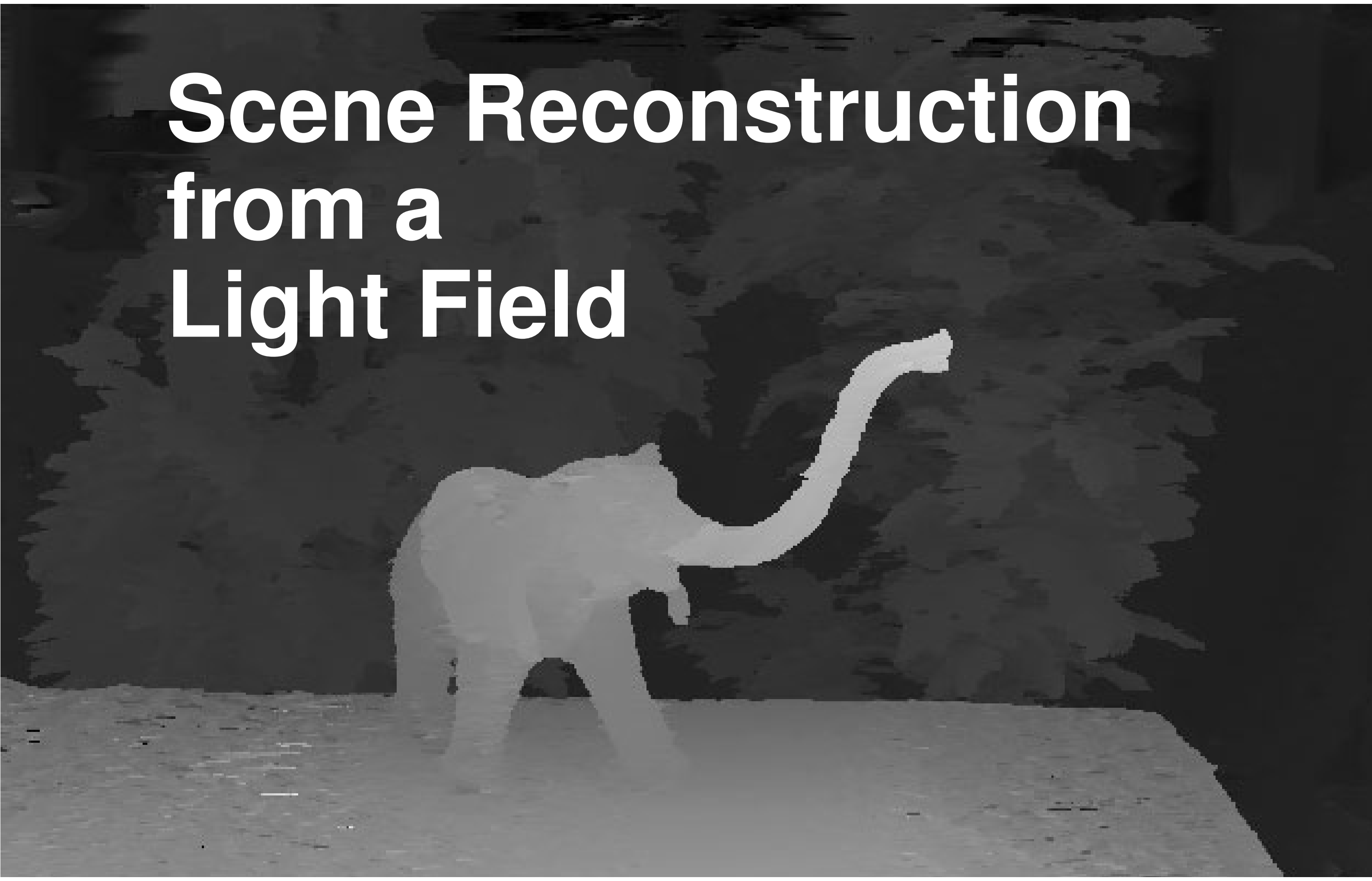
Scene Reconstruction from a Light Field



Changil Kim

Master Thesis
ETH Zürich
September 2010

Supervised by
Simon Heinzle,
Dr. Wojciech Matusik, *and*
Prof. Dr. Markus Gross

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Abstract

This thesis addresses novel methods to reconstruct the scene from a light field. Light fields are sampled radiances at all points along all directions in the scene. We propose to reconstruct the scene by analyzing the structures of rays in the light field without any geometric assumption about the scene. We first explore the general characteristics of light fields, and important operations such as digital refocusing and synthetic aperture. Using those operations, we derive a representation called a focal stack to describe the scene, and the mapping between the light field and the focal stack. We then devise tools to measure the existence of surfaces in the focal stack. Being equipped with these tools, we propose to reconstruct the depth of the scene, and then to reconstruct the whole 3D volume of the scene. We also discuss the effects of the occlusion, and seek to reduce its influence on the synthetic aperture imaging to make our methods more robust. Our methods are tested using captured and synthetic data sets, and the experimental results are presented and discussed.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

cgl
Computer Graphics Laboratory ETH Zurich
Prof. Markus Gross

Master thesis for Mr. Changil Kim

Scene pre-capturing for stereoscopic video

Introduction

3D filming and display attracts worldwide attention and experts from the field argue that the transition from 2D to 3D is the next milestone after colour TV, stereo sound, and high-definition. One of the main driving forces behind this trend is the digitalization of filming: digital 3D video sequences offer completely new means for pre- and post-processing . However, stereoscopic filming still requires extensive knowledge about the principles of stereography, and shooting 3D films still can only be performed by a handful of skilled operators.
Disney Research Zurich and the Computer Graphics Laboratory of ETH Zurich have developed a digital acquisition system for high-quality stereo video. The goal of the system is to aid the camera operator and stereographer with semi-automatic means to adjust the two most important parameters: baseline and convergence. As one enhancement of the pipeline, the system should be augmented by a mechanism to „pre-capture“ the depth of a static scene, and to use this depth for the subsequent acquisitions of the actors in front of the pre-captured scene.

Assignment

- Familiarization with and assessment of existing algorithms for dense stereo acquisition of static and for dynamic scenes, as well as basic concepts in 3D movie making.
- Evaluation of existing methods in terms of hardware requirements, scene assumptions, and quality of results.
- Development and implementation of a pre-capture method meeting following key requirements:
    1. A set of densely sampled input images should yield an accurate and densely sampled depth field of the scene.
    2. The method should work without physical modification of the current hardware system in real-time.
    3. The method can either be adapted from previous work, or be developed by the student.
- Integration into the existing hardware system.
- Evaluation and results.
- Extensions to this assignment are as follows:
    - Use pre-captured depth to determine a faithful depth of dynamic scene.
    - Use pre-captured depth for compositing with other captured scenes.

Remarks

A written report and an oral presentation conclude the work. The master thesis is overseen by Prof. Markus Gross and is supervised by Simon Heinzle, Institute for Visual Computing, and Dr. Wojciech Matusik, Disney Research Zurich.

Hand-Out: March 12, 2010
End-Date: September 11, 2010

Contents

1

Introduction

Since the light field was adopted in computer graphics it has been widely used in many applications, including image-based rendering, computational photography, and 3D displaying systems. A light field contains visual appearances of a scene observed from various viewing positions and directions. The virtue of the light field is that it is a simple and elegant representation of a 3D scene, and can be constructed from a collection of 2D images. The light field does not require scene geometry and complex models for the appearance of scene points, which are usually needed by geometry based scene representations. This characteristic makes operations on light fields simple and computationally efficient.

However, this advantage also comes with disadvantages. The scene geometry such as the depth is one of the most important notions of the scene, and is still required by many algorithms and applications. Despite its importance, the light field has no notion of the geometry. In addition, there is a practical difficulty that a large amount of data must be acquired, processed, and maintained to work with light fields. Since light fields reside in a high dimensional space and should be acquired in a fine resolution to avoid aliasing, they are spatially demanding and redundant. Thus, it is beneficial to convert the light field to another representation which is compact and can be fed into required applications. In this thesis, we focus on reconstructing the scene geometry by identifying the depth of visible scene points as well as the texture of scene points.

In computer vision, the scene reconstruction has long been a fundamental problem. Many approaches have been sought to reconstruct a good representation of a scene from images. Those approaches typically use two views or some small number of views to reconstruct the scene depth, and thus involve assumptions about prior knowledge about the scene and/or highly nonlinear operations such as complex optimizations. In this context, capturing and analyzing a light field of a scene can be understood as a mean of facilitating the scene reconstruction. Using the light field as a scene capturing method, we can capture more information about the scene in a structured way and exploit this additional information acquired by light field to reconstruct the scene with simpler operations. With light fields, a scene can be analyzed more systematically, since light fields are usually acquired in a regular sampling pattern.

In this thesis, we explore novel methods to reconstruct the scene from the light field. More specifically, we propose to reconstruct 3D volumes containing scene geometry and texture information of the scene. We first transform a light field to another scene representation called a focal stack, which has a more intuitive relation to the 3D scene. We then extract scene points lying on surfaces and their texture from the focal stack. In order to do this, we use focus measure criteria to determine the confidence of surface existence at each point in the focal stack. The influence of occlusions in the focal stack increases as the aperture (baseline) becomes wider. Therefore, the occlusions are explicitly handled.

Our contributions are as follows. First, we analyze the light field with respect to the depth, and devise depth measure criteria robust to occlusions. Second, we present a framework for the scene reconstruction from light fields, and provide specific methods to extract a view-dependent depth representation. Lastly, we propose an extended method to reconstruct the 3D scene volume from a light field, and present a detailed analysis of the results.

The proposed methods may be used in the shape and texture recovery of, for example, archaeological sites as a passive 3D reconstruction algorithm. It can also be used to analyze the scene in more specific purposes. For example, in 3D cinematography, the scene may be pre-captured before the actual filming. 3D cameras can easily capture a light field by adjusting the baseline of their two cameras. The acquired light field can then be used to analyze the scene to extract useful information to cinematographers such as the scene composition and depth distribution.

A rough scene geometry can be captured along with the video stream using a linear array of inexpensive cameras attached to a high quality main camera. The obtained scene geometry can then be used to render the second view to produce a stereoscopic 3D movie. Lastly, the proposed methods can be used to compress already acquired light fields by removing geometry-dependent redundancy, and also plan a second phase acquisition to efficiently acquire a finer resolution light field based on its coarse version.

The thesis starts with reviewing related previous work in Chapter 2. The theoretical background including important notions used throughout the thesis is presented in Chapter 3. Then, we present the methods to reconstruct the scene in Chapter 4. The experimental results and implementation details are reported in Chapter 5. Finally, Chapter 6 concludes the thesis with the summary and possible future work.

2

Related Work

The thesis was inspired by the previous works about epipolar-plane image analysis, range finding techniques, and light field photography. This chapter briefly reviews some of remarkable works about them.

Epipolar-Plane Image Analysis. Before the introduction of the light field to computer graphics and vision communities, its two dimensional slice known as epipolar-plane image (EPI) has been researched extensively.

Bolles et al. [BBM87] introduced the notion of a spatio-temporal volume of a static scene obtained by a camera motion. They investigated the techniques to analyze slices of this volume, the EPIs, to extract the 3D positions of objects in the scene as well as the relation between the objects such as occlusions of objects. They presented a method to estimate the depth of scene points from the EPI formed by a linear camera motion based on detecting edge features, which produces a map of *free space* not occluded by objects, yielding a sparse representation of the scene. Due to the sparsity of detected edge features, however, this scene reconstruction was also sparse, not generating a dense representation of the scene. They opened the possibility for their methods to be applied to more complex camera motions and moving objects by analyzing the projective duality, so that a free space map of the scene can be generated by a robot scanning the scene.

While most approaches focused on a sparse set of features in EPI, the following methods proposed to reconstruct a dense representation of the scene. Katayama et al. [KTOT95] proposed one of the earliest methods for dense representation of the scene from EPI in their approach to generate new views by the interpolation and reconstruction of multi-view images for autostereoscopic displays. They discussed the detection of trace lines of correspondence points on EPIs. To detect a trace line, they used the variance of color values on candidate straight lines, and accept the least steepest one among all the possible trace lines with predefined range of slope giving the variance smaller than a threshold. They also proposed to remove the pixels in the EPI which were already fitted to trace lines in order to deal with occlusions. However, their line fitting criterion used only the variance, which is too simple to handle complex occlusions. The decision based on thresholding is prone to errors, and may generate noisy reconstruction. Although their method was rather simple and they did not investigated further about the scene reconstruction since their interest was the view interpolation, their idea has been a starting point for the later researches about the dense scene reconstruction.

Intille and Bobick [IB94] proposed a notion of disparity-space image (DSI) to model occlusions and to help solve the stereo matching problem, which can be easily built from EPI. The DSI is a 2D slice of the discretized 3D volumetric model of the scene, which is similar with the focal stack we use to represent the 3D scene. Crimisini et al. [CKS+05] proposed a method to segment depth layers from EPI by exploiting the high degree of regularity found in the EPI. They also used the intensity variance to identify trace lines corresponding to surface points, and used DSI to identify coherent scene points constituting layers separated by occlusion boundaries. The method iteratively peels off occluding points from the EPI while updating the variance computation. They also proposed to segment EPI by detecting straight line and extracting the quadrilateral bounded by most slanted straight lines.

Their method is relevant to ours in that the use of EPI and DSI in tandem is similar with ours of the light field and the focal stack. However, they use DSI to compute occluded regions, and maintained a visibility weight mask based on this computation, whereas we propose to compute the focal stack and the focus measure free from the influence of occlusions. Although their method was more sophisticated and mathematically based, they also used a simple variance based criterion to identify trace lines. In addition, their ray removal scheme based on the visibility mask is sensitive to discretization, and prone to errors in practice.

In computer vision community, the depth estimation from multiple views has been actively studied. Among those researches, Kang and Szeliski's proposal [KS04] to extract view-dependent depth maps from image sequences is very relevant to our approach. In their proposal, a depth map associated to each view is independently estimated, and all those depth maps are combined to model the variation in object appearance with respect to the viewing position, which is similar with our multiple view projection to reconstruct 3D volumetric representation of the scene. They used the combination of shiftable windows and temporal selection to estimate depth maps, which is also relevant to our ray selection scheme.

Range Finding Techniques. In 1987, Pentland [Pen87] and Grossmann [Gro87] proposed independently new range finding techniques based on defocusing due to the finite depth of field. Their methods are called *depth from defocus*, and *depth from focus*, respectively. The methods based on defocusing have been applied to range finding such as auto-focusing as well as depth estimation. Both methods use images covering the same amount of the scene taken with different focus settings (depth from focus), or different aperture settings (depth from defocus). Depth from focus (DFF) [Gro87, Kro87, DW88, NN90] seeks the depth of a scene point by identifying the focus setting with the point sharpest focused. On the other hand, depth from defocus (DFD) [Pen87, PDTH89, SS94] directly computes the depth using the ratio of the spectral power of differently blurred images of the scene point. Thus, in general DFD needs less images (two or three images) than DFF (more than ten images).

The properties of defocusing and the measurement of the sharpness of focus, or the focus measure, were already explored for the servo-controlled auto-focusing cameras [Kro87], most of which can be applied to DFF methods. Schechner and Kiryati [SK00] investigated the fundamental relation between defocus based methods and stereo based methods, and built a bridge between them. In fact the defocus based methods share the same principles as the triangulation based stereo methods. Recently, Hasinoff and Kutulakos [HK09] proposed a range finding technique incorporating both the focus and aperture settings, called confocal stereo. They defined the aperture-focus image (AFI), which is computed for each pixel of the image and the best fit of the defocus profile is sought to yield the associated depth.

These defocus based methods are relevant to our method in that our method generates all differently focused images during the depth estimation, which can be directly fed to those methods. In principle, one of those methods may directly be applied to our synthetically refocused scene volume called a focal stack. However, most of those approaches use cameras with the aperture of a few centimeters at maximum, while our method uses a wide synthetic aperture. As a result, the blur properties and the effect of occlusions become more complex. Furthermore, we have to deal with the visibility problem due to the unmatched correspondences, which is usually the case in wide baseline stereo based algorithms [SK00]. Therefore, in order to apply the conventional defocus base methods, we first have to handle the difficulties caused by the wide aperture.

Light Field Photography. Adenson and Bergen [AB91] introduced the concept of the *plenoptic function* to describe the visual information available from every point. The plenoptic function is a 7D function of the radiance of the light rays measured at every possible location at every possible angle for every wavelength at every time. In 1996, Levoy and Hanrahan [LH96] and Gortler et al. [GGSC96] adopted the plenoptic function in a reduced dimension to image-based rendering using the names "the light field" and "the lumigraph", respectively. We use the term light field in this thesis. The authors of both papers described the representation for the 4D light field and its parameterization scheme, and proposed to use the light field to capture the visual appearance of the scene from many views and to generate new views from arbitrary camera positions by interpolating existing views from the light field without the need for scene geometry. They discussed practical considerations including the acquisition, rendering, and compression, and addressed typical applications. While the two proposals were almost the same, Gortler et al. addressed the discretization assisted by rough geometric information to reduce aliasing.

Since then, the light field has been widely adopted to many existing problems, for example, to extend the existing photography. The concept of imaging through a virtual aperture was first mentioned by Levoy and Hanrahan. Isaksen et al. [IMG00] proposed a method to compute photographs focused at a variable focus with a variable depth of field from light fields. The method is based on dynamically reparameterizing light fields, which were originally parameterized by two parallel planes, using a synthetic aperture lying on the camera plane and the frontoparallel focal plane that is placed at an arbitrary distance to the camera surface. Vaish et al. [VGT$^+$05] extended this method to be able to refocus from light fields with a tilted focal plane using the shear-warp factorization of planar homography. Several papers addressed how to capture light fields using lenslets [AW92, NLB$^+$05] and camera arrays [LH96, IMG00, WJV$^+$05]. Chai et al. [CTCS00] addressed the sampling strategy of light fields to achieve alias-free rendering as well as efficient utilization of the storage.

Ng et al. [NLB$^+$05] implemented a hand-held camera with a lenslet array inserted between the sensor and the main lens of the camera. As the applications of their plenoptic camera, they demonstrated the extended depth of field to generate refocused or all-in-focus photographs, and the view point manipulation within the extent of the camera aperture, which can be effective in the close-up macro photography. In particular, they discussed and analyzed the technique to digitally refocus from light fields. The work on these methods is often called synthetic aperture photography [VWJL04], which is used to dynamically refocus photographs after the acquisition with different camera parameters, and see through occluding objects by blurring them out using very shallow depth of field [LCV+04, JAMK07].

The use of scene geometry was mentioned mostly in the context of reducing the aliasing with a limited resolution of the light field, and compressing acquired light fields. Although the use of scene geometry can help reduce aliasing and compress light fields, the scene geometry has been assumed to be already available, and has not been sought to be extracted from the light fields. The reconstruction of 3D geometry from multiple images has been an active area of research in computer vision, and has much similarity to the geometry reconstruction from light fields. Thus, many approaches can be taken to be applied to light fields. However, there have not been active researches to relate those techniques to light fields. Recently, Ziegler et al. [ZBA+07] addressed a method for depth estimation from light fields in their paper about the transformation between light fields and holograms. Their method estimates the depth based on a criterion measuring the variance over the rays or Fourier power spectrum over the 2D slice spanning the rays, and peel depth layers from front to back to deal with occlusions. This method is very similar with previous approaches regarding the EPI analysis as well as one of our proposed methods. However, the ray removal from light fields that they used to deal with occlusion is prone to errors in real images due to the same reason as for the previous work on depth estimation from EPIs.

Vaish et al. [VSZ+06] compared several cost functions used for 3D reconstruction in the context of the synthetic aperture and the robustness to occlusions, which is relevant to our analysis on focus measures. The cost function in their context is roughly equivalent to the reciprocal of the focus measure described in this thesis. They also proposed two novel cost functions based on median and entropy. Although we took a different approach from theirs to handle the occlusion and to devise focus measures robust to occlusion—we used the selection and clustering schemes—their analysis about the cost function and ours can be complement to each other.

Lastly, microscopy has a notable similarity to our problem. Especially, deconvolution microscopy is worth noting in the sense that both the techniques deal with the projection of volumetric data and reconstruction of objects from them [MKCC99]. Levoy et al. [LNA+06] applied light field photography to microscopy. The focal stack of a specimen is constructed from a single shot light field photograph, and is deconvolved using the estimated point spread function to remove the light pollution. However, microscopy deals with transparent microscopic objects while we are interested in mostly opaque macroscopic objects. Thus, this difference must be first attacked in order for deconvolution microscopy techniques to be applied to the scenes we are interested in.

3

Background

This chapter presents important concepts that will be used throughout the thesis. The two most important concepts are the light field and the focal stack. First, we present the notion of the light field and its representations, followed by the definition of the focal stack. We discuss how synthetic imaging using digital refocusing and a synthetic aperture can be achieved, which basically is to generate new images with different aperture and focus than the original images constituting the light field. We describe how the focal stack can be built based on the synthetic imaging. We then introduce the fundamentals of the depth estimation from the focal stack, which will be the basis of the scene reconstruction and be discussed in a greater detail in the remaining of the thesis. We close the chapter with the discussion of the effect of the wide aperture and the presence of occlusions on the focal stack and the depth estimation.

3.1 Light Field

A light field is a collection of radiances sampled at every position in every direction of a space. The term originates from Gershun's paper [Ger39] about the radiometric property of light, and was used in his paper to denote the irradiance vector field. However, it is used slightly differently in computer graphics and vision communities to denote the function of radiance. The plenoptic function introduced by Adelson and Bergen [AB91] is equivalent to this notion.

The light field is in general a 5D[1] function of three coordinates to represent the position of a three-dimensional point and two angles to represent the direction leaving the point. Ignoring the effects of participating media such as scattering and absorption, and considering the region of

[1] It is a 7D function, when incorporating the time t and the wavelength λ, which is called a *plenoptic function* in Adelson and Bergen's paper [AB91].

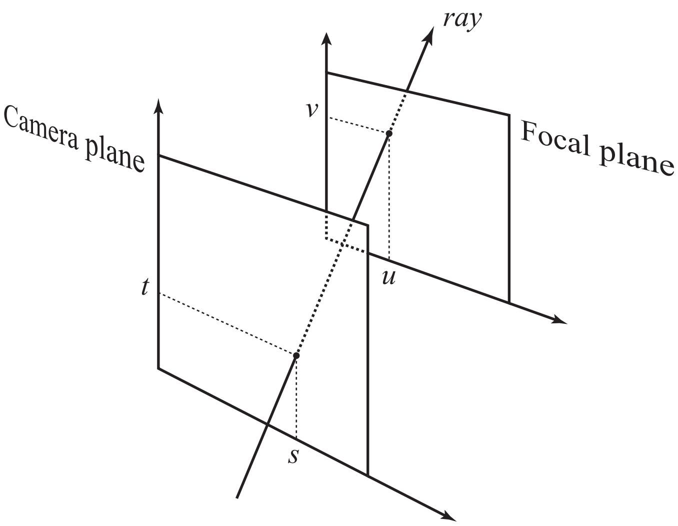


Figure 3.1: The two-plane parameterization of 4D light fields. The *st* plane parameterizes the directional dimension, and is called a camera plane. The *uv* plane parameterizes the spatial dimension, and is called a focal plane.

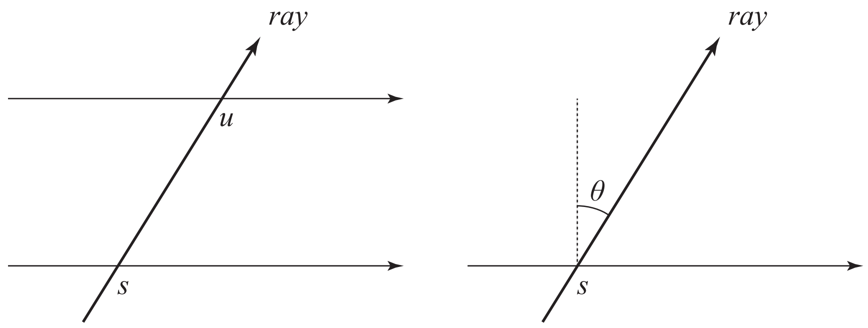
space free of occluders, the radiance can be assumed to be preserved along the ray conveying it. With this assumption, the dimension of the light field can be lowered to yield a 4D function, dropping the redundancy along 1D rays.

3.1.1 Parameterization

A 4D light field is often parameterized using 2 planes [LH96, GGSC96]. Although the two planes can be placed arbitrarily, they are usually placed in parallel. A ray is then parameterized by two points (s,t) and (u,v) on the two planes through which the ray passes (Figure 3.1). The *st* plane on which rays enter is called a camera plane, and the *uv* plane from which rays exit is called a focal plane. A point (s,t,u,v) in the light field is then defined as the radiance of the ray parameterized by the two points (s,t) and (u,v). With this parameterization, the light field can be conceptually constructed by taking 2D *uv* images at each *st* grid point in the camera plane, and inserting those images into the 4D structure.

Although light fields can be acquired using various methods, in practice there are two distinguished approaches categorizing those methods. The first uses a camera array [WJV$^+$05, LH96], and the second uses lenslets [NLB$^+$05, AB91]. In the first approach, 2D images are taken using a 2D camera array placed along the camera plane. Each image covers the focal plane, and yields a *uv* slice. Each camera position in the camera plane determines the *st*-coordinates of the *uv* slice. In the second approach using lenslets, a miniature lens grid is placed between the image sensor and the main lens of the camera, which effectively splits the rays focused at each *lenslet* of the lens grid. The sensor surface under a single lenslet contains all directional components at a particular spatial coordinate. Thus, the portion of the sensor surface covered by a lenslet constitutes an *st* slice, and the position of the lenslet in the grid determines the *uv*-coordinates of the *st* slice.

The two approaches have their own advantages and disadvantages. The acquisition using a



(a) Two-plane parameterization **(b)** Angular parameterization

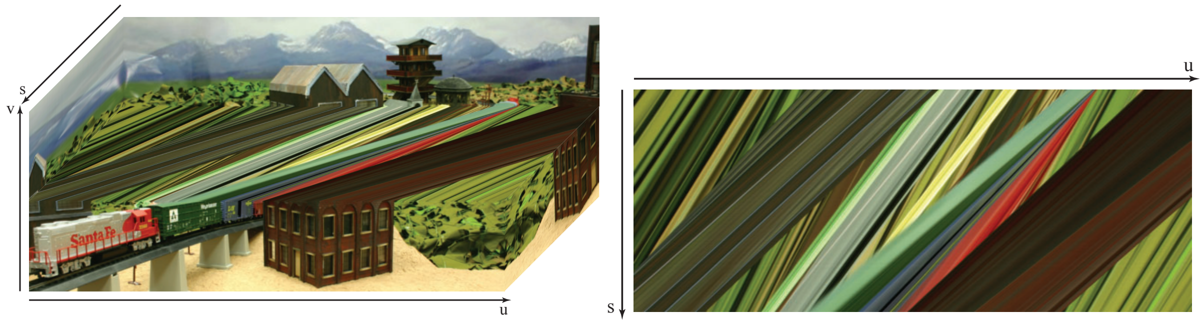
Figure 3.2: Comparison of 2D light field parameterizations. (a) With the two-plane parameterization, a ray is parameterized by (s,u), two points on the camera plane and the focal plane. (b) With the angular parameterization, a ray is parameterized by (s,θ), the position on the camera plane and the angle formed by the ray and the camera plane.

camera array can take advantage of a higher spatial resolution, and does not require the alteration of the camera. The acquisition using a lenslet array can obtain a higher and finer directional resolution, since the *st* grid resolution is not limited by the physical size of the camera. In addition, the calibration of a single camera as well as the acquisition step using the single camera is usually simpler than those with a camera array. One important factor with regard to the scene reconstruction is that we can exploit the advantage of a wide baseline when using a camera array. That is, the acquisition using a camera array can offer a wider range of directional differentiation of rays, which is better suited for the scene reconstruction. The maximum directional range of the light field acquired using the camera array spans the size of the camera array, while in the acquisition using a lenslet, the range spans the size of the single camera's aperture used in the acquisition.

3.1.2 Representations

It is difficult to find an intuitive way to visualize a 4D light field, but lower dimensional slices of the 4D light field provide meaningful representations. A *uv* slice of a 4D light field is a 2D perspective image with the center of projection placed at (s,t) and the plane of projection coinciding the focal plane. An *st* slice appears less straightforward, but resembles a hypothetical radiance function at the point (u,v) in the focal plane. See [LH96, Figure 6(b)] for an example of the *st* slice. Typically the (u,v) coordinates are called spatial dimension, and the (s,t) coordinates are called directional or angular dimension. The two-plane parameterization has several advantages. It works well with the conventional camera's ray sampling pattern, so that the acquisition of light fields is straightforward. Furthermore, the operations on the light field to extract a new image with different camera settings, such as focus and aperture, can be easily derived using the two-plane parameterization.

If the *uv* plane is placed at infinity, a ray may be more intuitively parameterized by a position and a direction. Since the direction is defined only by the (u,v) coordinates and is not influenced by the (s,t) coordinates, this setup has the same effect as the (u,v) coordinates defined in a local coordinate system affixed to each grid point in the *st* plane (Figure 3.4(a)). This setup has two



(a) 3D light field

(b) 2D light field

Figure 3.3: An example light field. (a) A 3D slice $L(s,u,v)$ of the 4D light field. The 3D volume is cut in the middle of v-axis to show a possible 2D slice. (b) A 2D slice $L(s,u)$ of the light field.

benefits. At the acquisition step, neither the camera at each *st* grid point has to be tilted nor the images have to be perspectively sheared when taken without tilting the camera. Second, each image can be taken with a fixed field of view as well as with other camera parameters fixed. This setup is useful especially to capture the scenes having a large depth range. For these reasons, this setup will be used throughout this thesis.

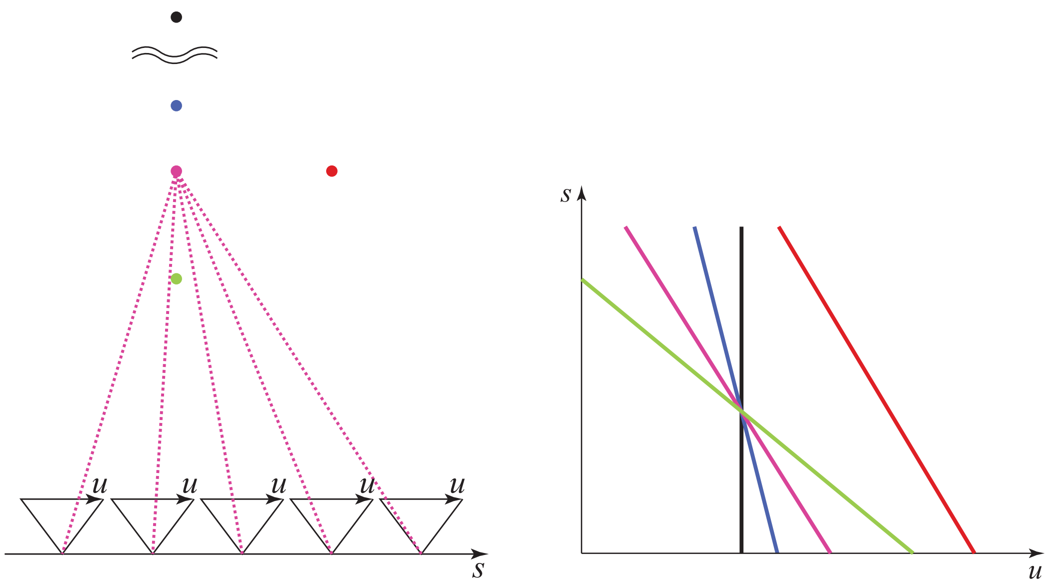
A similar parameterization is the angular parameterization, where a ray is defined by a 2D position in the camera plane and two angles between the ray and the camera plane. See Figure 3.2 for the comparison of two parameterizations in 2D light fields. The angular parameterization is different from the two-plane parameterization with the focal plane at infinity in that in the former, the direction is discretized in equally divided angles, whereas in the latter, it is usually discretized in rectilinear grids. We also use the angular parameterization to derive a focal stack in later sections.

Fixing each one of the directional and spatial coordinates, e.g. t and v, the remaining two coordinates (s,u) define a 2D light field, also called epipolar-plane image (EPI). This 2D light field contains the rays traveling across a 2D flat space captured by a single scanline of 2D *uv* images. Figure 3.3 visualizes 3D and 2D slices of an example light field. In a 2D light field, a scene point on a Lambertian surface appears as a straight line with the slope determined by its depth (Figure 3.4). In a 4D light field, the scene point forms a plane containing rays with a uniform color, instead of a line. This property will play an important role to the depth estimation of a scene point in the later parts of the thesis. For the sake of simplicity, we often use 2D light fields in the following discussions, which can be generalized to deal with 4D light fields in most cases.

3.2 Focal Stack

3.2.1 Digital Refocusing

In a conventional camera, the rays coming from a point in a range of distances are concentrated in a small region in the image plane, hence the objects in that distance range have their sharp images in the image plane. This process is called focusing. By changing focal length and placing



(a) Scene space: An example 2D scene with the acquisition setup shown together

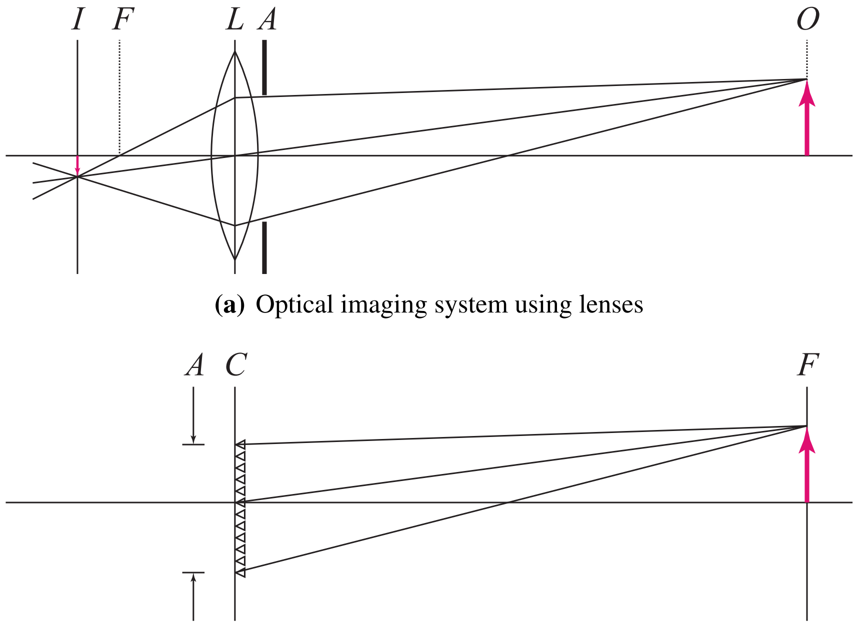
(b) Ray space: The 2D light field capturing the 2D scene

Figure 3.4: An example of a 2D light field, or epipolar plane image (EPI). (a) The parameterization of rays passing through scene points in a 2D flatland scene using two lines analogous to two-plane parameterization in 4D. The black dot is at infinity. (b) The appearance of scene points in the 2D light field. Scene points on Lambertian surfaces appear as straight lines in the 2D light field, whose slopes are proportional to the depths of the points.

the image plane at an appropriate position, we can select the rays which will be focused, and thus change the focusing. However, the rays arriving at the camera aperture with all different directions are already integrated (focused) at imaging time, and cannot be separated afterwards. In the light field, those rays are stored separately and can be differentiated after the acquisition. Therefore, we can selectively integrate the rays contained in the light field after the imaging time, and *refocus* the scene with a strategy of the ray selection determining the focusing. This process is called *digital refocusing*, and becomes the basis of the synthetic imaging system using the light field (Figure 3.5). A scene point is imaged (refocused) through the synthetic imaging system by integrating all the rays leaving the point and arriving within the set of sampling points on the camera plane. This set of sampling points defines a *synthetic aperture*.

For example, in a 2D light field, a scene point on a Lambertian surface appears as a straight line with a slope determined by its depth (Figure 3.4). If we integrate the entire light field along parallel lines of a particular slope, only the regions where there are surfaces in that distance will have clear images because the slopes of the straight lines corresponding to those regions match the slope of the integration lines. The scene points at the particular distance associated to the slope will have sharp shapes (*focused*), but the points off the distance will be blurred (*defocused*).

If the focal plane is placed at infinity, the integration over the aperture yields an image focused at infinity, since only the rays leaving a point at infinity form a straight line aligned along the camera plane. To focus the scene at a different distance, the focal plane must be moved to the desired distance and the light field must be reparameterized accordingly. Thus, the placing the focal plane determines the focusing in the synthetic imaging system (Figure 3.5(b)). There is no limitations



(a) Optical imaging system using lenses

(b) Synthetic imaging system aggregating multiple cameras

Figure 3.5: An optical imaging system using lenses vs. a synthetic imaging system aggregating multiple cameras. (a) An example of a typical imaging system using lens optics, with the image plane (I), focal plane (F), lens compound (L), aperture (A), an object (O). An object at O has its image at the image plane I. The rays leaving O and passing through the aperture A are integrated and focused at I by the lens L. (b) A synthetic imaging system, with the camera plane (C), synthetic aperture (A), and hypothetic focal plane (F). The rays are sampled by cameras located at sampling points in the camera plane C. The extent of the sampling points in the camera plane C constitutes a synthetic aperture A. An object can be digitally focused by placing hypothetic focal plane F at the position of the object and integrating the rays connecting appropriate points in C and F after the acquisition of the light field.

on the size and the shape of the aperture, as well as on the shape of the focal plane. The focal plane may have an arbitrary shape if the light field can be appropriately reparameterized[2]. The aperture can have an arbitrary shape, and can be extended to span many sampling positions in the camera plane. In the following discussion, however, we use a planar focal plane parallel to the camera plane. The reparameterization can then be achieved by shearing the light field parallel to the camera plane, as derived in the following.

3.2.2 Derivation of Focal Stack

We define a new representation of the scene captured in a light field using digital refocusing. By refocusing the light field with a range of distances between the camera plane and the focal plane, we obtain a set of differently refocused images. Stacking those images according to their distances yields a 3D volume, which we call a *focal stack*.

For the simplicity, we derive a 2D focal stack $FS(x,z)$ from a 2D light field $L(s,u)$. Without loss of generality, we define the coordinate system of a focal stack to be aligned to the coordinate system of a light field, such that x-axis and s-axis coincide and have the same metric. Then, a point in the focal stack $FS(x,z)$ is a point at the position x of the 1D image refocused at the distance z from the camera plane. Then $FS(x,z)$ is an integral of the rays passing through the point x in the new focal plane placed at z over a synthetic aperture defined in the camera plane:

$$FS(x,z) = \int_{\mathscr{A}} L\left(s, u(s,x,z)\right) \mathrm{d}s, \tag{3.1}$$

where $\mathscr{A}$ denotes the aperture, and $u(s,x,z)$ represents the reparameterization of the light field L according to the position of the new focal plane (See Figure 3.6).

The 2D light field $L(s,u)$ is centered at origin by normalizing each parameter of L to be within the interval $[-1,1]$, which is depicted in Figure 3.6(b). When using the full aperture, the aperture $\mathscr{A}$ becomes $s \in [-1,1]$. Then, by the trigonometry illustrated in Figure 3.6, we obtain

$$u(s,x,z) = \frac{x-s}{z} \cdot \frac{1}{\tan(\vartheta/2)}, \tag{3.2}$$

where ϑ is the field of view of the camera used to capture the light field.

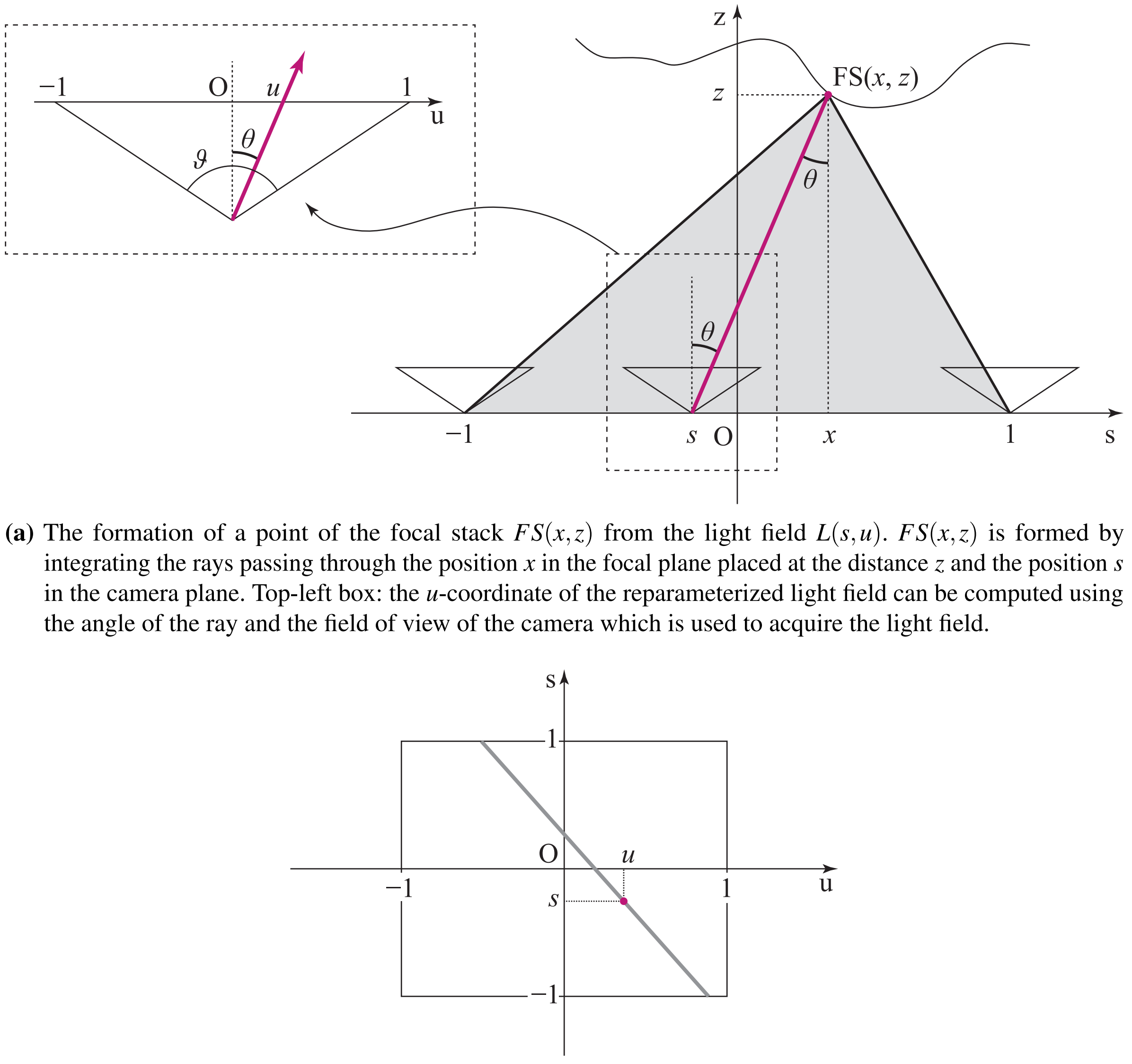
Substituting (3.2) into (3.1), we have

$$FS(x,z) = \int_{-1}^{1} L\left(s, \frac{1}{z\tan(\vartheta/2)}(x-s)\right) \mathrm{d}s. \tag{3.3}$$

The focal stack is defined in the region where a point has at least single ray passing through it (Figure 3.7(a)), that is, where $z > 0$ and $-1 - z\tan(\vartheta/2) \leq x \leq 1 + z\tan(\vartheta/2)$.

Let the field of view be the same for all images, and thus $c = \tan(\vartheta/2)$ be a constant. Here, the constant c, or the field of view ϑ, relates the metric of the u-axis to the metric of the s-axis.

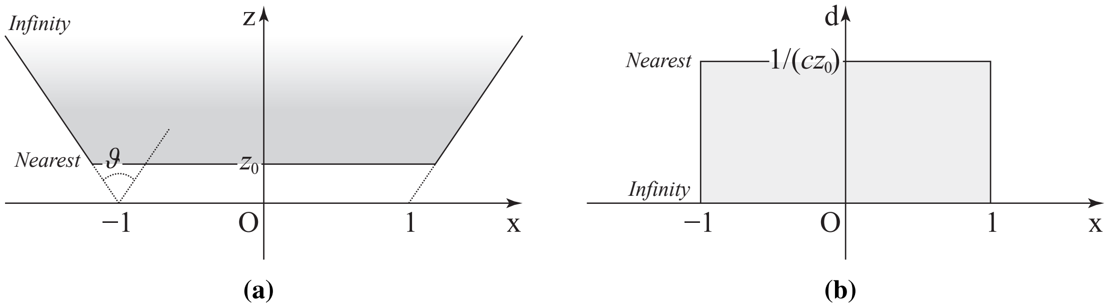
[2] There are several proposals for the light field reparameterization with a planar focal plane [IMG00, VGT$^+$05]. To our best knowledge, however, there has been no proposal of the reparameterization with an arbitrary shaped focal plane.



(a) The formation of a point of the focal stack $FS(x,z)$ from the light field $L(s,u)$. $FS(x,z)$ is formed by integrating the rays passing through the position x in the focal plane placed at the distance z and the position s in the camera plane. Top-left box: the u-coordinate of the reparameterized light field can be computed using the angle of the ray and the field of view of the camera which is used to acquire the light field.

(b) The 2D light field $L(s,u)$. The rays which are integrated to form the point in the focal stack $FS(x,z)$ in (a) are indicated in the light field.

Figure 3.6: Formation of the focal stack. (a) Rays in the shaded region are integrated to form a point at (x,z) in the focal stack. (b) The associated rays in the 2D light field appear as a straight line.



(a) **(b)**

Figure 3.7: Space defined by a 2D slice of a focal stack. (a) An orthographic focal stack for the depth range $z_0 \leq z < \infty$, where a point in the focal stack matches directly to an actual scene point up to scale. (b) A perspective focal stack covering the same depth range as (a). With cropping the areas not seen at the view point $s = 0$, the 2D slice fits to a rectangular region and is able to include points at infinity, but has perspective distortion. Note that disparity is used, instead of depth.

Then, (3.3) simplifies to

$$FS(x,z) = \int_{-1}^{1} L\left(s, \frac{1}{cz}(x-s)\right) \mathrm{d}s. \tag{3.4}$$

Inspecting (3.4) carefully, we find that it represents an integral of an appropriately sheared and then scaled light field, which means

$$FS(x,z) = \int_{-1}^{1} \left[\mathrm{Scale}_u\left(\frac{1}{cz}\right) \circ \mathrm{Shear}_u(-1) \circ L\right](s,x)\,\mathrm{d}s, \tag{3.5}$$

where $\mathrm{Scale}_u(\alpha)$ and $\mathrm{Shear}_u(\beta)$ denote the applications of a scale and shear transformation along the u-axis by the amount of α and β, respectively, and the operator $\circ$ denotes the function composition. The scaling factor $1/(cz)$ compensates the perspective projection, making the represented space view-independent. Thus, (3.4) defines a view-independent undistorted space, which reflects the scene space up to scale[3].

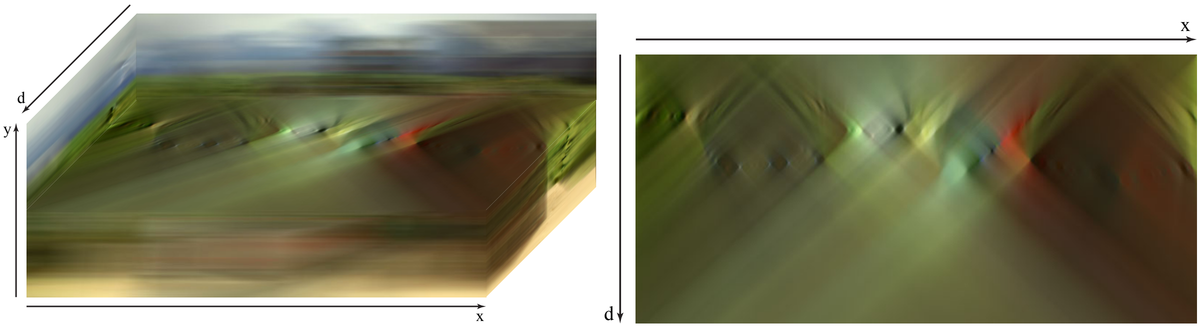
If we re-introduce the perspective distortion, and thus do not scale the light field along the u-axis, but only shear it, then the focal stack becomes

$$FS(cz \cdot x, z) = \int_{-1}^{1} L\left(s, x - \frac{1}{cz}s\right) \mathrm{d}s, \tag{3.6}$$

where $z > 0$ and $-1/(cz) - 1 \leq x \leq 1/(cz) + 1$. Furthermore, if we use the *disparity* $d = 1/(cz)$, instead of the depth z, and denote this volume as $FS_{s=0}$ then we have

$$\begin{align} FS_{s=0}(x,d) &= FS(\frac{1}{d}x, \frac{1}{cd}) \tag{3.7}\\ &= \int_{-1}^{1} L(s, x - d \cdot s)\,\mathrm{d}s \tag{3.8}\\ &= \int_{-1}^{1} [\mathrm{Shear}_u(-d) \circ L](s,x)\,\mathrm{d}s, \tag{3.9}\end{align}$$

[3] The scale for each axis does not have to be the same.



(a) 3D focal stack

(b) 2D focal stack

Figure 3.8: An example focal stack formed from the light field depicted in Figure 3.3. (a) A 3D focal stack $FS(x,y,d)$. The 3D volume is cut in the middle of y-axis to show a possible 2D slice. (b) A 2D slice $FS(x,d)$ of the 3D focal stack. The axes are depicted in both cases.

where $c = \tan(\vartheta/2)$ is a constant, $0 \le d < \infty$, and $-d-1 \le x \le d+1$.

$FS_{s=0}$ is perspectively distorted, and thus defines a view-*dependent* 2D focal stack which is seen at $s = 0$ (Figure 3.7(b)). We may only take x within its range at the infinite depth, i.e., zero disparity, which cuts the regions that are not seen at the current viewing position $s = 0$. Then, the range of x becomes $-1 \le x \le 1$. This 2D focal stack contains the 1D scene differently focused at a range of depths.

In 4D light fields with the assumption of the same metric of t and v coordinates and the square sampling grid of the uv plane—that is, the shape of the sensor elements of the camera used in the capture is square—(3.4) can be extended to

$$FS(x,y,z) = \int_{-1}^{1}\int_{-1}^{1} L\left(s,t,\frac{1}{cz}(x-s),\frac{1}{cz}(y-t)\right)\,\mathrm{d}s\,\mathrm{d}t, \tag{3.10}$$

and (3.8) can be extended to

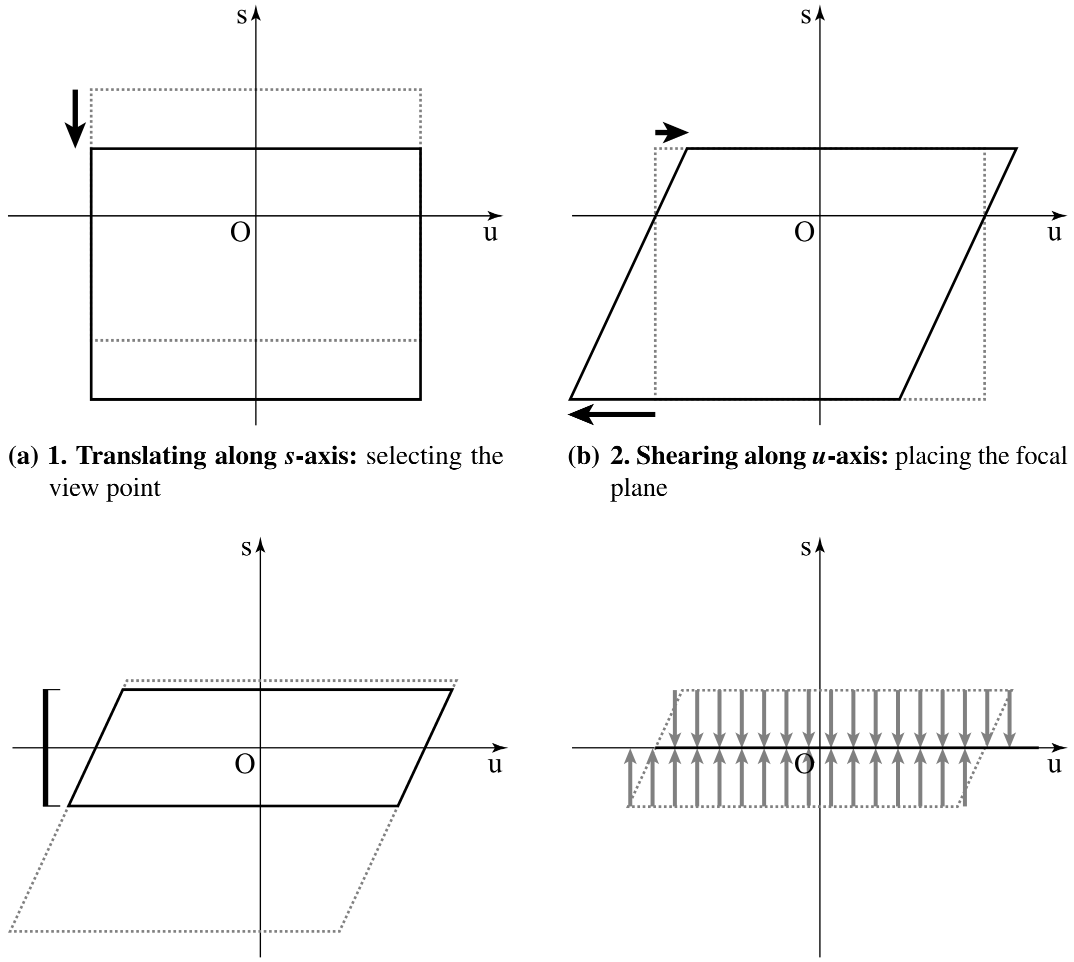
$$FS_{s=0,t=0}(x,y,d) = \int_{-1}^{1}\int_{-1}^{1} L\left(s,t,x-d\cdot s,y-d\cdot t\right)\,\mathrm{d}s\,\mathrm{d}t. \tag{3.11}$$

Equation 3.11 will be used throughout the thesis to select the rays from the light field to form a point in a focal stack. From this equation, a focal stack can be computed efficiently from a light field by a series of shear transformation and subsequent axis-aligned integration. Figure 3.8 shows an example focal stack.

3.2.3 Synthetic Imaging

The process derived in the previous section can be summarized as follows, and understood more intuitively in an analogy to the conventional photography.

1. Translating the light field in a direction parallel to the directional axes corresponds to selecting the view point. Then the origin of the st camera plane after the translation becomes the current view point, and the 2D uv-slice fixing $s = 0$ and $t = 0$ becomes the view at the point.



(a) 1. Translating along s-axis: selecting the view point

(b) 2. Shearing along u-axis: placing the focal plane

(c) 3. Selecting the interval of s: adjusting the size of aperture

(d) 4. Integrating along the s-axis: imaging the scene

Figure 3.9: Synthetic imaging in 2D light fields. The synthetic imaging can be achieved using digital refocusing given the size of a synthetic aperture, a distance in focus, and a view point. The digital refocusing can be illustrated by the four steps, which are depicted at (a)–(b).

2. Shearing the light field in a direction parallel to the spatial axes corresponds to placing the focal plane at a particular distance.
3. Selecting the interval of integral in the directional coordinates corresponds to adjusting the size of aperture. The 4D volume containing uv-slices in the selected interval of s and t is the rays gathered through the aperture.
4. Integrating the light field along the directional axes at each spatial point (u,v) in the interval defining the aperture corresponds to releasing the shutter and collecting photons to image the scene using the settings defined above. The integration for all spatial points produces a final 2D image that is focused at a given distance with a given aperture seen at a given view point.

Figure 3.9 illustrates these four steps in the case of a 2D light field.

In 2D light fields, the shear transformation reflects the tracking the slope of the straight line, and matching the integration lines to the slope. When a line becomes aligned to the directional axes after the transformation, this line integrates to a single point and the point is sharply focused. We therefore focus on the scene points at the distance corresponding to the slope. In sum, focusing a light field is a line integral over the light field, where the integration lines are parallel and their slope determines the distance to the focal plane.

3.3 Depth from Focal Stack

The focal stack can be used to determine the depth of a scene captured in a light field. The *view-independent* focal stack built from Equation 3.10 is a union of the view frustums at all viewing positions on the camera plane. The z dimension is bounded by the closest and the farthest distance in the range of z. The *view-dependent* focal stack built from Equation 3.11 is a perspectively distorted version of its view-independent counterpart, to be fitted to a cuboid. It is a bounded 3D space generated by deforming the view frustum at the viewing position into a rectangular box. In a view-dependent focal stack, the center of projection is at infinity, whereas it is on the camera plane in a view-independent focal stack. Thus, a 1D axis-aligned space of a view-dependent focal stack for each (x,y) is identical to a *hypothetical ray* shot from the center of projection in the corresponding direction in the undistorted view frustum. Finding the depth of the scene from a particular perspective can be achieved by determining the first intersection of each ray shot from the center of projection in a discretized direction in the view frustum. In the view-dependent focal stack, it is equivalent to finding the d-coordinate satisfying some *criteria* for each (x,y).

For example, if a scene point perspectively projected to a 2D point (x,y) is at a particular depth $z = 1/(cd)$, this point will be in focus and thus have the sharpest shape at the point (x,y,d) in the focal stack. Thus, conventional range finding techniques, such as depth from focus (DFF), may be applied to provide such a criterion to help extract a depth map from a focal stack.

A more flexible criterion is possible by examining the set of rays to be integrated to form each point in the focal stack. If a point resides on a Lambertian surface, the rays in the set will be of very similar color. If a point does not lie on a surface, the rays will have various colors. We can use characteristics of this set to infer higher level information, such as the probability of existence of surfaces. This analysis will form a basic framework of the methods explained in the following chapter.

3.3.1 Effects of Wide Aperture

Since a focal stack contains the scene focused at a range of different distances, it is inherently fit for the depth from focus (DFF) method [Gro87, DW88], where the most sharply imaged points are selected as scene depths among a set of differently focused images. The defocus based ranging techniques such as DFF can be interpreted as a small baseline stereo system [SK00], where the baseline corresponds to the diameter of the aperture. In a single lens stereo system [AW92], the parallax, also called the disparity, observed in the opposite sides of the aperture is directly used to measure the depth. In these methods, even when a very wide lens is used, the parallax induced by the lens aperture is an order of a few centimeters. Thus, the sensitivity—the depth resolution in this case—of those algorithms becomes low.

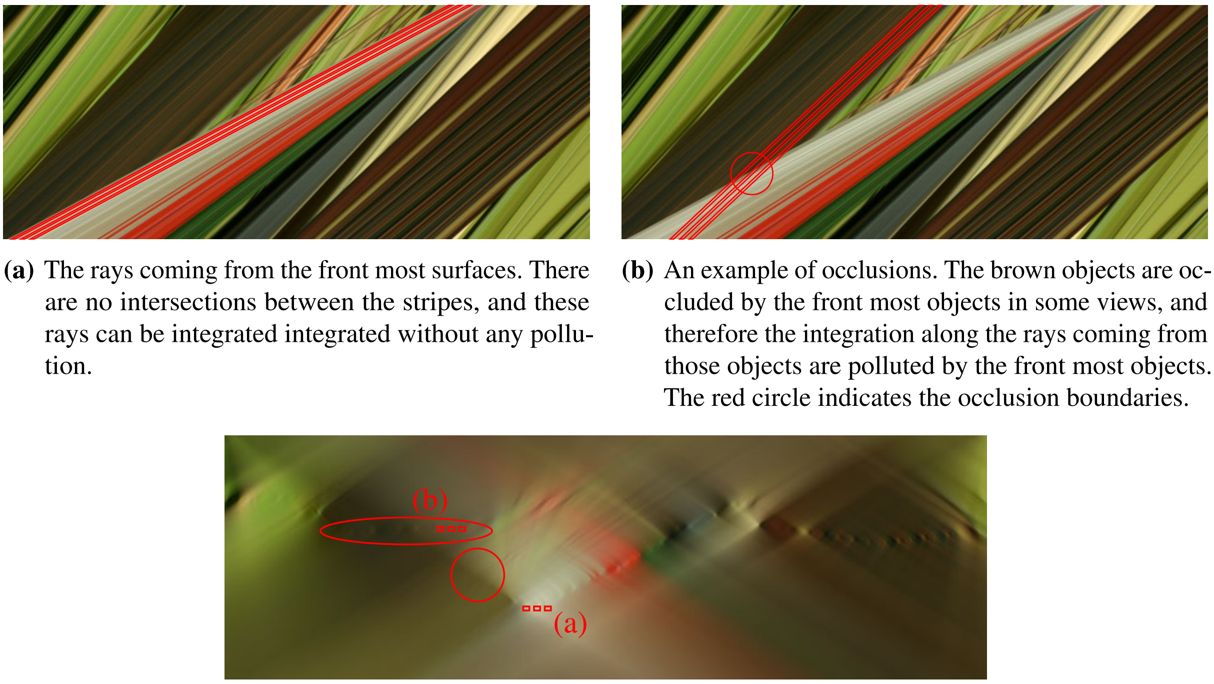
In light fields, the images of a scene point captured at different sampling points produce the parallax. The parallax can be used to infer the depth of the point as in the aforementioned methods, since the parallax depends on the depth under the perspective projection. Shearing light fields and trying to find the amount of shear transformation resulting the most uniform radiance distribution with the lowest variance can be understood as matching the correspondences of a scene point from different views and finding the parallax between the images of the scene point. This is the way the problem is approached in the computer vision literature [KS04, CKS+05].

In the synthetic imaging system with a wide synthetic aperture, as introduced in Section 3.2.3, the distance between two points in the aperture can be large, compared to the case of a typical camera lens. Hence the parallax observed in different points within the aperture also becomes large. This is advantageous to reconstruct the depth, and enables a higher depth resolution than a typical single lens stereo. In fact, the synthetic aperture can be interpreted as a baseline in a stereo system based on triangulations, and the synthetic imaging system inherits the merit of the wide baseline stereo system. In the synthetic imaging system, the aperture size can be adjusted within the size of the camera plane used to acquire the light field. With a large camera array setup for the acquisition, the synthetic aperture can be very wide so as to be comparable to the scene size.

However, the circle of confusion, although its shape may not be a circle any more, becomes large and rays coming from a wide directional range are mixed together, so that the blur properties become more complex. This also makes occlusion boundaries complicated. In addition, the aperture can be of any shape, which may be a line in the case of a 1D sampling grid or a rectangular region in the case of a 2D sampling grid. This irregular shape of the aperture may tend to introduce either a strange vignetting or unnecessary high frequency to the refocused images, which makes the direct application of existing algorithms difficult.

3.3.2 Effects of Occlusions

More importantly, a wider aperture introduces a stronger influence of occlusions, and causes a visibility problem, where a scene point which is visible at part of the aperture is not visible at the other part of the aperture. This limited visibility causes unmatched correspondences, and makes the parallax computation ambiguous or impossible for those scene points having unmatched correspondences. In a small aperture system (paraxial system) such as DFF and the single lens stereo discussed in Section 3.3.1, scene points which is visible at one end of the aperture are usually visible at the other end of the aperture. Thus, in most cases those methods can avoid the visibility problem. In a triangulation based stereo method using a wide baseline, the depth is reconstructed by identifying the correspondences of a scene point in all views and computing the parallaxes. With a wider baseline, the parallax increases for the scene point of the same depth, hence the depth resolution can also increase. However, each view can now have a more different perspective from other views than with a small baseline. It is more likely to be the case that scene points which are visible in some views are not visible in other views due to the occlusions, and thus do not have correspondences in all views. In such cases as unmatched correspondences,



(a) The rays coming from the front most surfaces. There are no intersections between the stripes, and these rays can be integrated integrated without any pollution.

(b) An example of occlusions. The brown objects are occluded by the front most objects in some views, and therefore the integration along the rays coming from those objects are polluted by the front most objects. The red circle indicates the occlusion boundaries.

(c) The focal stack. The points corresponding to the integration lines depicted in (a) and (b) are indicated. Note that the pollution caused by the front most objects inside the red circle and the color difference of the brown objects due to the occlusions inside the red ellipse.

Figure 3.10: The effect of occlusions in a 2D light field. The occlusion influences on the formation of the focal stack, and hampers the depth estimation based on the focal stack.

the scene point may not be reconstructed correctly. The wide synthetic aperture inherits these disadvantages as well.

The difficulty of the correspondence matching due to the occlusions and the visibility problem is equivalent to the difficulty of the finding of uniform rays along the integration lines during the ray integration over the light field. A naive approach will produce inaccurate depth estimations and polluted focal stacks. Light fields contain special structures depending on the depth and the visibility. As mentioned before, a scene point in a Lambertian surface appears as a straight line in the scene's associated 2D light field, and a plane in the 4D light field (see Figure 3.10(a) for the 2D case). The slope of the line, or the orientation of the plane, is determined by the depth of the point. In the 2D case, these straight lines can intersect each other when an occlusion occurs, and the line of the closer point always hides the line of the farther point (Figure 3.10(a)). As a consequence, the farther point in the focal stack has a different color from its original color, since the integration includes the rays originated from the closer point (Figure 3.10(c)). Thus, with the presence of a point occluded by others in some views, the digital refocusing derived in Section 3.2 produces a polluted focal stack. Since this pollution hampers accurate depth estimation, we are required to handle the pollution.

In the following chapter, we attack the problem in two ways—the first is to select only unoccluded rays based on the characteristics of the ray set, and the other is to split the light field into parts to make each of them occlusion-free.

4

Methods

In this chapter, we present detailed methods to estimate the depth of scene points and to reconstruct the 3D volume based on the notions and representations we developed in the previous chapter.

In the first two sections, we address how to build a 2D depth map from a light field. We especially focus on developing tools to measure the existence of surfaces which are robust to occlusions. In the third section, the methods are extended to deal with multiple depth layers which may not be represented from a single viewing position, resulting in a 3D volume occupancy. In the fourth section, we approach the problem from another side. We try to remove the artifacts caused by occlusions by appropriately partitioning the light field so that each part is free from occlusions. We briefly discuss the refinement of acquired volume occupancy in the final section.

4.1 Computing Depth for a Ray

We propose to estimate the depth of a scene by analyzing the rays that are focused to form a point in the focal stack. More specifically, we define a function that takes a set of rays coming from a given location, and returns the likeliness of a surface at that point. We call this function a *focus measure* in analogy to conventional range finding techniques. We will present several different focus measures in the subsequent sections.

Using such a focus measure function, we can then estimate the depth of a scene as follows. We first measure how likely there exists a surface for each point (x,y,z) in a focal stack, and then we select the points from the focal stack where surfaces are most likely to exist. For example, by selecting a depth z with the strongest focus measure for each (x,y) in the focal stack, we obtain a 2D depth map. This depth map is the one seen at the viewing position used to build the focal

Algorithm 4.1 Framework of the depth reconstruction

```
Input: 4D light field $L(s,t,u,v)$, focus measure $F(\cdot)$, and ray integration function $I(\cdot)$
Output: 3D perspective focal stack $FS(x,y,d)$, 2D disparity map $D(x,y)$, and 3D volume
    reconstruction $O(x,y,d)$
$O(\cdot,\cdot,\cdot) \leftarrow$ ‘empty’
for each $(x,y,d)$ do
    $\{R_{xyd}(s,t)\} \leftarrow L(s,t,x-ds,y-dt)$
    $V(x,y,d) \leftarrow F(\{R_{xyd}(s,t)\})$
    $FS(x,y,d) \leftarrow I(\{R_{xyd}(s,t)\})$
end for
for each $(x,y)$ do
    $D(x,y) \leftarrow \min_d\{V(x,y,d)\}$
    $O(x,y,D(x,y)) \leftarrow FS(x,y,d)$
end for
```

stack, and thus *view-dependent*. In the following discussion, we use disparity d instead of depth z. In this case, the reconstruction yields a 2D disparity map.

Algorithm 4.1 summarizes the framework of extracting a 2D depth map $D_{s=0,t=0}(x,y)$ at the viewing position $(s,t) = (0,0)$[1]. Let $L(s,t,u,v)$ be a 4D light field. The set of rays to form a point (x,y,d) in the view-dependent focal stack $FS_{s=0,t=0}$ is collected from L by taking

$$R_{xyd}(s,t) = L(s,t,x-ds,y-dt) \tag{4.1}$$

from (3.11), where we denote the ray set as $\{R_{xyd}(s,t)\}$. The ray set $\{R_{xyd}(s,t)\}$ is defined at each point (x,y,d) of a focal stack and parameterized by the directional coordinates s and t. Let $F(\cdot)$ be a function to compute the focus measure given a set of rays, and $I(\cdot)$ be a ray integration function associated to $F(\cdot)$ to compute the point in the focal stack $FS_{s=0,t=0}$. $V(x,y,d)$ is a focus measure volume[2] defined in the same space as the focal stack. The subscripts to indicate the viewing position of the focal stack and the depth map will be omitted hereinafter when there is no ambiguity. We often omit the subscript or the parameters of the ray set for the simplicity.

For each point (x,y,d) in the focal stack, we first collect the set of rays $\{R_{xyd}\}$. Then, the focus measure is computed using $F(\cdot)$ and the focal stack is computed using $I(\cdot)$. After building the focus measure volume V, we select the disparity d with the strongest focus measure at each (x,y), which generates a 2D depth map D. By taking the point at the reconstructed depth from the focal stack, we can reconstruct the radiometric volume representation O of the scene. However, this 3D volume reconstruction does not contain more information than the 2D disparity map in terms of the geometry, since we cannot reconstruct occluded surfaces. The reconstruction of the occluded surfaces are discussed in Section 4.3.

In the subsequent sections, we define several focus measures. The same framework will be used to build the 2D depth map using those focus measures.

[1] The viewing position is always assumed to be at the origin. A view-dependent focal stack can be built from any perspective by translating the light field in the direction parallel to directional axes before the integration (Section 3.2.3).

[2] When using variance as F, each xt-slice of V is known as disparity space image (DSI) [BI99].

4.1.1 Using Statistics of Rays

Our first focus measure is based on the statistics of the rays in the set $\{R\}$. If a set of rays comes from the same scene point, the rays are likely to have a similar color, and thus have a low variance. On the other hand, if the focal plane is not placed on the surface and the rays come from different points, they are likely to have different colors, hence higher variance. Therefore, the variance over the ray set contributing to each point in the focal stack can be used to determine the existence of a surface; the lower the variance, the more likely a surface exists. This results in a focus measure F defined as

$$F(\{R_{xyz}(s,t)\}) = \frac{1}{\mathrm{Var}(\{R_{xyz}(s,t)\})}, \tag{4.2}$$

and an integration function I defined as

$$I(\{R_{xyz}(s,t)\}) = \mathrm{Mean}(\{R_{xyz}(s,t)\}). \tag{4.3}$$

As a variant, the (normalized) sum of squared differences (SSD) between the ray seen at the viewing position of the focal stack $(s,t) = (0,0)$ and other rays can be used:

$$F(\{R_{xyz}(s,t)\}) = \frac{N_s N_t}{\sum_s \sum_t \left\{R_{xyz}(s,t) - R_{xyz}(0,0)\right\}^2}, \tag{4.4}$$

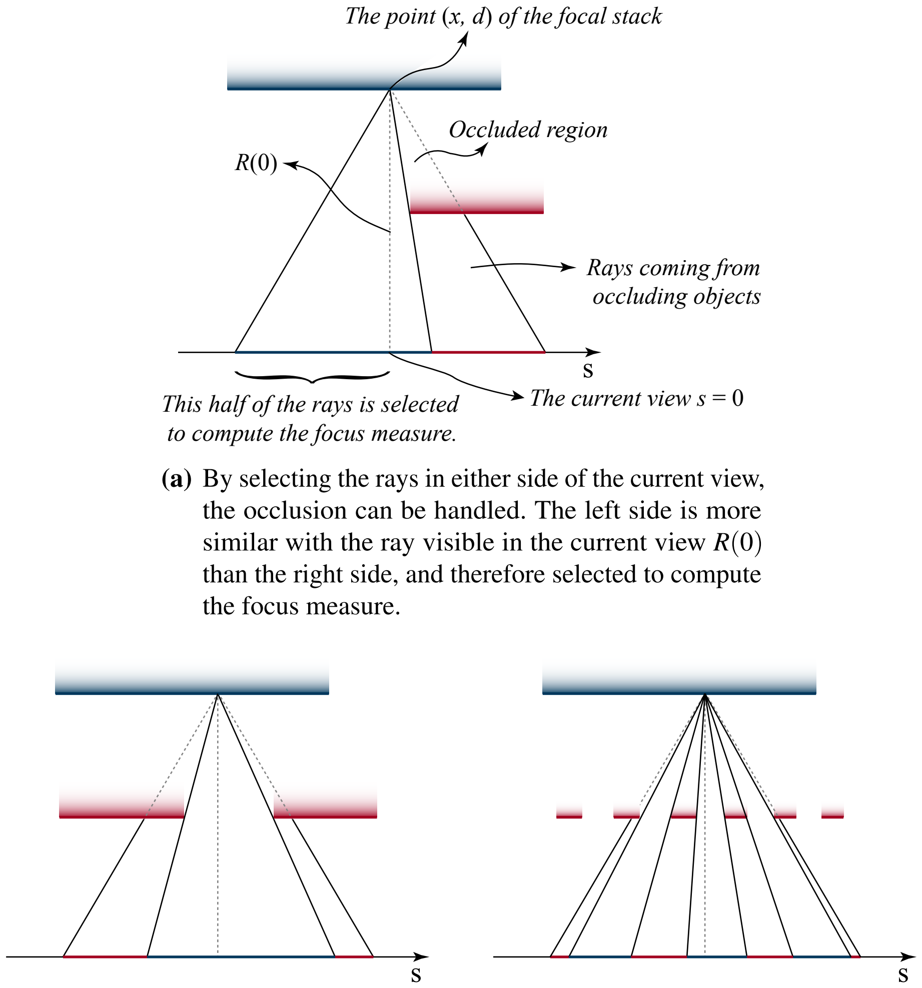
where N_s and N_t are the number of samples in each dimension. When an occluding object is a largely uniform surface, the variance can be undesirably low, since it computes the deviation from the mean. However, the SSD measures the deviation of rays from $R(0,0)$, the ray seen at the view position of the focal stack, and thus the SSD gives more reliable measurement. As explained later, the SSD works better with the ray selection and the ray clustering methods, since those methods select rays based on the similarity to $R(0,0)$. This step corresponds to estimating the orientation of the *plane of a uniform color* in the light field which contains the ray $R(0,0)$. The orientation estimation is discussed in Section 4.3.2.

4.1.2 Selecting Rays

The focus measure defined by Equation 4.4 may give a good guidance to estimate the depth of scene points *unless* the points are near to occlusion boundaries. That is because the rays coming from occluding objects are integrated together with the rays coming from the scene point we are interested in. Unless a point is open to all viewing positions in the camera plane, the focus measure for the point is *polluted*, and therefore inaccurate. In such a case, it is desirable to take only the rays coming from the object we want to focus. One choice is to select the rays sampled in either side of the current viewing position, which is expected to be less polluted than the other side (Figure 4.1(a)). The resulting focus measure in a 2D light field is

$$F(\{R_{xyz}(t)\}) = \min\left\{\sum_{t<0}\left\{R_{xyz}(t) - R_{xyz}(0)\right\}^2, \sum_{t>0}\left\{R_{xyz}(t) - R_{xyz}(0)\right\}^2\right\}. \tag{4.5}$$

In 4D light fields, only one quadrant of the $\{R(s,t)\}$ is selected. The influence of occluding objects can be avoided as long as the point is seen by at least half the viewing positions. However,


The point (x, d) of the focal stack
Occluded region
R(0)
Rays coming from occluding objects
s
This half of the rays is selected to compute the focus measure.
The current view s = 0
s
s

(a) By selecting the rays in either side of the current view, the occlusion can be handled. The left side is more similar with the ray visible in the current view $R(0)$ than the right side, and therefore selected to compute the focus measure.

(b) Although the scene point (x,d) has more than half the visibility to the camera plane, both sides of the rays do not give precise focus measures when using Equation 4.5. However, Algorithm 4.2 can handle this case correctly.

(c) Selecting the rays which are more similar with the ray $R(0)$ can handle more complex occlusions.

Figure 4.1: Selecting rays. With the presence of occluding objects the focus measure becomes inaccurate, since the rays coming from the occluding objects are mixed into the computation. By selecting fraction of the rays, the influence of the occluding objects can be considerably reduced.

Algorithm 4.2 Focus measure based on selecting rays

```
Input: set of rays $\{R(s,t)\}$, ray selection rate $r$
Output: value of the focus measure $F$, value of the ray integration function $I$
$i \leftarrow 0$
for each $(s,t)$ do
    $SD(i) = \{R(s,t) - R(0,0)\}^2$
    $RI(i) = R(s,t)$
    $i \leftarrow i + 1$
end for
sort $SD$ in ascending order
rearrange $RI$ in the same order as $SD$
$F \leftarrow \sum_{i=0}^{i<rN_sN_t} SD(i)$        // $N_s$ and $N_t$ are the number of samples in each dimension
$I \leftarrow \frac{1}{rN_sN_t} \sum_{i=0}^{i<rN_sN_t} RI(i)$
```

if there are more than one occluding object along the viewing positions, this measurement also suffers from the pollution (Figure 4.1(b)).

Instead of taking all rays on either side, we can take half the rays which are more similar to the ray $R(0,0)$ than the other half—no matter whichever side the ray is—in the hope that the rays coming from the same point will be of the similar color. By doing so, we can deal with multiple small occluding objects, or spatially periodic occluding objects like fences. Figure 4.1(c) illustrates an example case. However, this selection scheme also needs more than 50% of the visibility of the scene point. The focus measure $F(\cdot)$ is defined as follows. Algorithm 4.2 describes the procedure of computing $F(\cdot)$. For each ray in the set $\{R(s,t)\}$, the squared difference between the ray and $R(0,0)$ is computed. Half[3] the rays with smaller squared differences are selected. Then, the focus measure $F(\cdot)$ is the sum of the squared differences of the selected rays. The ray integration function $I(\cdot)$ is defined as the mean of the selected rays. In practice, the SSD is normalized when the number of rays change.

We can deal with partial occlusions, provided having the visibility of the scene point to more than half the viewpoints. This gives a focus measure more robust to occlusions than taking all the rays.

4.1.3 Clustering Rays

We further extend ray selection to ray clustering, so that we can deal with more complicated cases where only a limited degree of the visibility is available. Furthermore, we can incorporate more flexible criteria to select similar rays.

In a typical acquisition setup using RGB color images, rays are distributed in the RGB color space[4]. If there is a portion of rays coming from the same point, those rays will distribute closely and form a cluster. The remaining rays will probably be scattered. If we can find such a cluster

[3] Or a fixed portion r of the rays. Then, the focus measure is robust up to having a portion r of the visibility.

[4] Clustering the rays in the L*u*v* or L*a*b* color space may give a better result, since in the RGB space, the color difference and the Euclidean distance in the color space show a relatively large gap.

Algorithm 4.3 Focus measure based on clustering rays

```
Input: set of rays $\{R(s,t)\}$, bandwidth parameter $h$, small number $\varepsilon$
Output: value of the focus measure $F$, value of the ray integration function $I$
$R_c \leftarrow R(0,0)$          // initial guess of the cluster center
repeat
    $m \leftarrow \frac{\sum_s \sum_t R(s,t) \exp\left(-\frac{1}{2}\left\|\frac{R_c - R(s,t)}{h}\right\|^2\right)}{\sum_s \sum_t \exp\left(-\frac{1}{2}\left\|\frac{R_c - R(s,t)}{h}\right\|^2\right)} - R_c$          // compute the mean shift vector
    $\hat{p} \leftarrow \sum_s \sum_t \exp\left(-\frac{1}{2}\left\|\frac{R_c - R(s,t)}{h}\right\|^2\right)$          // compute the density estimate
    $R_c \leftarrow R_c + m$          // translate the cluster center
until $\|m\| < \varepsilon$
$F \leftarrow \hat{p}$          // density estimate of the cluster
$I \leftarrow R_c$          // center of the cluster
```

and the cluster is dense enough for the member rays to be closely correlated, the rays in the cluster can be regarded as originated from the same point lying on a surface.

A clustering algorithm well suited to our purpose is the mean shift clustering algorithm [CM02, Che95]. The mean shift algorithm is a mode finding algorithm based on the non-parametric density estimation in a feature space. The mode means the peak, or the most densely populated point, of a distribution. Given data points, the algorithm seeks the mode of the probability density using its gradient. The probability density of discrete data points is estimated using the kernel density estimation, also known as Parzen's window method [DHS01, Chapter 4]. The mean shift clustering algorithm finds all local modes (local maxima) as the cluster centers and identifies the cluster where each data point is contained.

The mean shift clustering algorithm fits to our requirements for the following reasons. It does not require the number of clusters to be known beforehand. It seeks all the peaks where data points are densely populated, and provides a measure of how densely the data points are gathered at each peak. That is, we can find the cluster centers, their members, and the density of the clusters with little prior knowledge about data points.

We use the density of the cluster containing the ray $R(0,0)$ observed at the current viewing position. The higher the density is, the more similar the rays are. The focus measure $F(\cdot)$ and the integration function of rays $I(\cdot)$ are computed as follows. The mean shift clustering algorithm is run over the set of rays $\{R(s,t)\}$. We used a Gaussian kernel $k(\mathbf{x}) = \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)$ with the bandwidth parameter $h = 0.01$ for the estimation of the probability density. The mean shift algorithm does not explicitly compute the density, but its gradient. However, the probability density estimated using a Gaussian kernel is proportional to the gradient of the density up to a constant, and thus can also be obtained without additional cost during the clustering. After all clusters are found, the cluster containing the ray $R(0,0)$ is selected. Then, we use the probability density of the selected cluster as the value of $F(\cdot)$, and the cluster center as the value of $I(\cdot)$ at the point. Algorithm 4.3 sketches this procedure.[5] In practice, only the cluster containing $R(0,0)$ is sought, and the cluster membership is not explicitly computed.

[5] In Algorithm 4.3, although R_c, $R(s,t)$, and m are actually vectors in such a multi-dimensional feature space as color space, they are not notated as a bold face for the notational consistency.

In the following section, we use neighboring points in the focal stack as well to increase the robustness of the focus measures discussed here.

4.2 Depth Estimation Using Multiple Rays

Up to now, we only considered a single point (x,y) of the focal stack to infer the disparity d at that point. In this section, we extend our method to use the neighborhood of the point in two different ways. First, the spatial frequency of each xy-slice of the focal stack is used to infer the depth. This method is known as depth from focus (DFF). Then, we introduce a focus measure as an aggregator of another focus measure. This focus measure combines the focus measures computed at the points within a small window in the xy-slice of the focal stack.

4.2.1 Using Spatial Frequency

The method described in this subsection uses the spatial frequency information of a 2D xy slice of the focal stack at each disparity d. Assuming the scene has sufficient textures, the high frequency information of the 2D slice can be used to measure how sharply a scene point is focused. We employ a high-pass filter on the image to measure the high frequency information. A high-pass filter can be approximated by the difference of two Gaussian images with different spatial parameters[6]. Thus, the high frequency information is obtained by taking the image convolved with a difference of Gaussians (DoG) filter.

The focus measure described here is not a function of rays, but can be computed at each point (x,y,d) of a pre-built focal stack FS using Equation 4.3. Thus, the focus measure volume V is computed as

$$V(x,y,d) = \left| \left[FS(\cdot,\cdot,d) * \{G(\sigma_1) - G(\sigma_2)\} \right](x,y) \right|^2, \tag{4.6}$$

where $FS(\cdot,\cdot,d)$ is a 2D slice at disparity d of the pre-built focal stack, $G(\sigma)$ is a Gaussian kernel with spatial parameter σ, and $*$ denotes a 2D convolution.

In practice, each 2D slice of the focal stack is convolved with a DoG filter, and its magnitude image is taken. Then, the image is convolved with a Gaussian filter to propagate the high frequency information to neighbors in order to fill the possible frequency gaps. After the high frequency images for all focal stack slices are computed, the maximum value for each point (x,y) along the d-axis is selected to produce a 2D disparity map. This method is an application of the depth from focus algorithm to the focal stack. Since a focal stack is a collection of all differently focused images, the depth from focus method can be applied without additional cost.

Various approaches to measure the sharpest focus have been proposed, which include high-pass filtering, gradient magnitude, gray-level variance, etc., and they have their own advantages and drawbacks [Kro87]. Among them, high-pass filter was employed since it can be efficiently implemented and computed using convolution, while providing a good measurement power.

[6] The difference of two Gaussians is in fact a band-pass filter, where the frequency bounded by two Gaussian filters is preserved. In a discrete setup, appropriate selection of two spatial parameters for Gaussian filters will effectively act as a high-pass filter, since the high frequency in a discrete image is already bounded by the sampling frequency of the image.

4.2.2 Frequency vs. Variance

Before proceeding to the second approach, we briefly discuss the relation between the frequency based focus measure (Section 4.2.1) and the variance based focus measure (Section 4.1.1). Both focus measures are simple and can be computed very efficiently. Since they do not handle the occlusions explicitly, they fail to extract the scene points with a limited visibility. However, they exhibit different characteristics.

Frequency based methods and variance based methods are complement to each other. The spatial frequency of a surface texture is often too low in the *regions with smooth color change*, so that those regions are not well reconstructed by frequency based methods. However, the variance based methods can detect this subtle clue unless the color is exactly uniform, since the variance can be sufficiently low in those regions, so that they can be detected as surfaces.

On the other hand, the variance based methods often fail in the *regions with high frequency textures*, since a very small misalignment between the stripes in the light field and the resampling pattern, or imperfection of the acquired light field such as noise and geometric distortion, can cause the variance to be high. However, it is rare that the frequency based methods miss those regions, since regardless of the existence of misalignment or data imperfection, the regions are still likely to have enough high frequency after the integration.

As the frequency information is a good complement to the variance based method, they can be combined to produce a better focus measure, such as

$$V(x,y,d) = \frac{1}{V_v(x,y,d)} + \alpha V_f(x,y,d), \tag{4.7}$$

where V_v is the focus measure volume computed using a variance based method discussed in Section 4.1.1, V_f is the focus measure volume computed using the method discussed in Section 4.2.1, and α is a pre-determined constant to balance the influences of both measures.

In the next subsection, we discuss a different approach to incorporate the spatial information present at each 2D slice of the focal stack to make our focus measures more robust.

4.2.3 Using Focus Measures of Neighbors

The focus measures discussed in Section 4.1 can be further improved if the neighboring rays are considered as well. If we assume a smooth surface model of the scene, the focus measures for the neighbors of a point in the focal stack probably have similar values to the focus measure at the point. Considering spatial neighbors of the point helps remove the influence of noise or data imperfection, and makes the focus measure more robust. The focus measure then can be formulated as

$$F(\{R_{xyz}(s,t)\}) = f(\{R_{xyz}(s,t)\}) + \sum_{(x_i,y_i)\in\mathcal{N}_d(x,y)} f(\{R_{x_iy_iz}(s,t)\}), \tag{4.8}$$

where $\mathcal{N}_d(x,y)$ denotes the neighborhood of a point (x,y) in the 2D slice of the focal stack at the given disparity d, and f is the base focus measure function.

This aggregated focus measure, however, suffers at occlusion boundaries. If the point and its 2D neighbors lie across an occlusion boundary, and indeed they are *not* neighbors in the 3D space, the focus measure becomes inaccurate. To deal with this, we can adopt a similar approach to the ray selection discussed in Section 4.1.2, so that we can remove from the sum the points that are neighbors in the xy slice, but not in the d dimension. Similar to the ray selection, only n-closest focus measures of its neighbors are summed to result the aggregated focus measure F, where $0 < n < |\mathcal{N}_d(x,y)|$. The base focus measure f can be any of those presented in Section 4.1.

Throughout Section 4.1 and 4.2, we focused on eliminating the effect of occlusions on the focus measure, such as a mixed rays coming from the different sides of an occlusion boundary. Using one of those focus measures, we selected only one depth d having the maximum focus measure for each (x,y) in the focal stack. However, there may be more than one depth layer for an (x,y) in the focal stack. Multiple depth layers are observed as multiple peaks (local maxima) in the focus measure profile $V(x,y,\cdot)$ for a given (x,y). By selecting the maximum, only the closest and thus visible depth layer is reconstructed, since we compute the focus measure based on the comparison to the ray seen at the viewing position of the focal stack.

In the following sections, we address multiple depth layers for each (x,y) in the focal stack to reconstruct the 3D scene volume.

4.3 Volumetric Scene Reconstruction

One common limitation of the methods described so far is that only a single depth layer is reconstructed per ray. We present three approaches to handle multiple depth layers in this section. In the first approach, we warp all view-dependent depth reconstructions into the same 3D space, and then reconstruct the scene volume. In the second approach, we reconstruct the occluded scene point by iteratively removing the rays coming from the occluding point, once we identify its presence. In the last approach, we partition the light field to minimize occlusions, so that each partition is free from the occlusions. We discuss the first approach in this section, and the other two in the next section.

4.3.1 Projection of Multiple Views

With the methods discussed in Section 4.1 and 4.2, we estimate the depth using the view of the focal stack[7] as a reference, and thus only visible points to the current view are reconstructed. More specifically, we seek the depth d of a point (x,y) by searching the ray set containing most similar rays to the ray $R_{xyd}(0,0)$—the ray observed at the current viewing position. If a scene point is occluded in the current view, this point thus cannot be reconstructed. However, if we can see this point in another viewing position (s,t), the point could be reconstructed in the focal stack $FS_{s,t}$. Ideally, all the scene points which are seen from at least two viewing positions within the synthetic aperture can be reconstructed in one of those viewing positions' associated focal stacks.

[7] All xy slices of a focal stack $FS_{s,t}$ have the same view, which is the perspective projection of the seen at the viewing position (s,t) used to build the focal stack.

In this section, we reconstruct the depth from all available viewing positions within the aperture using the methods from the previous sections, which produces the scene volume with a single depth layer seen at each viewing position. We then warp each of them into a common 3D space which represents the scene. We use the perspective view volume seen at $(s,t) = (0,0)$, which is again *view-dependent*.

The rationale behind the use of a view-dependent volume is that the rays sampled in our light fields are not uniform over the 3D space. We place the second parameterization plane of the light field at infinity, but the resolution of the *uv* plane is finite and fixed. Thus, the deeper the depth, the sparser the samples. That also means that the *uv* slice is a perspective projection of the 3D space, but the view frustum is not bounded. Thus, if we use undistorted space, we will have very sparse samples in the regions far from the camera. On the other hand, one advantage of the use of a view-dependent volume is that the infinite depth can be represented in the finite 3D volume. If we have enough samples, however, there is no restriction to chose the space into which the view-dependent 3D volumes are warped. In fact, we can warp the scene volume into a view-independent volume which can be directly related to the actual 3D object space up to scale, provided the field of view ϑ of the camera used to acquire the light field.

We now derive the algorithm to construct the 3D volume from the 2D depth reconstructions. The 3D volume used here is a discretized voxel space in which the occupancy is marked. Thus, this can be thought of as a 3D histogram, where the warped surface points falling into each bin are counted.

We first apply Algorithm 4.1 for the depth reconstruction using one of the focus measures described in Section 4.1 and 4.2 to all views[8] in the light field. As a result of the algorithm, we have the 3D volume reconstruction O at each viewing position. We warp all occupied points (x,y,d) of each 3D volume O at the view (s,t) into the common view volume O_{vol}. For each occupied point (x,y,d) at the view (s,t), the warped coordinates (x',y',d') in O_{vol} can be computed as

$$x' = x + ds, \tag{4.9}$$

$$y' = y + dt, \text{ and} \tag{4.10}$$

$$d' = d, \tag{4.11}$$

where the view point of the the volume O_{vol} is at origin. After warping 3D volumes of all views, the occupancies in O_{vol} with fewer votes than a threshold are discarded to remove outliers. A 2D depth map can be again derived from the 3D occupancy volume O_{vol} by taking the depth d of the occupied voxel closest to the camera plane for each (x,y). Algorithm 4.4 illustrates the framework for generating a 3D occupancy volume using multiple views.

4.3.2 Direct Analysis of Light Field

Estimating the depth for an (x,y) in the focal stack by computing the focus measure is equivalent to examining the ray set corresponding to each disparity d. The ray set for a point (x,y) and a

[8] Since those algorithms run assuming the viewing position is at origin, the light field has to be appropriately translated along the directional axes before the application of the algorithms. For example, if the depth should be reconstructed at the viewing position (s,t), then the light field must be translated in the direction parallel to the s and t axes, so that the point (s,t) is located to $(0,0)$.

Algorithm 4.4 Framework of the 3D volume reconstruction

```
Input: 4D light field $L(s,t,u,v)$, focus measure $F(\cdot)$, ray integration function $I(\cdot)$, threshold $T$
Output: 3D volume reconstruction $O_{vol}(x,y,d)$
$H_{vol}(\cdot,\cdot,\cdot) \leftarrow 0$            // 3D histogram to count occupancies
$O_{vol}(\cdot,\cdot,\cdot) \leftarrow$ 'empty'      // 3D volume reconstruction with texture information
for each $(s,t)$ do
    // loop over Algorithm 4.1 for each view
    for each $(x,y,d)$ do
        $\{R_{xyd}(s',t')\} \leftarrow L(s'-s,t'-t,x-d(s'-s),y-d(t'-t))$
        $V(x,y,d) \leftarrow F(\{R_{xyd}(s',t')\})$
        $FS(x,y,d) \leftarrow I(\{R_{xyd}(s',t')\})$
    end for
    // warp the view to the common space
    for each $(x,y)$ do
        $d' \leftarrow \min_d\{V(x,y,d)\}$
        $H_{vol}(x+d's,y+d't,d') \leftarrow H_{vol}(x+d's,y+d't,d')+1$
        $O_{vol}(x+d's,y+d't,d') \leftarrow O_{vol}(x+d's,y+d't,d')+FS(x,y,d)$
    end for
end for
// post-process the occupancy volume
for each $(x,y,d)$ do
    if $H_{vol}(x,y,d) < T$ then
        $O_{vol}(x,y,d) \leftarrow$ 'empty'                              // discard the occupancy
    else
        $O_{vol}(x,y,d) \leftarrow O_{vol}(x,y,d)/H_{vol}(x,y,d)$       // will contain the texture of the point
    end if
end for
```

disparity d is a plane containing the point (x,y), slanted by $1/d$ with respect to both u and t axes. That means, determining the depth is changing the plane's orientation with keeping it to pass through (x,y) and finding the orientation which makes the plane span the most uniform area in the light field.

In short, the depth estimation of a ray can be interpreted as an orientation estimation of the *plane of a coherent color* containing the ray in the light field. This suggests that we can achieve the depth estimation without explicit mapping between ray space and scene space. Estimating the local orientation of a plane at each ray of the light field corresponds to estimating depth at each point in each view. The discretization of depth is equivalent to the discretization of orientation. Using this relation, the depth estimation can be run only in ray space. The problem of depth estimation can be cast to the problem to find dense planar structures in a 4D volume.

All the focus measures discussed in Section 4.1 and 4.2 are equivalent to reshaping the aperture where the rays are collected, which in turn is equivalent to adjusting the 4D window over the light field where the orientation is computed. This window over the light field is the same as the synthetic aperture. Then, the orientation estimation can be localized to cope with the occlusions as well as the imperfection of captured light fields by adjusting the shape and size of the window.

By reducing the window size, we simulate a smaller aperture with which the effect of occlusions can be alleviated, and the imperfection of the acquired light field can be tolerated.

4.4 Handling Occlusions

The effect of occlusions is more influential when a wider aperture is used, and the integration of rays over the occlusion boundaries are more polluted. In this section, we deal with this effect by splitting the light field so that each part of the light field is free of occlusions. First, we describe a method to peel off each depth layer from front to back so that closer layers do not influence on farther layers. Then, we present a method to partition the light field to make each partition occlusion free.

4.4.1 Peeling Off Depth Layers

The idea starts from the fact that the scene points open to all sampling points in the camera plane are not influenced by occlusions, and therefore their depth can be computed accurately. Once the depths of those points are computed, their influence on the other scene points can be reduced by removing the rays coming from those points. Then, the scene points which were previously occluded become visible in more views.

The rays in the light field are iteratively removed based on the value of the focus measure as each focal plane in the focal stack is computed *from front to back*. For example, for a point in the focal stack, the focus measure is computed based on a set of rays sampled from the light field. Then, based on the focus measure, it is determined whether the rays which were used to compute the focus measure will be removed or not. If the focus measure is so high that there is very likely to be a surface, the associated rays in the light field can be removed so that they are not considered any more in the later iterations. By doing so, the influence of those rays to the farther depth layers can be removed. This step is repeated until all the depth layers are reconstructed and no more rays remain in the light field. Algorithm 4.5 describes this method.

This method was also discussed in Ziegler *et al.*'s paper [ZBA$^+$07]. However, the method has two limitations. First, the method is sensitive to the resolution of discretization. If the resolution of the depths is not fine enough, the rays which do not exactly match to a depth layer may be incorrectly removed, which affects the depth reconstruction for the next depth layers. Second, the method is sensitive to the imperfection of the acquired light field, notably radial distortion. The rays lying on the plane corresponding to a depth should match to the rays coming from the same scene point. If it is not the case due to the radial distortion, the rays unrelated to the scene point will be removed.

4.4.2 Partitioning Light Field

The pollution due to occlusions can also be avoided by partitioning the light field so that the ray integration does not span across occlusion boundaries. This ensures that a *subvolume* of the focal

Algorithm 4.5 Scene reconstruction by peeling off depth layers

```
Input: 4D light field $L$, focus measure $F(\cdot)$, ray integration function $I(\cdot)$, threshold $T_w$
Output: 3D volume occupancy $O$
$O(\cdot,\cdot,\cdot) \leftarrow$ 'empty'
for $d =$ front to back do
    for each $(x,y)$ do
        $\{R_{xyd}\} \leftarrow \{L(s,t,x-ds,y-dt)\}$
        $w \leftarrow 1/F(\{R_{xyd}\})$
        if $w < T_w$ then
            $\{L(s,t,u-ds,v-dt)\} \leftarrow 0$        // a surface is detected. remove the rays from $L$
            $O(x,y,d) \leftarrow I(\{R_{xyd}\})$        // mark the occupancy
        end if
    end for
end for
```

stack associated to a light field partition has only one surface point along each ray. In other words, this makes the focus measure profile have a single peak. Thus, once a light field is partitioned in a proper way, extracting 3D occupancy volume can be greatly simplified; the occupancy volume can be obtained by merging the occupancy volume associated to each partition.

There are following relations between a light field and its focal stack. Here, we assume 2D light fields for the sake of simplicity.

- A line in the light field corresponds to a point in the focal stack. A point in the focal stack is an image formed by integrating rays leaving that point and arriving within the synthetic aperture. Those rays are the images of the point projected to viewing positions in the camera plane, and are aligned along a straight line in the light field.
- A point, or a ray, in the light field corresponds to a straight line in the focal stack, which is the path of the ray. The ray leaves some point lying in that straight line, and the color of the ray will be the same as the color of the point.

Inspired by these relations, we can find a candidate scene point where a ray may have left, by examining the color of all points lying in the ray's associated straight line in the focal stack, and picking the point with the closest color to the ray.

If the focal stack is under the influence of occlusion, however, a point in the focal stack may have a mixed color with some other points. This makes the process inaccurate to determine the point in the focal stack from which the ray comes. We use this inaccuracy to determine whether a scene point is influenced by occlusions. If a ray does not have a point in its path in the focal stack with sufficiently close color, the ray comes from an occluded point. The integration of rays including such a ray is always polluted because the integration lines passing through the ray must span across occlusion boundaries. In order for such a ray to be used to reconstruct the scene, either the integration line must be adjusted not to pass across the occlusion boundary (Section 4.1), or the occlusion boundary must be removed by modifying the light field (Section 4.4).

This property itself can be used to infer the existence of surfaces. By computing all the points in the focal stack where rays come from, we can find all candidate points in the focal stack

Algorithm 4.6 Light field partitioning

```
Input: 4D light field $L$, threshold $T$
Output: Light field partitions $\{LP_1, LP_2, \dots, LP_k\}$
$i \leftarrow 1$
while there is a ray in $L$ do
    $\widetilde{L}(\cdot,\cdot,\cdot,\cdot) \leftarrow 0$          // the light field to be reassembled
    build $FS$ from the current $L$ using Algorithm 4.1
    for each available $(s,t,u,v)$ do
        $\{P_{stuv}(d)\} \leftarrow FS(s-dx, t-dy, d)$
        $p \leftarrow$ the member in $\{P_{stuv}(d)\}$ which has the most similar color with $L(s,t,u,v)$
        $\widetilde{L}(s,t,u,v) \leftarrow p$       // reassemble the light field $\widetilde{L}$ using $p$
    end for
    $\Delta(\cdot,\cdot,\cdot,\cdot) \leftarrow \|L(\cdot,\cdot,\cdot,\cdot) - \widetilde{L}(\cdot,\cdot,\cdot,\cdot)\|$
    take the rays from $L$ in the region where $\Delta(\cdot,\cdot,\cdot,\cdot) < T$ to create $LP_i$
    remove $LP_i$ from $L$
    $i \leftarrow i+1$
end while
```

that are maybe on surfaces. This inference is correct provided the elimination of the effect of occlusions. Once we partition the light field to remove occlusions, the depth per ray may be computed accurately using this property. However, in general, it is better to apply one of depth estimation algorithms discussed in Section 4.1 and 4.2 to each partition. It is due to the fact that the color comparison is a weaker constraint than other criterion used to determine the depth.

The light field partitioning can be achieved by the following algorithm. For each ray (s,t,u,v) in the light field L, the points $\{P\}$ lying in the path of the ray in the focal stack FS are collected. Each point in $\{P\}$ can be computed as

$$P_{stuv}(d) = FS(u - ds, v - dt, d), \tag{4.12}$$

where the points on the straight line are parameterized by the depth d. Then, the point p having the closest color to $L(s,t,u,v)$ among $\{P\}$ is identified. A new light field $\widetilde{L}$ with the same parameterization with L is assembled using the color of the point p for each (s,t,u,v). The difference between the reassembled light field $\widetilde{L}$ and the original light field L contains the information about the influence of occlusions. After thresholding the difference using some threshold T, the rays in the light field whose differences are less than T are taken as the first partition. The rays in the partition are the rays whose associated scene points are predicted in the focal stack up to the threshold T, and are considered not influenced by occlusions. These steps are repeated for the remaining rays in light field until no more rays remain. The algorithm is sketched in Algorithm 4.6.

For the partitions of the light field, one of the methods to reconstruct the depth described in Section 4.1 and 4.2 is applied. Since the partitioned light fields are free from the influence of occlusions, a simple reconstruction algorithm such as the one with a variance based focus measure in Section 4.1.1 can be used to estimate the depth. Moreover, because of the absence of the occluding depth layers within each partition, previously occluded surfaces can now be reconstructed as they become visible to the current viewing position. After finding the occupancy volume independently, the set of volumes can be simply merged to form a complete one.

Another measurement can also be considered to identify the points partially occluded. As an example of the orientation estimation in Section 4.3.2, for each ray in the light field and for all orientations per ray, the maximum measure that gives the best orientation estimation can be used as an error measure. If the maximum measure value for a ray among all possible orientations is relatively low, the orientation measure for the ray is not confident enough, and under the influence of other rays coming from occluding objects. In this case, the estimated orientation is not reliable. Therefore, the maximum measurement value can be used as an indicator that shows how confident the estimation is, and can be used as the replacement for the color difference in Algorithm 4.6.

4.5 Reprojection to Light Field

After reconstructing the scene using the methods aforementioned, the resulting occupancy map can be refined by reprojecting the occupancy map back into the light field representation and examining the residual error from the original light field. Then, the volume occupancy is iteratively refined by minimizing the residual error, enforcing the constraints on the value of the volume occupancy.

$$\begin{aligned} O^\star = &\operatorname*{argmin}_O \left\| L - \widetilde{L}(O) \right\|_E \\ &\text{s.t. } 0 \leq O(x,y,z) \leq 1 \quad \text{for all } (x,y,z), \end{aligned} \tag{4.13}$$

where $\widetilde{L}(O)$ is the reprojected light field of the volume occupancy O, and $\|\cdot\|_E$ is the norm of the residual defined in some metric E.

However, this optimization step was not fully explored in this thesis. This step is mentioned here for completeness, and is left for the future work.

5

Experimental Results

To test our approaches, experiments were performed on three data sets, including two captured data sets and a synthetic data set. Both the depth reconstruction (Section 4.1 and 4.2) and the volume reconstruction (Section 4.3 and 4.4) were tested. The results of those experiments are reported in this chapter. First, we discuss the light field acquisition, and then we present the experimental results of the selected algorithms. We also provide an analysis about the influence of the parameter selection.

5.1 Data Acquisition

For the experiments, a sequence of 2D images of the scene was taken along a 1D translation of a camera. The camera movement was aligned to the horizontal image axis. The set of camera positions constitutes the baseline. The camera was placed at equally spaced positions in the baseline, looking at the same direction perpendicular to the baseline. Figure 5.1 shows the acquisition set up.

This acquisition produces 3D light fields which are spatially 2D and directionally 1D. 3D light fields are advantageous over 4D light fields in that the acquisition and the calibration becomes less complicated, and that each 2D slice of the 3D light fields can be processed independently in parallel.

Two captured data sets and one synthetic data set were used for experiments. The captured data sets were taken from UCSD/MERL light field repository, which were acquired for the work of Zwicker *et al.* [ZMDP06]. One is a scene consisting of a toy train and buildings ("train" data set). This data set consists of 500 images, and the image resolution is 1255×473. The other is a scene of an elephant model in front of plants ("elephant" data set). This data set consists of

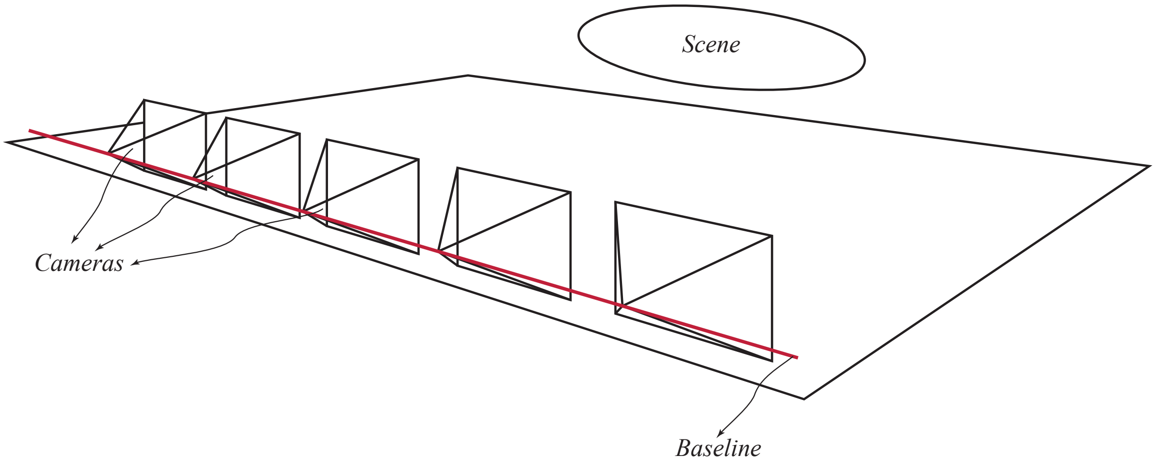


Figure 5.1: Acquisition of a 3D light field. 2D images of the scene are taken at equally spaced camera positions. The images stacked together form a 3D light field.

460 images, and the image resolution is 1280×853. The images are linear RGB images. We used half the images in each sequence, smoothed and downsampled to half the original size. Figure 5.2 shows some of the used images. The data sets were taken using a camera on a 1D linear translating gantry. All images of each data set were taken with the same camera, and the camera positions were equally spaced. The images were geometrically calibrated and rectified, but the non-linear distortion was not corrected. Radiometric falloff (vignetting) was corrected to ensure that the image of equally illuminated surface would produce a flat image. No other photometric calibration was performed. Therefore, the images exhibit slight intensity differences, and have radial distortion.

We alleviated the radial distortion by picking several scene points manually and estimating the coefficients of the polynomial distortion model [Bro66]. The coefficients were estimated up to the fourth order. Tangential distortion was not corrected. The correction, however, did not remove the distortion completely due to the unknown intrinsic camera parameters. Therefore, the data sets still had geometric distortion. Although the data sets were not correctly calibrated, our methods performed well as can be seen in the remainder of this chapter. Both data sets have a few specular or translucent objects, but most surfaces are Lambertian and the scenes have no transparent or reflective objects.

The synthetic data set ("cube" data set) contains a few geometric primitives such as cubes, spheres, and pyramids. The data set was ray traced using Maya. It consists of 200 color images, and the resolution is 640×209. See also Figure 5.2 for the synthetic data set.

5.2 Experiment Setup

The algorithms were implemented using MATLAB. The image interpolation routine and the mean shift clustering algorithm were most time consuming, and thus implemented using CUDA on NVIDIA graphics cards. MATLAB routines and CUDA routines interface through MATLAB's MEX binary files. We used a PC with an Intel i7 2.8 GHz CPU.

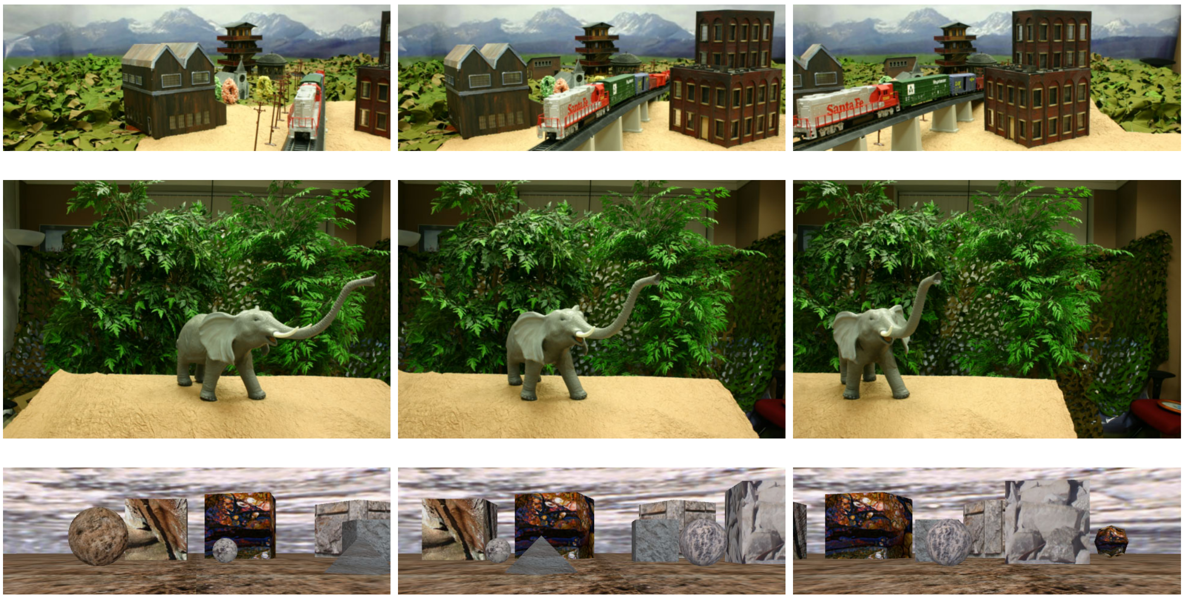


Figure 5.2: The three data sets used in the experiments, including two captured data sets (the first and the second rows) and one synthetic data set (the third row). First row: "train" data set. Second row: "elephant" data set. Third row: "cube" data set. The first column shows the images taken at the leftmost camera position, the second column shows the images taken at the center, and the third column shows the images taken at the rightmost camera position.

We tested the depth reconstruction algorithms (Algorithm 4.1) using SSD (Equation 4.4), ray selection (Algorithm 4.2), and ray clustering (Algorithm 4.3) with or without using neighboring rays (Equation 4.8), the depth from focus method without occlusion handling (Section 4.2.1), and the volume reconstruction algorithms (Algorithm 4.4) using the same focus measures. The depth peeling algorithm (Algorithm 4.5) was only applied to demonstrate the occlusion-free refocusing. The results of the light field partitioning algorithm (Algorithm 4.6) are not reported in the thesis, since our implementation was not complete, and hence did not produce better results than other algorithms.

The resampling from 3D light fields involves a 2D shear transformation. The images are interpolated to deal with subpixel shearing. Since shearing is axis-aligned, and we do not oversample along the directional dimension, 1D linear interpolation over each 1D subimage was employed. Except for the algorithms using neighboring rays discussed in Section 4.2, each ray can be processed independently throughout the entire pipeline. In addition, the interpolation and the clustering routine can be designed not to access the same ray at the same time, which is well suited for the parallel processing. Although in our implementation, only a small portion was implemented using CUDA, the whole pipeline may be implemented more efficiently using parallel processing to further reduce the processing time. We believe there is much room for improvement, which is left for future work.

Since the algorithm runs independently for each 2D slice of the 3D light field, we processed each slice at a time. In our implementation, it took about 1.5 seconds to process one slice of the 3D light field of the "train" data set with the discretization of 300 disparities using occlusion handling based on ray clustering described in Section 4.1.3. It took about three minutes to process the whole light field and to generate the 2D disparity map in an "embarrassingly parallel" processing mode, where the algorithm runs in parallel simply in multiple MATLAB instances.

5.3 Depth Reconstruction

The depth reconstruction algorithms using different focus measures were applied to the three data sets. Figure 5.3–5.5 show brief comparisons between the methods. Each figure shows the 2D disparity maps reconstructed by the depth reconstruction algorithm (Algorithm 4.1) with three different focus measures (left page), and the reconstruction using neighboring rays, multiple-view projection, and spatial frequency (right page). For "elephant" scene, the last two results were omitted.

Figure 5.6 shows the effects of different ray selection rates. When all the rays are used, partially occluded objects are not reconstructed correctly. By reducing the portion of selected rays, we can reconstruct occluded objects which are seen by at least the same portion of camera positions within the aperture. For example, if half the rays are selected to compute the focus measure, the objects visible at more than half the camera positions can be reconstructed. With a low selection rate, the focus measure can tolerate the imperfect acquisition. However, as the selection rate becomes lower, the reliability of the focus measures also decreases. It also becomes less discriminative and less robust to noise. Thus, too low selection rate produces noisy outputs. In Figure 5.7, the relation between the ray selection rate and the visibility rate is clearly observed. The reconstruction quality of the background area between the leftmost cube and the second leftmost cube improves as the ray selection rate decreases.

The aperture size also affects the quality of reconstruction (Figure 5.8). With a wider aperture, the reconstruction is more smooth, and has less outliers. However, the effects of occlusions become more noticeable, so that surfaces with a limited visibility are not well reconstructed. With a smaller aperture, the reconstruction is less influenced by occlusions, but shows a similar tendency observed with a low ray selection rate, resulting in noisy outputs. In addition, it is more difficult to reconstruct the area with a uniform color when using a smaller aperture. The wall above the plants is correctly reconstructed when the full aperture is used (Figure 5.6(a)), but is not when a small aperture is used (Figure 5.6(b)). If a very wide aperture is used, the depth of a uniform region is bounded by the depth of the neighboring textured regions. With a smaller aperture, however, this is not the case. It is more intuitive to cast the depth estimation to the orientation estimation of the light field stripe. If we use a wide aperture and thus consider more views, the slope of the uniform stripe is more tightly bounded by the neighbor stripes. On the other hand, if the aperture size is smaller than the width of the uniform region, the orientation is not bounded at all, and the region cannot be reconstructed correctly.

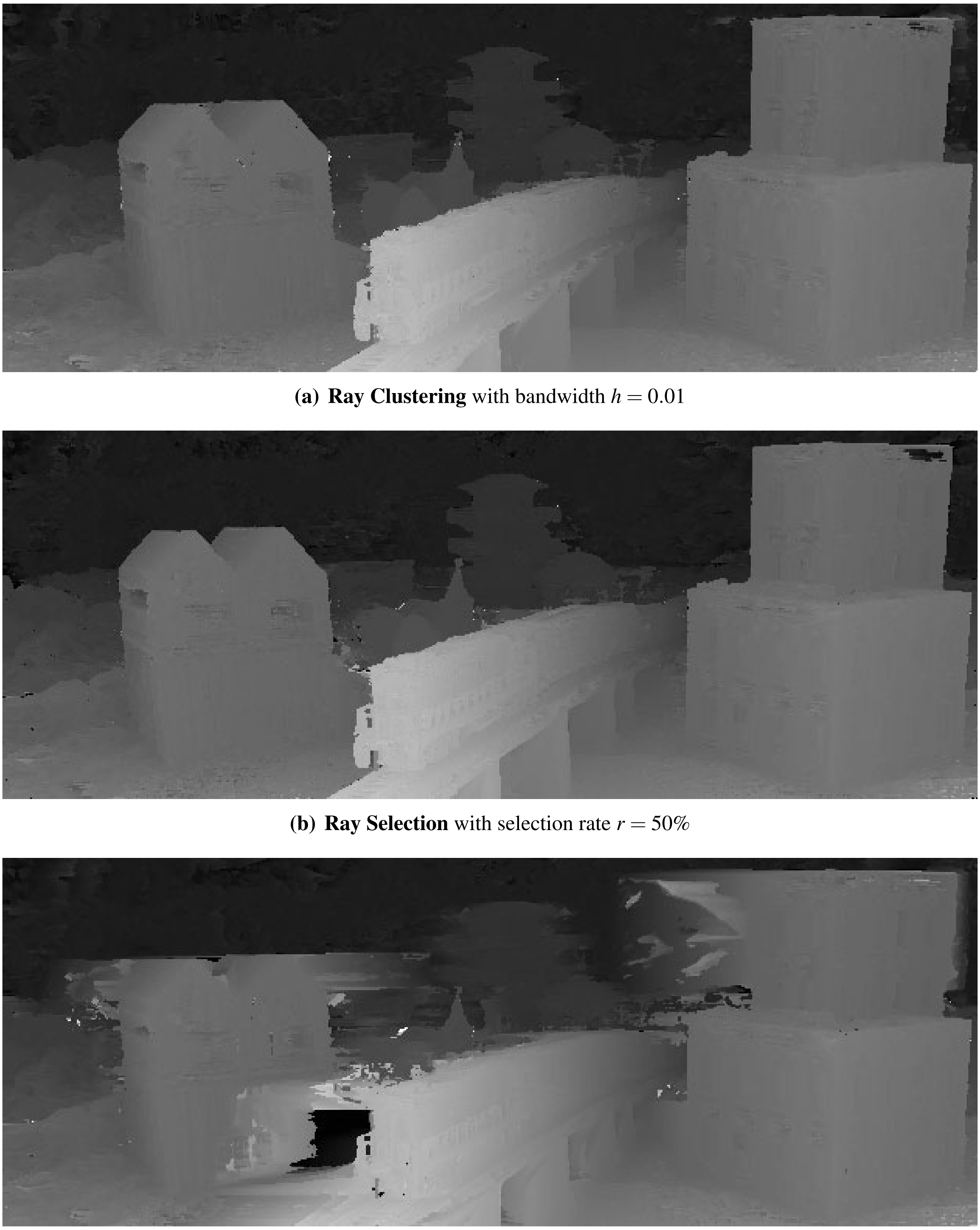
Similar to the ray selection rate, reducing the aperture size can be beneficial to compensate the imperfect data set, such as radial distortion. The reconstruction with the full aperture is slightly worse in the region of close objects than with a reduced aperture (see the table in the front in Figure 5.6(a) and (b)), since a closer object exhibits a larger disparity, hence spans larger extent of the image plane between images for different viewing positions.

Using the focus measures of the neighboring rays, the reconstruction produces smoother results.

In particular, many outliers are removed, compared to the reconstruction without considering neighbors. However, thin structures narrower than the width of the spatial window are eroded by the objects surrounding them. Figure 5.9 shows the effect of the use of neighboring rays.

Figure 5.3(a), 5.4(a), and 5.5(a) present the reconstruction using ray clustering, showing that the method can adapt to a wide range of visibility conditions, compared to the fixed rate of ray selection. The mean shift clustering has only one parameter called bandwidth parameter which determines the window size used for the kernel density estimation (Section 4.1.3). Figure 5.10 shows the comparison with varying this parameter. If we increase the bandwidth parameter, the results becomes more similar to the result using all rays without any discrimination. If we set the bandwidth parameter as large as to cover the whole extent of the feature space, all the data points will be in the same cluster, and the cluster center becomes the (weighted) mean of the samples. The feature space in our case is the RGB color space, which is a cube with a unit volume. On the other hand, if too small a bandwidth is used, most data points form their own clusters. Thus, clustering loses its discriminating power.

In sum, the reconstruction using the ray selection with the full aperture produced good results, and the ray clustering method gave the best results. If no occlusion handling is used, the aperture size should be very small. However, it trades off the confidence of the estimated depth.

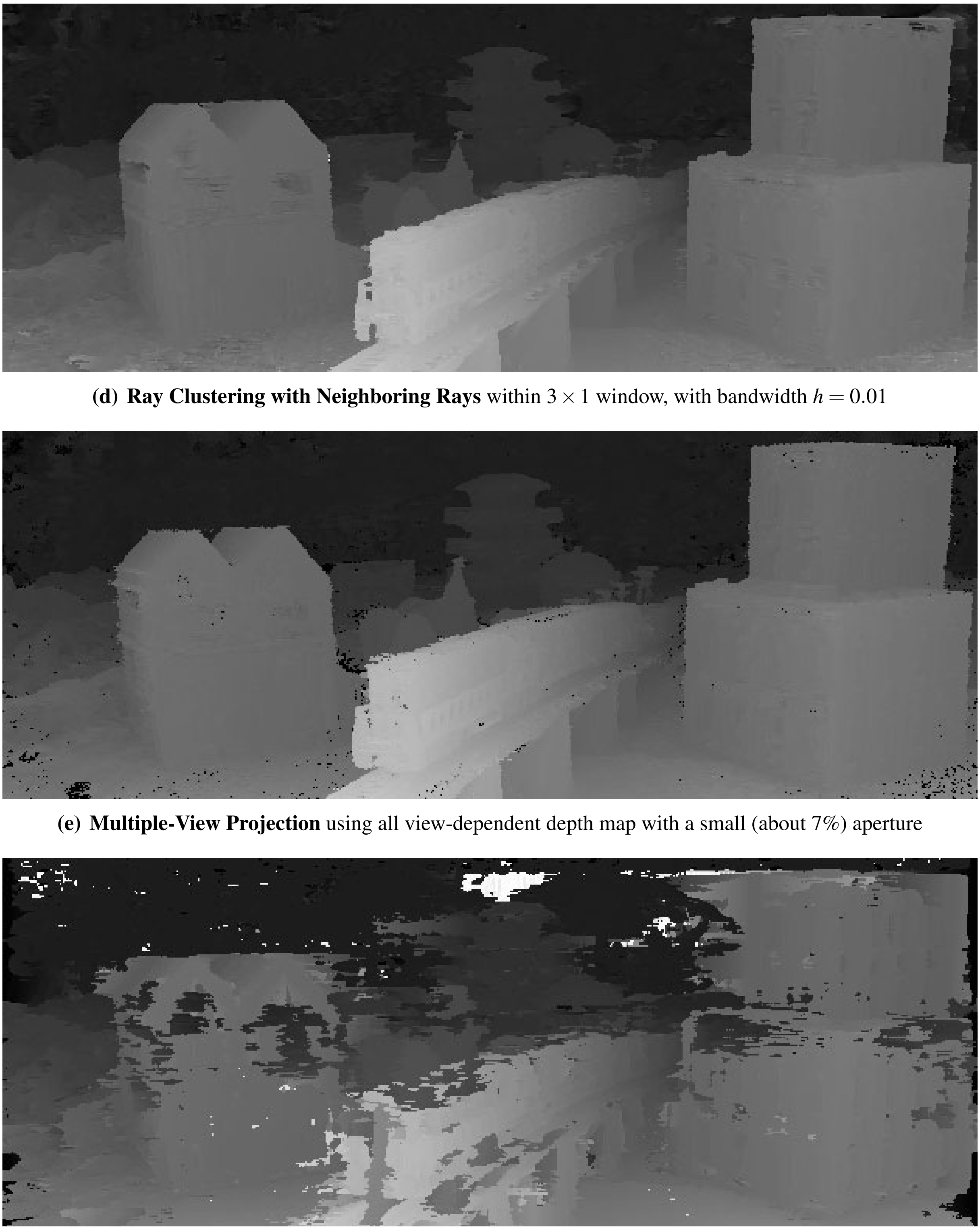


(a) Ray Clustering with bandwidth $h = 0.01$

(b) Ray Selection with selection rate $r = 50\%$

(c) SSD

Figure 5.3: Comparison of Focus Measures. The comparison of the depth reconstruction of the “train” data set using different focus measures. The methods using SSD and DFF do not have an occlusion handling mechanism. Their respective results (c) and (f), however, are included to show the importance of the occlusion handling.

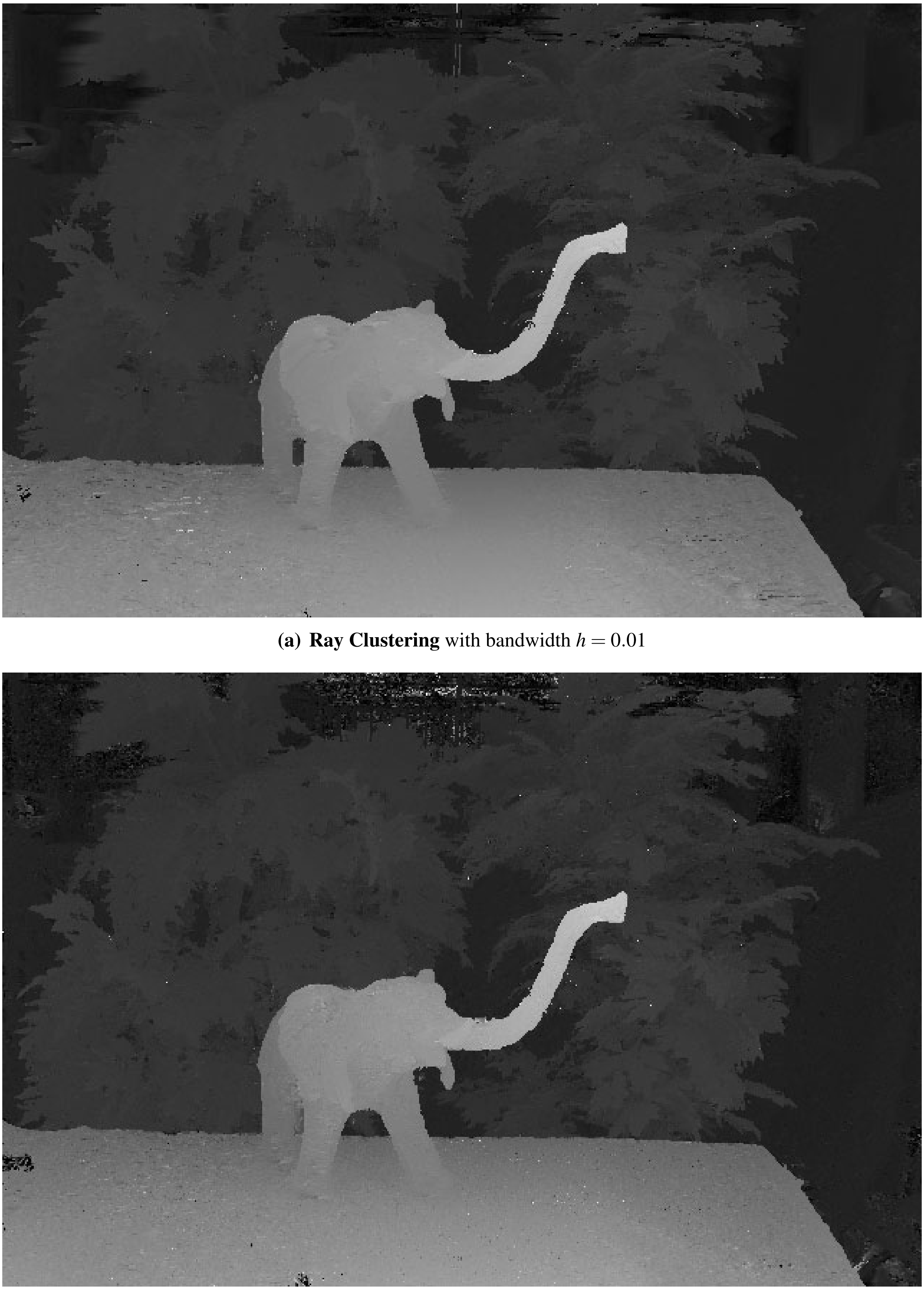


(d) Ray Clustering with Neighboring Rays within 3×1 window, with bandwidth $h = 0.01$

(e) Multiple-View Projection using all view-dependent depth map with a small (about 7%) aperture

(f) DFF using spatial frequency

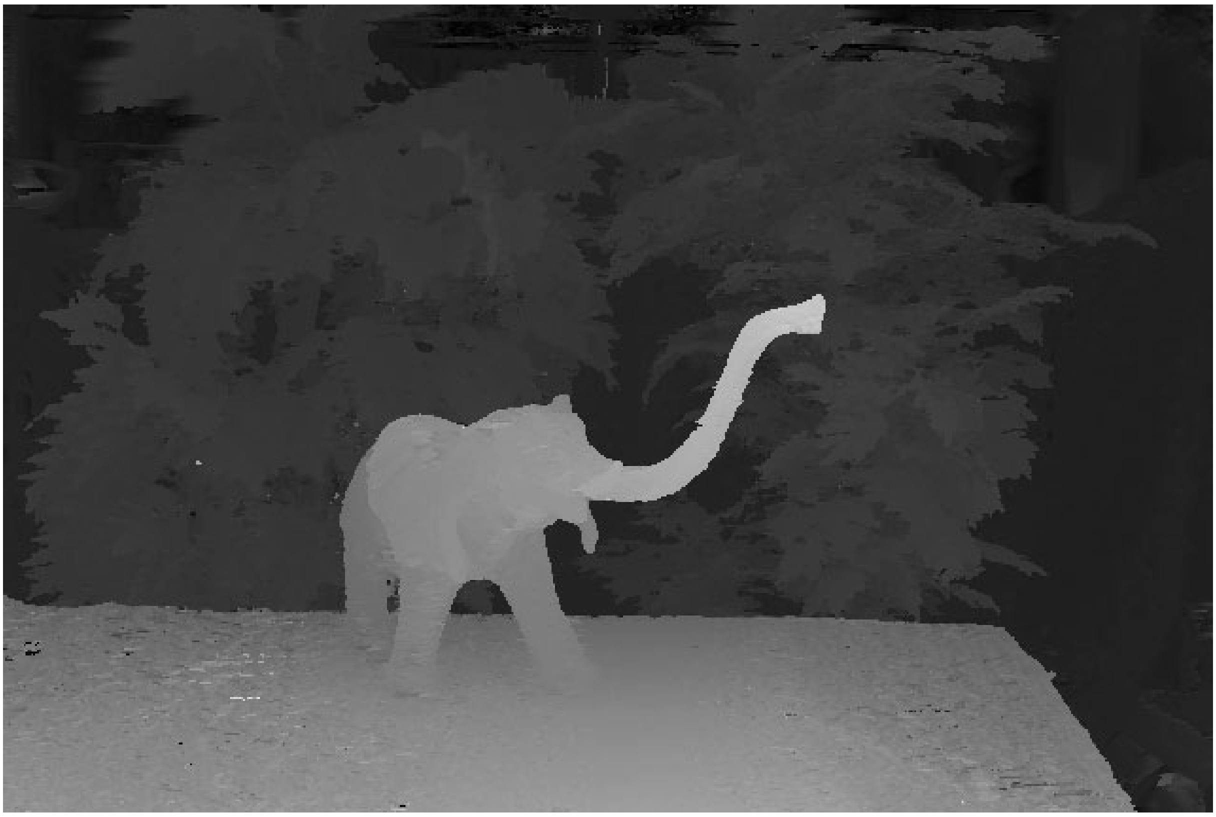
Figure 5.3: *(continued)* **Comparison of Focus Measures.** The comparison of the depth reconstruction of "train" data set using different focus measures. The methods using SSD and DFF do not have an occlusion handling mechanism. Their respective results (c) and (f), however, are included to show the importance of the occlusion handling.



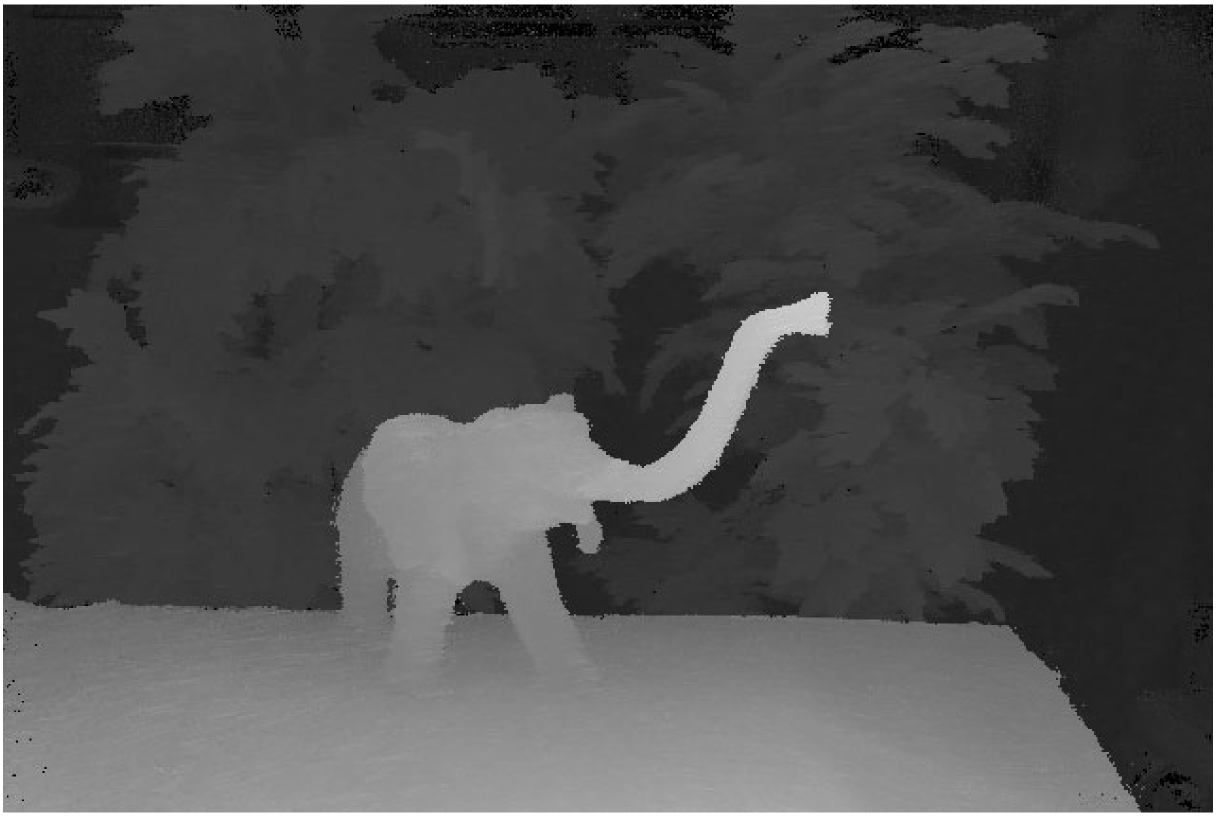
(a) Ray Clustering with bandwidth $h = 0.01$

(b) Ray Selection with selection rate $r = 50\%$

Figure 5.4: Comparison of Focus Measures. The comparison of the depth reconstruction of “elephant” data set using different focus measures. The results using SSD and DFF are omitted due to the limited space here.

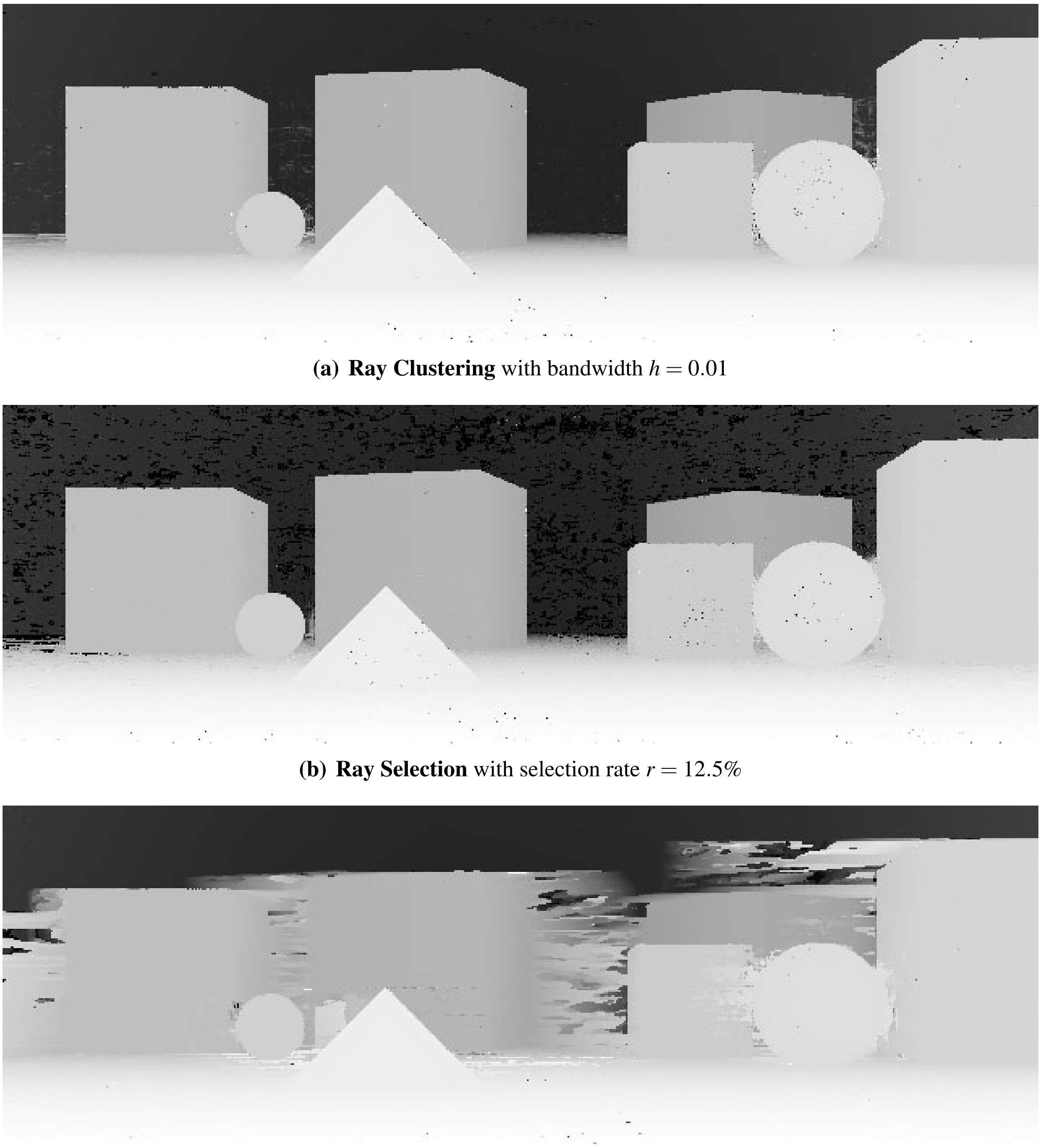


(c) Ray Clustering with Neighboring Rays within 3×1 window, with bandwidth $h = 0.01$



(d) Multiple-View Projection using all view-dependent depth map with a small (about 7%) aperture

Figure 5.4: *(continued)* **Comparison of Focus Measures.** The comparison of the depth reconstruction of "elephant" data set using different focus measures. The results using SSD and DFF are omitted due to the limited space here.

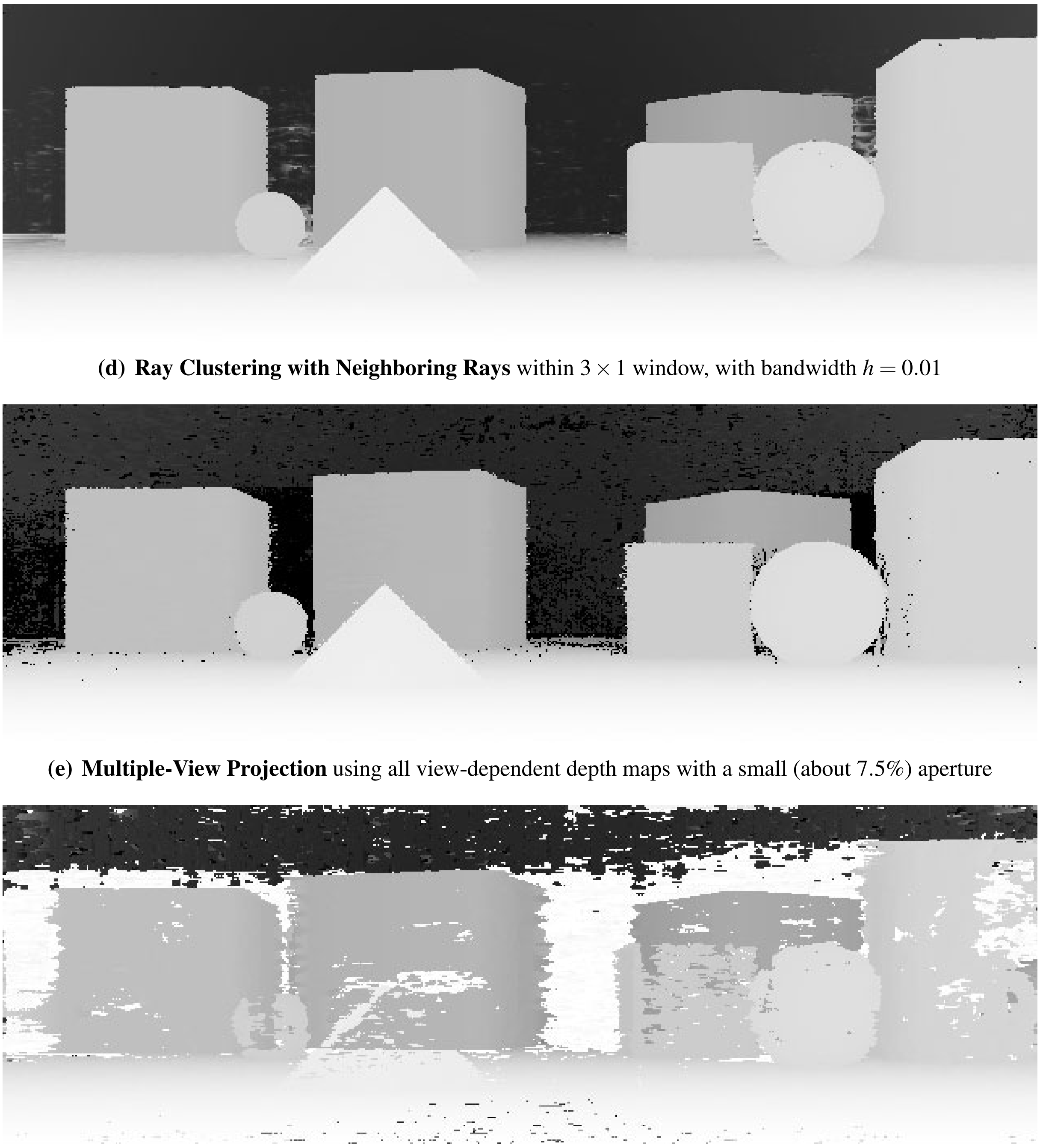


(a) Ray Clustering with bandwidth $h = 0.01$

(b) Ray Selection with selection rate $r = 12.5\%$

(c) SSD

Figure 5.5: Comparison of Focus Measures. The comparison of the depth reconstruction of "cube" data set using different focus measures. The methods using SSD and DFF do not have an occlusion handling mechanism. Their respective results (c) and (f), however, are included to show the importance of the occlusion handling.

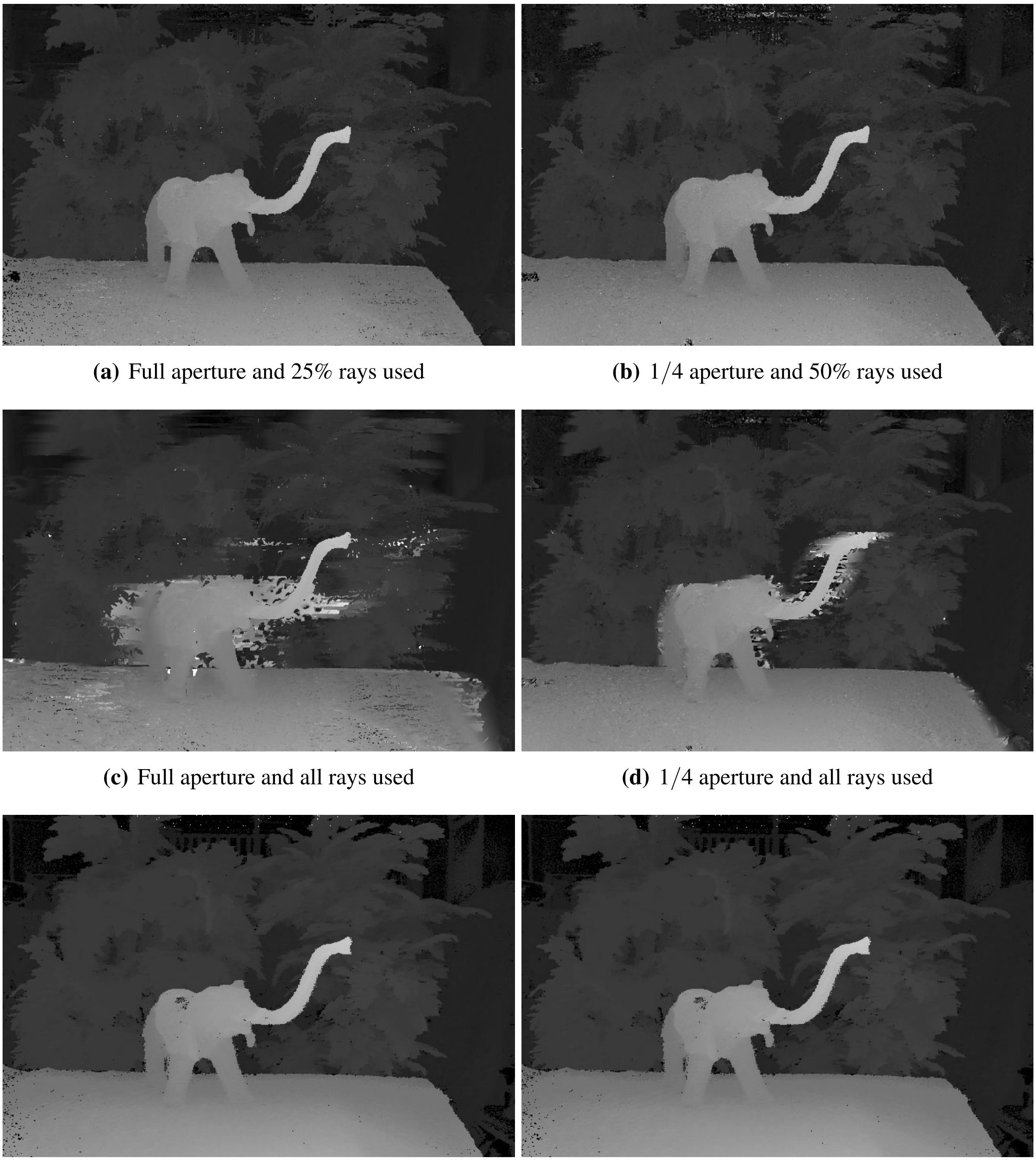


(d) **Ray Clustering with Neighboring Rays** within 3×1 window, with bandwidth $h = 0.01$

(e) **Multiple-View Projection** using all view-dependent depth maps with a small (about 7.5%) aperture

(f) **DFF** using spatial frequency

Figure 5.5: *(continued)* **Comparison of Focus Measures.** The comparison of the depth reconstruction of “cube” data set using different focus measures. The methods using SSD and DFF do not have an occlusion handling mechanism. Their respective results (c) and (f), however, are included to show the importance of the occlusion handling.



(a) Full aperture and 25% rays used

(b) $1/4$ aperture and 50% rays used

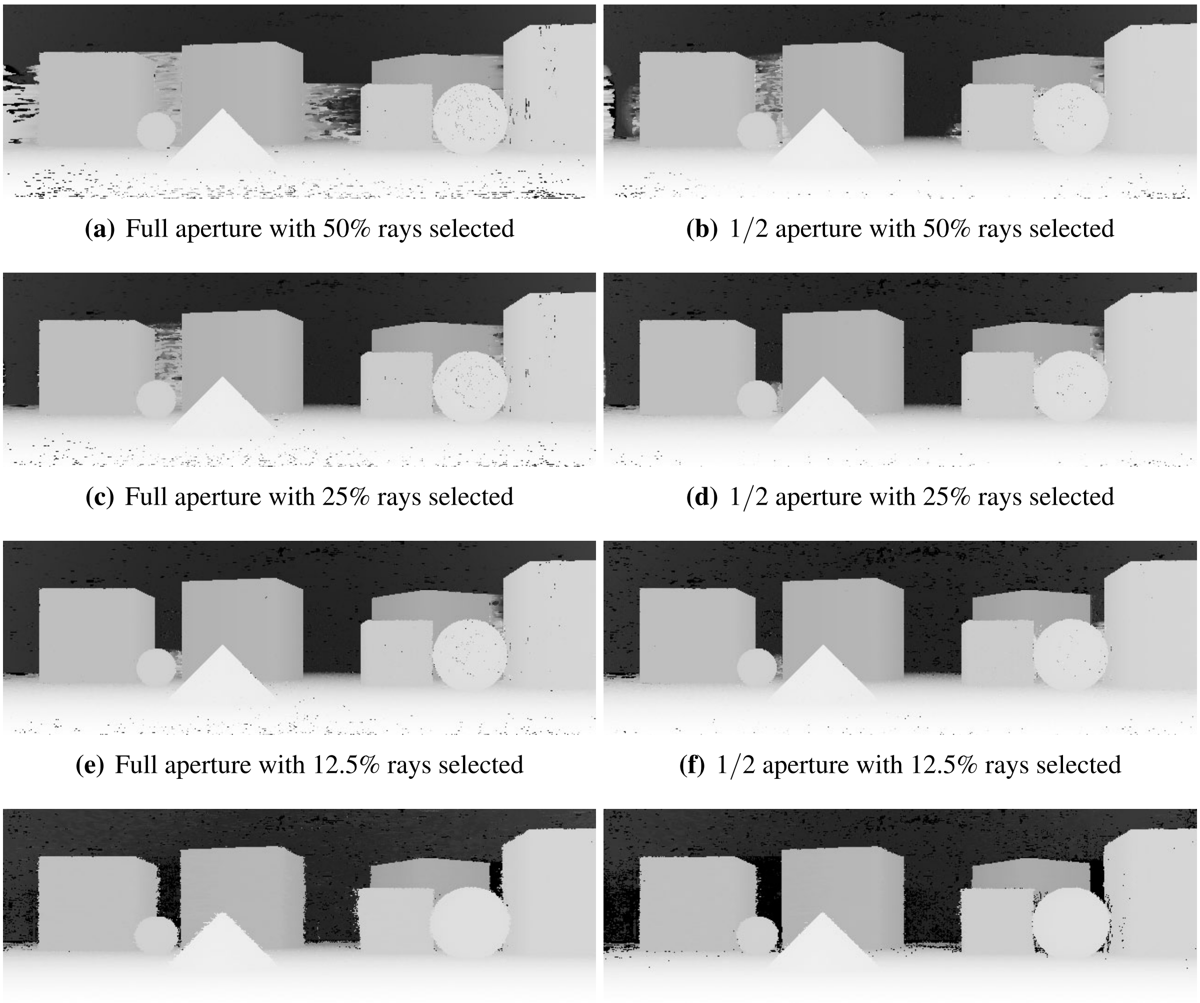
(c) Full aperture and all rays used

(d) $1/4$ aperture and all rays used

(e) A depth map extracted from the 3D occupancy map built using the full aperture and 50% rays selected

(f) A depth map extracted from the 3D occupancy map built using 7.5% aperture and 50% rays selected

Figure 5.6: Effect of Aperture Size. The reconstruction of "elephant" data set. The scene was reconstructed using a single view in (a)–(d), and using the multiple-view projection in (e) and (f). For (e) and (f), the depth estimations from all views were projected, and the voxels with less than 7 votes were considered as outliers and discarded.



(a) Full aperture with 50% rays selected

(b) 1/2 aperture with 50% rays selected

(c) Full aperture with 25% rays selected

(d) 1/2 aperture with 25% rays selected

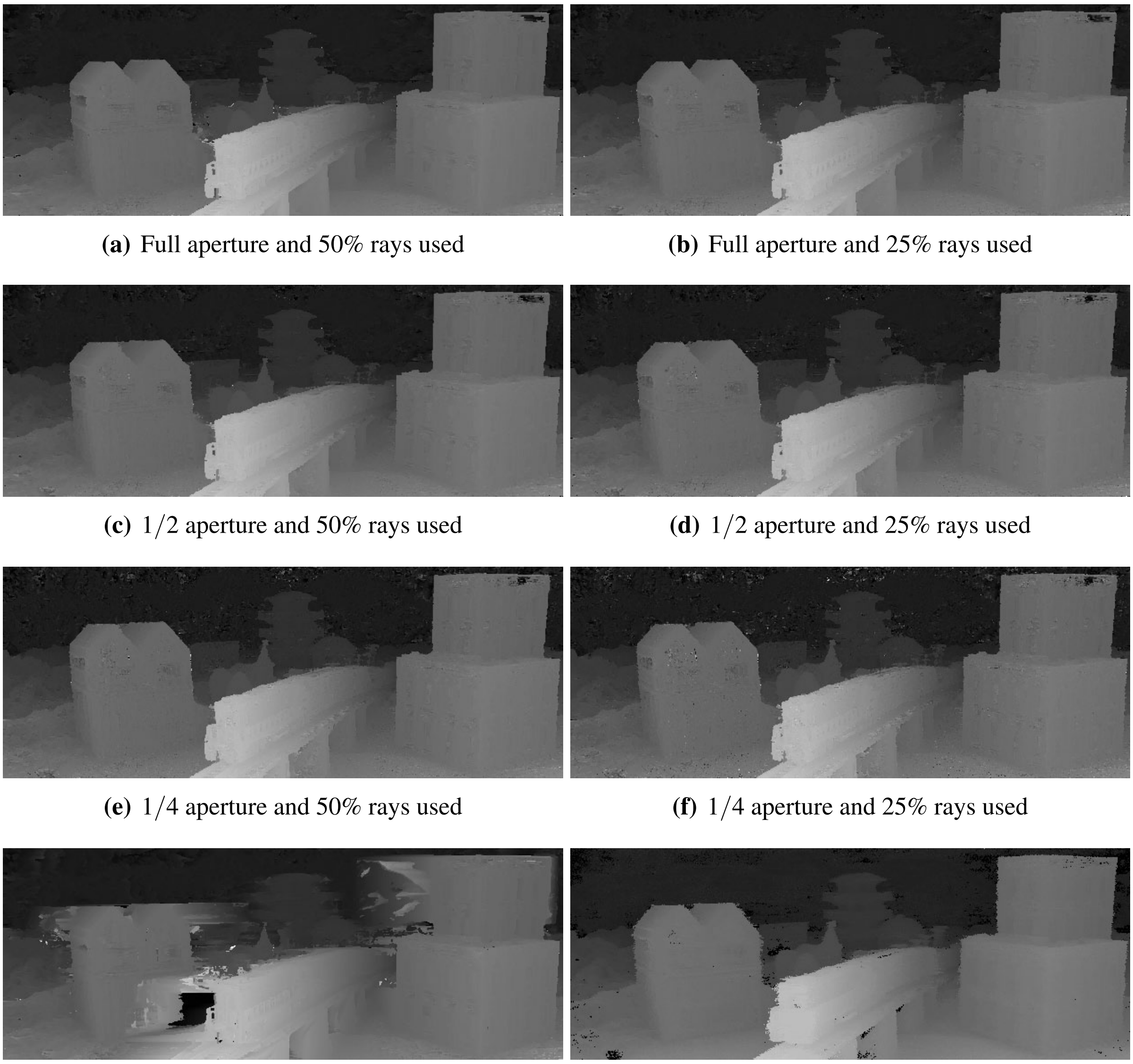
(e) Full aperture with 12.5% rays selected

(f) 1/2 aperture with 12.5% rays selected

(g) Depth map from the 3D occupancy map, thresholded by 5. 50% rays within about 1/10 aperture were used to build the 3D occupancy map

(h) Depth map from the 3D occupancy map, thresholded by 10. The same setting was used as (g).

Figure 5.7: Effect of Ray Selection Rate. The reconstruction of 'cube' data set. The first three rows show the direct reconstruction of 2D depth maps. The last row shows the 2D depth map extracted from the 3D occupancy map. (a), (c), and (e) used the full aperture, whereas (b), (d), and (f) used half the aperture. (g) and (h) show the effect of the thresholding. The threshold value for (g) was 5, and (h) was 10.



(a) Full aperture and 50% rays used

(b) Full aperture and 25% rays used

(c) 1/2 aperture and 50% rays used

(d) 1/2 aperture and 25% rays used

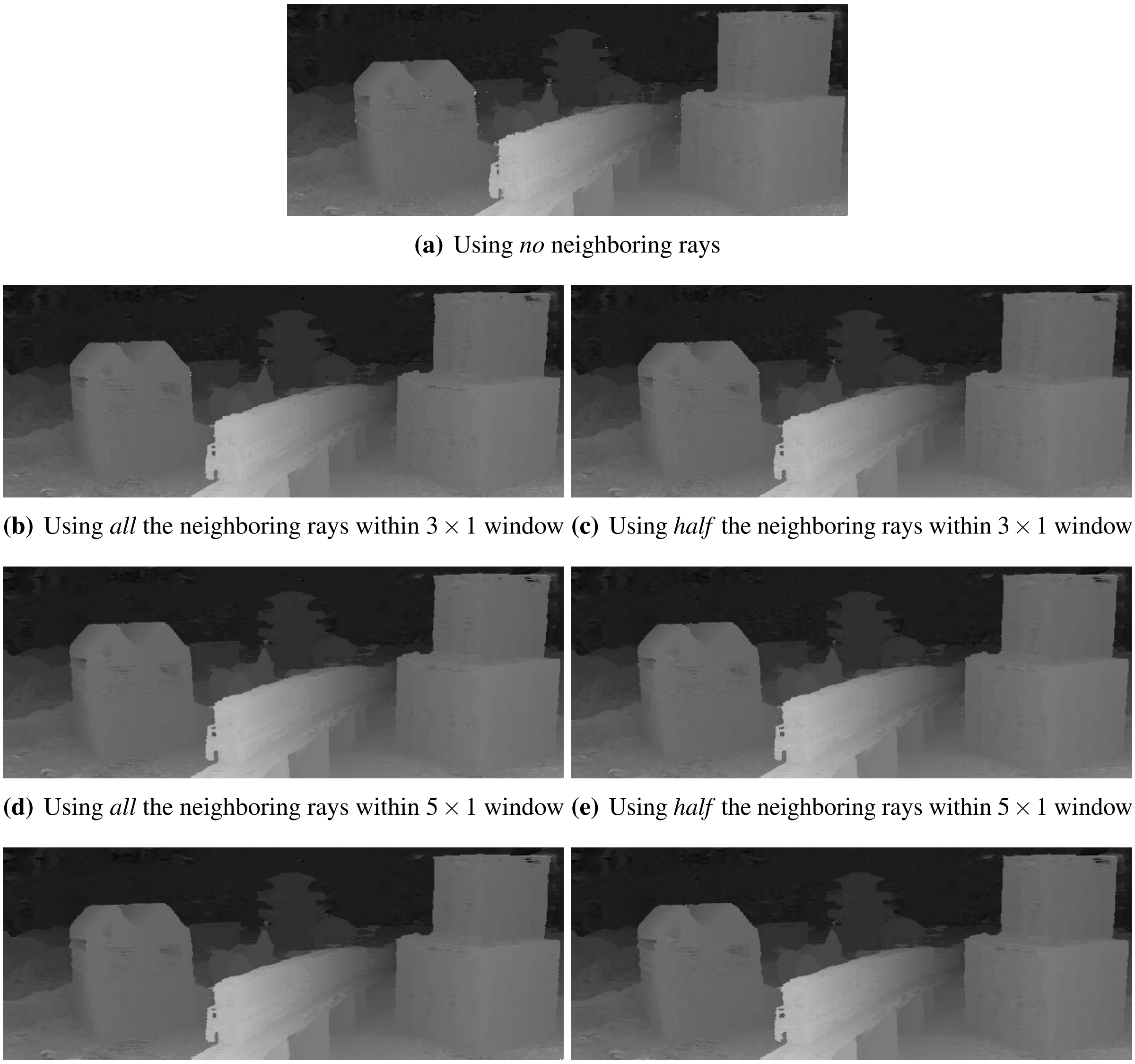
(e) 1/4 aperture and 50% rays used

(f) 1/4 aperture and 25% rays used

(g) Full aperture and all rays used

(h) A depth map extracted from the 3D occupancy map built using 7.5% aperture and 50% rays selected

Figure 5.8: Effect of Ray Selection Rate. The reconstruction of 'train' data set. The scene was reconstructed using a single view in (a)–(g), and using the multiple-view projection in (h). The aperture size decreases from the first row to the third row. (g) used the full aperture and all rays within the aperture, and thus does not handle occlusions. (h) was extracted from a 3D occupancy map.



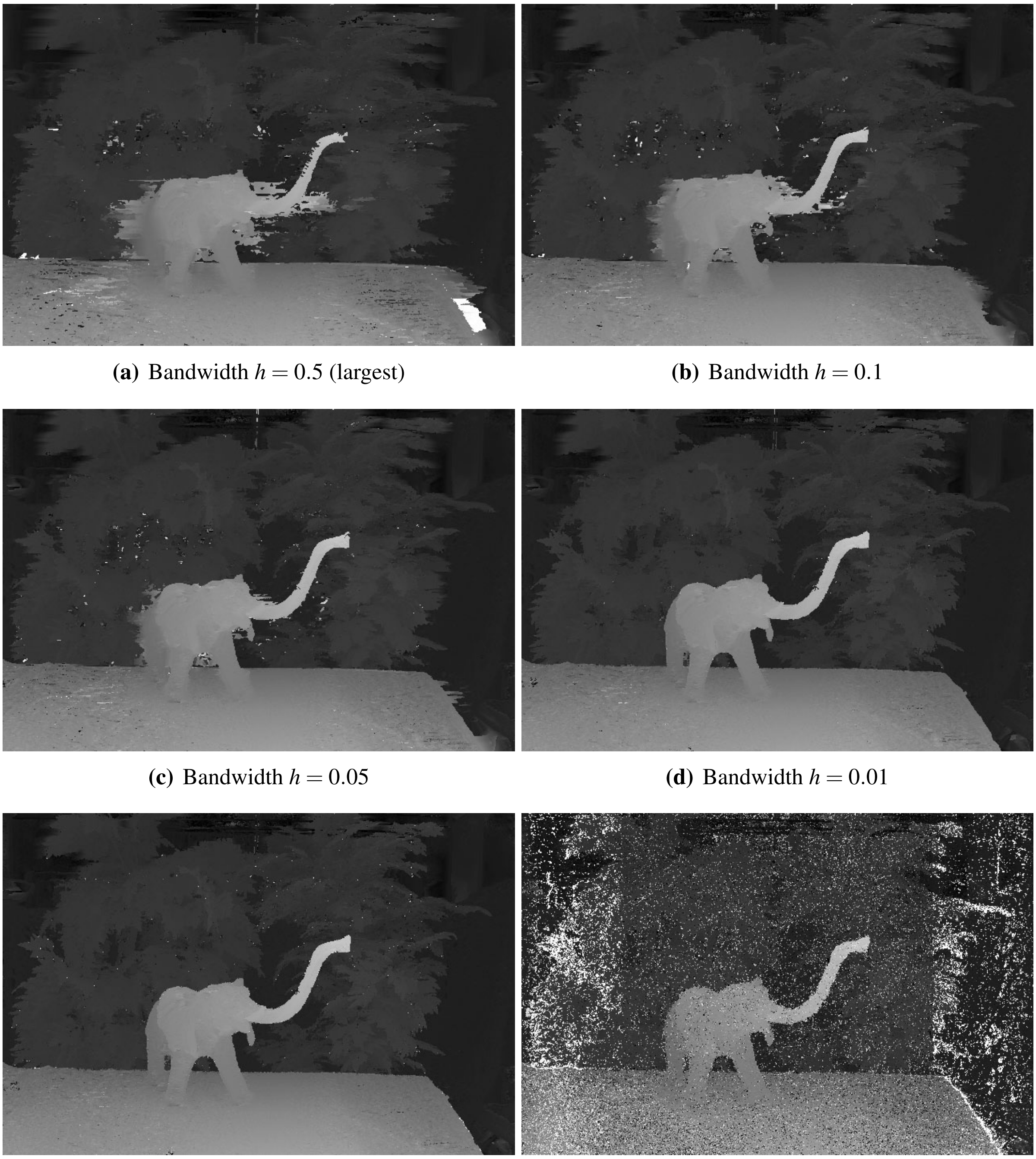
(a) Using *no* neighboring rays

(b) Using *all* the neighboring rays within 3 × 1 window **(c)** Using *half* the neighboring rays within 3 × 1 window

(d) Using *all* the neighboring rays within 5 × 1 window **(e)** Using *half* the neighboring rays within 5 × 1 window

(f) Using *all* the neighboring rays within 7 × 1 window **(g)** Using *half* the neighboring rays within 3 × 1 window

Figure 5.9: Using Neighboring Rays. The reconstructed 2D depth maps of "train" scene using ray clustering with bandwidth $h = 0.01$. The scene was reconstructed with different window size and strategy over the neighboring rays. The window size increases from top to bottom. All the neighboring rays in the window were used in the left column, whereas only half the rays were used in the right column. As the window size increase, the outliers are reduced. However, the objects are eroded, and some thin structures disappear.



(a) Bandwidth $h = 0.5$ (largest)

(b) Bandwidth $h = 0.1$

(c) Bandwidth $h = 0.05$

(d) Bandwidth $h = 0.01$

(e) Bandwidth $h = 0.005$

(f) Bandwidth $h = 0.001$ (smallest)

Figure 5.10: Effect of Bandwidth Parameter in Ray Clustering. The reconstructed 2D depth maps of "elephant" scene using ray clustering with various bandwidth parameters. With a larger bandwidth, the scene points with limited visibility are not well reconstructed, and the reconstruction becomes more similar with the case of no occlusion handling. On the other hand, with a smaller bandwidth, the reconstruction becomes noisier.

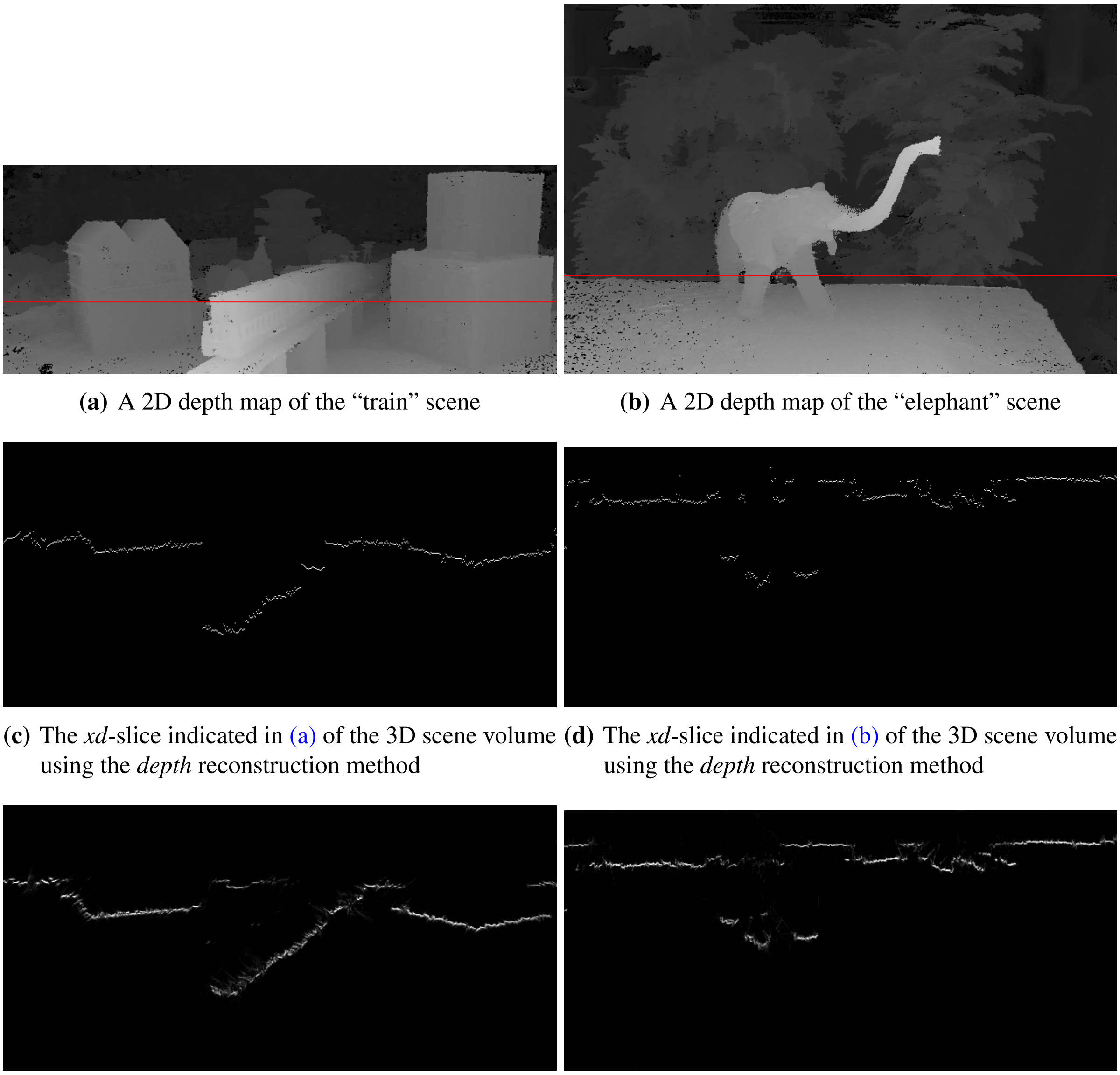
5.4 Volume Reconstruction

Figure 5.11 shows the volume reconstruction based on the projection of multiple view-dependent depth maps (Algorithm 4.4). At each viewing position, the aperture spanning 50% of viewing positions were used to create the 2D depth map. 50% of rays within the aperture were selected to compute the focus measure. Figure 5.11(e) and (f) are the horizontal slices of the 3D occupancy volumes of "train" and "elephant" data sets, respectively. With the comparison to the cross section of the single view reconstruction (Figure 5.11(c) and (d), respectively), the surfaces that are occluded from the given viewing position can be observed. However, for the captured data sets, the reconstructed surfaces become *fatter*, because the depth estimations in different views are slightly different, and thus do not map to the same scene point. We suspect this is caused by radial distortion existing in the captured data sets.

In Figure 5.12, the 2D depth maps extracted from the 3D occupancy map are presented with the directly computed depth maps. Taking the closest points to the camera plane in the 3D occupancy map yields a 2D depth map. The reconstructed surfaces are more smooth, but have more holes, compared to their correspondences in the direct 2D depth maps. Note the difference of the aperture size. The full aperture was used for the depth reconstruction, whereas less than 10% of the aperture was used for the volume reconstruction. With the volume reconstruction method, the aperture size can be more reduced than the depth reconstruction methods, because the reconstructed scene points are accumulated in the 3D voxel space. Thus, the surfaces with more limited visibility can be handled. However, the object boundaries are usually more noisy, because small errors in depth estimation are amplified during the view warping. If there is some error in a few number of views, this erroneous estimation affects the occupancy map, and is more conspicuous in the object boundaries. The holes are due to the thresholding.

5.5 Occlusion-Free Refocusing

Figure 5.13 shows the examples of refocused images, which are 2D *xy* slices of 3D focal stacks. In the refocused images without occlusion handling, the trace of the occluding object (e.g. the elephant in Figure 5.13(g)) is present in the object (e.g. the net in Figure 5.13(i)) behind it. With occlusion handling using the depth peeling method described in Algorithm 4.5, the occluded object can be more clearly focused. That is, we can remove most of the rays coming from the occluding objects instead of blurring them out. Thus, the rays coming from the object behind the occluding objects are integrated to form the scene point. Consequently, with the occlusion handling, we can see the scene through the occluders. In the Figure 5.13(i) and (l), the elephant was mostly removed from the scene, and the net and plants are seen more clearly. However, the gray area (Figure 5.13(l)) is still observed near to the center of the area where the elephant was. This area is a completely occluded region by the elephant. The scene point in that region is not visible from any camera position, and thus cannot be reconstructed.



(a) A 2D depth map of the "train" scene

(b) A 2D depth map of the "elephant" scene

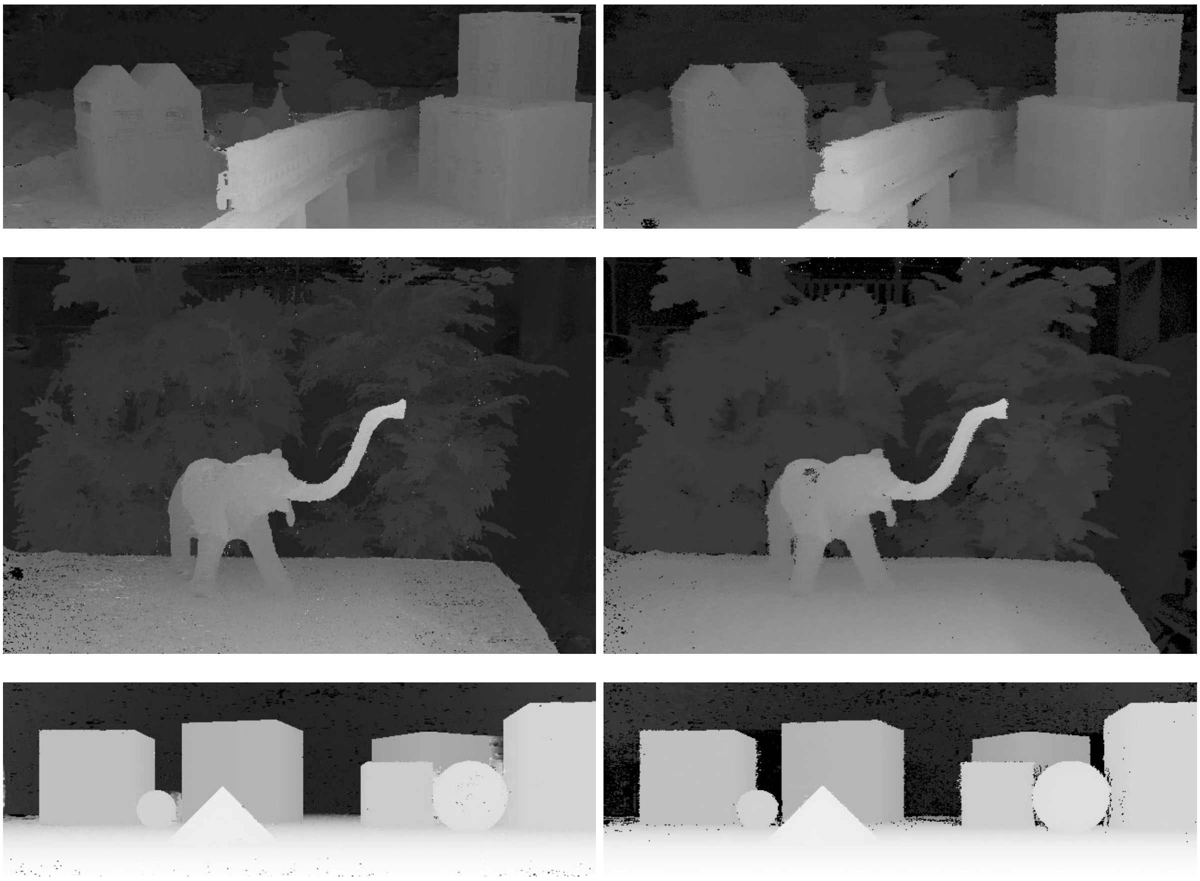
(c) The *xd*-slice indicated in (a) of the 3D scene volume using the *depth* reconstruction method

(d) The *xd*-slice indicated in (b) of the 3D scene volume using the *depth* reconstruction method

(e) The *xd*-slice indicated in (a) of the 3D scene volume using the *volume* reconstruction method

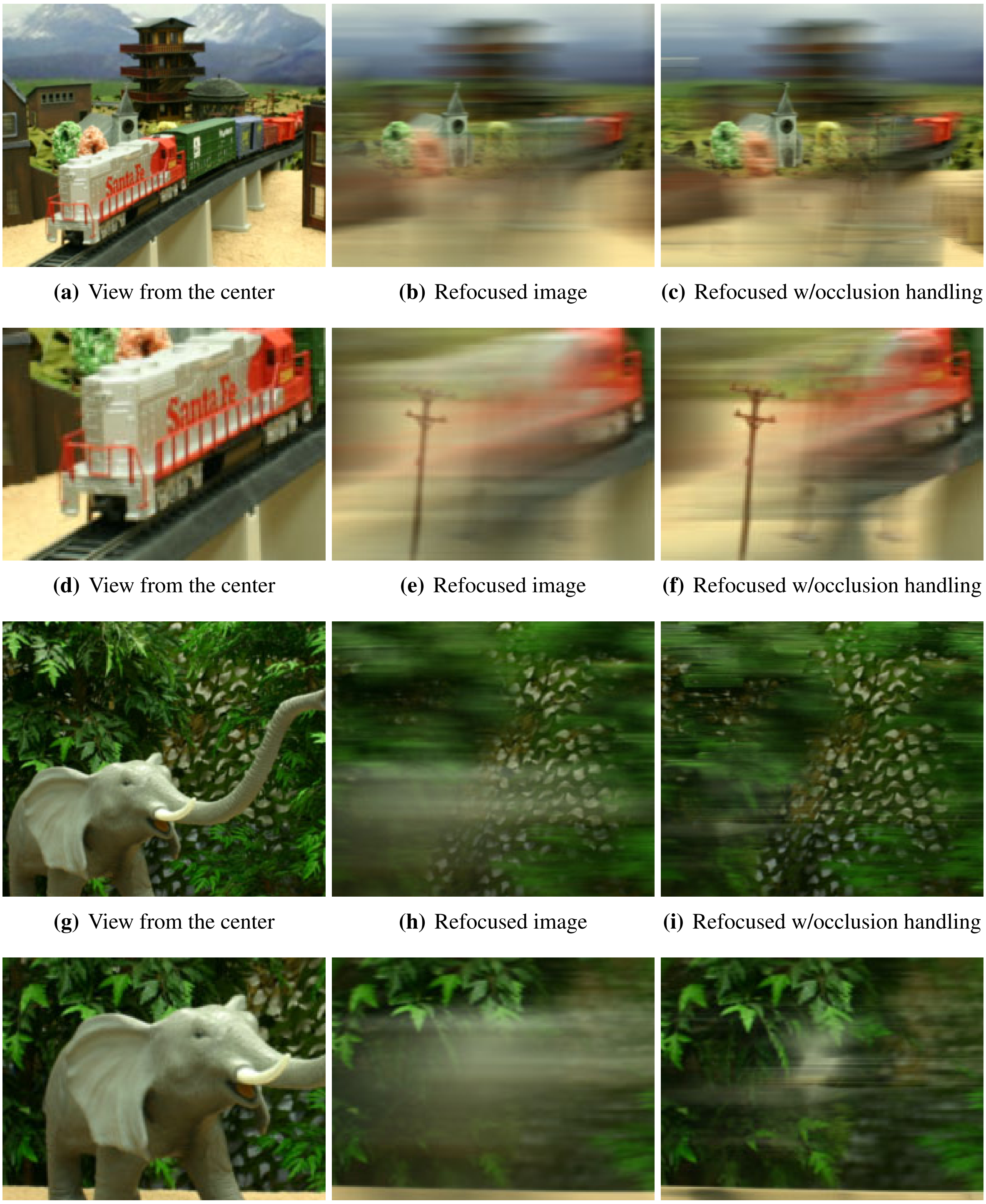
(f) The *xd*-slice indicated in (b) of the 3D scene volume using the *volume* reconstruction method

Figure 5.11: 3D volume reconstruction. *xd*-slices of the reconstructed 3D volume. First row: 2D depth maps of the "train" and "elephant" data sets, where the *y*-coordinates of the *xd*-slices are indicated as a red line. (a), (c), and (e) are from the "train" data set, and (b), (d), and (f) are from the "elephant" data set. Second row: The *xd*-slice of the reconstructed 3D volume from the *depth* reconstruction algorithm (Algorithm 4.1), where only a single depth layer for each *x*-coordinate is reconstructed. Third row: The *xd*-slice of the reconstructed 3D volume from the *volume* reconstruction algorithm (Algorithm 4.4), where multiple depth layers are reconstructed as well as the scene depth is more faithfully reconstructed.



(a) 2D depth maps **(b)** 2D depth maps extracted from 3D occupancy maps

Figure 5.12: 2D depth map vs. 3D occupancy map. The comparison between 2D depth maps directly computed from the depth reconstruction algorithm (left column) and 2D depth maps extracted from 3D occupancy maps (right column). For the captured data sets, the 2D depth maps extracted from the 3D occupancy maps show smoother surfaces, but more holes. For the synthetic data set, however, the surfaces of the single shot 2D depth map look better.



(a) View from the center

(b) Refocused image

(c) Refocused w/occlusion handling

(d) View from the center

(e) Refocused image

(f) Refocused w/occlusion handling

(g) View from the center

(h) Refocused image

(i) Refocused w/occlusion handling

(j) View from the center

(k) Refocused image

(l) Refocused w/occlusion handling

Figure 5.13: Occlusion-free refocusing. The first column shows the view from the camera position at center. The images in the second column were refocused *without* occlusion handling, whereas the images in the third column were refocused *with* occlusion handling using the depth peeling method (Algorithm 4.5). The church ((b) and (c)) and a pole ((e) and (f)) in the "train" scene are observed more clearly with occlusion handling. Especially, the pole is not visible at the given view point. In the "elephant" scene, the elephant can be almost removed from the scene ((i)) by refocusing with occlusion handling. Note that in (h), there is the trace of the elephant when refocused. However, the regions that are not visible from any camera position cannot be reconstructed. In (l), most of the elephant was removed, but there is a gray region at the center of images, which is completely occluded.

6

Conclusions

6.1 Summary

In this thesis, we have explored novel methods for the depth reconstruction of the scene captured in a light field. Digital refocusing with synthetic aperture was used as a main tool to build a scene representation called the focal stack. We employed the concepts of measuring the depth from differently focused images, taken from the depth from focus methods. These methods, however, did not handle the increased influence of occlusions due to the wide synthetic aperture. In light fields, we have access to all rays which are not yet integrated to form an image of the scene, enabling more flexible and powerful methods to determine the existence of surfaces.

Using digital refocusing, we first defined the focal stack from the light field as a mean to represent the scene. We then introduced focus measure functions to determine how likely there exists a surface, given a collection of rays to be focused at a point. The focus measures are computed for all points in the focal stack using one of such functions. Picking the points with the strongest measure along the depth, we can reconstruct the depth of the scene.

Due to the wide synthetic aperture, however, the effect of occlusions in the synthetic imaging is more noticeable and complicated than the conventional photography. With a limited visibility condition, it is often difficult to accurately measure the existence of surfaces. Therefore, we proposed novel approaches to attack this problem. First, focus measures based on selecting the rays not influenced by occlusions were presented. Those new focus measure functions are more robust to the occlusions. Then, we discussed how to systematically extract the occluding objects from the scene. This was approached twofold. The first was to remove occluding objects before focusing on objects behind them. The second was to partition the light field to make each partition occlusion-free to ease the depth reconstruction on each partition.

We then extended the depth reconstruction to the volume reconstruction. Each view-dependent depth map is warped to the common scene space, where the volume occupancy is marked. If a voxel is marked as a surface point enough times, then this voxel is accepted as occupied. Then again, a depth map can be constructed from the volume occupancy map by taking the closest occupied voxel to the camera. When we fail to reconstruct the scene in some regions due to the limited visibility or a large uniform area, we can try to extract the depth map from the volume occupancy, in the hopes of the area being better seen in some other view point.

We showed the effectiveness of our methods by the results presented in the previous chapter. The results from our methods showed good reconstructions of real and synthetic scenes. We presented a thorough evaluation of our methods and an analysis of the selection of parameters as well.

6.2 Limitations and Future Work

Although many approaches to reconstruct the scene from light fields were explored throughout the thesis, still there are areas to be further explored.

The obtained 3D occupancy map can be further refined by projecting the volume occupancy back into the light field as briefly described in Section 4.5. The residual error between the backprojected light field and the original light field can be obtained. Then, an optimization scheme can be applied to refine the volume occupancy in the direction to reduce the residual error, while enforcing some constraints on the volume occupancy. This capstoning step was not fully addressed in the thesis, but will be one of the direct extensions to our work.

Another adaptive ray selection scheme can be devised. In the ray selection method (Section 4.1.2) presented in the thesis, the acceptance rate of the rays is a fixed parameter. This limitation is relaxed in the ray clustering method (Section 4.1.3), so that any degree of visibility can be handled. However, the mean shift clustering algorithm we employed is an iterative nonlinear algorithm, and may be prohibitive for large scale data sets. Thus, an efficient adaptive algorithm is required for the method to be widely used in practice.

The light field partitioning discussed in Section 4.4.2 was limited in our implementation. It often suffers from the mixed rays that blend two or more object from different depth layers. Those rays arise mainly due to the limited discretization resolution. Thus, for the accurate partitioning of the light field, some method based on the alpha matting technique is required. The method has to deal with a multiple layer alpha matting, which is a challenging problem. This was not explored thoroughly in the thesis. Future work may be inspired by the previous work about the alpha matting using a camera array [JMA06].

Implementation-wise, the algorithms presented in this thesis are both computationally and spatially intensive. However, they are highly parallelizable, and well suitable for the current parallel computing framework such as GPU computing. Although some of the most time consuming parts already run on GPUs, more efficient implementation using GPU will greatly reduce the execution time. This is also left for future work.

We then want to pay attention to the remarkable similarity between the scene reconstruction from the focal stack and the reconstruction of a microscopic object from volumetric data. Microscopy deals with a similar type of problems with ours. Especially, in deconvolution microscopy [MKCC99], the volumetric representation of the space consisting of the specimen is acquired, and then the specimen is reconstructed from the acquired light polluted volume data. The point spread function of the optical device is first estimated, and then the volume data is deconvolved with the point spread function. As a result, the blurring caused by the “occluding” parts residing in the other focal plane is alleviated. There are similarities between the scene reconstruction and the deconvolution microscopy in the complex blur properties, the effect of occlusion, and the shape of the point spread function. In fact, the volumetric representation in the deconvolution microscopy is nothing more than a focal stack.

However, there is a crucial difference between the two. The specimen in the microscopy is translucent. Thus the volume can be integrated. However, most objects in the macroscopic world are opaque, and thus the light rays are blocked and scattered. If the opacity of the scene can be handled, the problem of the macroscopic scene will take the advantages of the microscopy techniques. The microscopy using the light field representation of the specimen was addressed in Levoy et al.’s work [LNA+06], but the opaque object was not handled.

In addition, computed tomography also has notable relation to our work as well as microscopy. The integration of rays over the light field is the same as the line integral used in tomographic reconstruction. Computed tomography also deals with the reconstruction from volumetric data. The Radon transform and the integration of the light field are only different in the parameterization of the projection line.

Finally, the methods described in this thesis can be applied to the light fields acquired by a lenslet based apparatus [NLB+05]. Then, hand-held cameras will be able to capture the depth image of the scene along with the familiar photograph.

Bibliography

[AB91] 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, 1991.

[AW92] E.H. Adelson and J.Y.A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):99–106, 1992.

[BBM87] R.C. Bolles, H.H. Baker, and D.H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[BI99] A.F. Bobick and S.S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200, 1999.

[Bro66] D.C. Brown. Decentering distortion of lenses. *Photogrammetric Engineering*, 32(3):444–462, 1966.

[Che95] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(8):790–799, 1995.

[CKS$^+$05] 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, 2005.

[CM02] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(5):603, 2002.

[CTCS00] J.X. Chai, X. Tong, S.C. Chan, and H.Y. Shum. Plenoptic sampling. In *Proceedings* *of ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 307–318. ACM, 2000.

[DHS01] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification*. John Wiley & Sons, 2001.

[DW88] T. Darrell and K. Wohn. Pyramid based depth from focus. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 504–509. IEEE, 1988.

[Ger39] A. Gershun. The light field. *Journal of Mathematics and Physics*, 18:51–151, 1939.

[GGSC96] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. The lumigraph. In *Proceedings of ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 43–54. ACM, 1996.

[Gro87] P. Grossmann. Depth from focus. *Pattern Recognition Letters*, 5(1):63–69, 1987.

[HK09] S.W. Hasinoff and K.N. Kutulakos. Confocal stereo. *International Journal of Computer Vision*, 81(1):82–104, 2009.

[IB94] S. Intille and A. Bobick. Disparity-space images and large occlusion stereo. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 179–186. Springer-Verlag, 1994.

[IMG00] A. Isaksen, L. McMillan, and S.J. Gortler. Dynamically reparameterized light fields. In *Proceedings of ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 297–306. ACM, 2000.

[JAMK07] N. Joshi, S. Avidan, W. Matusik, and D.J. Kriegman. Synthetic aperture tracking: Tracking through occlusions. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.

[JMA06] N. Joshi, W. Matusik, and S. Avidan. Natural video matting using camera arrays. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 25(3):779–786, 2006.

[Kro87] E. Krotkov. Focusing. *International Journal of Computer Vision*, 1(3):223–237, 1987.

[KS04] S.B. Kang and R. Szeliski. Extracting view-dependent depth maps from a collection of images. *International Journal of Computer Vision*, 58(2):139–163, 2004.

[KTOT95] A. Katayama, K. Tanaka, T. Oshino, and H. Tamura. Viewpoint-dependent stereoscopic display using interpolation of multiviewpoint images. In *Proceedings of SPIE*, volume 2409, pages 11–20, 1995.

[LCV$^+$04] M. Levoy, B. Chen, V. Vaish, M. Horowitz, I. McDowall, and M.T. Bolas. Synthetic aperture confocal imaging. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 23(3):825–834, 2004.

[LH96] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 31–42. ACM, 1996.

[LNA+06] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz. Light field microscopy. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 25(3):924–934, 2006.

[MKCC99] J.G. McNally, T. Karpova, J. Cooper, and J.A. Conchello. Three-dimensional imaging by deconvolution microscopy. *Methods*, 19(3):373–385, 1999.

[NLB+05] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Stanford University Computer Science Technical Report CSTR 2005-02*, 2005.

[NN90] S.K. Nayar and Y. Nakagawa. Shape from focus: An effective approach for rough surfaces. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 218–225. IEEE, 1990.

[PDTH89] A. Pentland, T. Darrell, M. Turk, and W. Huang. A simple, real-time range camera. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 256–261. IEEE, 1989.

[Pen87] A.P. Pentland. A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 9(4):523–531, 1987.

[SK00] Y.Y. Schechner and N. Kiryati. Depth from defocus vs. stereo: How different really are they? *International Journal of Computer Vision*, 39(2):141–162, 2000.

[SS94] M. Subbarao and G. Surya. Depth from defocus: A spatial domain approach. *International Journal of Computer Vision*, 13(3):271–294, 1994.

[VGT+05] V. Vaish, G. Garg, E.V. Talvala, E. Antunez, B. Wilburn, M. Horowitz, and M. Levoy. Synthetic aperture focusing using a shear-warp factorization of the viewing transform. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, page 129. IEEE, 2005.

[VSZ+06] V. Vaish, R. Szeliski, C.L. Zitnick, S.B. Kang, and M. Levoy. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2331–2338. IEEE, 2006.

[VWJL04] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy. Using plane + parallax for calibrating dense camera arrays. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2–9. IEEE, 2004.

[WJV+05] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 24(3):765–776, 2005.

[ZBA+07] R. Ziegler, S. Bucheli, L. Ahrenberg, M. Magnor, and M. Gross. A bidirectional light field - hologram transform. *Computer Graphics Forum*, 26(3):435–446, 2007.

[ZMDP06] M. Zwicker, W. Matusik, F. Durand, and H. Pfister. Antialiasing for automultiscopic 3D displays. In *Eurographics Symposium on Rendering (EGSR)*, 2006.