi}(\omega)H_{\vec{\alpha}}(e^{j\omega}), X(\omega)\rangle \\ &\overset{(b)}{=} \frac{1}{2\pi}\big\langle e^{-j\omega k}\hat{\varphi}(\omega)\hat{\beta}_{\vec{\alpha}}(\omega)L_{\vec{\alpha}}(j\omega), X(\omega)\big\rangle \\ &= \langle L_{\vec{\alpha}}\{\varphi(t-k) * \beta_{\vec{\alpha}}(t-k)\}, x(t)\rangle \\ &\overset{(c)}{=} \langle \varphi(t-k) * \beta_{\vec{\alpha}}(t-k), L_{\vec{\alpha}}\{x(t)\}\rangle, \end{aligned}$$

where (a) follows from Parseval's identity, (b) from (6.9) and (c) from integration by parts and the fact that $\varphi * \beta_{\vec{\alpha}}$ is of finite support. This means that the coefficients $y_{ann}[k]$ represent the samples given by the inner-product between a modified $x(t)$ that we call

[2] This case was also presented in [4]

$x_\delta(t) = L_{\vec{\alpha}}\{x(t)\}$ and a new sampling kernel $\varphi_{equ}(t) = \varphi(t) * \beta_{\vec{\alpha}}(t)$. Now assume that the sampling kernel $\varphi(t)$ has compact support W. Then the equivalent kernel $\varphi_{equ}(t)$ is of compact support $W + 2DN$. Furthermore, according to (6.10), $x_\delta(t)$ is made only of differentiated Diracs of maximum order $2DN - 1$ in the discontinuities. That is, we are left with a signal of the type $x_\delta(t) = \sum_{d=1}^{D+1} \sum_{r=0}^{2DN-1} a_{d,r}\delta^{(r)}(t - t_d)$ for which a sampling theorem exists [28] [29]. Hence given that the hypotheses of the theorem are met, we are able to perfectly reconstruct $x_\delta(t)$ and retrieve the exact locations t_d. The theorem states that a signal made of differentiated Diracs can be sampled and perfectly reconstructed using a sampling kernel that is able to reproduce exponentials or polynomials. Since, this reproduction capability is preserved through convolution [74], the equivalent kernel $\varphi_{equ}(t)$ is able to reproduce the same exponentials or polynomials as $\varphi(t)$. Hence the classes of kernels used in [28] are also valid in our context.

Similarly to the previous method, the retrieval of the locations t_d and the weights $a_{d,r}$ of $x_\delta(t)$ is based on the annihilating filter method. As shown in [28] [29], these parameters can be found using appropriate linear combinations of the samples $y_{ann}[k]$. Indeed, using an exponential reproducing kernel, we have the moments

$$\begin{aligned} \tau[m] &= \sum_k c_{m,k} y_{ann}[k] = \langle \sum_k c_{m,k}\varphi_{equ}(t-k), x_\delta(t)\rangle && (6.18)\\ &= \int_{-\infty}^{\infty} e^{\alpha_m t} x_\delta(t)dt = \sum_{d=1}^{D} \sum_{r=0}^{2DN-1} b_{r,d} m^r e^{\lambda t_d m} && (6.19) \end{aligned}$$

which are made of weighted exponentials. Therefore a filter of the type $H(z) = \prod_{d=1}^{D}(1 - e^{\lambda t_d} z^{-1})^{2DN}$ will annihilate $\tau[m]$ and the problem of finding the locations t_d is reduced to that of finding the multiple root of $H(z)$. This filter can be determined using a system of equations similar to the one in (6.6). A more detailed description of the location retrieval can be found in [29]. The general result is summarized as follows:

Theorem 2. *Assume a sampling kernel $\varphi(t)$ that can reproduce exponentials $e^{\alpha_0 + \lambda m}$ or polynomials t^m with $m = 0, 1, \ldots, M$ and of compact support W. An infinite-length piecewise sinusoidal signal with a maximum of N sinusoids in each piece is uniquely determined by the samples $y[k] = \langle \varphi(t - kT), x(t)\rangle$ if the frequencies $\omega_{d,n}$ are known, there*

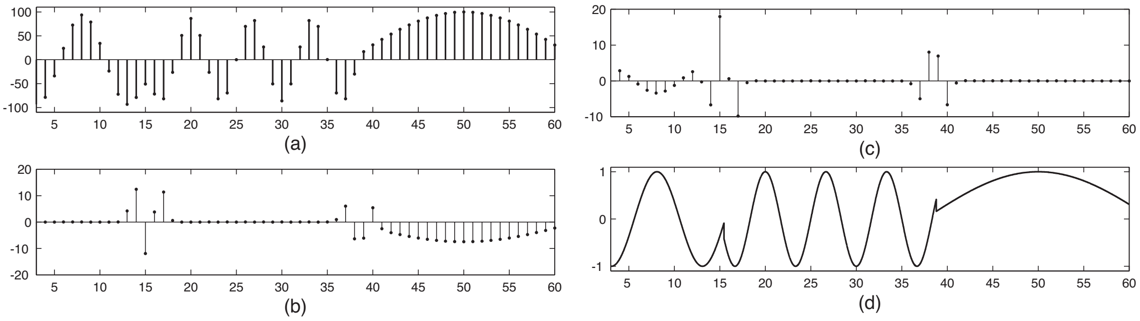


Figure 6.5: Sequential recovery of a piecewise sinusoidal signal. The observed discrete signal is illustrated in Figure 6.5(a). In this example, we have one sine wave per piece and frequencies $\omega_{1,1}$, $\omega_{2,1}$ and $\omega_{3,1}$ in the first, second and third piece respectively. The frequencies are determined using the annihilating filter method. The annihilated signal $y_1^{ann}[k] = (y * h_{\vec{\alpha}_1} * h_{\vec{\alpha}_2})[k]$ where $\vec{\alpha}_1 = (j\omega_{1,1}, -j\omega_{1,1})$ and $\vec{\alpha}_2 = (j\omega_{2,1}, -j\omega_{2,1})$ is shown in Figure 6.5(b). The non zero samples in the vicinity of the discontinuity are sufficient to recover the first breakpoint. The second breakpoint can be found by looking at $y_2^{ann}[k] = (y * h_{\vec{\alpha}_2} * h_{\vec{\alpha}_3})[k]$ where $\vec{\alpha}_2 = (j\omega_{2,1}, -j\omega_{2,1})$ and $\vec{\alpha}_3 = (j\omega_{3,1}, -j\omega_{3,1})$ which is depicted in Figure 6.5(c). The recovered continuous-time signal is shown in Figure 6.5(d).

are at most D sinusoidal discontinuities and in an interval of length 2D(W+2DN)T and $M \geq 4DN(D+1) - 1$.

6.6.2 Local reconstruction with unknown frequencies

In the previous section, we saw that the exact locations of the switching points t_d of a piecewise sinusoidal signal can be estimated from its sampled version. The number of moments required in this case was less than in the global method presented in Section 6.5 since in essence the estimation of the breakpoints is separated from that of the sine waves. In this section, we show how the local method presented above may be applied even if the frequencies of the sine waves are unknown. The basic idea is to impose that the discontinuities are sufficiently far apart such that a classical spectral estimation method can be run in each piece.

Assume, for the moment, an original continuous-time signal that is purely sinusoidal with a maximum of N sinusoids. The signal is sampled with a sampling kernel $\varphi(t)$ and

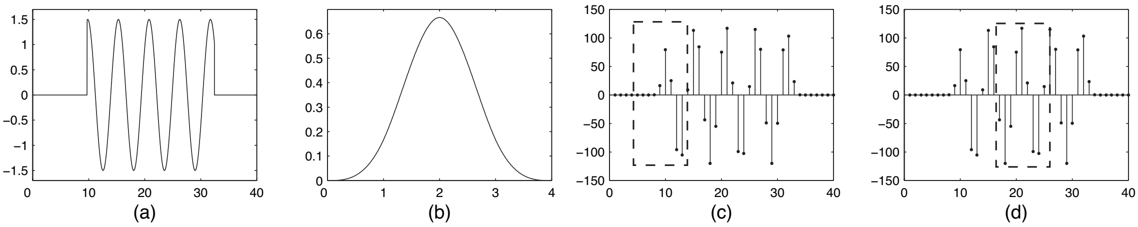


Figure 6.6: Determining the sinusoidal part of the pieces. Figure 6.6(a) illustrates a truncated sinusoid. Assume, for example, a B-spline sampling kernel $\varphi(t) = \beta_3(t)$ that is of compact support $W = 4$ as is depicted in Figure 6.6(b). Since the kernel has a certain support, the samples in the vicinity of the discontinuities are not pure discrete sinusoids. Therefore the rank of matrix S is full when S is constructed with the samples in the dashed window depicted in Figure 6.6(c). However, S is rank deficient when the window is chosen as shown Figure 6.6(d) since the samples are not influenced by the discontinuities.

the samples are given by

$$y[k] = \sum_{n=1}^{N} \frac{\tilde{A}_n}{2} \left(e^{j(\omega_n k + \Phi_n)} + e^{-j(\omega_n k + \Phi_n)} \right)$$

with $\tilde{A}_n = \hat{\varphi}(\omega_n) A_n$. From Section 6.4.1, we know that $4N + 1$ samples are sufficient to construct the matrix **S** in (6.6) and solve the system of equations in order to determine the annihilating filter $H_{\vec{\alpha}}(z)$. The ω_ns are found using the roots of the filter. Clearly, in this case, the classical Nyquist condition $\omega_n < \pi/T$ holds otherwise there is no distinction between ω_n and $\pi/T - \omega_n$. In order to find the amplitudes A_n and the phases Φ_n, we use $2N$ consecutive samples of $y[k]$ in order to construct a Vandermonde system. For example, in the case where $N = 1$, we have the following system:

$$\frac{1}{2} \begin{bmatrix} e^{j\omega_0 k} & e^{-j\omega_0 k} \\ e^{j\omega_0 (k+1)} & e^{-j\omega_0 (k+1)} \end{bmatrix} \begin{bmatrix} \tilde{A}_0 e^{j\Phi_0} \\ \tilde{A}_0 e^{-j\Phi_0} \end{bmatrix} = \begin{bmatrix} y[k] \\ y[k+1] \end{bmatrix}$$

where the unicity of the solution is guaranteed since the exponents are distinct. Notice that determining the parameters of the sinusoids is a classical spectral estimation problem [72].

In the piecewise sinusoidal case, the discontinuities influence the samples. Indeed, if the kernel has support W, the samples in the interval $[t_d - TW/2, t_d + TW/2]$ are not pure discrete sinusoids. Hence, the sampling period T must be such that there are at least $4N+1$ samples that are not influenced by the discontinuities in each interval $[t_d, t_{d+1}]$. This enables one to use classical spectral estimation methods. The only apparent difficulty lies in finding the right samples in each piece that are not perturbed by the breakpoints. Recall from Section 6.4.1 that the $2N+1$ by $2N+1$ matrix $\mathbf{S}$ admits an annihilating filter when $Rank(\mathbf{S}) = 2N$. However, the rank is full when $\mathbf{S}$ is constructed with samples that are influenced by the discontinuities. It follows that the samples that contain only frequency atoms can be found by running a window along the k-axis constructing successive matrices and looking at the rank of $\mathbf{S}$. Figure 6.6 illustrates the sliding window. In Figure 6.6(c), the window contains samples that are influenced by the discontinuity and the rank of $\mathbf{S}$ is full. However, in Figure 6.6(d), the matrix is rank deficient and the annihilating filter method is run to retrieve the parameters of the sinusoids. Once the frequencies have been estimated, the locations of the discontinuities may be found using the method in Section 6.6. Note that in this case, we impose that the discontinuities are sufficiently far apart to retrieve each t_d separately. We therefore have $D = 1$. The above derivation is summarized with the following theorem:

Theorem 3. *Assume a sampling kernel $\varphi(t)$ of compact support W and that can reproduce exponentials $e^{\alpha_0+\lambda m}$ or polynomials t^m with $m = 0, 1, \ldots, M$. An infinite-length piecewise sinusoidal signal is uniquely determined by the samples $y[k] = \langle \varphi(t-kT), x(t) \rangle$ if there are at most N sinusoids with frequency $\omega < \pi/T$ in a piece of length $T(4N+W+1)$ and $M \geq 8N-1$.*

An overview of the algorithm for the local recovery of piecewise sinusoidal signals is illustrated in Figure 6.7. A simulation recovering a piecewise sinusoidal signal with three pieces containing one sinusoid each is illustrated in Figure 6.5. We use a classical B-spline sampling kernel $\beta^7(t)$ as it is capable of reproducing polynomials of maximum degree $8N-1=7$. The reconstructed signal is exact within machine precision.

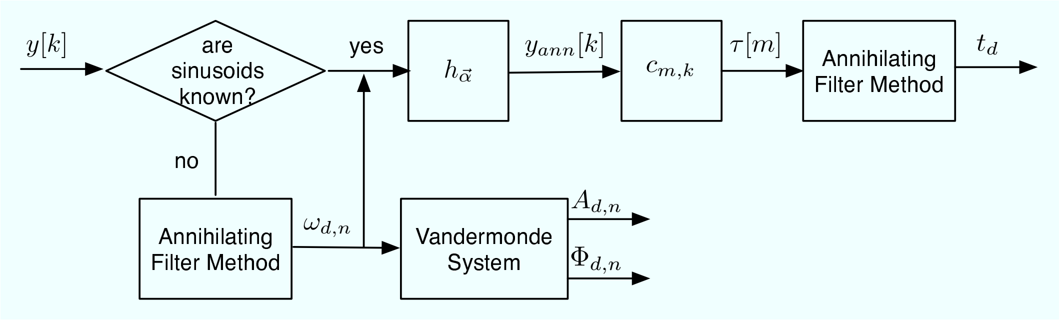


Figure 6.7: Overview of the algorithm for the local recovery of a piecewise sinusoidal signal.

6.7 Joint recovery of piecewise sinusoidal and polynomial signals

Sampling piecewise sinusoidal signals using the schemes presented above is not based on the fact that the signals of interest are bandlimited but on the fact that they can be represented with a finite number of parameters. It is worth mentioning that signals that are a combination of piecewise sinusoidal and polynomials pieces are also defined by a finite number of parameters and they can also be recovered from their sampled versions using the same algorithms. These signals are of the type:

$$x(t) = \sum_{d=1}^{D} x_d(t)\xi_{d,d+1}(t)$$

where $x(t) = 0$ for $t < t_1$, $\xi_{d,d+1}(t)$ is as previously defined and

$$x_d(t) = \sum_{n=1}^{N} A_{d,n}\cos(\omega_{d,n}t + \Phi_{d,n}) + \sum_{p=0}^{P-1} B_{d,p}t^p.$$

That is, we have a maximum of N sinusoids and polynomials of maximum degree P in each piece. In the following, we will briefly discuss the basic steps to recover the parameters as they are analogous to the piecewise sinusoidal cases presented in Sections 6.5 and 6.6.

Clearly, the Pth order derivative of $x(t)$ is

$$x^{(P)}(t) = \sum_{d=1}^{D}\sum_{n=1}^{N} \frac{d^P}{dt^P}[A_{d,n}\cos(\omega_{d,n}t + \Phi_{d,n})]\xi_{d,d+1}(t) + \sum_{d=1}^{D+1}\sum_{p=0}^{P-1} c_{d,p}\delta^{(p)}(t - t_d)$$

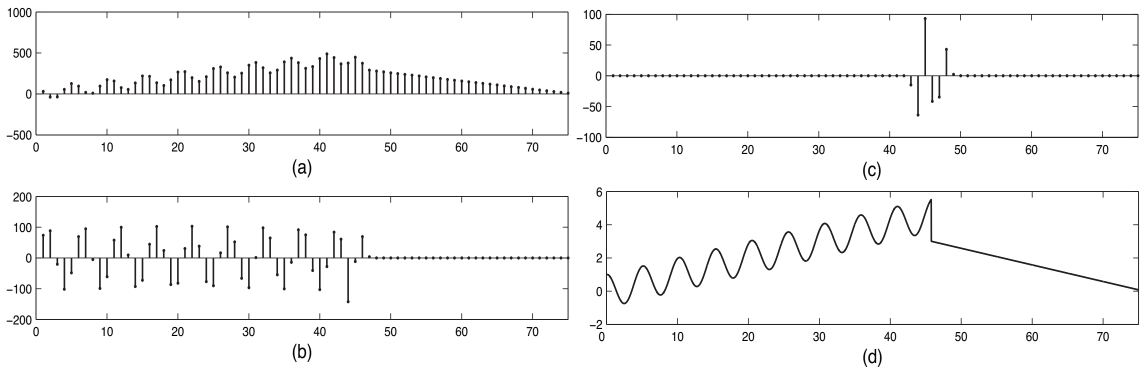


Figure 6.8: Sampling a combination of piecewise polynomials and sinusoids. The observed samples are depicted in Figure 6.8(a). In the first step, we annihilate the polynomial part by applying the finite difference operator. As shown in Figure 6.8(b), we are left only with a piecewise sinusoidal part. The parameters characterizing the sinusoid are retrieved and the annihilating filter is applied. The samples depicted in Figure 6.8(c) contain all the information necessary to find the discontinuity. The recovered continuous-time signal is shown in Figure 6.8(d).

which is a piecewise sinusoidal signal with differentiated Diracs in the discontinuities. Both the global and the local schemes presented above are able to cope with these signals. Therefore if we are able to relate the observed samples $y[k]$ with the samples $y^{(P)}[k]$ that would have been obtained from $x^{(P)}(t)$, we will be able to recover $x^{(P)}(t)$. The $x(t)$ will then be obtained by integration which is uniquely defined since we assume that $x(t) = 0$ for $t < t_1$. The relation between the samples $y[k]$ and $y^{(M)}[k]$ is related to B-spline theory and was demonstrated in [29]. Assume we apply the finite difference $y^{(1)}[k] = y[k+1] - y[k]$ to the observed samples. The new set of samples $y^{(1)}[k]$ are equivalent to $\langle \varphi(t-k) * \beta_0(t-k), \frac{d}{dt}x(t)\rangle$ where $\beta_0(t)$ is the B-spline of order zero. Similarly, the Pth order finite differences lead to the samples

$$y^{(P)}[k] = \langle \varphi(t-k) * \beta_{P-1}(t-k), \frac{d^P}{dt^P}x(t)\rangle$$

which means the obtained samples are equivalent to the ones that would have been observed from sampling $x^{(P)}(t)$ with the kernel $\varphi(t) * \beta_{P-1}(t)$. Moreover, since the polynomial and exponential reproduction capability are preserved through convolution, the new kernel is able to reproduce the polynomials or exponentials as well. Hence the sampling schemes presented above are also valid for piecewise sinusoidal and polynomial signals. An example of the piecewise polynomial and sinusoidal case is depicted in Figure 6.8.

6.8 Examples

6.8.1 Parametric plenoptic functions

Consider slit or one-dimensional pinhole cameras that are placed along a line in known locations. These cameras observe the scene in the direction perpendicular to the camera location line. In effect, these cameras are setup in the EPI configuration. The observed scene is made of several fronto-parallel planar objects with periodic bandlimited textures as illustrated in Figure 6.1. Therefore the signals observed by each camera are piecewise sinusoidal. Now assume that the point-spread function of the cameras are B-splines or E-splines.[3] That is, the sampling kernels belong to the family of kernels that can be used in our sampling framework. Then, given that the hypotheses on the rate of innovation of the scene are satisfied, we are able to perfectly reconstruct each view. Therefore, in the absence of occlusions, it is straightforward to reconstruct the continuous EPI using projective transformations.

6.8.2 Electronic circuits as acquisition devices

Following the previous work on sampling signals with finite rate of innovation [29], it is possible to use sampling kernels that have a rational Fourier transform. These include many electrical systems including the classical RC circuits. This is due to the fact that the observed samples can be converted into those that would have been obtained using an E-spline as sampling kernel. In this section, we show with a simple example how to retrieve the exact jump location of a truncated sinusoidal signal.

Consider an electrical system with transfer function $\hat{\varphi}(\omega) = \prod_{m=1}^{4} \frac{1}{j\omega+\nu_m}$ and $\nu_m = m\lambda$ with $\lambda \in \mathbb{C}$. Such a system would be, for instance, a fourth order RC circuit. The samples observed at the output of the system are given by $y[k] = \langle \varphi(t-k), x(t) \rangle$ where we assume for simplicity that the sampling period T is unity. In this example, the input

[3] These kernels approximate very well the usual Gaussian point-spread function.

voltage is a truncated sine function $u(t) = A_1 \cos(\omega_1 t) u(t - t_1)$ with known frequency ω_1 and we wish to retrieve the exact switching time t_1.

This first step consists in applying the digital filter with z-transform $H_{\vec{\nu}}(z) = \prod_{m=1}^{4}(1 - e^{\nu_m} z)$. The new set of samples are given by [29]

$$y_E[k] = \langle \beta_{\vec{\nu}}(t-k), x(t) \rangle$$

which are equivalent to those that would have been observed using an E-spline sampling kernel. Therefore the methods described in the previous sections may be applied. In this particular example, we assume that the frequency of the sine wave is known and we are interested only in the recovery of t_1. We therefore use the algorithm presented in the first part of Section 6.6. The next step consists in applying the digital filter $H_{\vec{\alpha}}(z) = (1 - e^{j\omega_1} z^{-1})(1 - e^{-j\omega_1} z^{-1})$ to the samples $y_E[k]$. The resulting samples are

$$y_{ann}[k] = \langle \beta_{\vec{\nu}}(t-k) * \beta_{\vec{\alpha}}(t-k), L_{\vec{\alpha}}\{x(t)\} \rangle$$

with $L_{\vec{\alpha}}\{x(t)\} = -2A_1\omega_1 \sin(\omega_1 t)\delta(t - t_1) + A_1 \cos(\omega_1 t)\delta^{(1)}(t - t_1)$. The new kernel is an E-spline that can reproduce exponentials including those in the subspace spanned by $\{e^{\nu_1 t}, e^{\nu_2 t}, e^{\nu_3 t}, e^{\nu_4 t}\}$. Therefore by an appropriate linear combination of the samples $y_{ann}[k]$ as in (6.18), we have access to the continuous-time moments

$$\tau[m] = \int_{-\infty}^{\infty} e^{\nu_m t} x_\delta(t) dt = A_1 e^{\nu_m t_1}[-3\omega_1 \sin(\omega_1 t_1) + \cos(\omega_1 t_1)\nu_m]$$

from which we can retrieve t_1. Indeed, the signal $\tau[m]$ is annihilated by the filter $H(z) = (1 - e^{\lambda t_1} z^{-1})^2$ since by hypothesis $\nu_m = \lambda m$. Posing $h[0] = 1$, we may write the annihilation of the moments $\tau[m]$ as

$$\begin{bmatrix} \tau[2] & \tau[1] \\ \tau[3] & \tau[2] \end{bmatrix} \begin{bmatrix} h[1] \\ h[2] \end{bmatrix} = \begin{bmatrix} -\tau[3] \\ -\tau[4] \end{bmatrix}$$

which will provide a solution for the filter taps of $H(z)$. Therefore the exact location of the switching point is given by $t_1 = \ln(z_r)/\lambda$ where z_r is the multiple root of the annihilating

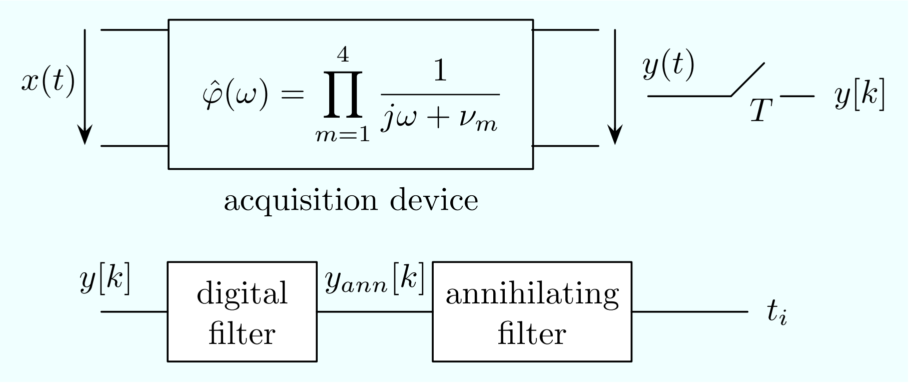


Figure 6.9: Sampling of a truncated sine function using an electrical circuit. The input is filtered and sampled with uniform sampling period T. The reconstruction involves applying a digital filter and running the annihilating filter method in order to retrieve the parameters of the original continuous-time signal $x(t)$.

filter. Figure 6.9 illustrates the steps of the reconstruction.

6.9 Summary and key results

We have set out to show that piecewise sinusoids belong to the family of signals with finite rate of innovation and can be sampled and perfectly reconstructed using sampling kernels that reproduce exponentials or polynomials. These classes of kernels are physically realizable and are of compact support. Moreover, combinations of piecewise sinusoids and polynomials also have a finite rate of innovation and can be dealt with using similar sampling schemes. This combination gives rise to a very general type of signal.

Since the sampling scheme is limited by the rate of innovation rather than the actual frequency of the continuous-time signal, we are, in theory, capable of retrieving piecewise sine waves with an arbitrarily high frequency along with the exact location of the switching points. We are therefore able to sample and perfectly reconstruct scenes in which there are several fronto-parallel planes with periodic bandlimited textures pasted onto them. However, this case remains limited to synthetic scenes. Therefore, we believe that the sampling scheme presented may find applications, for example, in spread spectrum and wide band communications where the signals are of the type presented in this chapter and precise time and frequency localization are of crucial importance.

Finally, we assumed deterministic signals throughout the chapter. The noisy case scenario is not in the scope of this work but will be tackled in the future.

Chapter 7

Conclusions and future work

7.1 Summary of Thesis Achievements

Image based rendering is in essence the problem of sampling and interpolating the seven-dimensional plenoptic function. For many applications, the four-dimensional light field representation is sufficient. Many works have studied the structure of light fields and showed that the band of the function is spread by occlusion and large depth variations. However, an accurate geometry of the scene is often not necessary.

In this thesis, we looked into extracting regions in the light fields that are coherent and can be rendered without aliasing. In doing so, we looked into the coherence of multiview images from the plenoptic function point of view emphasizing that looking at the problem from this angle provides a nice framework for studying the data in a global manner and imposing a coherent segmentation. Using this representation, we looked into the nature of the function and suggested in Chapter 2 that in extension to the object tunnels in videos and EPI-tubes in multi-baseline stereo data, layers carve multi-dimensional hypervolumes in the plenoptic function. We called these regions plenoptic hypervolumes. Just like in the three-dimensional cases, the hypervolumes contain highly regular information since they are constructed with images of the same objects. There is therefore clearly potential for robust analysis and efficient representation.

In Chapter 4, we proposed a novel multi-dimensional segmentation scheme based on a variational framework for the extraction of coherent regions in light fields. Since the formulation is global, coherence and consistency is enforced on all the dimensions. The method presented also takes fully advantage of the constraints namely that points in space are mapped onto particular trajectories (i.e. lines in light fields) and occlusions occur in a specific order. Therefore occlusions are naturally handled and all the images are treated equally. The variational framework is flexible in terms of the number of dimensions (i.e. depending on the camera setup), the depth estimation and the descriptors used. This flexibility is important for several reasons. First, the same framework can be used for different camera setups. Second, most applications in image based rendering do not necessitate an accurate depth reconstruction. Third, possible extensions to take into account large textureless regions and specular effects, for instance, may be incorporated into the descriptors.

We illustrated in Chapter 5 some applications in image based rendering using the extracted regions. View interpolation and scene manipulation as well as augmented reality were demonstrated in order to illustrate the benefits of extracting these plenoptic hypervolumes. Since the scene is not represented as layers in a traditional sense but as a combination of coherent hypervolumes, accurate geometry is often not necessary and fronto-parallel depth models were used to represent the regions.

In Chapter 6, we looked at sampling and interpolation in a more theoretical manner and investigated sampling and perfectly reconstructing signals that follow piecewise models. New finite rate of innovation sampling schemes are able to sample and perfectly reconstruct Diracs and piecewise polynomials. In Chapter 6, we showed that piecewise sinusoidal signals also belong to the family of signals that can be sampled and perfectly reconstructed using FRI principles. Moreover, combinations of piecewise sinusoidal and polynomial signals can be dealt with in the same way. Perhaps most interestingly, this chapter showed that it is theoretically possible to sample and perfectly reconstruct piecewise sinusoidal signals with sine waves having an arbitrarily high frequency.

7.2 Future research

In conclusion, we discuss some open questions and possible directions for future research. The variational framework derived in this thesis analyzes a sparse light field for image based rendering applications. Through experimental results, we have shown that it is capable of rendering photorealistic images of complicated and cluttered scenes. There are, however, some further developments that are possible:

- **Extensions to higher dimensions** The global analysis of light fields is powerful and enables one to take advantage of the coherence of the plenoptic function throughout all the images. However, such an analysis inherently implies that the cameras be well calibrated and structured in order for the plenoptic constraints to be imposed. For example, in Chapter 4 we derived a four-dimensional variational framework for the extraction of coherent regions in light fields. In future work, one may explore more complicated and unstructured camera setups as well as dynamic scenes and non rigid objects. For instance, an interesting direction for future work is to explore the potential of the global formulation to automatically extract coherent regions in dynamic light fields. This would imply using the constraints and imposing coherence in the four spatial dimensions as well as the time dimension, effectively leading to a five-dimensional framework.

- **Adaptive layer-based representation** Layer-based representations are powerful but suffer form several disadvantages. It is, indeed, difficult for these representations to deal with complicated layers such as trees, branches and leaves. That is, the rendered image might suffer from artifacts due to errors in segmenting these complicated layers. Moreover, the algorithm assumes that each individual layer can be individually rendered free of aliasing. That is, if the number of cameras is not sufficient to render a layer, the resulting interpolated viewpoint will suffer from aliasing regardless of the segmentation. In our current implementation, the choice of the number of depth layers that represent the light field is given to the user or is predetermined. This condition was imposed in order to offer the tradeoff between computational complexity and rendering quality. It would, however, be useful to devise an algorithm capable of adaptively choosing the number of layers for a given scene and application.

The sampling scheme presented in Chapter 6 is fundamental in nature and proposes novel theoretical results in sampling and perfect reconstruction of piecewise sinusoidal and polynomial signals. However, in practice there are some issues that affect the reconstruction accuracy. These issues are discussed in the following:

- **Model mismatch** The proposed reconstruction scheme is based on parameters and models. That is, the input signals and sampling kernels such as exponential splines have predefined characteristics. If one considers an input signal that is not an exact piecewise sinusoidal signal or a sampling kernel that does not exactly reproduce exponentials or polynomials then a range of model mismatch errors are possible. Therefore, the reconstruction of the continuous-time signal will not be perfect. Note that, the reconstruction error depends on the degree of model mismatch, and in many cases, this error can be reduced by best-fit solutions.

- **Noise** The proposed reconstruction algorithms rely on continuous and noise-free moments computed from the observed samples. In practice, these observed samples will inevitably be corrupted by noise. Therefore the estimated signal moments will be corrupted as well and the proposed algorithms will not achieve perfect reconstruction. It would therefore be useful to study quantitatively the effects of noise on the sampled piecewise sinusoidal signals. Several solutions such as over-sampling [29] and Cadzow methods [11] have been proposed for the Dirac and polynomial cases. We believe therefore that similar methods to deal with noise should be successful in the piecewise sinusoidal case as well. Finally, it would be interesting to apply these methods to wideband communications where precise time and frequency localization are of crucial importance.

Appendix A

Proof of (4.10)

Theorem 1. *Assume a four-dimensional hypersurface parameterized by*

$$\vec{\Gamma}(s, v_x, v_y, \tau) = \begin{pmatrix} x(s,\tau) - v_x p(s,\tau) \\ y(s,\tau) - v_y p(s,\tau) \\ v_x \\ v_y \end{pmatrix},$$

where $\vec{\gamma}(s,\tau) = [x(s,\tau), y(s,\tau)]$ is the contour of the surface in $v_x = v_y = 0$. The τ is the evolution parameter and $\vec{v}_\Gamma = \partial\vec{\Gamma}/\partial\tau$ and $\vec{v}_\gamma = \partial\vec{\gamma}/\partial\tau$ are the velocities. Then the normal speed $\vec{v}_\Gamma \cdot \vec{n}_\Gamma$ of $\vec{\Gamma}$ projected onto the subspace in $v_x = v_y = 0$ is related to the normal speed $\vec{v}_\gamma \cdot \vec{n}_\gamma$ of $\vec{\gamma}$ by the relation $\vec{v}_\Gamma \cdot \vec{n}_\Gamma = \chi(s, v_x, v_y, \tau)(\vec{v}_\gamma \cdot \vec{n}_\gamma)$ with

$$\chi(s, v_x, v_y, \tau) = \frac{(1 + \frac{\partial p}{\partial x} v_x + \frac{\partial p}{\partial y} v_y)\sqrt{(\frac{\partial x}{\partial s})^2 + (\frac{\partial y}{\partial s})^2}}{\sqrt{(\frac{\partial x}{\partial s} + (\frac{\partial p}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial s})v_x)^2 + (\frac{\partial y}{\partial s} + (\frac{\partial p}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial s})v_y)^2}}.$$

Proof. Let $\vec{\Gamma}(s, v_x, v_y, \tau)$ be a 4D hypersurface parameterized by

$$\vec{\Gamma}(s, v_x, v_y, \tau) = \begin{pmatrix} x(s,\tau) + p(s,\tau)v_x \\ y(s,\tau) + p(s,\tau)v_y \\ v_x \\ v_y \end{pmatrix},$$

where $\vec{\gamma}(s,\tau) = [x(s,\tau), y(s,\tau)]$ is the curve defined by the intersection of $\vec{\Gamma}$ and the plane in $v_x = v_y = 0$. The outward unit normal $\vec{n}_\Gamma = \hat{\vec{n}}_\Gamma / |\hat{\vec{n}}_\Gamma|$ to the hypersurface must be orthogonal to its tangent vectors $\frac{\partial \vec{\Gamma}}{\partial s}$, $\frac{\partial \vec{\Gamma}}{\partial v_x}$ and $\frac{\partial \vec{\Gamma}}{\partial v_y}$. Therefore, it must satisfy

$$\begin{bmatrix} \frac{\partial \vec{\Gamma}}{\partial s}^{tr} \\ \frac{\partial \vec{\Gamma}}{\partial v_x}^{tr} \\ \frac{\partial \vec{\Gamma}}{\partial v_y}^{tr} \end{bmatrix} \hat{\vec{n}}_\Gamma = \begin{bmatrix} \frac{\partial x}{\partial s} + \frac{\partial p}{\partial s} v_x & \frac{\partial y}{\partial s} + \frac{\partial p}{\partial s} v_y & 0 & 0 \\ p(s,\tau) & 0 & 1 & 0 \\ 0 & p(s,\tau) & 0 & 1 \end{bmatrix} \hat{\vec{n}}_\Gamma = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{A.1}$$

where the superscript tr denotes the transpose. The solution to this system of equations can be seen as a four-dimensional extension of the cross product as shown in [10] and gives

$$\hat{\vec{n}}_\Gamma = \begin{pmatrix} -\frac{\partial y}{\partial s} - \frac{\partial p}{\partial s} v_y \\ \frac{\partial x}{\partial s} + \frac{\partial p}{\partial s} v_x \\ p(\frac{\partial y}{\partial s} + \frac{\partial p}{\partial s} v_y) \\ p(\frac{\partial x}{\partial s} + \frac{\partial p}{\partial s} v_x) \end{pmatrix}. \tag{A.2}$$

The orthogonality of $\hat{\vec{n}}_\Gamma$ with each of the gradient vectors may easily be verified by substituting (A.2) in (A.1). The speed of $\vec{\Gamma}$ is given by

$$\vec{v}_\Gamma = \frac{\partial \vec{\Gamma}}{\partial \tau} = \begin{pmatrix} \frac{\partial x}{\partial \tau} + \frac{\partial p}{\partial \tau} v_x \\ \frac{\partial y}{\partial \tau} + \frac{\partial p}{\partial \tau} v_y \\ 0 \\ 0 \end{pmatrix}. \tag{A.3}$$

Consider now the outward normal $\hat{\vec{n}}_\gamma$ and speed function $\vec{v}_\gamma$ of the 2D contour $\vec{\gamma}(s,\tau)$. Clearly, we have

$$\hat{\vec{n}}_\gamma = \begin{pmatrix} -\frac{\partial y}{\partial s} \\ \frac{\partial x}{\partial s} \end{pmatrix} \qquad \vec{v}_\gamma = \begin{pmatrix} \frac{\partial x}{\partial \tau} \\ \frac{\partial y}{\partial \tau} \end{pmatrix}.$$

We may now explicitly derive $\hat{\vec{n}}_\Gamma \cdot \vec{v}_\Gamma$ using (A.2) and (A.3). After expansion, the normal speed gives

$$\hat{\vec{n}}_\Gamma \cdot \vec{v}_\Gamma = (1 + \frac{\partial p}{\partial x} v_x + \frac{\partial p}{\partial y} v_y) \hat{\vec{n}}_\gamma \cdot \vec{v}_\gamma,$$

where we have used $\frac{\partial p}{\partial s} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial s}$ and $\frac{\partial p}{\partial \tau} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial \tau} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial \tau}$. Finally, using the unit norm vectors $\hat{\vec{n}}_\Gamma = \vec{n}_\Gamma |\hat{\vec{n}}_\Gamma|$ and $\hat{\vec{n}}_\gamma = \vec{n}_\gamma |\hat{\vec{n}}_\gamma|$ gives:

$$\vec{n}_\Gamma \cdot \vec{v}_\Gamma = \frac{(1 + \frac{\partial p}{\partial x} v_x + \frac{\partial p}{\partial y} v_y)|\hat{\vec{n}}_\gamma|}{|\hat{\vec{n}}_\Gamma|} (\vec{n}_\gamma \cdot \vec{v}_\gamma)$$

which shows that

$$\chi(s, v_x, v_y, \tau) = \frac{(1 + \frac{\partial p}{\partial x} v_x + \frac{\partial p}{\partial y} v_y)\sqrt{(\frac{\partial x}{\partial s})^2 + (\frac{\partial y}{\partial s})^2}}{\sqrt{(\frac{\partial x}{\partial s} + (\frac{\partial p}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial s})v_x)^2 + (\frac{\partial y}{\partial s} + (\frac{\partial p}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial s})v_y)^2}}.$$

Note that since we consider the projection on the subspace in $v_x = v_y = 0$, we take only the (x, y) components of the normal $\hat{\vec{n}}_\Gamma$.

□

Bibliography

[1] Special issue on multiview imaging and 3DTV. *IEEE Signal Processing Magazine*, 24, November 2007.

[2] 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. MIT Press, Cambridge, MA, 1991.

[3] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 434–441, 1998.

[4] J. Berent and P. L. Dragotti. Perfect reconstruction schemes for sampling piecewise sinusoidal signals. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 377–380, May 2006.

[5] J. Berent and P. L. Dragotti. Segmentation of epipolar plane image volumes with occlusion and dissocclusion competition. In *IEEE Int. Workshop on Multimedia Signal Processing*, pages 182–185, October 2006.

[6] J. Berent and P. L. Dragotti. Plenoptic manifolds. *IEEE Signal Processing magazine*, 24(7):34–44, November 2007.

[7] J. Berent and P. L. Dragotti. Unsupervised extraction of coherent regions for image based rendering. In *British Machine Vision Conference*, September 2007.

[8] S. Beucher. Watersheds of functions and picture segmentation. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume 7, pages 1928–1931, 1982.

[9] R.E. Blahut. *Theory and Practice of Error Control Codes.* Addison-Wesley, 1983.

[10] J. F. Blinn. Lines in space. 1. the 4D cross product. In *IEEE Computer Graphics and Applications*, volume 23, pages 84–91, May-June 2003.

[11] T. Blu, P. L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot. Sparse sampling of signal innovations: theory, algorithms and performance bounds. *IEEE Signal Processing Magazine*, 25(2):31–40, March 2008.

[12] R.C. Bolles, H. H. Baker, and D.H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *Int. Journal of Computer Vision*, 1:7–55, 1987.

[13] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *IEEE Conf. on Computer Vision*, volume 1, pages 377–384, 1999.

[14] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, and M. F. Cohen. Unstructured lumigraph rendering. In *Computer graphics (SIGGRAPH '01)*, pages 425–432, 2001.

[15] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. on Information Theory*, 52(2):489–509, February 2006.

[16] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. Journal of Computer Vision*, 1(22):61–79, 1997.

[17] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. Minimal surfaces based object segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):394–398, April 1997.

[18] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong. Plenoptic sampling. In *Computer graphics (SIGGRAPH '00)*, pages 307–318, 2000.

[19] S. C. Chan, K. T. Ng, Z. F. Gan, K. L. Chan, and H. Y. Shum. The plenoptic video. *IEEE Trans. on Circuits and Systems for Video Technology*, 15:1650–1659, 2005.

[20] T. F. Chan and L. Vese. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–277, February 2001.

[21] C. L. Chang, X. Zhu, P. Ramanathan, and B. Girod. Light field compression using disparity-compensated lifting and shape adaptation. *IEEE Trans. on Image Processing*, 15(4):793–806, April 2006.

[22] A. Chebira, P. L. Dragotti, L. Sbaiz, and M. Vetterli. Sampling and Interpolation of the Plenoptic Function. In *IEEE International Conference on Image Processing*, pages 917–920, September 2003.

[23] S. E. Chen. Quicktime VR - an image based approach to virtual environment navigation. In *Computer graphics (SIGGRAPH '95)*, pages 29–38, August 1995.

[24] A. Criminisi, S. B. Kang, R. Swaminathan, R. Szeliski, and P. Anandan. Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Computer Vision and Image Understanding*, 97(1):51–85, January 2005.

[25] W. Dahmen and C. A. Micchelli. *On theory and application of exponential splines.* New York: Academic, 1987.

[26] M. Do, D. Marchand-Maillet, and M. Vetterli. On the bandlimitedness of the plenoptic function. In *IEEE International Conference on Image Processing*, pages 14–17, September 2005.

[27] D. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 52(4):1289–1306, April 2006.

[28] P. L. Dragotti, M. Vetterli, and T. Blu. Exact sampling results for signals with finite rate of innovation using the Strang-Fix conditions and local kernels. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, March 2005.

[29] P. L. Dragotti, M. Vetterli, and T. Blu. Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix. *IEEE Trans. on Signal Processing*, 55(5):1741–1757, May 2007.

[30] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. *IEEE Trans. on Image Processing*, 3:336–344, 1998.

[31] I. Feldmann, P. Eisert, and P. Kauff. Extension of epipolar image analysis to circular camera movements. In *IEEE Int. Conf. on Image Processing*, pages 697–700, September 2003.

[32] T. Fujii, T. Kimoto, and M. Tanimoto. Ray space coding for 3D visual communication. In *Picture Coding Symposium*, pages 447–451, 1996.

[33] Z. F. Gan, S. C. Chan, K. T. Ng, and H. Y. Shum. An object-based approach to plenoptic videos. In *IEEE Int. Symp. on Circuits and Systems*, volume 4, pages 3435–3438, May 2005.

[34] N. Gehrig and P. L. Dragotti. Distributed sampling and compression of scenes with finite rate of innovation in camera sensor networks. In *Data Compression Conference (DCC 2006)*, March 2006.

[35] B. Goldlücke and M. Magnor. Space-time isosurface evolution for temporally coherent 3D reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 350–355, June 2004.

[36] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Computer graphics (SIGGRAPH '96)*, pages 43–54, 1996.

[37] A. Gouze, C. De Roover, B. Macq, A. Herbulot, E. Debreuve, and M. Barlaud. Watershed-driven active contours for moving object detection. In *IEEE International Conference on Image Processing*, pages 818–821, 2005.

[38] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004.

[39] A. Isaksen, L. McMillan, and S. J. Gortler. Dynamically reparameterized light fields. In *Computer graphics (SIGGRAPH '00)*, pages 297–306, 2000.

[40] S. Jehan-Besson, M. Barlaud, and G. Aubert. Detection and tracking of moving objects using a new level set based method. In *Int. Conf. on Pattern Recognition*, pages 1112–1117, September 2000.

[41] S. Jehan-Besson, M. Barlaud, and G. Aubert. Video object segmentation using Eulerian region-based active contours. In *IEEE Int. Conf. on Computer Vision*, pages 353–361, 2001.

[42] S. Jehan-Besson, M. Barlaud, and G. Aubert. DREAMS: Deformable regions driven by an Eulerian accurate minimization method for image and video segmentation. *Int. Journal of Computer Vision*, 53(1):45–70, 2003.

[43] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. Journal of Computer Vision*, 1(4):321–331, 1988.

[44] Q. Ke and T. Kanade. A subspace approach to layer extraction. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 255–262, 2001.

[45] J. Konrad. Videopsy: dissecting visual data in space-time. *IEEE Communications magazine*, 45:34–42, 2007.

[46] J. Konrad and M. Halle. 3-D displays and signal processing. 24(6):97–111, November 2007.

[47] A. Kubota, K. Aizawa, and T. Chen. Reconstructing dense light field from a multi-focus images array. In *IEEE International Conference on Multimedia and Expo*, volume 3, pages 2183–2186, 2004.

[48] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang. Multiview imaging and 3DTV. In *IEEE Signal Processing magazine*, volume 24, pages 10–21, November 2007.

[49] K. Kutulakos and S. Seitz. A theory of shape by space carving. *Int. Journal of Computer Vision*, 3(38):199–218, 2000.

[50] M. Levoy and P. Hanrahan. Light field rendering. In *Computer graphics (SIGGRAPH '96)*, pages 31–42, 1996.

[51] M. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 710–717, 2003.

[52] D. Lischinksi and A. Rappoport. Image-based rendering for non-diffuse synthetic scenes. In *Proc. Ninth Eurographics Workshop on Rendering*, pages 301–314, 1998.

[53] A. Mansouri and J. Konrad. Multiple motion segmentation with level sets. *IEEE Trans. on Image Processing*, 12(2):201–220, February 2003.

[54] B. J. McCartin. Theory of exponential splines. *Journal of Approximation Theory*, 66:1–23, 1991.

[55] L. McMillan and G. Bishop. Plenoptic modeling: an image-based rendering system. In *Computer graphics (SIGGRAPH '95)*, pages 39–46, 1995.

[56] A. Mitiche, R. Feghali, and A. Mansouri. Motion tracking as a spatio-temporal motion boundary detection. In *Robotics and autonomous systems*, volume 43, pages 39–50, 2003.

[57] A. S. Ogale and Y. Aloimonos. Shape and the stereo correspondence problem. *Int. Journal of Computer Vision*, 65(3):147–162, December 2005.

[58] M. Ristivojevic and J. Konrad. Space-time image sequence analysis: object tunnels and occlusion volumes. *IEEE Trans. on Image Processing*, 15(2):364–376, February 2006.

[59] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journal of Computer Vision*, 47(1):7–42, 2002.

[60] H.-Y. Schum and L.-W. He. Rendering with concentric mosaics. In *Computer graphics (SIGGRAPH '99)*, pages 299–306, 1999.

[61] H.-Y. Schum and S. B. Kang. A review of image-based rendering techniques. In *IEEE/SPIE Visual Communications and Image Processing (VCIP)*, pages 2–13, June 2000.

[62] L. L. Schumaker. *Spline Functions: Basic Theory.* Wiley, 1981.

[63] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 519–528, June 2006.

[64] J. Sethian. *Level Set Methods.* Cambridge University Press, 1996.

[65] J. Shade, S. Gortler, L. W. He, and R. Szeliski. Layered depth images. In *Computer graphics (SIGGRAPH '98)*, pages 231–242, 1998.

[66] C.E. Shannon. Communications in the presence of noise. *Proc. of the IRE*, 37:10–21, January 1949.

[67] Y. Shi and W. Karl. A fast level set method without solving pdes. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 97–100, March 2005.

[68] P. Shukla and P. L. Dragotti. Sampling schemes for multidimensional signals with finite rate of innovation. *IEEE Trans. on Signal Processing*, 55(7):3670–3686, July 2007.

[69] H. Y. Shum, J. Sun, S. Yamazaki, Y. Li, and C. K. Tang. Pop-up light field: An interactive image-based modeling and rendering system. *ACM Trans. Graph.*, 23(2):143–162, April 2004.

[70] J.E. Solem and N.C. Overgaard. A geometric formulation of gradient descent for variational problems with moving surfaces. In *Int. Conf. on Scale Space and PDE methods in Computer Vision, Scale Space 2005*, 2005.

[71] T. Stich, A. Tevs, and M. Magnor. Global depth from epipolar volumes - a general framework for reconstructing non-lambertian surfaces. In *3DPVT*, pages 1–8, 2006.

[72] P. Stoica and R. Moses. *Introduction to Spectral Analysis.* Prentice Hall, 2000.

[73] G. Strang and G. Fix. A Fourier analysis of the finite element variational method. In *Constructive Aspects of Functional Analysis*, pages 796–830, 1971.

[74] M. Unser and T. Blu. Cardinal exponential splines: Part I—Theory and filtering algorithms. *IEEE Trans. on Signal Processing*, 53(4):1425–1438, April 2005.

[75] M. Vetterli, P. Marziliano, and T. Blu. Sampling signals with finite rate of innovation. *IEEE Trans. on Signal Processing*, 50:1417–1428, June 2002.

[76] J. M. N. Vieira and P. J. S. G. Ferreira. Interpolation, spectrum analysis, error-control coding and fault-tolerant computing. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, April 1997.

[77] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991.

[78] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Trans. on Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638, September 1994.

[79] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph.*, 24(3):765–776, 2005.

[80] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. In *IEEE Trans. on Pattern Analysis and Machine intelligence*, volume 27, pages 1644–1659, 2005.

[81] J. C. Yang, M. Everett, C. Buehler, and L. McMillan. A real-time distributed light field camera. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering*, pages 77–86. Eurographics Association, 2002.

[82] J. D. Young. Numerical applications of hyperbolic spline functions. *Logistics Rev.*, 4:17–22, 1968.

[83] C. Zhang and T. Chen. Generalized plenoptic sampling. Technical Report AMP 01-06, Advanced Multimedia Processing Lab, Electrical and Computer Enginering, Carnegie Mellon University, Pittsburgh, PA 15213, September 2001.

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

[85] C. Zhang and T. Chen. A survey on image-based rendering. Technical Report AMP 03-03, Advanced Multimedia Processing Lab, Electrical and Computer Enginering, Carnegie Mellon University, Pittsburgh, PA 15213, June 2003.

[86] C. Zhang and T. Chen. A self-reconfigurable camera array. In *Eurographics symposium on rendering*, pages 243–254, 2004.

[87] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000.

[88] S. C. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(9):884–900, September 1996.

[89] C. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *Computer graphics (SIGGRAPH '04)*, pages 600–608, 2004.