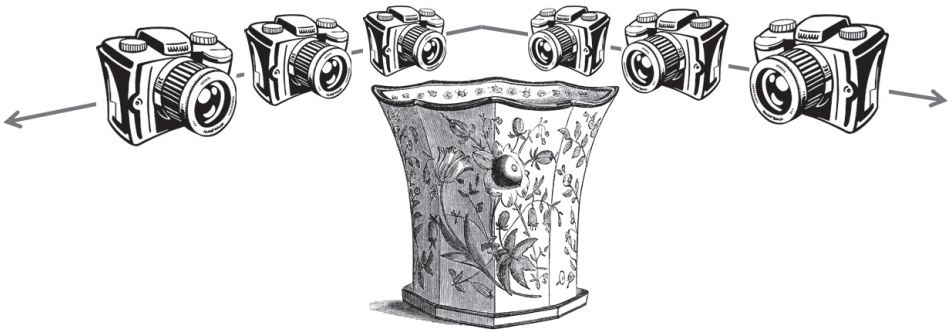
Plenoptic Layer-Based Modeling for Image Based Rendering



by
James Pearson

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

Communications & Signal Processing Group
Department of Electrical & Electronic Engineering
Imperial College London
2013

Statement of originality

I declare that this thesis, and the research it contains, is the product of my own work under the guidance of my thesis supervisors: Dr. Pier Luigi Dragotti and Mike Brookes. Any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline. The material of this thesis has not been submitted for any degree at any other academic or professional institution.

Copyright declaration

Abstract

Image based rendering is an attractive alternative to model based rendering for generating novel views due to its lower complexity and potential for photo-realistic results. In order to reduce the number of images necessary for alias-free rendering, some geometric information for the 3D scene is normally necessary.

Because the assumptions underlying Plenoptic theory are not fully met in practice, some aliasing is always present in real world examples. We will describe how we can mitigate these errors and achieve the performance predicted in plenoptic theory on real world data.

We will present a fast unsupervised layer-based method for synthesising arbitrary new view of a scene from a set of existing views. Our algorithm takes advantage of the knowledge of the typical structure of multiview data in order to perform occlusion-aware layer extraction. Moreover, the number of depth layers used to approximate the geometry of the scene is chosen using Plenoptic sampling theory. We further generalise this theory to allow the use of angled layers and multiple camera planes. The rendering is achieved by using a probabilistic interpolation approach and by extracting the depth layer information on a small number of key images.

Simulation results show that our method is only 0.25 dB away from the ideal performance achieved when having access to the ground truth pixel based geometric information of the scene and comparisons are also made to alternative methods. These results demonstrates the effectiveness of our method and the validity of the layer-based model.

Acknowledgement

There are many people I would like to thank for helping me through this PhD. First and foremost, I would like to thank my supervisors Pier Luigi Dragotti and Mike Brookes. I am eternally grateful for their unstinting support and guidance throughout my PhD and allowing me to to take advantage of the invaluable asset that is their combined wisdom. Pier Luigi has always known when to rein in my excesses to keep me on track, while Mike encouraged me to keep pursuing the crazy ideas that might just lead to something; together they make a superb team. I feel privileged that they were always willing to take time out of their busy schedule to see me to discuss ideas; they have contributed immeasurably to my work and my growth as a researcher. I would also like to thank them for their patience (especially when writing journals and this thesis) and enlightening me to the fact that spagetti should never be served with bolognese. It has been my privilege and pleasure to work with both of you, thank you.

I would like to thank all my friends from the CSP group who have made the lab a fun and enjoyable place to work, especially Sira Gonzalez[2] and Daniel Jarrett who helped keep me sane. I would also like to thank the Rabin Ezra scholarship fund for their support in my first two years.

For their support and encouragement every step of the way I would like to thank my family. Last and by no means least I would like to thank In San, for putting up with me and supplying me with victory biscuits.

Contents

List of Figures

Abbreviations

FAST . . . . . Features from Accelerated Segment Test

GT . . . . . Ground Truth

EPI . . . . . Epipolar Planar Image

MSC . . . . . Minimum Sampling Criterion

FOV . . . . . Field of View

RGB-D . . . . . Red, Green, Blue and Depth

IR . . . . . Infra Red

2D . . . . . two dimensional

3D . . . . . three dimensional

FV-TV . . . . . Free Viewpoint TV

IBR . . . . . Image Based Rendering

DG . . . . . Disparity Gradient

ELV . . . . . EPI Line Volume

3D-TV . . . . . three dimensional TV

SAD . . . . . Sum of Absolute Differences

PSNR . . . . . Peak Signal to Noise Ratio

CSS . . . . . Colour and Spatial Segmentation

PDSNR . . . . . Peak Disparity Signal to Noise Ratio

MSE . . . . . Mean Squared Error

MS . . . . . Mean Shift

GC . . . . . Graph Cut

SL . . . . . Structured Light

SD-SC . . . . . Single Depth Single Colour

MD-MC . . . . . Multiple Depth Multiple Colour

DIBR . . . . . Depth Image Based Rendering

SLAM . . . . . Simultaneous Location And Mapping

CODAC . . . . . Compressive Depth Acquisition Camera

CCTV . . . . . Closed Circuit TV

Symbols

i . . . . . Pixel position in camera plane - horizontal distance from optical axis

j . . . . . Pixel position in camera plane - vertical distance from optical axis

V_X . . . . . Camera position in X axis

V_Y . . . . . Camera position in Y axis

Z_{min} . . . . . Minimum depth within a scene

Z_{max} . . . . . Maximum depth within a scene

d . . . . . Disparity value

S_n . . . . . The n^{th} segment of the image

g . . . . . Disparity gradient value

V_s . . . . . Synthesised camera position

P_p . . . . . A bracketing pixel intensity

V_s^- . . . . . Closest input camera position adjacent to the synthesised view in the negative direction

V_s^+ . . . . . Closest input camera position adjacent to the synthesised view in the positive direction

V_Z . . . . . Camera position in Z axis

$L_{\min}$ . . . . . Minimum required number of layers for alias free synthesis

ϕ . . . . . The angle between two camera planes about the axis specified by $\widehat{\boldsymbol{\vartheta}}$

g_l^+ . . . . . The upper limit of the layer occlusion boundary

g_l^- . . . . . The lower limit of the layer occlusion boundary

$\mathcal{P}$ . . . . . The Plenoptic function

λ . . . . . Wavelength of lightray

t . . . . . Time

Z . . . . . Z axis

f . . . . . Camera focal length

X . . . . . X axis

Y . . . . . Y axis

h . . . . . Reciprocal depth range within a scene

ω_X . . . . . Camera plane axis of the Fourier domain spectrum

u . . . . . Pixel spacing

ω_i . . . . . Image plane axis of the Fourier domain spectrum

ΔV_X . . . . . Camera spacing

$\mathcal{B}$ . . . . . Highest image bandwidth

u . . . . . Pixel spacing

l . . . . . Layer index

Z_l . . . . . Depth of layer l

g_l . . . . . Disparity gradient of layer l

$g_{\max}$ . . . . . Maximum disparity gradient value in the scene

$g_{\min}$ . . . . . Minimum disparity gradient value in the scene

M . . . . . Number of images in the dataset

N . . . . . Number of segments

K_n . . . . . Number of pixels in a segment n

k . . . . . Pixel index within a segment n

I . . . . . Input image

g_n . . . . . Disparity gradient assignment of segment n from non occlusion aware error minimisation

$\epsilon(S_n,g)$ . . . . . Matching confidence for a segment S_n when projected with disparity gradient g using a non occlusion aware method

I_m . . . . . The m^{th} input image in the set

$g^{(n)}$ . . . . . Proposed disparity gradient for segment n

V_m . . . . . Camera motion between the synthesised and key images

m . . . . . Image index in the dataset

L . . . . . The number of layers used to model the scene

$O_k^{(n)}$ . . . . . Visibility mask for $\bar{\epsilon}$

$\bar{g}_n$ . . . . . Disparity gradient assignment of segment n from occlusion aware error minimisation

$\bar{\epsilon}(S_n,g)$ . . . . . Matching confidence for a segment S_n when projected with disparity gradient g using an occlusion aware method

$\mathfrak{t}$ . . . . . Occlusion aware matching combination threshold

$\hat{g}_n$ . . . . . Disparity gradient assignment of segment n from combining both error minimisations

$P(S_n)$ . . . . . Perimeter length of segment S_n

$A(S_n)$ . . . . . Area of segment S_n

ϖ_n . . . . . Measure of how spidery a segment S_n is

Y . . . . . Luminance component of a pixel

C_r . . . . . First chrominance component of a pixel

C_b . . . . . Second chrominance component of a pixel

$\epsilon_m(S_n,g)$ . . . . . Matching confidence map for a segment S_n when projected with disparity gradient g using a non occlusion aware method

$\bar{\epsilon}_m(S_n,g)$ . . . . . Matching confidence map for a segment S_n when projected with disparity gradient g using an occlusion aware method

$\bar{Y}$ . . . . . Scaled luminance component of a pixel

$\tilde{g}_n$ . . . . . Updated disparity gradient assignment of segment n based on surrounding segment values

$\eta(S_n, g)$ . . . . . The flattened assignment confidence for a segment S_n at disparity gradient g

$B(S_n, g_l)$ . . . . . The number of pixel assigned to layer g_l bordering Segment S_n

$\mathfrak{fj}(S_n, g)$ . . . . . The confidence response over different possible disparity gradients

ζ . . . . . Damping term for minimising depth discontinuities

$\bar{\eta}(S_n, g, k+1)$ . . . The damped flattening metric for a segment S_n at disparity gradient g based on previous iteration k

$\widetilde{DG}$ . . . . . Maximum possible disparity gradient value in the dataset

DGM_{GT} . . . . . The ground truth disparity gradient map

DGM_M . . . . . The ground truth disparity gradient map

I . . . . . Image width

J . . . . . Image height

β . . . . . Distance between EPI line and P_3

γ . . . . . Distance between the synthesised camera position and the closest input camera position adjacent to the synthesised view in the negative direction

α . . . . . Distance between EPI line and P_1

G_p . . . . . The disparity gradient for a synthesised pixel

p . . . . . A pixel

g_s . . . . . The disparity gradient for a synthesised pixel

g_p . . . . . The disparity gradient for a real pixel

$\mathcal{A}(p,g_l)$ . . . . . The alpha blending profile for a pixel at position p

p_l . . . . . The minimum distance of a pixel to the edge of its layer

$p_{\max}$ . . . . . The number of pixels extended for the layer

o . . . . . A pixel on the edge of a layer

$\mathbb{E}(g_l)$ . . . . . The set containing all pixels at the edge of layer g_l

$||\cdot||_2$ . . . . . The ℓ_2 norm

$B(p, g_l)$ . . . . . The number of pixel assigned to layer g_l bordering pixel p

$\tilde{I}$ . . . . . Maximum possible image value in the dataset

$^{(1)}$ . . . . . Superscript denoting that the axis or layer is in the first camera plane

$^{(2)}$ . . . . . Superscript denoting that the axis or layer is in the second camera plane

i' . . . . . Shifted pixel position in camera plane

j' . . . . . Shifted pixel position in camera plane

$\mathbf{K}$ . . . . . The camera matrix

f_i . . . . . Camera focal length in i dimension

$\bar{\imath}$ . . . . . Optical centre of the image in i

f_j . . . . . Camera focal length in i dimension

$\bar{\jmath}$ . . . . . Optical centre of the image in i

$\mathbf{R}$ . . . . . The rotation matrix

$\widehat{\boldsymbol{\vartheta}}$ . . . . . The unit vector specifying the axis of the intersection of the two camera planes

$\widetilde{\vartheta}$ . . . . . The antisymmetric matrix form of $\widehat{\boldsymbol{\vartheta}}$

ϑ_X . . . . . The X axis component of $\widehat{\boldsymbol{\vartheta}}$

ϑ_Y . . . . . The Y axis component of $\widehat{\boldsymbol{\vartheta}}$

ϑ_Z . . . . . The Z axis component of $\widehat{\boldsymbol{\vartheta}}$

$\dot{\boldsymbol{\vartheta}}$ . . . . . The fixed unit vector specifying the Y axis as the camera plane intersection

σ . . . . . The angle between two camera planes

$\Upsilon(\Delta V_X, \phi, l^{(1)}, l^{(2)})$ The switchover point for the occlusion ordering of $l^{(1)}$ and $l^{(2)}$

$l^{(1)}$ . . . . . A layer from the $^{(1)}$ plane model

$l^{(2)}$ . . . . . A layer from the $^{(2)}$ plane model

$\bar{\varrho}(S_n, i_k^{(n)}, g_l)$ . . . . An angled disparity gradient for segment n, where the disparity for a pixel i is dependent on its position and the assigned layer l

$\hat{\varrho}_n$ . . . . . Angled disparity gradient assignment of segment n from combining all error minimisations

$\breve{\epsilon}\,(S_n,\varrho)$ . . . . . Matching confidence for a segment S_n when projected with angled disparity gradient ϱ using an occlusion aware method

$\varrho(S_n, i_k^{(n)}, g_l)$ . . . . An angled disparity gradient for segment n, where the disparity for a pixel i is dependent on its position and the assigned layer l

Chapter 1

Introduction

1.1 Motivation

From Magic Lanterns in the 17th century, the Thaumatrope (Fig. 1.1(a)) of the early 19th century, the silent black and white films of the Cinematographe (Fig. 1.1(b)) of the late 19th Century, to the 'talkies' of the early 20th and the glorious full colour extravaganza of todays cinema, visual media is in constant flux, surging forwards on a wave of technology and pulling consumer expectations along with it. We are currently

(a) A Thaumatrope (b) A Cinematographe

Figure 1.1: The Thaumatrope (a) was a Victorian toy that showed a simple animation by spinning a disk whereas the Cinematographe (b) was a complete system capable of recording and playing film back.

undergoing yet another transition, from two dimensional (2D) to three dimensional (3D) content and displays. Users are demanding greater immersion and there has been

an explosion of 3D technology used in films, TV and even consumer devices. However for many people this is not enough. Although properly used 3D display technology can draw people into a scene it still has its limitations: there is no interaction and a viewer is tied to the whims of a director. The next evolution will be Free Viewpoint TV (FV-TV) allowing viewers to immerse themselves fully in the experience, giving them the freedom to choose where they look, finally invoking the feeling of 'being there'.

As cameras and processors grow cheaper and more powerful, it becomes feasible to deploy large numbers of cameras and treat the entire array as a single sensor. To do this, we require fast and robust algorithms that can combine the camera outputs to create high quality images from arbitrary viewpoints.

1.2 Problem statement

View synthesis is the process of generating an arbitrary new view of a scene from a set of existing views. One approach to view synthesis is to create a textured 3D model, for example [1, 2], of the entire scene and to use this for synthesising new views. This approach allows freedom in the final rendering but creating the complex 3D model in the first place can often be computationally intensive. Moreover, the synthesised output images, in particular for cluttered scenes, are often noticeably artificial. An alternative approach is Image Based Rendering (IBR) [3, 4], in which new views are generated by combining individual pixels from a densely sampled set of input images. This approach requires little geometric information and can give potentially photo-realistic results but requires many more input images [5, 6]. These two approaches can be thought of as opposite extremes of a spectrum where a reduction of one resource, geometric completeness, requires a corresponding increase in another, the number of images, to maintain a consistent quality.

Plenoptic sampling theory [7, 8] gives us a theoretical framework to understand this trade-off. In particular, Plenoptic sampling shows that, in the absence of occlusions, the number of views necessary for alias free rendering does not depend on the geometrical complexity of the scene but only on the depth variation within the scene [9]. Conse-

quently, a layer-based representation [10–12], in which the scene is split into separate depth layers each with a reduced depth range, is an effective way of introducing a variable amount of geometric complexity to allow accurate view synthesis from a moderate number of input images. In particular, the trade-off between geometric information and rendering quality reduces, in this way, to a trade-off between the number of images, the depth variation within the scene and the number of layers. A layer based model also has other advantages including implicit occlusion ordering and scalability.

The direct application of Plenoptic sampling theory to IBR relies on several assumptions which include the absence of occlusions, an infinite field of view and a perfect reconstruction filter. These assumptions are often not met in real world examples but Plenoptic theory is still useful as a guide. This has been shown for example for cases where many of the assumptions hold true with small, [13], and large, [14], numbers of input images. However this connection has yet to be shown for complex scenes with occlusions and multiple objects. The further a scene diverges from these assumptions the more aliasing occurs, understanding the cause of these errors allows us to mitigate their effect and use our resources as effectively as possible to achieve high quality rendering within the guidelines set down by Plenoptic theory.

1.3 Original contributions

1.3.1 Connecting Plenoptic theory to the real world

We have shown that Plenoptic theory is a real and valid guide to determining the trade-off between the quality of the output versus the complexity of the geometry for complex real-world scenes. Although some of the key assumptions are no longer valid most of its prediction remain true, provided the IBR method deals with the inevitable consequences of the divergence from the assumptions.

1.3.2 Scene adaptive layer extraction algorithm

In this thesis we present a fast automatic algorithm for IBR from a set of input images where Plenoptic sampling theory is used as a guide to the required number of layers for

alias free rendering. The layer positions are then selected non-uniformly to take advantage of the distribution of objects within the scene. The algorithm handles occlusions effectively by performing the layer assignment in two non-iterative stages. Finally, the performance is improved by a post-processing step merging adjacent small regions with neighbouring layer assignments when appropriate.

1.3.3 Probabilistic view synthesis algorithm

The rendering is performed using a probabilistic interpolation method. Moreover we propose a method of using multiple depth maps in a master-slave approach that is effective and scalable. The overall algorithm scales naturally with the number of input images, can be adaptive in the choice of the number of layers and can be used on different camera arrays such as the EPI volume [15] or the Lightfield [16, 17].

1.3.4 Arbitrary virtual camera positions

Our algorithm is robust and flexible and can be expanded beyond the normal cases of a line or plane of input cameras. We have shown how the relaxation of the fronto-parallel layer constraint improves performance without a large impact on complexity and how this leads to a parametrisation of the scene via a series of connected camera planes allowing us to relax the position constraints for the output synthesis position.

1.3.5 Publications

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

J1 J. Pearson, M. Brookes, and P. L. Dragotti, "Plenoptic layer-based modelling for image based rendering," in *IEEE Trans. on Image Processing*, vol. Special Issue on 3D video, 2013. [18]

C2 C. Gilliam, J. Pearson, M. Brookes, and P. L. Dragotti, "Image based rendering with depth cameras: How many are needed?" in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 2012. [19]

C1 J. Pearson, P.-L. Dragotti, and M. Brookes, "Accurate non-iterative depth layer extraction algorithm for image based rendering," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Processing*, May 2011, pp. 901–904. [20]

1.4 Thesis outline

The thesis is structured as follows:

In Chapter 2 we discuss the core Plenoptic theory that underlies the work described in this thesis and review the literature in the area of IBR. Plenoptic theory is useful because it shows that alias-free rendering can be achieved with limited geometric information and input images, importantly it allows us to characterise the tradeoff between the density of cameras and the amount of geometry required.

We will describe the concept of the seven dimensional Plenoptic function and how it parametrises the rays emanating from a scene and how by making certain assumptions it can be reduced to a five dimensional form. We will describe the camera model and the geometric relationship between the five components of the reduced Plenoptic function, leading to the Epipolar Planar Image (EPI) line setup. We will investigate how spectral analysis of the EPI structure leads to the conclusion that alias-free synthesis is possible even with reduced geometry given certain conditions and these conditions are only related to the camera spacing and the depth range of the scene. We show how the layer model is a robust and effective option for representing the geometry within the scene and meshes well with Plenoptic theory.

Chapter 3 details our algorithm for layer extraction, extracting the right amount of geometry from the scene to allow us to perform the view synthesis, we cover the problems that arise and how we have solved them. The first step is to choose the number of depth layers required for our geometric model. We will then describe how we assign each pixel to one of these layers, introducing our methods for efficiently dealing with the effects of object occlusions and discuss our post-processing methods to improve the final Disparity Gradient (DG) map. Finally we evaluate all the proposed methods and improvements against the Ground Truth (GT) geometry.

In Chapter 4 we describe our view synthesis algorithm. To perform synthesis we need layer based geometry for all of the input images and the view to be synthesised. This geometry allows us to use the EPI line structure to interpolate a new image from existing images. As described previously, we calculate the layer models for a few key images and then use these to predict the geometry for all the other views. This chapter will show that the predictions made by Plenoptic theory hold true for real world scenes.

In Chapter 5 we describe how our algorithm can be expanded to allow greater freedom in our input and output camera positions by relaxing certain constraints. We will explain how multiple connected camera-planes can be modelled and detail the changes to the algorithm necessary to allow output camera rotation and movement outside of the camera plane. An essential part of this expansion is the relaxation of our assumptions about the fronto-parallel nature of modelling the scene which also significantly improves the synthesis quality. Importantly all of this can be achieved while still adhering to the conditions that allow us to use Plenoptic theory as a valuable guide. We will show how by relaxing our constraints not only do we allow more freedom in our output synthesis position and pose but we also improve synthesis quality.

Finally in Chapter 6 we summarise the achievements of the work, discuss our conclusions and present some possible future extensions of the approach.

Chapter 2

Image based rendering and the Plenoptic function

2.1 Introduction

We will start this chapter, in Sec. 2.2, with an overview of Plenoptic theory, explaining why it is important to our Image Based Rendering (IBR) approach. We will then move on to review, in Sec. 2.3, some of the current approaches to IBR that have inspired us.

Plenoptic theory is a way of parametrising the visual world around us by considering the light rays emanating from the scene rather than the objects themselves. By using Plenoptic theory we can frame the IBR question in terms of a more traditional sampling and interpolation problem where new images are generated by interpolating between existing images which can be considered as samples of the Plenoptic function.

Plenoptic sampling theory is important for IBR because it gives us a theoretical framework to understand the tradeoff between geometric completeness and the number of images necessary to maintain a consistent quality. In particular, Plenoptic sampling shows that, in the absence of occlusions, the number of views necessary for alias free rendering does not depend on the geometrical complexity of the scene but only on the depth variation within it, as will be shown in Sec 2.2.1.

Consequently, a layer-based representation, detailed in Sec 2.2.2, where the scene

is split into separate depth layers each with a reduced depth range is a good model of many scenes and lends itself to Plenoptic theory. In particular, the trade-off between geometric information and rendering quality reduces, in this way, to a trade-off between the number of images, the depth variation within the scene and the number of layers. A layer based model also has other advantages including implicit occlusion ordering and scalability.

2.2 The Plenoptic function

A convenient way of regarding a multiview image set is to consider the collection of light rays emanating from the scene. The complete seven dimensional parametrization of the rays at any position and time is known as the Plenoptic function, introduced by Adelson and Bergen [21]. It expresses the intensity, $\mathcal{P}$, of a light ray as

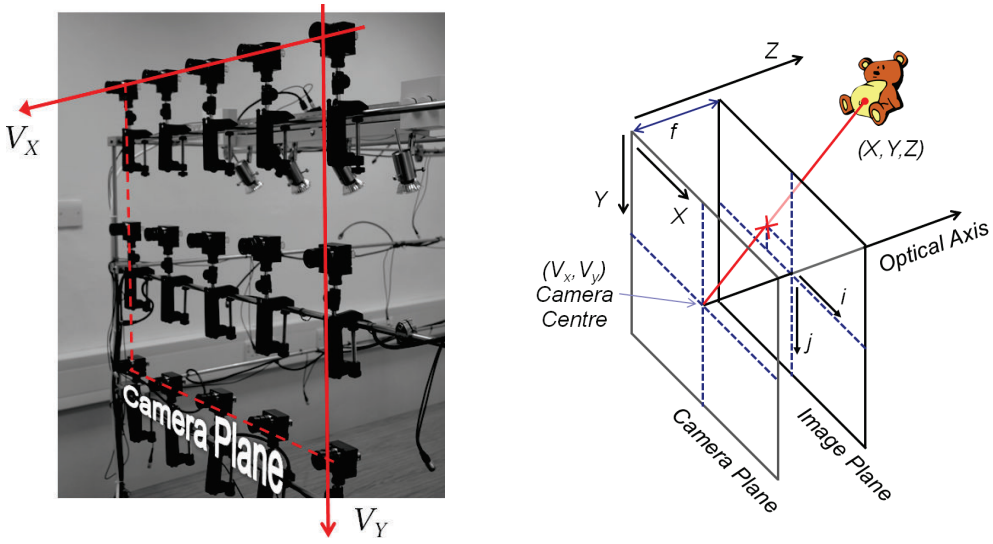
$$\mathcal{P} = \mathcal{P}_7(i, j, \lambda, t, V_X, V_Y, V_Z), \tag{2.1}$$

in which λ is the wavelength, t is the time, (V_X, V_Y, V_Z) is the position of the camera centre and (i, j) a point in the image. The dimensionality of the Plenoptic function can be reduced by imposing restrictions on the acquisition setup. Thus we can omit t for a static scene and we can eliminate λ by considering separate red, green and blue images. A convenient parametrization, the Light Field or Lumigraph, introduced in [16, 17], assumes the light ray intensity is constant along its length and the cameras are restricted to the plane $V_Z = 0$. It defines a light ray by the coordinates of its intersections with two parallel planes, the image plane (i, j) and the camera plane (V_X, V_Y). This leaves us with the four dimensional parametrisation,

$$\mathcal{P} = \mathcal{P}_4(i, j, V_X, V_Y). \tag{2.2}$$

In this thesis, we will assume that a static scene is sampled by an array of identical pinhole cameras whose optical centres lie on a camera plane perpendicular to their optical axes as illustrated in Fig. 2.1(a). We define a right-handed world coordinate

system with its origin at the optical centre of the upper left camera position and the Z-axis pointing towards the scene.



(a) Camera Array (b) Pinhole model

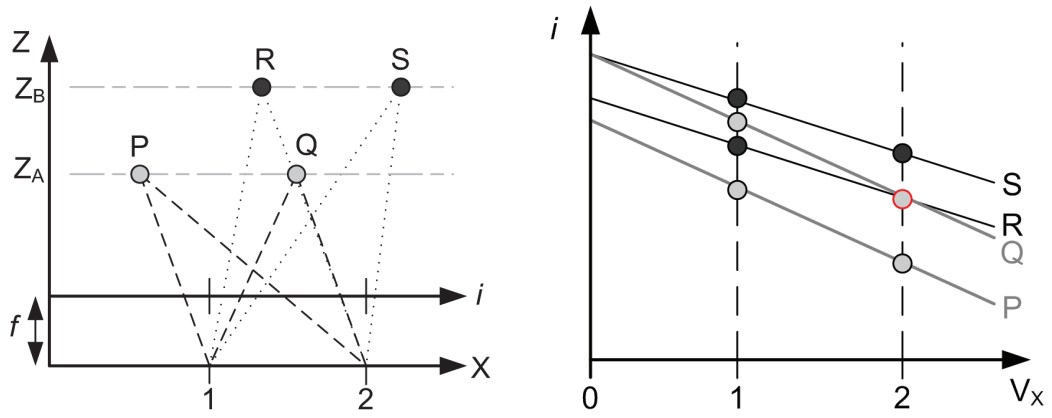
Figure 2.1: (a) Our array of cameras allows us to sample the Plenoptic function in the image, (i, j), and camera, (V_X, V_Y), planes. (b) The pinhole camera model of how the rays within a scene are captured by a camera, with the lens modelled as a single point, and the ray vector described as the intersection with two planes.

The geometry of the pinhole camera Lightfield is illustrated in Fig. 2.1(b). The camera centre location is (V_X, V_Y) on the camera plane which is separated from the image plane by the focal length f. The image plane for each camera has a separate coordinate system (i, j), centred on the optical axis. For a light ray that originates at point (X, Y, Z) in real world space and passes through the camera position (V_X, V_Y), the intersection with the image plane (i, j) is given by,

$$(i, j) = \frac{f}{Z}(X - V_X, Y - V_Y). \tag{2.3}$$

The Plenoptic function can be further simplified by fixing V_Y, thereby restricting the camera positions to a horizontal line. This set-up results in,

$$\mathcal{P} = \mathcal{P}_3(i, j, V_X), \tag{2.4}$$



(a) Layered view (b) EPI lines

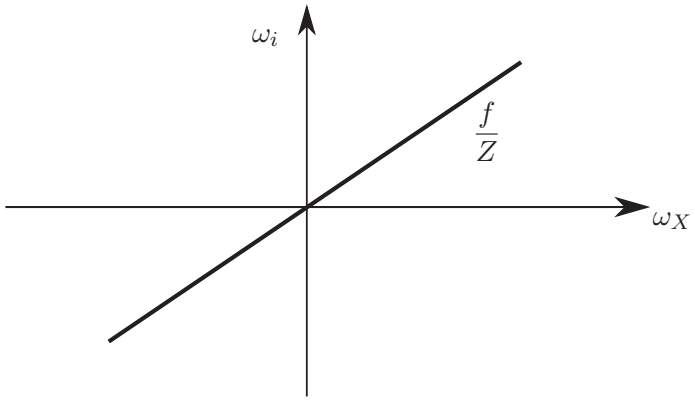
Figure 2.2: Four points at two different depths, Z_A and Z_B observed by a camera in positions $V_X = 1$ and $V_X = 2$, (a) shows the top down real world scene and (b) shows the EPI plot.

the three dimensional (3D) Epipolar Planar Image (EPI) line [15]. Figure 2.2(a) shows the view from above of four points in a scene, P, Q, R and S at two different depths, Z_A and Z_B, from the camera line. The figure shows the light rays from the four points that are received at two different camera positions $V_X = \{1, 2\}$. For the light rays from each of the four points to the camera, Fig. 2.2(b) plots i, the intersection with the image plane as a function of the camera position, V_X. The locus corresponding to each scene point is known as its EPI line [15]. Each EPI line has a constant gradient, the Disparity Gradient (DG) that is inversely proportional to the depth, Z, of its scene point; thus the lines corresponding to P and Q have a steeper gradient than those corresponding to R and S. From Fig. 2.2(a) we can see that when the camera is at $V_X = 2$, point Q occludes point R; this occlusion is predicted by the intersection of the EPI lines shown in Fig. 2.2(b) since lines with a steeper gradient occlude lines with a shallower gradient when they intersect. When we consider a full scene with many points and hence many EPI lines we call the whole an EPI Line Volume (ELV) [22, 23].

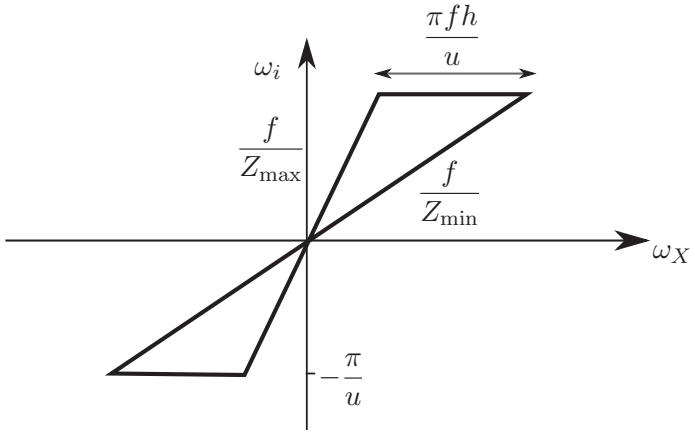
2.2.1 Plenoptic spectrum

In [7], Chai et al. use spectral analysis to investigate the EPI structure described above. The two dimensional Fourier transform of a line in the EPI domain is a line perpendicular to the original and with a gradient f/Z. This is shown in Fig. 2.3(a) for a

point at depth Z. In the more general case of a scene with varying depth, each point leads to a line in the EPI spectrum and all the line gradients are bounded by the minimum and maximum depths of points within the scene. For a scene comprising points with $Z_{\min} \leq Z \leq Z_{\max}$, we end up with a band-limited spectrum with a characteristic bow-tie shape support as shown in Fig. 2.3(b).



(a) Fourier transform

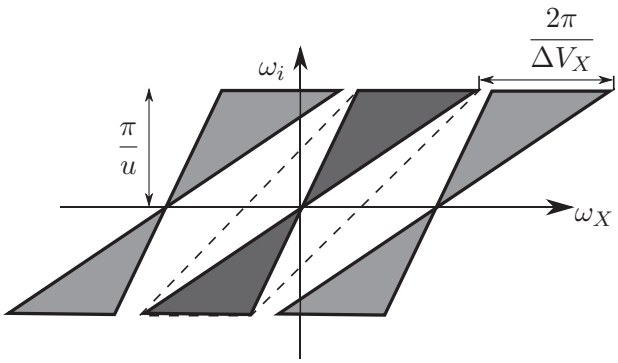


(b) Bow-tie bounding

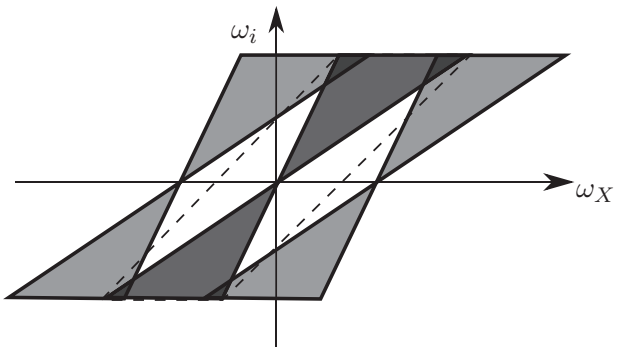
Figure 2.3: (a) Shows the Fourier transform of an EPI line. (b) Taking the minimum, $Z_{\min}$, and maximum, $Z_{\max}$, depths bounds the bundle of EPI lines into a characteristic bow-tie shape.

If the EPI is uniformly sampled with cameras spaced ΔV_X apart, the spectrum repeats at intervals of $2\pi/\Delta V_X$ in ω_X, as shown in Fig. 2.4(a), where u, the pixel spacing, determines the maximum unaliased frequency in the ω_i direction. An optimal reconstruction filter (dotted line) can be constructed around the fundamental section of the spectrum defined by $Z_{\max}$ and $Z_{\min}$. This allows us to pick a sufficiently low

camera spacing ΔV_X such that aliasing does not occur. If ΔV_X is made too large, aliasing will occur as the repeated spectra overlap; this is shown in Fig. 2.4(b).



(a) No aliasing



(b) Aliasing occurs

Figure 2.4: (a) Using an optimal reconstruction filter (dotted line) and a finite depth of field we can calculate a sufficiently small sampling spacing to avoid aliasing effects. (b) A higher ΔV_X leads to aliasing as parts of the repeated spectrum lie within the optimal reconstruction filter (shaded regions).

By combining the relationships shown in Fig. 2.3 and Fig. 2.4 we determine the maximum non-aliasing camera spacing [7] as follows:

$$\Delta V_X = \frac{1}{\mathcal{B} f h} \tag{2.5}$$

where $h = [1/Z_{\text{min}} - 1/Z_{\text{min}}]$ and $\mathcal{B} \leq 0.5/u$ is the highest image bandwidth given a pixel spacing u.

2.2.2 Layer model

The Plenoptic model describes a scene in terms of light rays emanating from points within a scene. A geometric model helps us describe and store the position of these points. One method of achieving this is a full 3D model in which every point has its own individually recorded position in (X,Y,Z). An alternative is a layer based model where the volume in which the points reside is partitioned into a set of constant-depth layers parallel to the camera plane and each point is assigned to the closest layer.

In this work we use a layer based geometric model because it is robust, offers a good description of many real scenes and is computationally efficient. Fig. 2.5 shows the layer model of a simple scene, where each surface point is projected along the Z axis onto the nearest layer to form a series of fronto-parallel planes. Associated with each layer l, at depth Z_l, is a unique DG,

$$g_l = \frac{d}{\Delta V_X} \tag{2.6}$$

$$= \frac{f}{Z_l}, \tag{2.7}$$

for a disparity shift d between the same scene point in two cameras with a spacing of ΔV_X.

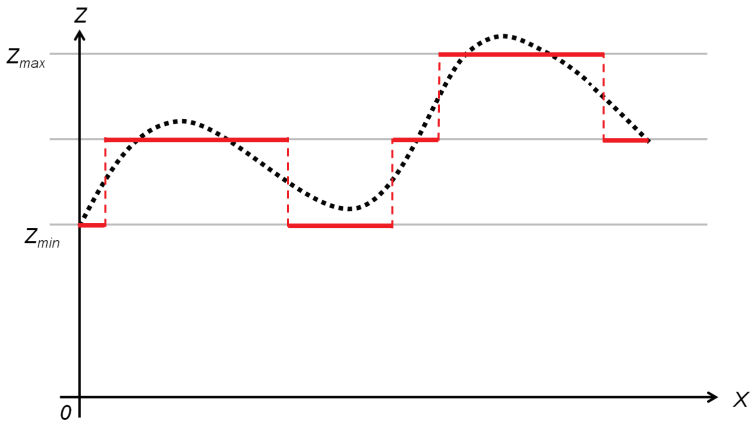


Figure 2.5: Layer model, each point in the continuous real world (dotted) is projected onto the nearest layer to give a series of planes (solid).

By partitioning the scene into layers, we can reduce the depth range within any given layer; this reduces h in (4) and therefore allows sparser sampling in V_X. Conversely if we have a fixed camera spacing, ΔV_X, we can determine the value of h that will result in alias free rendering. Assuming the layers are uniformly spaced in Z^{-1} with a pixel spacing of u, this allows us to determine the minimum number of layers,

$$L_{\min} = f\Delta V_X \mathcal{B} h \tag{2.8}$$

$$= \frac{f\Delta V_X}{2u}\left(\frac{1}{Z_{\min}} - \frac{1}{Z_{\max}}\right) \tag{2.9}$$

$$= \frac{\Delta V_X}{2}(g_{\max} - g_{\min}), \tag{2.10}$$

necessary for successful rendering without aliasing, known as the Minimum Sampling Criterion (MSC). This equation allows us to extract the best result for a given situation. Generally the range of Z for a scene will be constrained by the real world geometry, so if we are given a fixed camera spacing we can determine the optimal number of layers, or conversely if we have a fixed number of layers we can determine the corresponding maximum camera spacing.

We use this Plenoptic sampling framework to inform our layer extraction algorithm. In the initial stage of our algorithm we calculate $Z_{\min}$, $Z_{\max}$ and ΔV_X in order to determine the necessary $L_{\min}$. Since this computation can be performed on any number of input images, our algorithm allows us to adaptively modify the number of layers extracted as the visible scene depth range or camera spacing change.

2.3 IBR literature review

Two major areas of research within IBR that are of particular interest are the initial geometric model construction and the synthesis method. The major problems faced in the geometric assignment are the robustness and accuracy of the pixel assignment and dealing with any errors that appear, there are many different approaches to dealing with these problems. Many of these techniques can be split into two groups, working

at a pixel by pixel level or working with particular groups of pixels. Many different synthesis processes have been put forwards, the main problems they face are dealing with occlusions and rendering artifacts due to incomplete or erroneous geometric data.

We have also covered several useful multiview compression papers that have a different slant on the problem. In particular the benefits to designing the geometric model assignment and the synthesis as a mutually supporting pair that work in harmony with each other.

2.3.1 Geometric assignment

In a layer based system, each pixel in each input image needs to be assigned to a specific depth layer. This is normally achieved by matching points in two or more images and combining the pixel position shift and the camera position shift to obtain the depth of the pixel. Several methods have operated at a local pixel level, often with high speed (e.g., [24] and their accuracy can be improved by expanding the matching scope, for example by utilising a semi-global approach to improve the edge accuracy (e.g., [25]).

A popular alternative to a pixel based method is to assign depths to entire blocks of pixels [26]. Although more robust to noise and requiring a less iterative approach it introduces the problems of blockiness and poor reproduction of object edges. Various post-processing methods have been proposed to refine coarse depth geometry with reference to the original images [27, 28]. An alternative to dealing with the issue of matching object edges is through the use of a collection of sub-blocks processed together, as suggested in [29, 30], or the use of segments based on the image content rather than on a regular grid [31]. Although an initial segmentation step is required and some assumptions are made about the selected regions, there are several advantages to this approach as discussed by Zhang et al. [32, 33]. One is that it results in a higher robustness to noise, another is that it allows good edges to be formed without requiring a highly iterative approach. Segments have also been used to good effect to smooth out assignment noise from pixel based methods while preserving object edges, [34–36] or by comparing the results of adjacent segments [37]. An extension to the general segmentation method is over-segmentation [31, 38] or the use of high level

object segmentation [39–41] often based on human intervention [11]. Another approach to improving robustness is using structures within images [42] to help validate depth assignments by other methods.

2.3.2 Synthesis

There are many ways to use the layer model to synthesise new image views, various image surface warping techniques applied to the entire image have been proposed in [43–45]; although the resultant output is a complete image, it may be significantly distorted and often fails to fully take into account the occlusions and disocclusions inherent in the set-up. An alternative approach is a rigid layer shift accounting for the occlusion ordering on the layers, this models a scene more accurately but dissoclusion may lead to gaps in the final output which need to be filled as discussed in [46].

Depending on the type of rendering and the quality of the depth geometry a number of rendering artifacts can arise in layer based IBR. Various ways to mitigate these have been proposed such as enhancing depth geometry by using the images to refine the edges of layers, for example using weighted mode filtering [47], or merging multiple sets of geometry together [48]. One way to mitigate the effects of these artifacts is the use of alpha matting [11] to blend between layer boundaries, as most geometric artifacts will be most evident on the edges of layers.

One major, though inevitable, difficulty with the use of rigid layer shifts is the introduction of holes in the output image due to regions in the output image that are not visible in any input image. Several innovative approaches have been suggested to solve this for specific situations with varying degrees of complexity, for example [49–52]. Work has also been done to measure and predict the extent of errors in a system [53] to pick the particular approach to be used.

2.3.3 Multiview compression

Another popular and related field of study is compression schemes for three dimensional TV (3D-TV), [54,55]. The emphasis on absolute accuracy over speed or perceived quality may be different but many multi-view techniques are used to increase compression

performance by utilising the predictable geometric redundancy in multi-view video. Some of these approaches have utilised Plenoptic theory [56–58]. One area of particular interest is the accurate prediction of depth geometry using techniques such as boundary filters, [59], and Wavelets [60–62].

2.3.4 Similar work

Tong et al. [63] have investigated the trade-off between geometry and the number of input images. Their approach is similar in several respects to that taken in this thesis; these include the use of a layered geometric model and the combination of discrete input images to directly synthesise the output rather than using a pre-generated unified reference image model. However, [63] uses a stereo-matching algorithm to extract layers whereas we use a two-stage approach which allows us to handle occlusions effectively. Moreover, they have investigated situations in which the trade-off between geometry and number of images can have several optimal points and experimentally determine their validity. In contrast, we have used only a single operating point, as given by Plenoptic sampling theory, based on a fixed input image spacing, and have investigated the behaviour either side of this operating point.

2.4 Conclusions

In this chapter we have discussed the core Plenoptic theory that underlies our thesis and reviewed some of the key papers in the area of IBR. Plenoptic theory is useful because it shows that alias-free rendering can be achieved with limited geometry and input images, importantly it allows us to parametrize the tradeoff between the input image spacing and the amount of geometry required. Plenoptic theory is a good framework to understand IBR, but in practice it needs to be adapted to work with real world scenes.

We have introduced the concept of the seven dimensional Plenoptic function and how it parametrises the rays emanating from a scene and how, through certain assumptions, it can be reduced to a three dimensional form. We have described the camera model and the geometric relationship between the five components of the re-

duced Plenoptic function, leading to the EPI line setup. Spectral analysis of the EPI structure leads to the conclusion that alias-free synthesis is possible even with reduced geometry given certain conditions and these conditions are only related to the camera spacing and the depth range of the scene.

We have also shown how the layer model is a robust and effective option for representing the geometry within the scene and meshes well with Plenoptic theory.

Chapter 3

Layer extraction and assignment

3.1 Introduction

From Plenoptic sampling theory, Chapter 2, we know that it is possible to obtain an alias-free representation of a scene by representing its geometry as a set of fronto-parallel layers. There are also geometric based arguments, as mentioned in Chapter 2, that support this approach. This chapter is concerned with the choice of layer depths and the assignment of each input image pixel to a specific layer. We explained in Sec. 2.2.2 why we aim to use a geometric model comprising a finite number of layers and how Plenoptic sampling theory indicates the number of layers that are needed.

The theory shows that, provided certain assumptions are met, alias-free rendering can be achieved by spacing the layers uniformly in inverse depth and by using a number of layers that exceeds the minimum, $L_{\min}$, given in (2.10). In practice however, these assumptions, which include the absence of occlusions, an infinite field of view and a perfect reconstruction filter, are not fully met and some aliasing is inevitable. In Sec. 3.5.1, we will demonstrate that this residual aliasing distortion can be reduced by placing the layers closer together than the minimum spacing predicted by Plenoptic sampling theory. Conversely, if we fix the number of layers, the impact of the residual aliasing on rendering quality can be reduced by choosing the layer positions appropriately. Accordingly, our algorithm selects non-uniformly spaced layer positions according to the depth distribution of objects within the scene by increasing the density of layers at depths

that occur frequently while reducing the density at depths that occur infrequently.

Our view synthesis is dependent on the depth layer model generated from a collection of camera views. We assume that the only available inputs to the system are the input images and the required camera positions for the synthesised output images. For the sake of convenience and to simplify explanation we have assumed that the input images have already been rectified, [64]. We will initially describe the layer extraction and assignment for the algorithm for the case where the input camera positions are uniformly spaced along a line and extend this to the case of non-uniformly spaced cameras and planar camera arrays in Secs 3.2.2 and 3.3 respectively.

As discussed in Sec. 2.2, the required number of layers can be determined from the depth range, ($Z_{\min}$, $Z_{\max}$) of the scene. This can be calculated from a sparse estimate of the scene geometry, which is used to initialise the next step. At this point we diverge from the Pleoptic theory which suggests evenly spaced layers. For the case of a precisely bandlimited Plenoptic spectrum with an ideal reconstruction filter, this results in alias-free rendering with the minimum number of layers.

Because the assumptions underlying Plenoptic theory are not fully met in practice, some aliasing is always present. Because objects within a real scene are not uniformly distributed in depth there are advantages to assigning the output layers with uneven spacings. To do so a more detailed knowledge of the scene geometry is needed to assign layers to the best positions, this is discussed further in Sec. 3.2.4. Once the layer positions have been chosen we can assign each pixel within an image to a particular layer, this flat representation of the geometry is known as a Disparity Gradient (DG) map. This gives us a final version of the EPI Line Volume (ELV) assigned to the chosen layers that we then use to synthesise new views.

This chapter will show that the predictions made by Plenoptic theory hold true for real world scenes and explain what enhancements can be made to take advantage of properties of a particular scene. We will describe our novel method of picking layer positions and assigning pixels to these layers.

This chapter is organised as follows : The first part of the chapter, Sec. 3.2, covers our algorithm for layer extraction, problems that arise and how we have solved them.

An overview of the geometry estimation algorithm is presented in Sec. 3.2.1 and the estimation of the Z range of the scene is discussed in Sec. 3.2.2 along with estimating ΔV_X for the input cameras. We present the layer assignment method in Sec. 3.2.3, discuss the reasons behind our non-uniformly spaced layer scheme in Sec. 3.2.4, introduce our methods for efficiently dealing with the effects of object occlusions in Sec. 3.2.5 and elaborate on how we deal with planar camera setups in Sec. 3.3. In the second part of the chapter, Sec. 3.4, we discuss what methods we have used to enhance the effectiveness of our layer assignment. In Sec. 3.4.1 we discuss how we deal with problem segments that span multiple layers, in Sec. 3.4.2 we discuss our post-processing methods to improve the final DG map. Finally we evaluation of all the proposed methods and improvements against the Ground Truth (GT) geometry in Sec. 3.5 and present our conclusions in Sec. 3.6.

3.2 Layer extraction

3.2.1 Layer extraction algorithm overview

The goal of our algorithm is to take in a series of images as inputs and use these to construct sufficient geometry to allow us to synthesis high quality output images. The layer extraction and assignment is illustrated in Fig. 3.1 and described in detail below, comprises the following main stages:

A *Depth range estimation*: the depth range within a scene ($Z_{\min}$, $Z_{\max}$) and the camera spacing ΔV_X are found by examining the depth estimation of features within the scene, using all input images.

B *Disparity gradient histogram*: a more detailed estimate of the distribution of depths within the scene, bounded by the previously calculated $Z_{\min}$ and $Z_{\max}$, is obtained using all available input images.

C *Layer depth selection*: the distribution of depths estimated in step B is used to determine the detailed depth distribution estimate from the previous step is used to determine the optimum layer positions that will minimise the total error.

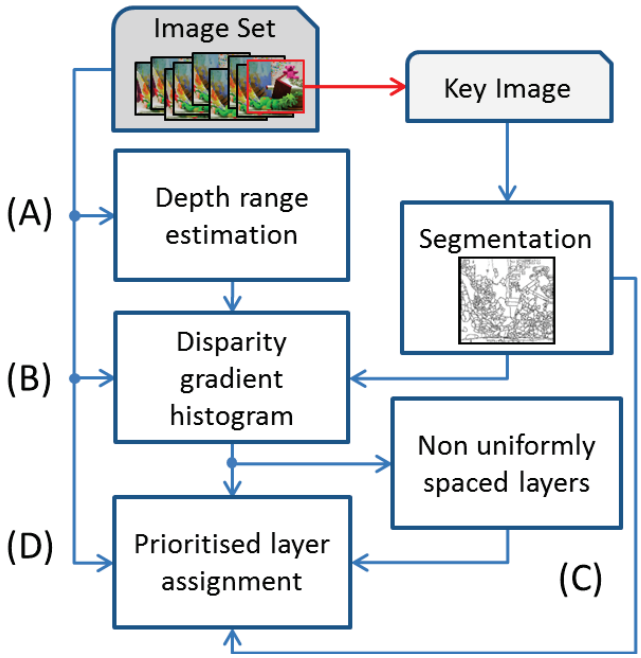


Figure 3.1: Flow diagram of the layer extraction and assignement algorithm. The main stages of the algorithm are (A) estimating the depth range of the scene (Sec. 3.2.2), (B) calculate an accurate disparity gradient histogram (Sec. 3.2.3), (C) assign the best layers using the Lloyd-Max algorithm (Sec. 3.2.4) and (D) assign segments to layers (Sec. 3.2.5).

D *Prioritised layer assignment*: pixels are assigned to layers in a single pass taking into account occlusions within the scene.

Although the algorithm can potentially compute a separate DG map for each available input image, we found that for all the sequences tested, the DG map only needs to be calculated for a small number of "key" images, typically two images. Using a larger number of key images increases the computational complexity but normally results in only a small improvement in the rendered images. We discuss the use and choice of the key images in more detail in Sec. 4.4.2.

Finally, the algorithm to the outlined above and discussed in detail in detail below is for the EPI case in which the camera positions lie along a line which, for rectified images, we take as the X axis. The extension for the more general case of camera motion in two dimensions is straightforward. When our input is a 2-dimensional camera array we can parallelise the calculation along the V_X and V_Y axes, as shown in Fig. 3.2, and then combine the results. An advantage of this approach is that we can also use the

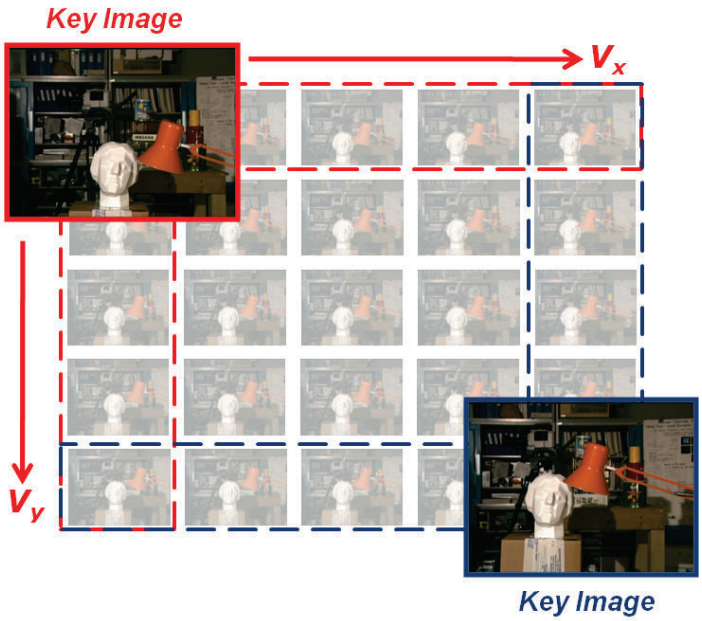


Figure 3.2: With a 2-dimensional camera array the EPI sets for a key image can be separately calculated along both V_X and V_Y axis in parallel with a shared key image. Calculations along both separate axis can then be combined for a more robust and accurate result. Shown here are two key images at (V_X, V_Y) = $(0, 0)$ and (V_X, V_Y) = $(4, 4)$.

same algorithm for many types of camera array. Choosing two EPI subsets from the camera array that intersect makes it possible for them to share a common key image to ensure consistent segmentation. Additionally by choosing two perpendicular EPI sub-sets we maximise the Field of View (FOV) diversity and hence the coverage of the scene.

For example, Fig. 3.2 shows the camera positions in a 5x5 planar array. If we select (V_X, V_Y) = (0, 0) as a key image, we would apply our linear algorithm separately to the horizontal line, $V_Y = 0$, and to the vertical line,$V_X = 0$. Selecting (V_X, V_Y) = (4, 4) we would apply our linear algorithm separately to the horizontal line, $V_Y = 4$, and to the vertical line, $V_X = 4$. This extension to two dimensions and the way in which we utilise the information from the extra dimension to improve matching robustness is covered in detail in Sec. 3.3.

3.2.2 Step A : Depth range estimation

The first stage of the algorithm is to determine the $Z_{\min}$ and $Z_{\max}$ for the visible scene. To achieve this aim as efficiently as possible we match a limited number of distinctive features between image pairs, we try all the available image pairs and combine the results. Features from Accelerated Segment Test (FAST) [65] are extracted from the key image and matched to an adjacent image using the pyramidal Lucas-Kanade feature tracker [66, 67]. The implementation for both these algorithms is taken from the OpenCV 2.4 library [68]. An example scene is shown in the left image of Fig. 3.3 (Image 0 from the Teddy sequence, see Table 3.1) and the positions of the extracted feature points is shown in the right image. Associated with each matched pair of features is a disparity, d, and we can form a histogram showing the distribution of these disparities.

The histogram of feature-point d between images 0 and 1 of the Teddy sequence is shown as the solid line in Fig. 3.4 (scaled by a factor of 8 for visibility). For comparison, the dotted line shows the corresponding histogram obtained for all pixels using the GT disparity. It can be seen that, although the two histograms are similar in shape, there are several noticeable differences. The most obvious of these is the large peak, Fig. 3.4(i), between the d values 3.8 - 4.2 which is only partially represented by a small spike in the FAST points at $d = 4$, in addition the spike at $d = 9.4$ is also missing, Fig. 3.4(ii). The reason for this difference is that, as can be seen in Fig. 3.3, the FAST points are not uniformly distributed in the image but cluster around distinctive features, e.g. region (H) in Fig. 3.3(H) and are sparse in low texture regions in the background (DG values 3.8 - 4.0) and the roof area, region (L) in Fig. 3.3, at DG = 9.4.

If, instead, we compare the disparity histogram for the FAST feature points with the GT disparity histogram of the same pixels, we obtain the graph shown in Fig. 3.5 where we see that the two histograms are very similar. Although there are not enough FAST points to determine the final layer positions robustly, we can reliably estimate $Z_{\min}$ and $Z_{\max}$ from this and move onto the next stage of the algorithm.

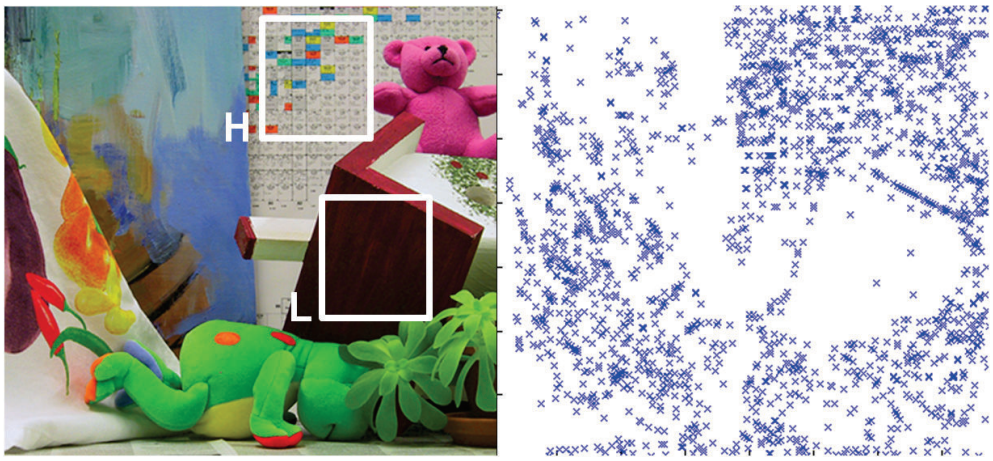


Figure 3.3: Teddy image 0 and the corresponding FAST features. The features are not uniformly distributed, there are (H)igh concentrations of points within highly textured areas and (L)ow concentrations within regions having little texture variation.

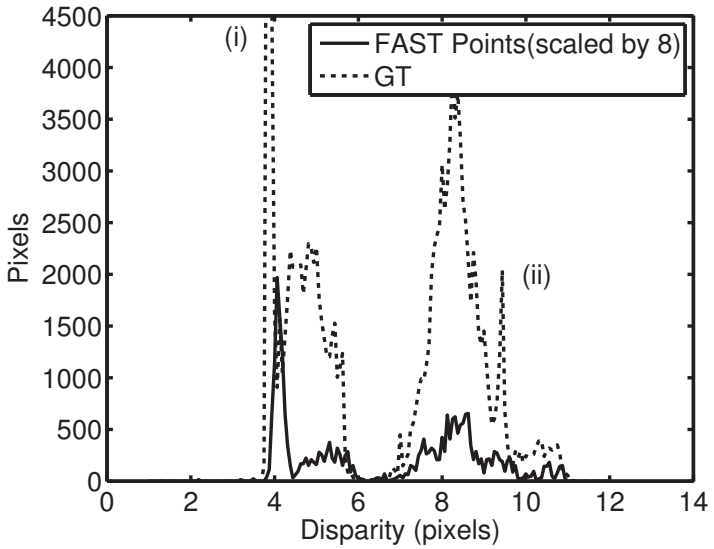


Figure 3.4: Comparison of the DG histograms for image 0 from the Teddy sequence; the ground truth (dotted line) and the FAST features (solid line scaled by a factor of 8). Peaks in the ground truth histogram that correspond to regions with few FAST points (e.g. at (i) $d = 3.9$ and (ii) $d = 9.4$) are missing from the FAST point histogram.

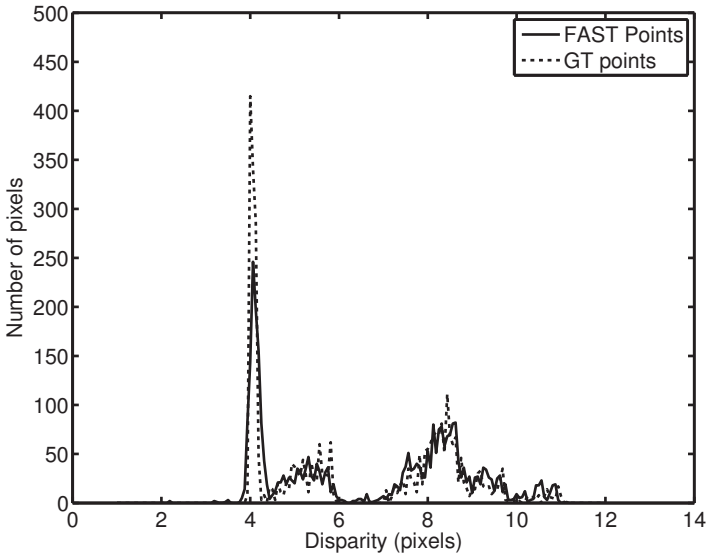
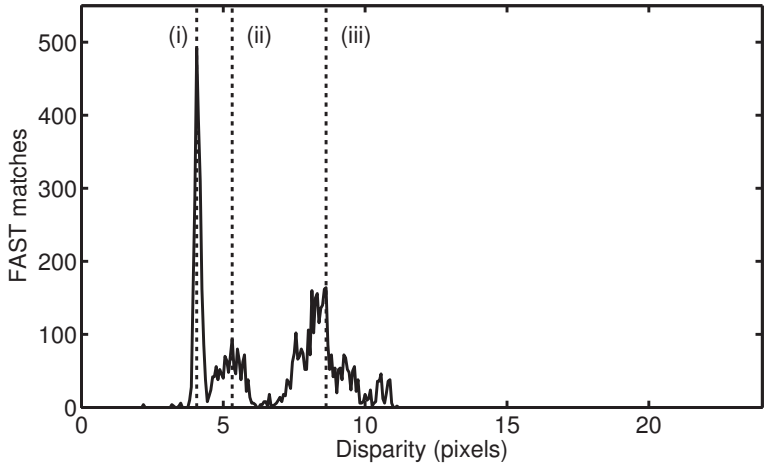


Figure 3.5: The solid line show the disparity gradient histogram for the FAST points, the dotted line shows the disparity histogram distribution for the ground truth at the same points.

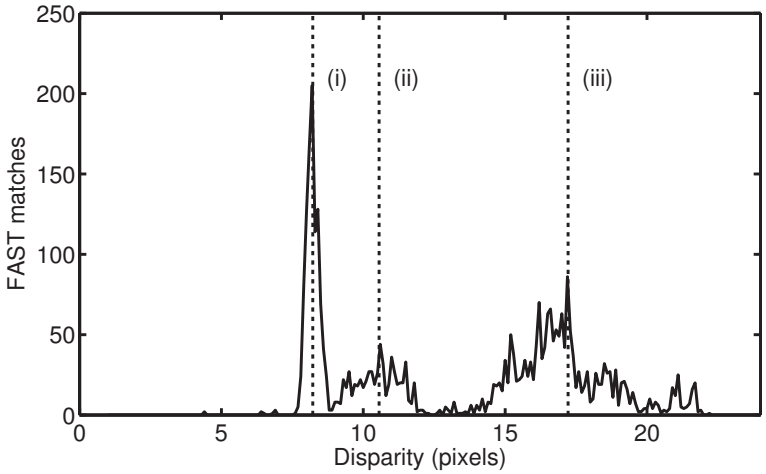
This method can be extended to deal with input images with unknown and possibly non-uniform camera spacing in V_X. The point disparities are calculated for an image pair as normal, as shown in Fig. 3.6(a) for the case of $V_X = 0 \Rightarrow V_X = 1$, then using the same points and the same initial image we calculate the point disparities to the next image, as shown in Fig. 3.6(b) for the case of $V_X = 0 \Rightarrow V_X = 2$. Because we are using the same points in each estimate we can work out the relative ΔV_X between the different images by comparing the change in disparities. If we look at three FAST points (i) - (iii) in $V_X = 0$, in Fig. 3.6(a) (i) $d = 4.063$, (ii) $d = 5.313$, (iii) $d = 8.625$ and in Fig. 3.6(b) (i) $d = 8.219$, (ii) $d = 10.56$, (iii) $d = 17.22$. The relative scale difference is 2.02, 1.98 and 1.997. If we calculate this for all the points (excluding outliers) we can get an accurate estimate for the relative ΔV_X between the images.

3.2.3 Step B : Disparity gradient histogram

Matching the features between images gives a good estimate for the DG range but a more detailed estimate of the scene disparities is needed to assign layers. Although we want an estimate of the DG, g, for each pixel we do not determine this on a pixel by pixel basis. Rather than assigning each pixel to a layer individually, we segment the



(a) $V_X = 0 \Rightarrow V_X = 1$. (i) $d = 4.063$, (ii) $d = 5.313$, (iii) $d = 8.625$



(b) $V_X = 0 \Rightarrow V_X = 2$. (i) $d = 8.219$, (ii) $d = 10.56$, (iii) $d = 17.22$

Figure 3.6: Disparity histograms for two pairs of images with different ΔV_X. In each case the first member of the pair is the same. The vertical dashed lines (i) - (iii) indicate the disparity of a particular pixel position in V_X.

images, using a 2D spatial and colour based procedure (eg. [69]), then assign entire segments to a particular layer. This has two advantages: it makes the algorithm more robust to noise and since object edges are normally aligned to segment boundaries, results in sharp and consistent edges.

For each segment in the image we need an estimate of the g with sufficient granularity, Δd, that we can project between the two furthest images in the sequence with an accuracy of one pixel so

$$\Delta d = \frac{1}{M-1} \tag{3.1}$$

where M is the total number of images in the sequence and $L_{\min}$ is the estimated minimum number of layers (2.10).

Since we assume that the images have been rectified, using for example [70], we know that correct feature point matches must lie on the same horizontal line so we can discount any features whose match shifts are not along the V_X axis. Matches can be consistent with this requirement yet still be incorrect. To account for this we can compare the estimate from several features within the same segment and if they agree we can conclude that there is sufficient evidence to estimate a particular g value for that segment.

As Fig. 3.7 shows the more features that are tracked within a segment the more reliable this method is. However as shown in Fig. 3.8 the more features we require, the fewer segments are valid. We found experimentally that a threshold of 10 feature points in a segment was a good compromise between the number of valid segments and the assignment reliability . The remaining segments are assigned using the following method.

We have an estimate for the $Z_{\min}$ and $Z_{\max}$ of the scene and hence their inverse relation $g_{\max}$ and $g_{\min}$. We need to calculate the best match for each remaining, unassigned segment within this DG range. For any given camera pair, with separation ΔV_X, we can calculate the expected disparity shift d of a segment with gradient g. We can evaluate the result of assigning a segment to a particular g and see how well

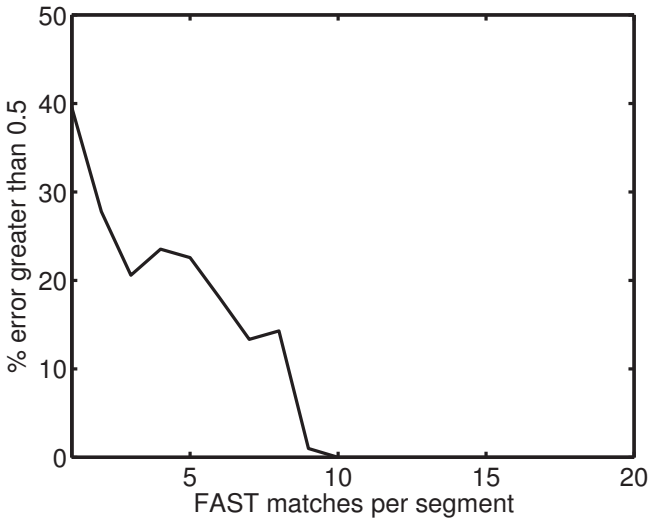


Figure 3.7: A graph showing for each number of FAST matches the percentage of segments with an assignment error of more than 0.5 pixel from the GT disparity.

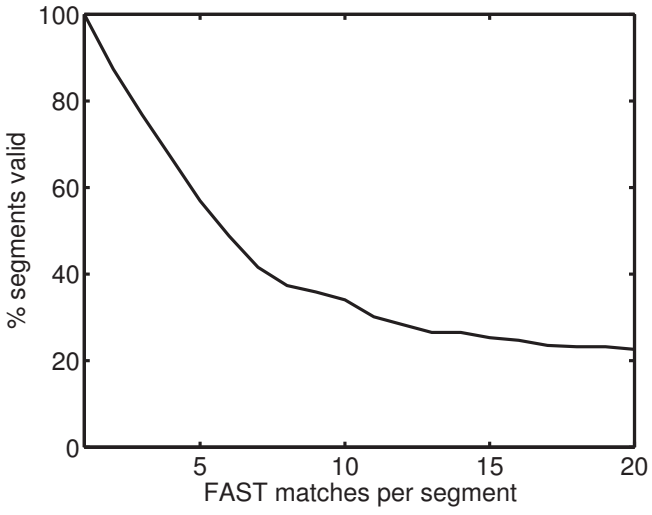


Figure 3.8: A graph showing the remaining percentage of segments as the required number of FAST matches is increased.

the predicted shift of a segment from the key image applies as a prediction for the other images. We sample the DG histogram uniformly between $g_{\min}$ and $g_{\max}$, with sufficient resolution to represent pixel-accurate disparities between the images of the most widely spaced cameras. Following [71] our matching metric is based on Sum of Absolute Differences (SAD) where we are trying to maximise the confidence function ϵ for each of the N segments. Each segment,S_n, contains K_n pixels each of which has a position index $(i_k^{(n)}, j_k^{(n)})$ for $0 \leq k < K_n$ within an image I_0. We select the layer assignments that will maximise the global ϵ for a scene so $g_n = \underset{g}{\operatorname{argmax}} \left(\epsilon(S_n,g)\right)$ where the matching confidence ϵ is

$$\epsilon\left(S_n, g\right) = \frac{M}{\sum_{k=0}^{K_n-1} \sum_{m=1}^{M-1} \left| I_0\left(i_k^{(n)}, j_k^{(n)}\right) - I_m\left(i_k^{(n)} + gV_m, j_k^{(n)}\right) \right|}, \tag{3.2}$$

where K_n is the total number of pixels within the segment S_n which is being evaluated over M images. I_0 is the current key image and I_m is the target image. $g^{(n)}$ is the proposed DG and V_m is the V_X position of image m so the ϵ value is a sum over all available images.

3.2.4 Step C : Non uniformly spaced layers

Previous authors [71] have selected layers that are uniformly spaced in disparity as suggested by Plenoptic theory. For the case of a precisely bandlimited Plenoptic spectrum with an ideal reconstruction filter, this results in alias-free rendering with the minimum number of layers. Because the assumptions underlying Plenoptic theory are not fully met in practice, some aliasing is always present and its impact on rendered output images can be reduced by increasing the layer density beyond that indicated by the theory. As will be shown in Sec. 3.5.1, the geometric modelling of the scene for a given number of layers can be improved by increasing the layer density at depths that occur frequently in the observed scene while decreasing it at depths that occur less often. If layers can be placed non-uniformly, the potential improvement in performance for a given number of layers is several dB, as will be shown for ground truth DG map data

in Sec. 3.5.1 and for image synthesis in Sec. 4.6.1.

This assignment requires some geometric knowledge of the scene, so we use the DG histogram from step B, shown in Fig. 3.9, and use it to assign the layers. We want to minimise the error from quantising disparities to the layer positions so the Lloyd-Max algorithm [72] with a quadratic cost function is used to find the values of g_l, the DG for layer l, where $1 \geq l \geq L_{\min}$.

The DG histogram for the Teddy sequence is shown in Fig. 3.9 with vertical lines showing the selected layer DGs when $L = 8$ layers are used. It can be seen that these cluster around the regions with a higher density of pixels, minimising the assignment error when using the layer model.

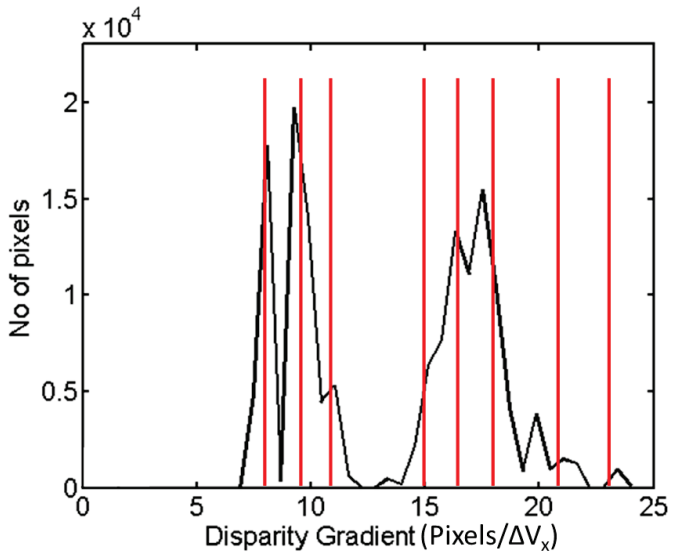
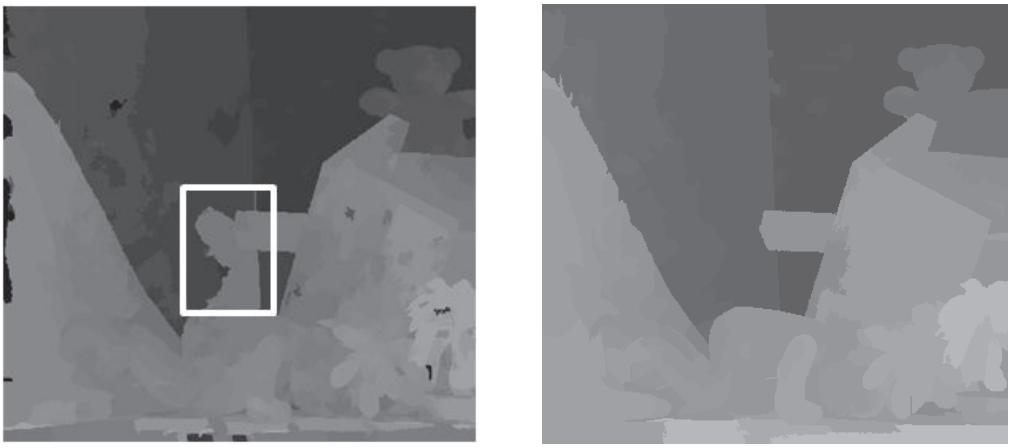


Figure 3.9: Disparity gradient distribution (black curve) for Teddy sequence with its associated DG layers (vertical red lines), where L is 8.

The use of non-uniform layer spacing represents a trade-off in which the aliasing error at frequently occurring scene depths is reduced at the expense of increased aliasing error at rarely occurring scene depths. This trade-off is controlled by the cost function used in the Lloyd-Max algorithm; we have found that the use of a quadratic cost function consistently gives the greatest improvement in Peak Signal to Noise Ratio (PSNR) on our evaluation sequences.

3.2.5 Step D : Prioritised layer assignment

We know from the Plenoptic theory that occlusions are hierarchical and predictable in that segments with higher g always occlude those with a lower g. We refine the DG assignment in a separate step, [73], initially analysing each segment in isolation (as discussed in Sec. 3.2.3) and then taking into account the predicted occlusions from surrounding segments to refine the initial estimate. The improvements can be seen in Fig. 3.10.



(a) Original DG map. (b) Refined DG map.

Figure 3.10: Using the prioritised segment assignment improves the accuracy of assignment for the whole DG map, especially for segments (marked) that are occluded by foreground objects.

Plenoptic sampling theory suggests that only a limited number of layers are required for alias free synthesis, so we can conduct the final occlusion-aware segment assignment using the layers calculated with the Lloyd-Max algorithm from Sec. 3.2.4 (eg. 8 layers shown in Fig. 3.9) with little loss of quality. We select the occlusion-aware layer assignments that will maximise the global ϵ for a scene so

$$\bar{g}_n \;=\; \underset{g}{\operatorname{argmax}} \left(\bar{\epsilon}\left(S_n, g\right)\right) \tag{3.3}$$

where the new matching confidence $\bar{\epsilon}$ is

$$\bar{\epsilon}\left(S_n, g\right) = \frac{M\left(\sum_{k=0}^{K_n-1} O_k^{(n)}\right) \log\left(\sum_{k=0}^{K_n-1} O_k^{(n)}\right)}{\sum_{k=0}^{K_n-1}\sum_{m=1}^{M-1} O_k^{(n)} \left| I_0\left(i_k^{(n)}, j_k^{(n)}\right) - I_m\left(i_k^{(n)} + gV_m, j_k^{(n)}\right)\right|}, \tag{3.4}$$

where $O_k^{(n)}$ is a visibility mask and

$$O_k^{(n)} = \begin{cases} 1 & \text{if } I_m(i_k^{(n)} + gV_m, j_k^{(n)}) \text{ is visible;} \\ 0 & \text{if } I_m(i_k^{(n)} + gV_m, j_k^{(n)}) \text{ is occluded.} \end{cases} \tag{3.5}$$

This matching metric is similar to (5.24), the main difference is that the effects of occlusions are modelled and occluded pixels are masked out, via the $O_k^{(n)}$, and are not included in the match. The numerator has been modified to account for the number of pixels considered to preserve the mean matching confidence measure. As the segments were previously matched independently without considering occlusions the disparity estimates were independent of the assignment order. However we can use the previous results to aid us in re-calculating the segment disparity in a more efficient manner.

The DG of each segment has already been provisionally assigned in step B of the algorithm (Sec. 3.2.3). In this second pass we process segments in order of decreasing DG, since a segment cannot be occluded by another segment with a lower DG. For each segment in turn, we determine $\bar{g}_n$ from (3.3) and also its matching confidence $\bar{\epsilon}(S_n,g)$. If $\bar{\epsilon}$ is less than a threshold, $\mathfrak{t}$, the segment is added to a cumulative occlusion map so that, for subsequent segments, the pixels it occupies will be excluded from the matching confidence calculation in (5.21). If, on the other had, $\bar{\epsilon} \leq \mathfrak{t}$ the segment's layer assignment is regarded as unreliable and it is omitted from the occlusion map. This process is repeated for each layer until $g_{\min}$ is reached. Segments with a poor matching confidence are ignored until the very end at which point they are then assigned using the most recent and complete occlusion map. The benefits of this prioritised procedure is that occlusions are estimated for all new assignments, rather than the

less accurate assignments of (5.24), and that unreliably assigned segments are ignored when estimating occlusions. We note that this prioritised approach does not increase the complexity of the method in that it only changes the order in which segments are tested but it does improve the quality of the occlusion map and hence the final reliability of the algorithm. The weighting in the SAD (5.21) is biased towards preferring larger segments whenever possible, so the increased reliability of large segments is reflected in the confidence metric.

As discussed previously in Sec. 3.2.2, a significant minority of the segments have sufficient feature tracking g estimates. To save computation they are not re-scanned, but merely assigned to the nearest layer.

The matching confidence $\epsilon\,(S_n, g_n)$ determined from (5.24) in step B (Sec. 3.2.3) will normally be lower than $\bar{\epsilon}\,(S_n, g)$ from 5.21.

The S_n error results are compared with a negative bias weight of 0.8 applied to ϵ results to give the final DG value of $\hat{g}_n$ where

$$\hat{g}_n \;=\; \underset{g}{\operatorname{argmax}}\left(\epsilon\,(S_n, g_n)\cdot 0.8,\;\; \bar{\epsilon}\,(S_n, \bar{g}_n)\right). \tag{3.6}$$

This is because although the occlusion aware assignment is generally more accurate and reliable in some cases, as shown in Fig. 3.11, it can give a mistaken estimate. If the DG response has a very defined peak that is between layers then the estimate might not be accurate. In this case even with the weighting the $\bar{\epsilon}$ would have a significantly higher error and g_n would thus be used in preference.

3.3 Layer assignment for 2D camera arrays

For the two dimensional (2D) camera array case the two intersecting camera lines are calculated separately and then combined afterwards. This combination is simple as the camera lines intersect at the shared key image camera, as seen in Fig. 3.2. This means that only one image needs to be segmented and that the matching error for each segment can be minimised in both directions. By choosing to use an additional camera

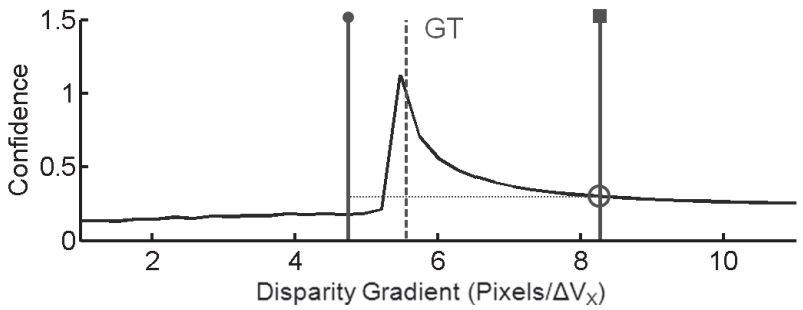


Figure 3.11: Due to the sparse nature of the refinement step when there are only a few layers local minima can cause miss assignment. In this example sampling at the closest layer (circular end) gives a worse result than a further away layer (square end).

line perpendicular to the first we maximise the diversity of the segment matching as some objects may be largely occluded or contain poor texture in a certain direction but these problems might not be apparent in the orthogonal direction. For example in

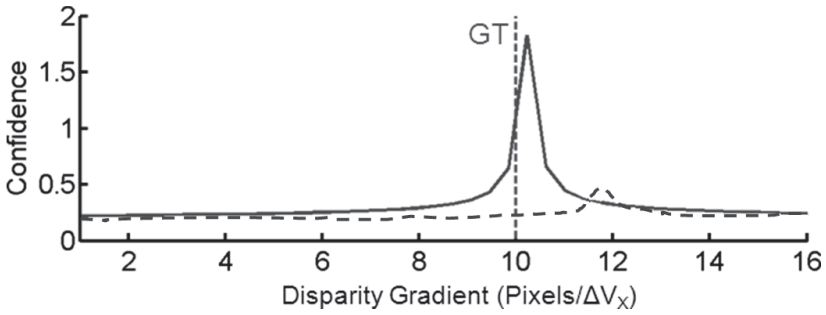


Figure 3.12: For this segment there is a small (incorrect) peak when matching along V_X (dashed line) but along V_Y (solid line) there is a distinct peak in the segment assignment confidence close to the marked GT.

Fig. 3.12 we look at the matching confidence (inverse error) of a segment for different potential g_n and we note that the confidence along V_X (dashed line) shows a small peak while that along V_Y (solid line) shows a large distinct peak which is closer to the GT. We have found that that the most robust and reliable improvement comes from choosing either one direction or the other based on the strength and sharpness of the peak, rather than combining and possibly exacerbating any errors. As both EPI sub-sets have the same key image, combining the results is very simple.

3.4 Layer enhancements

3.4.1 Section splitting

In scenes with lots of shadows and dark objects there is a risk that the shadows and the objects are segmented into a single segment. This sometimes leads to spidery 'legs' extending out from the main object. An illustration of this is shown in Fig. 3.13 where the black tripod in the foreground has wrongly been placed in the same segment as the dark shadows in the bookcase behind it. If we look at the underlying GT DG map, Fig. 3.14(b) we can see that there are two distinct layers that the segment covers. This is due to two issues, firstly the camera tripod is dark and very similar is colour to the surrounding bookcase shadows. In addition due to a restriction on the minimum segment size some small brighter regions surrounded by shadow have been absorbed.

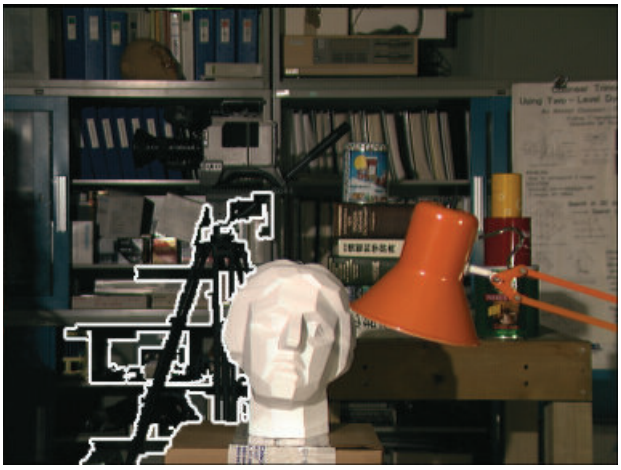


Figure 3.13: Spidered segment shown here highlighted with white border.

(a) Spidered segment spread between two layers (b) True DG map

Figure 3.14: Segment with many narrow splayed "spidered" outcrops.

We can identify potential examples where this has happened as segments with a high number of boundary pixels. We would like to subdivide these segments that are more compact. Our proposed solution is to split up the original segment if the spidered regions are on different layers.

3.4.1.1 Segment Identification

The ratio of the segment perimeter $\mathfrak{P}(S_n)$ and the area $\mathfrak{A}(S_n)$ is a good measure of any spidery segments as the 'legs' will increase the perimeter with little effect on the area. We use the dimensionless metric ϖ_n,

$$\varpi_n = \frac{(\mathfrak{P}(S_n))^2}{\mathfrak{A}(S_n)} \tag{3.7}$$

for each segment S_n leading to a ϖ map for the object, as shown in Fig. 3.15. A

Figure 3.15: ϖ map for the Tsukuba sequence, spidery segments are clearly visible.

dynamic threshold is used to filter out which segments are potentially spidered as the metric scales with segment size. Spidered sections can be excluded from the hierarchical patch assignment till the end as they are potentially unreliable.

3.4.1.2 Disparity Identification

Spidered segments are more likely to be spread over two or more layers so we analyse the disparity histogram to try and identify multiple potential disparities. If there is only one clear peak, such as in Fig. 3.11, then no further steps are taken. However

after we apply increasingly spread smoothing low pass filters to the result if two clear peaks are still visible as, shown in Fig. 3.16 we need to take further steps.

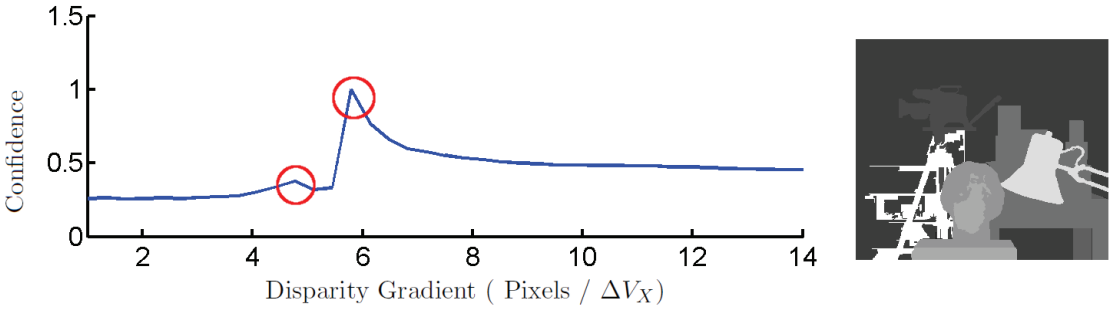


Figure 3.16: Analysing the ϵ distribution to detect multiple peaks using a combined V_X and V_Y, peaks are highlighted in red.

Scanning in both V_X and V_Y lets us find the two peaks clearly as shown in Fig. 3.17. If this is the case we need to split the segment into small segments that each lie on a single layer. This is covered in Seg. 3.4.1.3.

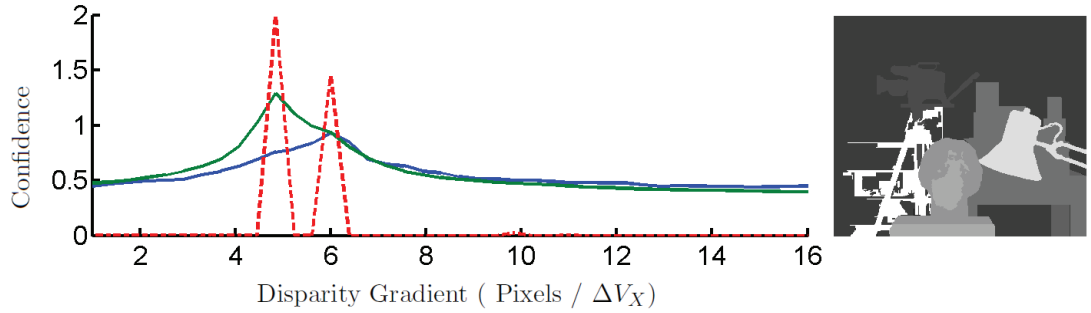


Figure 3.17: Analysing the ϵ distribution to detect multiple peaks using a separated V_X and V_Y.

3.4.1.3 Splitting method

In step B (Sec. 3.2.3) the key image I_0 was segmented using a Colour and Spatial Segmentation (CSS) algorithm,

$$CSS(S_n, i, j, Y, C_r, C_b) \tag{3.8}$$

using the spatial position (i, j) and colour information (Y, C_r, C_b) of each pixel for a segment S_n. If a segment is split over two layers the confidence map, $\epsilon_m(S_n,g)$ (5.24) individually for each pixel, is not evenly distributed as shown in Fig. 3.18. As discussed

(a) $\epsilon_m(S_n,g)$ along V_X (b) $\epsilon_m(S_n,g)$ along V_Y

Figure 3.18: Confidence map when $S_n = 84$ and $g = 3.75$ with no occlusions. Lighter indicated a higher confidence.

previously, Sec. 3.2.5 by taking into account occlusion in our model we can get a more reliable confidence measurement, $\bar{\epsilon}_m(S_n,g)$ (5.21) individually for each pixel, as shown in Fig. 3.19. We have previously determined the two layers that the segments lie on,

(a) $\bar{\epsilon}_m(g)$ (b) $\bar{\epsilon}_m(g)$ after applying low pass filter

Figure 3.19: Confidence map when $S_n = 84$ and $g = 3.75$ with occlusions. Lighter indicated a higher confidence.

in this case $g_1 = 3.75$ and $g_2 = 6$ so we can calculate the confidence for each pixel at these two layers and then use this instead of the two chroma components C_r and C_b for our CSS,

$$CSS(S_n, i, j, \bar{Y}, \bar{\epsilon}_m(g_1), \bar{\epsilon}_m(g_2)) \tag{3.9}$$

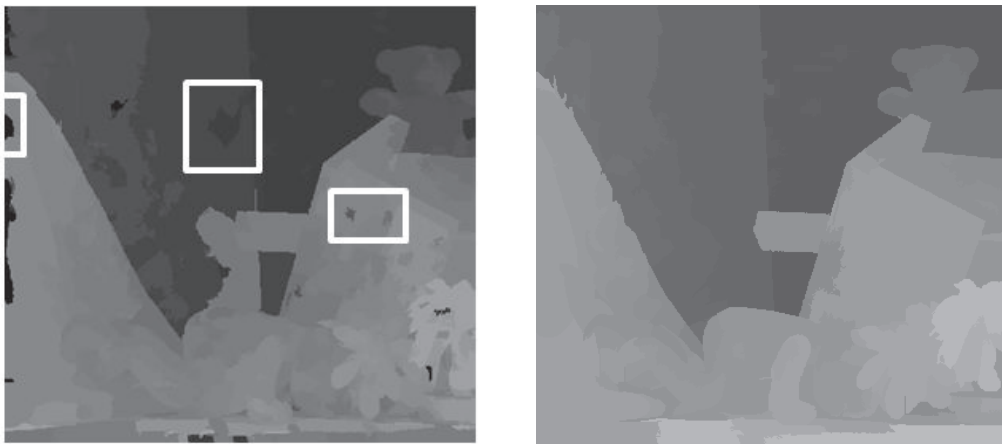
or the normalised form

$$CSS\left(S_n, i, j, \bar{Y}, \left(\frac{\bar{\epsilon}_m(g_1)}{\bar{\epsilon}_m(g_1) + \bar{\epsilon}_m(g_2)}\right)\right) \tag{3.10}$$

where $\bar{\epsilon}_m(S_n, g_1)$ and $\bar{\epsilon}_m(S_n, g_2)$ are the confidence match maps for disparity gradients $g_1 = 3.75$ and $g_2 = 6$ for segment S_n. $\bar{Y}$ is the scaled luminance value, which allows us to adjust the dependency of the metric to the error difference vs image information. The original segment is replaced by the new collection of segments.

3.4.2 Minimising depth discontinuities

The prioritised segment matching step described in Sec. 3.2.5 is effective in avoiding the types of errors shown in Fig. 3.10 where a segment is grossly mis-assigned due to an occlusion. Fig. 3.20 illustrates an example of a few types of error that are not resolved. Segments that are small and affected by frame occlusions or a segment wrongly assigned to a slightly different DG will cause a minor but unsightly artefact in the final synthesis. An additional step is required to deal with this issue.



(a) Original DG map (b) Enhanced DG map

Figure 3.20: Using the prioritised segment assignment and applying the flattening algorithm with an α of 0.4 and β of 0.01 per iteration allows us to deal with unassigned and slightly miss-assigned segments.

Previously we have discussed maximising the matching confidence to calculate $\hat{g}_n$ (3.6) for each segment, taking into account occlusions from other segments. To reduce

the artefacts in the final output we can include the g of the surrounding layers as a weight in the segment assignment maximising our new flattened assignment confidence η giving us a new estimate for the segment DG, $\tilde{g}_n$, where

$$\tilde{g}_n = \underset{g}{\operatorname{argmax}} \left(\eta(S_n, g) \right), \tag{3.11}$$

and the assignment confidence $\eta(S_n, g)$ (3.13) is dependent on both the highest confidence combined matching assignment $\hat{g}_n$ and the g of the surrounding segments.

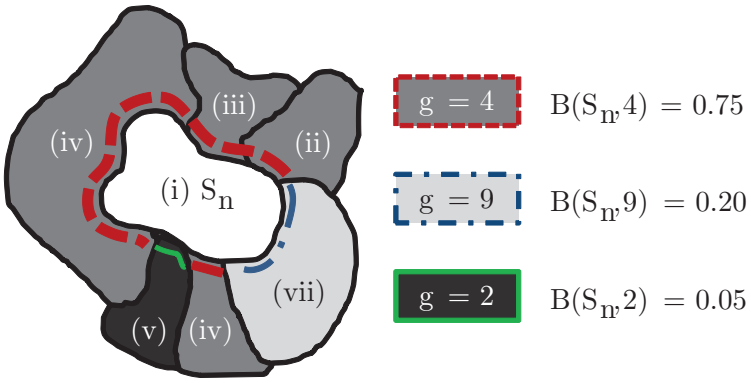


Figure 3.21: For segment (i) there are three different adjacent disparity gradients, $g = 2$, $g = 4$ and $g = 9$. Segments (ii) - (iv) and (vi) all have $g = 4$ and their combined contiguous border ratio with (i) is 0.75 so $B(S_n, 4) = 0.75$, similarly from segment (vii) $B(S_n, 9) = 0.20$ and segment (v) $B(S_n, 2) = 0.05$.

The diagram in Fig. 3.21 shows an example in which a segment S_n (labelled (i)) is surrounded by six other segments (labelled (ii) to (vii)). Each of these segments has been assigned to a layer l, with a DG value g_l. So the perimeter of (i) will be bounded by by other segments whose DGs equal one or more values of g_l.

The first step is to find the proportion of the segment border bounded by each of the g_l, giving us the border ratio $B(S_n, g_l)$ and disparity gradient g_l. This border ratio allows us to determine the best layer to assign S_n to in order to minimise discontinuities in the depth map.

The second step is to determine the cost of such a disparity re-assignment. We do this by looking at the DG confidence response $\text{ʃj}(S_n, g)$ where

$$\text{ʃj}(S_n, g) = \underset{\epsilon}{\operatorname{argmax}} (\epsilon \left(S_n, g \right), \ \bar{\epsilon} \left(S_n, g \right)), \tag{3.12}$$

for each segment S_n. If we assign a segment to a g_l other than $\hat{g}_n$ there will be an increase in the $\bar{\epsilon}(S_n,g)$ matching error (3.6), resulting in a drop in confidence, Δʃʃ, this drop is the cost of re-assignment.

For a low texture background segment with a wide peak (which is the main type of segment to have slight variations of assignment), as shown in Fig. 3.22, a small shift in g leads to a small shift of ʃʃ so there is little cost in the re-assignment. Conversely a

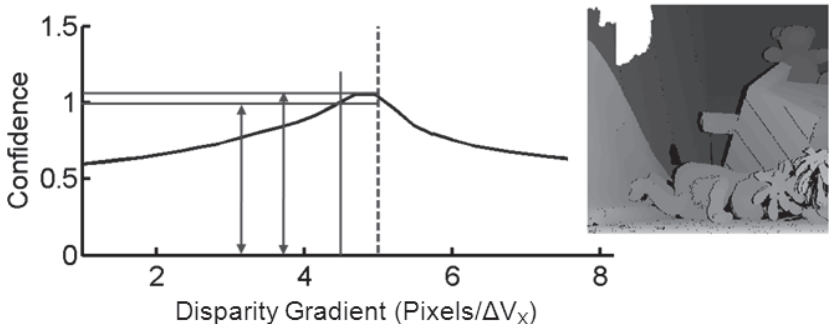


Figure 3.22: When the peak is shallow and smooth, slight changes in g do not lead to a large change in confidence.

highly textured foreground object with a sharply defined peak which we do not want to flatten with surrounding segments, such as Fig. 3.23, has a high cost for the same degree of re-assignment. Combining these gain and cost functions together gives us the

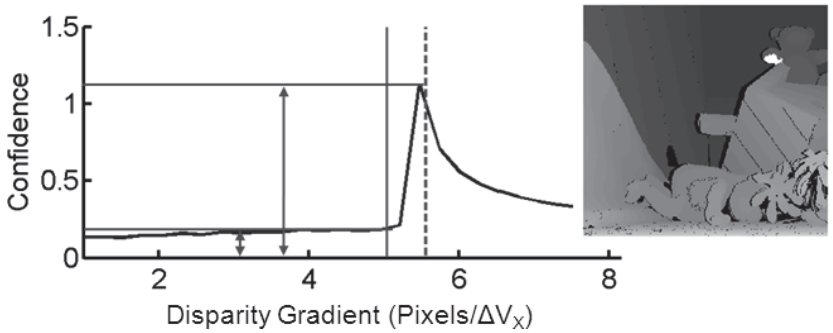


Figure 3.23: When the peak is steep and sharp, slight changes in g lead to a large change in confidence.

flattened assignment confidence metric $\eta(S_n, g)$,

$$\eta(S_n, g) = B(S_n, g) - \alpha \left(\frac{ʃʃ(S_n, \hat{g}_n) - ʃʃ(S_n, g)}{ʃʃ(S_n, \hat{g}_n)} \right), \tag{3.13}$$

Table 3.1: This table lists the sequences [74, 75] that were used in our evaluation.

Sequence	Image resolution	Number of images	$\widetilde{DG}$
Teddy	450×375	9	16
Cones	450×375	9	16
Barn1	432×381	7	8
Sawtooth	432×380	7	16

which balances the gain of flattening a segment to the surrounding segment DG, based on the border length, versus the cost of a less confidence assignment with a weighting term α to allow fine tuning. The segment, S_n, will be assigned to the layer associated with the highest $\eta(S_n, g)$, as long as it is above a empirically determined re-assignment threshold of 0.6. The process is iterative with all calculations occurring with the current segment assignments and a simultaneous re-assignment of all the segments after the round of calculations has finished. However in certain cases the segments can end up in periodic pattern, flip-flopping between a series of states. To force the system to stabilise a damping term $\zeta\ (k)$ is added to the equation, giving us a damped matching metric $\bar{\eta}(S_n, g, k+1)$ where

$$\bar{\eta}(S_n, g, k+1) = B(S_n, g) - \zeta(k) - \alpha \left(\frac{\text{ﬦ}(S_n, \hat{g}_n) - \text{ﬦ}(S_n, g)}{\text{ﬦ}(S_n \hat{g}_n)} \right), \tag{3.14}$$

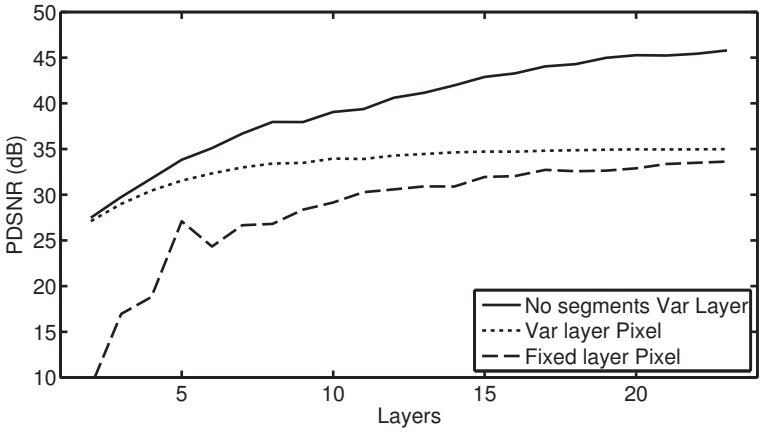
k is the iteration number and $\zeta = \beta \cdot k$ so a high ζ will stabilise the system in fewer iterations. We have found empirically that values of $\alpha = 0.4$ and $\beta = 0.01$ gives us good results.

3.5 Evaluation

For our evaluation we used the sequences [74, 75] shown in Table 3.1. The key images were segmented using the mean shift algorithm [69, 76]. These sources are provided with GT DG maps with a granular resolution of $\frac{1}{16}$ pixel/ΔV_X for all cases except for Barn1 which has a granular resolution of $\frac{1}{32}$ pixel/ΔV_X. This results in the calculated maximum possible image disparity measure $\widetilde{DG}$ for the GT DG maps.

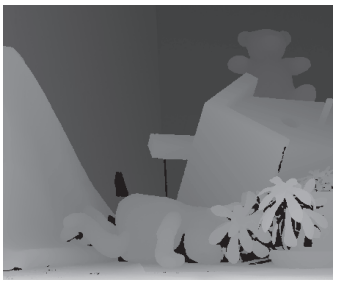
3.5.1 Evaluation of the layer model

The performance of the layer model can be assessed by studying the error when the layer model is used to estimate the DG map against the 255 layer GT results provided with the sequences.



(a) Teddy sequence geometric model comparison.

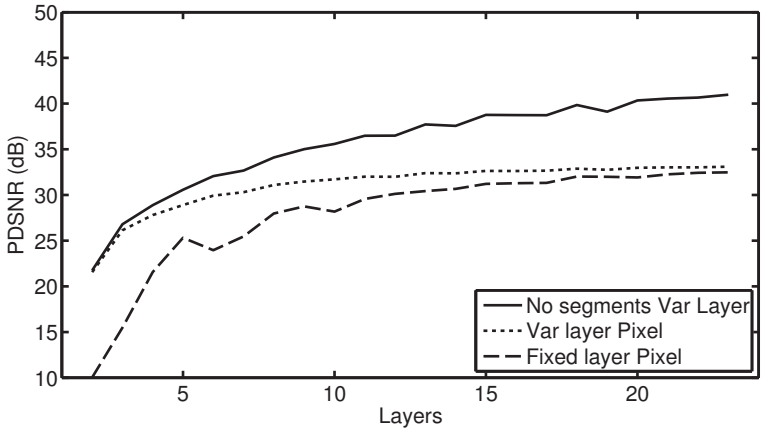
(b) Image.



(c) GT DG map.

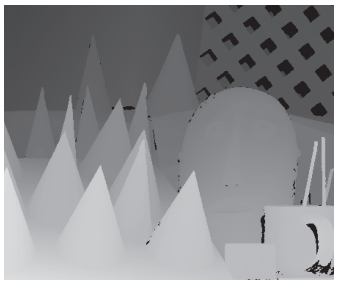
Figure 3.24: In this figure we have (a) the comparison graph between different geometric models and (b)(c) examples from the dataset.

Figures 3.24-3.27 show the error in estimating the DG map using our layer-based method against the GT map for each of the sequences. The assignment error from applying the different layer models to the GT DG map, single pixels assigned to non-uniformly spaced layers (solid line), segments assigned to non-uniformly spaced layers (dotted lines) and segments assigned to uniformly spaced layers (dashed lines). The calculated $L_{\min}$ for each sequence is shown by the vertical dotted line. We also show an example image and disparity map for each dataset. We have measured the similarity of the two DG maps using a Peak Disparity Signal to Noise Ratio (PDSNR) measure



(a) Cones sequence geometric model comparison.

(b) Image.



(c) GT DG map.

Figure 3.25: In this figure we have (a) the comparison graph between different geometric models and (b)(c) examples from the dataset.

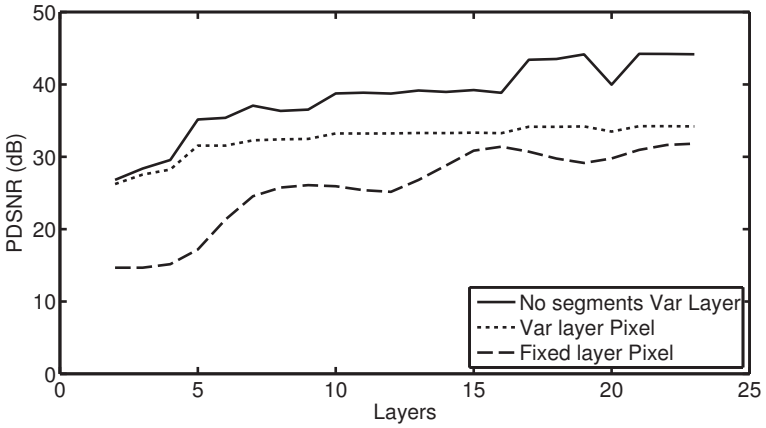
$$PDSNR = 10 \cdot \log_{10}\left(\frac{\widetilde{DG}^2}{MSE}\right), \tag{3.15}$$

where the Mean Squared Error (MSE),

$$MSE = \frac{1}{I \cdot J}\sum_{i=0}^{I-1}\sum_{j=0}^{J-1}\left|DGM_{GT}(i,j) - DGM_M(i,j)\right|^2, \tag{3.16}$$

is the squared pixel difference between the GT DG map, DGM_{GT}, and the layer model DG map, DGM_M, we are investigating. $\widetilde{DG}$ is the maximum disparity value possible for the scene and I and J are the image width and height, as detailed in Table 3.1.

We compare three different models: single pixels assigned to non-uniformly spaced layers (solid line), segments assigned to non-uniformly spaced layers (dotted line) and segments assigned to uniformly spaced layers (dashed line). In 3.24(a) we can see that for the single pixel assignment (solid) the quality of the output increases with the



(a) Barn1 sequence geometric model comparison.

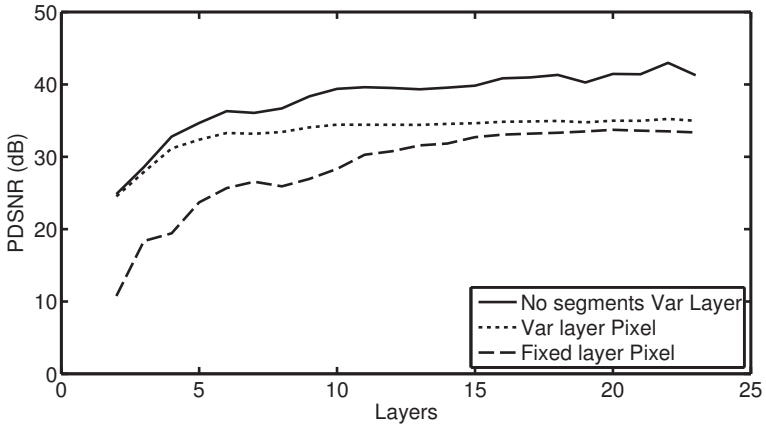
(b) Image.

(c) GT DG map.

Figure 3.26: In this figure we have (a) the comparison graph between different geometric models and (b)(c) examples from the dataset.

number of layers used until the improvement plateaus with no further improvement with additional layers, this has a higher final PSNR compared to the segment based methods. However as we have described earlier, the segmentation step is essential for our quick robust layer assignment method. Comparing the two segment based models, the non-uniformly spaced layer model and the uniform layer spacing model do eventually converge but the non-uniformly spaced model (dotted) plateaus much faster, supporting our argument in Sec. 3.2.4. It is also important to note that the non-uniformly spaced layer model is also much smoother. In all cases the non-uniformly spaced model (dotted) plateau point is at or before the calculated minimum requried layers, $L_{\min}$ (2.10).

All four cases Figs 3.24-3.27 show similar characteristic curves with some scene specific differences. For example in Figs 3.25-3.27 there is less difference between the segment and pixel based methods because the assumption that we can model the scene



(a) Sawtooth sequence geometric model comparison.

(b) Image.

(c) GT DG map.

Figure 3.27: In this figure we have (a) the comparison graph between different geometric models and (b)(c) examples from the dataset.

using fronto-parallel planes is closer to the true scene geometry. Also in Fig. 3.25 the curves plateau faster because the scene depths are more highly clustered.

The behaviour of the segmented models support our arguments that Plenoptic theory is valid as a guideline for selecting the right number of layers to allow sufficiently accurate geometry of the scene, that there are diminishing returns from increasing the geometry beyond this point and most importantly that our non-uniformly spaced layers allows us to efficiently allocate resources to improve the modelling of the scene.

3.5.2 Evaluation of the segmentation method

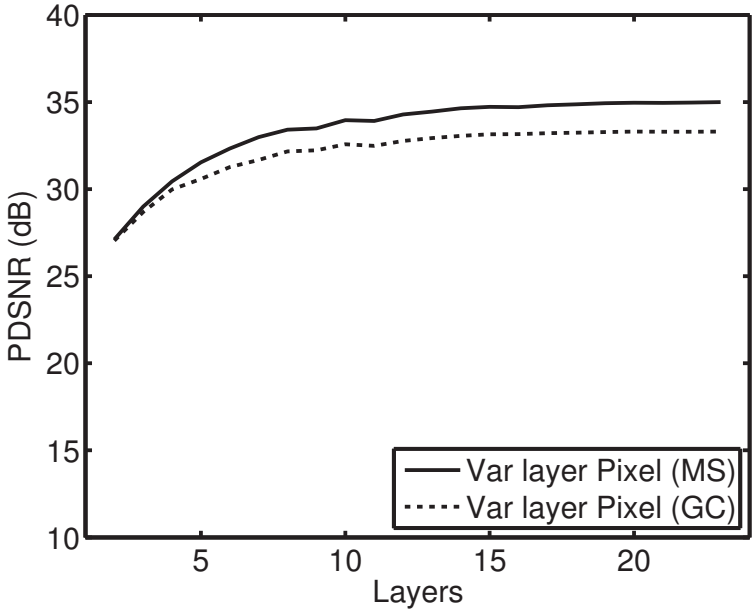
Our method is independent of the segmentation method used, as long as it successfully gives us continuous regions that lie within the same layer of sufficient size to be matched robustly. Most segmentation methods of this type are based on spatial and colour similarities between pixels, two examples are the Mean Shift (MS) [69, 76] and Graph

Cut (GC) [77–79] algorithms.

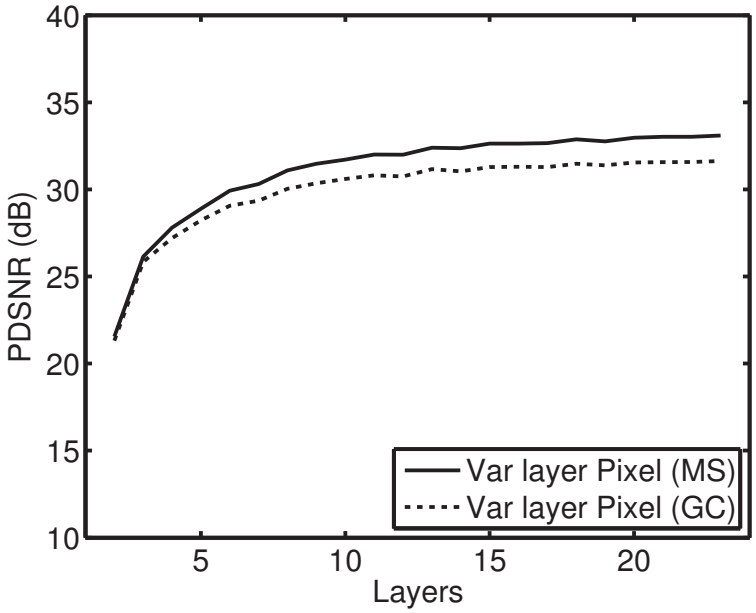
The performance when using these two segmentation algorithms with non-uniformly spaced layers is shown in Fig. 3.28 and we see that, for these two data sets, the mean shift algorithm is consistently better by approximately 2 dB. The performance when using uniformly spaced layers is shown in Fig. 3.29 and we see that difference between the segmentation algorithms remains the same even though the overall performance is worse.

3.6 Conclusions

In this chapter we have presented a novel layer assignment algorithm. Our approach uses Plenoptic sampling theory to infer the amount of geometric information required for artefact-free rendering. Guided by this prediction it takes advantage of the typical structure of multiview data in order to perform a fast occlusion-aware non-uniformly spaced layer extraction. We have shown that our novel non-uniformly spaced layer placement model gives a major improvement in quality and robustness over the uniform spacing layer model. Moreover, our layer extraction algorithm is independent of the segmentation method used. We have also shown that many mis-assignments or inconsistencies can be solved by smoothing and splitting the segments.

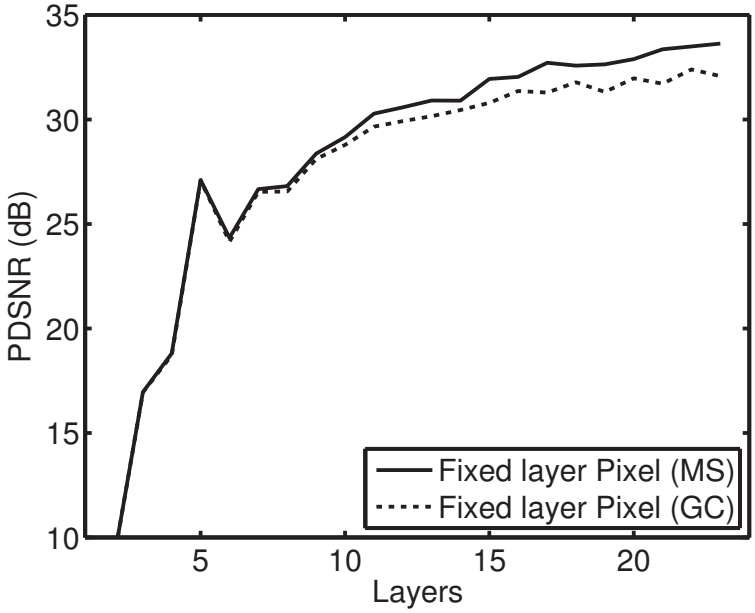


(a) Teddy sequence.

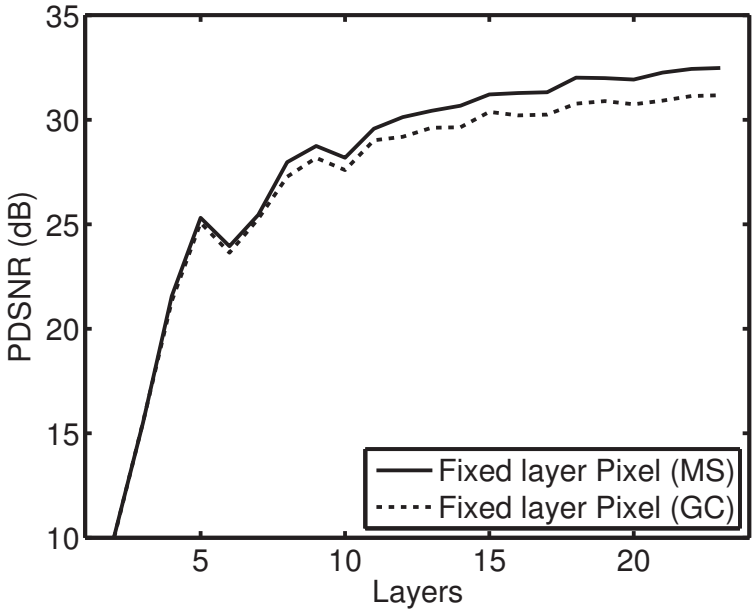


(b) Cones sequence.

Figure 3.28: The assignment error from applying the different segmentation methods to a non-uniform spacing layer scheme.



(a) Teddy sequence.



(b) Cones sequence.

Figure 3.29: The assignment error from applying the different segmentation methods to a uniform spacing layer scheme.

Chapter 4

View synthesis

4.1 Introduction

View synthesis is the creation of novel views of a scene based on existing images. Our synthesis algorithm consists of the following steps: First we need layer based geometry for all of the input images and the view to be synthesised. As described previously, we calculate the layer models for a few key images and then use these to predict the geometry for all the other views. This geometry allows us to use the Epipolar Planar Image (EPI) line structure to interpolate a new image from existing images. Generally to minimise errors the closest two images either side of the new view are used for the synthesis, as described in Sec. 4.4.1. We explained in Sec. 2.2.2 why we aim to use a geometric model comprising a finite number of layers and how Plenoptic sampling theory indicates the number of layers that are needed. As discussed in Chapter 3 we have calculated the required amount of geometry and assigned every pixel in the key images to a fixed fronto-parallel layer, this flat representation of the geometry is known as a Disparity Gradient (DG) map. This geometry and the input images is used to perform the synthesis of new views. This chapter will show that the predictions made by Plenoptic theory hold true for real world scenes.

This chapter is organised as follows : In Sec. 4.2, we describe how using the EPI line structure we can predict the intersection in adjacent images of the EPI line that passes through each new output image pixel, accounting in this way for occlusions.

An overview of the issues with synthesising real world scenes is presented in Sec. 4.3. We then discuss how we have solved these issues. In particular, improvements to the synthesis algorithm are presented in Sec. 4.4, covering both pixel level rendering based on their spatial positions and pixel value similarity a probabilistic estimate is made to interpolate the new pixel position, Sec. 4.4.1 and how multiple key images are utilised to fill in any gaps in the output image, Sec. 4.4.2. Post processing improvements are covered in Sec. 4.5 dealing with missing information, Sec. 4.5.2 and edge based errors in Sec. 4.5.1 are applied to the image on a pixel by pixel basis to remove minor rendering artefacts.

Finally we evaluate all the proposed methods and improvements against the Ground Truth (GT) and alternative competing algorithms in Sec. 4.6 and present our conclusions in Sec. 4.7.

4.2 Plenoptic synthesis

From Plenoptic theory (see Sec. 2.2), the function $\mathcal{P}_3(i, j, V_X)$ gives the intensity of pixel (i, j) in the image from camera position V_X. Each point in the scene corresponds to an EPI line in the three dimensional (3D) space (i, j, V_X). If the scene is Lambertian, all light rays from a scene point have the same intensity and, in the absence of occlusions, the intensity $\mathcal{P}_3$, will be constant along each EPI line. Novel views are generated by interpolating the sample points provided by the other input images along the corresponding EPI line. In Fig. 4.1 we illustrate a simplified two dimensional (2D) case with four EPI lines on two layers (i.e. j is constant). The two points P and Q, lie on the layer closest to the cameras while points R and S lie on a more distant layer and are occluded by point Q at $V_X = 1.4$ and $V_X = 0.2$ respectively. The new sample on an EPI line, at position $V_X = 1.7$, is interpolated from the samples provided by input images, $V_X = 1$ and $V_X = 2$, either side. For points P, Q and S the EPI line is un-occluded on both sides so the new sample will be interpolated as a blended distance-dependant mixture of the two input images. In the case of R only one side of the EPI line is un-occluded so only the sample from $V_X = 2$ will be used.

We synthesise the image on a layer by layer basis, starting with the lowest disparity and hence the most distant layer, and move through the layers progressively closer to the camera to preserve the occlusion ordering.

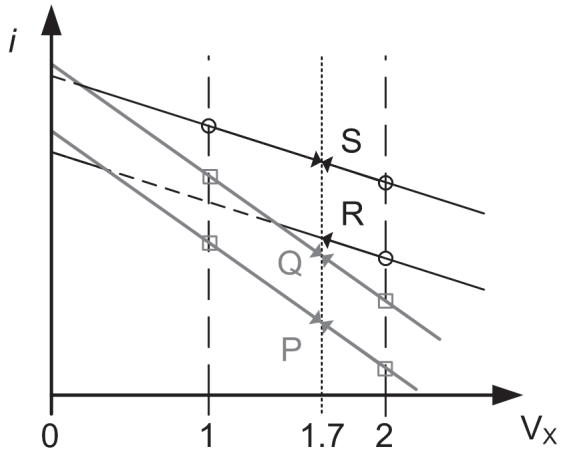
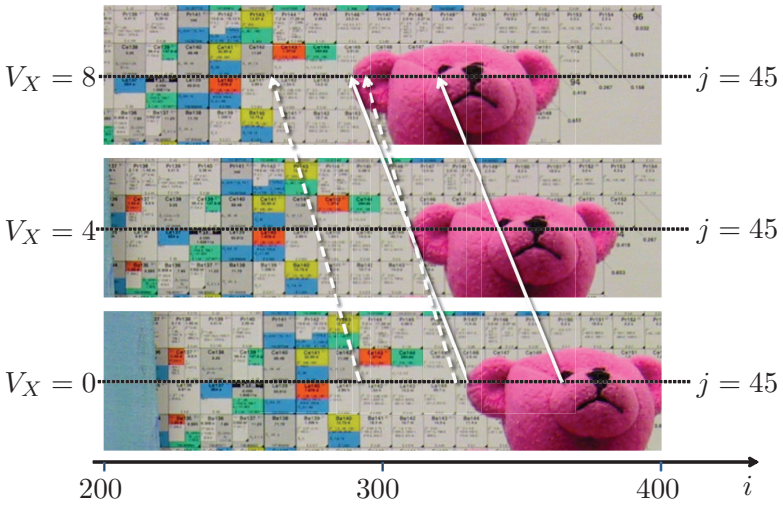


Figure 4.1: To synthesise a new view at $V_{1.7}$ we take pixels along the EPI line from bracketing views V_1 and V_2 and combine them to form a new interpolated value. If a potential source pixel is occluded it is not included in the interpolation.

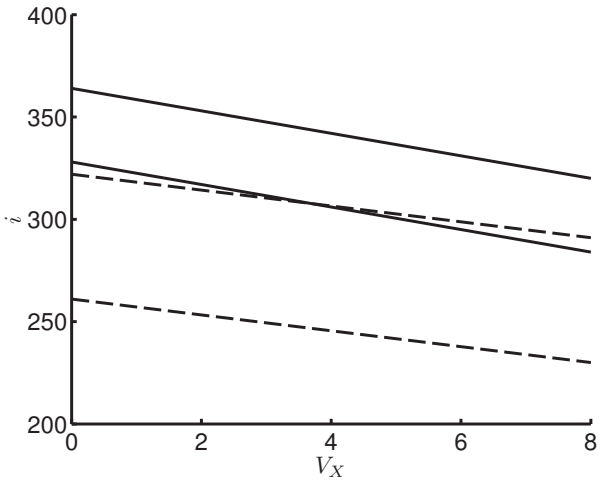
In Fig. 4.2 we show an example of real world Plenoptic synthesis along EPI lines. The two solid lines in Fig. 4.2(a) are from the foreground teddy object and the dashed lines the background periodic table. The real world example demonstrates the occlusion occurring at $V_X = 4$ as predicted by the intersection in the EPI line graph in Fig. 4.2(b). So at $V_X > 4$ three of the points can be interpolated from two directions and one can only be interpolated from one direction.

4.3 Layer geometry approximations

In the previous chapter we discussed how we estimated the required amount of geometric information necessary for high quality synthesis. We use this geometric information combined with the input images to perform Plenoptic synthesis, as described in Sec. 4.2, which should result in alias-free reconstruction of any view. In a real world situation this is not the case because many of the assumptions do not hold true in reality. These lead to errors in the synthesised output; however by understanding the cause of these issues their effect can be reduced.



(a) EPI lines between points in Teddy sequence.



(b) EPI lines

Figure 4.2: An example of EPI lines in a real world (a) and 1D (b) case. All points are along a slice through the image at $j = 45$.

4.3.1 Model inconsistencies

Errors in the synthesis are due to inconsistencies between the geometric model and reality. These errors arise because some of our assumptions such as for example fronto-parallel planes or infinite field of view are not valid in a real world case. Although with sufficient layers a flat model of the scene is a good representation for many cases there will always be differences from reality. In addition the finite sampling resolution of the camera means that the scene textures are not band limited within an object and the discontinuities at object edges violate band limiting. We introduce a novel enhancement to Plenoptic sampling in Sec. 4.4.1 that deals with many of these problems.

4.3.2 Geometric misassignment

It is inevitable that mistakes in the layer assignment process will sometimes result in pixels being assigned to an incorrect layer. Such mistakes may arise either from errors in segmentation or from the assignment of a segment to the wrong layer. It is important that a view synthesis algorithm is robust to such mistakes and we discuss ways of dealing with them in Sec. 4.5.1.

4.3.3 Missing information

One of the key assumptions made in Plenoptic theory is that there are no occlusions. Once we are dealing with a scene with occlusions and cameras with a limited field of view there will be regions of the scene that are only visible in only some or even in none of the available input images. Due to the differing amounts each layer is shifted, regions of one layer may move to occlude a layer with a lower DG. Consequently when the layers are shifted, regions of the scene also become disoccluded leaving gaps. This is illustrated in Fig. 4.3 where the left column shows the effect of projecting segments from a key image, (i) at $V_X = 0$ to other camera positions, (ii) - (iv) with $V_X = \{2, 6, 8\}$. As the DG map is projected further from the key image the effects of the occlusions and dissocclusions becomes more and more obvious as more holes appear in the image. If a key image is taken from the opposite end of the sequence, as shown in the right

column of Fig. 4.3, the same can be seen to happen in reverse, as the key image (iv) is projected onto (iii)-(i). These holes occur because of the cluttered nature of the scene, as these regions are not visible from a key image so there is no DG information available. However as we discuss in Sec. 4.4.2, it is possible to eliminate the gaps by combining the two key images and filling the gaps in one view with information from the other.

It is important to understand the causes of different types of occlusion/disocclusion as different approaches are required to deal with them. Three types of possible disocclusion are illustrated in Fig. 4.4; (A) shows tearing, where a missing region appears in a oblique surface which spans multiple depth layers; (B) shows a region of inter object disocclusion. Type (C) errors demonstrate disocclusions due to the lack of available image information outside the field of view. Type (A) and (B) errors can be in-filled directly, either from surrounding pixels or different image sources if available; this is discussed in Sec. 4.4.2. Type (C) holes can cause problems if in-filled directly, as described in the latter part of Sec. 4.4.2.

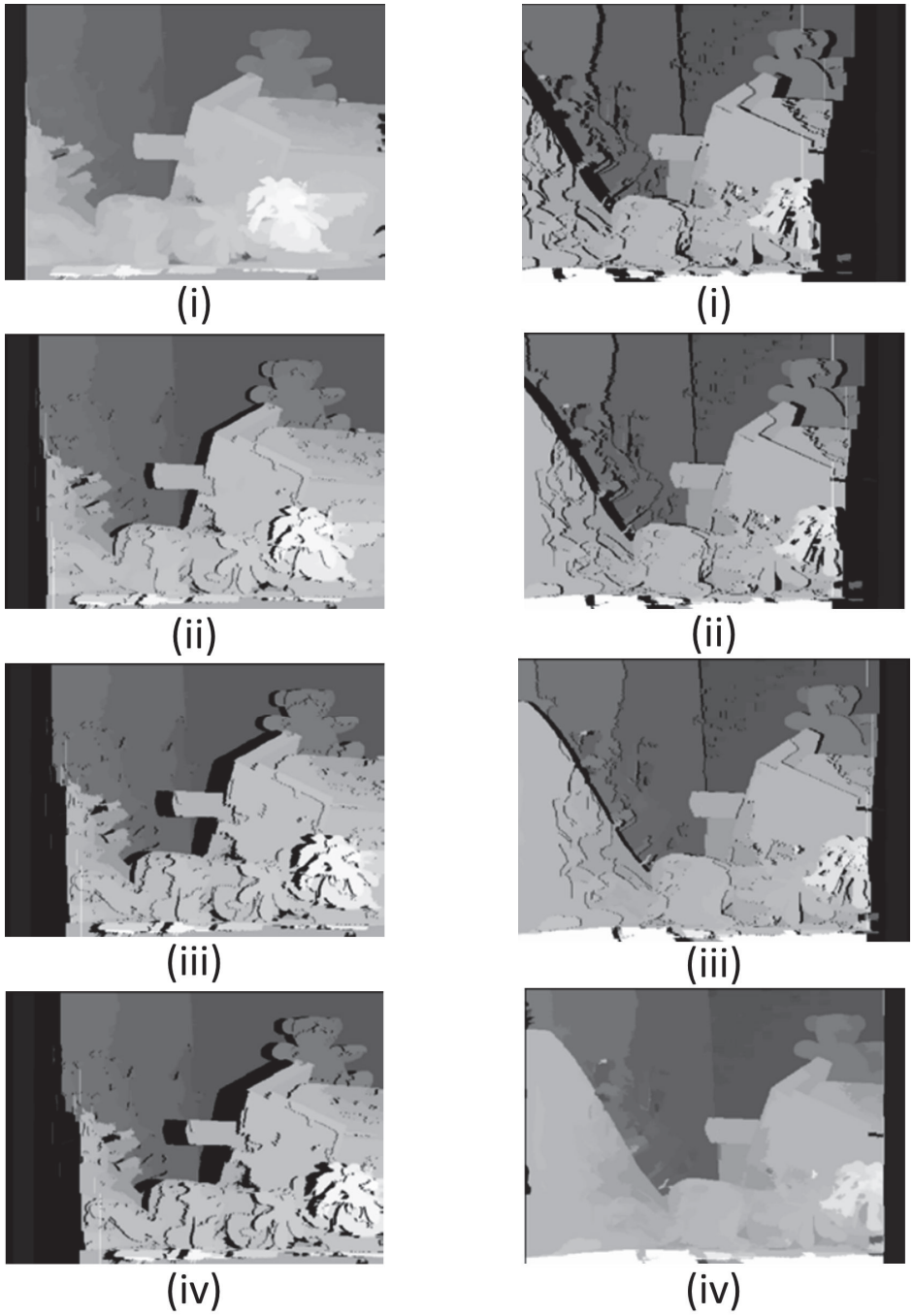
Sometimes a region of the scene is not visible from any of the input sources; in this case we need to extrapolate from the surrounding image and our knowledge of a typical scene in order to fill the missing region, this is described in more detail in Sec. 4.5.2.

4.4 Rendering enhancements

4.4.1 Probabilistic pixel interpolation

To synthesise a new view we scan through all the empty output pixels synthesising each individually by interpolating along the EPI lines using sample pixels, P_p, from the two closest bracketing views, as shown in the top down view in Fig. 4.5.

Because the $g \cdot V_X$ for a point has a sub-pixel precision the projection to the bracketing images will not normally lie exactly on a pixel. The most straightforward approach would be to linearly interpolate the intensity of the intersection point from the pixels either side of the intersection based on their spatial separation. For example using



(a) Key $V_X = 0$ (b) Key $V_X = 8$

Figure 4.3: Disparity map projection for two key images (a) $V_X = 0$ and (b) $V_X = 8$. Position (i) is at $V_X = 0$, (ii) $V_X = 2$, (iii) $V_X = 6$ and (iv) $V_X = 8$.

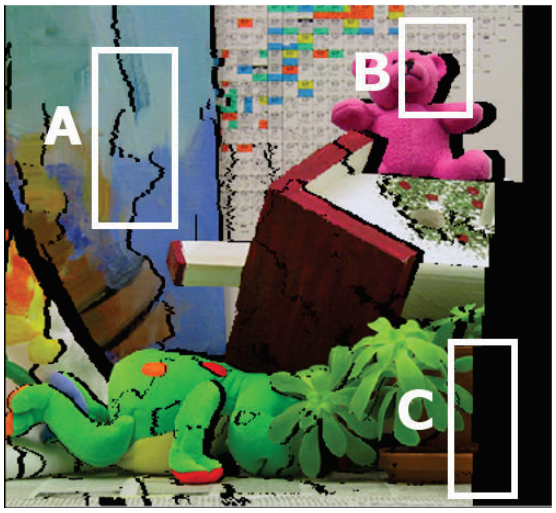


Figure 4.4: The view from the Teddy sequence at $V_X = 0$ is projected layer by layer to $V_X = 8$, with resulting disocclusions left as black pixels. Three different types of disoclusion are highlighted.

linear interpolation for the synthesised point in Fig. 4.5 we obtain

$$P_{1,2,3,4} = (1 - \gamma)P_{1,2} + \gamma P_{3,4}, \tag{4.1}$$

where at $V_X = V_s^-$,

$$P_{1,2} = (1 - \alpha)P_1 + \alpha P_2, \tag{4.2}$$

similarly for $V_X = V_s^+$ $P_{3,4}$ is calculated using β. Here

$$\gamma = \frac{V_s - V_s^-}{V_s^+ - V_s^-}, \tag{4.3}$$

is the distance between the synthesised image V_s and the lower bracket camera position, V_s^-, normalised relative by the total distance, $(V_s^+ - V_s^-)$. Moreover α and β are the distances in pixels from the EPI line to P_1 and P_3 respectively.

In some cases, however, the pixels are not all equally valid as sample points. For example, we need to make sure our interpolation only uses pixels from the current layer and that we account for any potential error in our layer assignment. So rather than the fixed interpolation scheme of (4.1), we use a probabilistic method, weighting each input pixel based on its estimated reliability. First we set a very low weight to any

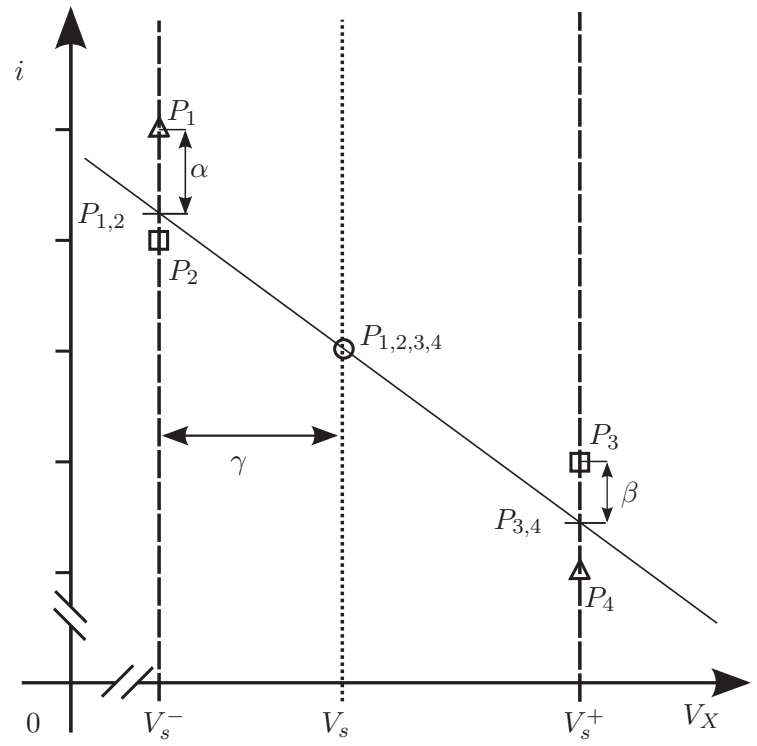


Figure 4.5: When synthesising a new view (dotted line) at V_s we interpolate along the EPI line using sample pixels, P_p, from bracketing views (dashed lines) V_s^- and V_s^+. Because the sample point in i for the existing views will not normally lie exactly on a pixel we have to use the two closest pixels from each bracketing view. The pixel $P_{1,2,3,4}$ is interpolated from pixels P_1, P_2 from V_s^- and pixels P_3, P_4 from V_s^+.

of the four input pixels which are not on the same layer as the output pixel. Second we compare the diagonally opposite pixel pairs (i.e. P_1 with P_4 and P_2 with P_3); if a diagonally opposite pair of pixels has similar intensities, then they are likely to match the target pixel and so are given a high weight.

So the probabilistic prediction for the interpolated pixel now becomes,

$$\hat{P}_{1,2,3,4} = \frac{(1-\gamma)(G_1\tau(1-\alpha)P_1 + G_2\chi\alpha P_2) + \gamma(G_4\tau\beta P_4 + G_3\chi(1-\beta)P_3)}{\sum_{p=1}^{4} G_p}, \quad (4.4)$$

where

$$\chi = \frac{|P_1 - P_4|}{|P_1 - P_4| + |P_2 - P_3|}, \quad (4.5)$$

and

$$\tau = \frac{|P_2 - P_3|}{|P_1 - P_4| + |P_2 - P_3|}. \tag{4.6}$$

G_p is the weighting for a synthesised pixel, $\hat{P}_{1,2,3,4}$, for points $p = \{1, 2, 3, 4\}$ where

$$G_p = \begin{cases} 1 - |\mathrm{g_s} - \mathrm{g_p}| & \text{if } |\mathrm{g_{diff}}| \leq \dfrac{\Delta \mathrm{V_X}}{2}; \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

g_p is the g for a bracketing pixel and g_s is the g for the EPI line; α and β are ratios of the intersection distance of the EPI line in relation to the pixel pair either side, as shown in Fig. 4.5. In the special case where $|P_1 - P_4| = |P_2 - P_3| = 0$ we set $\chi = \tau = 0.5$. The benefits of this approach are an improvement in PSNR and visual quality due to unreliable pixels having less effect on the interpolation.

4.4.2 Multiple key images

For complex scenes all regions of the scene may not be visible from a single key image. Using more key images increases the coverage of the scene and allows reliable assignment of these regions. For the EPI sequences tested, with between 5 and 9 images, we use two key images as we found that increasing the number of key images beyond this point provides little additional benefit to the output quality. By selecting images at opposite ends of the sequence we can increase the parallax and hence maximise coverage. A similar reasoning leads to choosing key images at opposite corners when using a Lightfield source.

When using multiple key images it is important that all the calculated key image DG maps have the same layer positions. The DG histograms, Fig. 3.9, are estimated for each key image independently. These results are then combined before the Lloyd-Max algorithm is applied jointly to both in order to estimate a common set of layer disparity gradients. This allows easy and smooth combination of the key image DG maps as well as making sure that the layer positions are placed efficiently even for objects that are only visible in some of the key images.

When synthesising a novel view, because of the consistent layer model used in all

key images we use them in a master-slave relationship. For each output view the closest key image is set as the master and any other available key images as slaves. Priority is given to the information from the master image in the case of any conflicts, so that information from the slave images is only used to fill holes in the resulting projection.

The three types of occlusion shown in Fig. 4.4 may be divided into two groups. Occlusion types (A) and (B) are caused by objects occluding other objects within the scene, known as internal occlusions. As these occlusions are consistent within the scene they can be filled in from other slave images. Type (C) errors are more problematic, because these framing occlusions are not consistent within the scene as they will be unique for each image position, so they will therefore cause problems when they are projected beyond the camera position. For example Fig. 4.6 shows a few examples of continuous objects that are occluded by the image framing but would be visible as a continuous surface in other views.

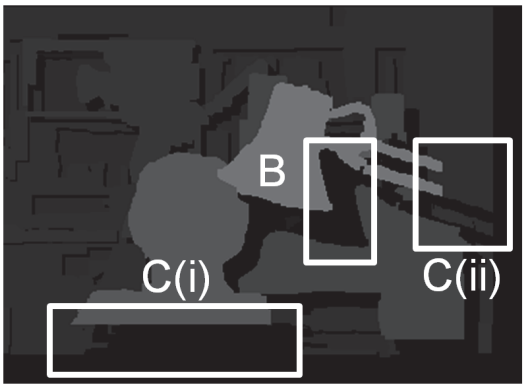
Figure 4.6: A few examples of contiguous regions within the scene that extend beyond the image framing and would therefore be occluded by the field of view.

Fig. 4.7(a) shows the DG map directly calculated for the camera position $(V_X, V_Y) = (4, 4)$ from the Tsukuba sequence. Figure 4.7(b) shows the prediction for the same camera position, based on the calculation for camera position $(V_X, V_Y) = (0, 0)$. If we compare the two there are a number of errors.

In this case all the disocclusions are type B or C, as shown in Fig. 4.4. For type B disocclusions such as Fig. 4.7(b)(B) the error is a hole in the DG map so it can be filled

in by using the DG map of another key image if available, if not it can be in-filled, as explained in Sec. 4.5.2. Some type C disocclusions can be dealt with in a similar way, for example in the case of Fig. 4.7(b)(C-i), although the error is caused by framing rather than internal occlusion none of the layers project into the disoccluded region so it can be in-filled as previously described for a type B error. Figure 4.7(b)(C-ii)) on the other hand poses a problem, although the region is missing part of the table lamp, due to the framing occlusion, there is already a lower layer present so no in-filling will occur even though the lower layer should actually be occluded by the table lamp. Because of their higher g layer, the foreground objects near the edge of the field of view are very vulnerable to this effect.

(a) $(V_X, V_Y) = (4, 4)$



(b) Projected from $(V_X, V_Y) = (0, 0)$

Figure 4.7: Comparing the original DG map for (a) and the DG map for $(V_X, V_Y) = (0, 0)$ projected to the same position shows that some regions (C-ii) cannot accurately be predicted without accounting for framing occlusion effects, whereas some can: (B), (C-i).

Our method to prevent this is to project the slave DG maps onto the master and

record which regions fall outside the frame and hence correspond to regions unseen from the master map. An example of this is shown in Fig. 4.8, showing the regions of image $(V_X, V_Y) = (4, 4)$ which are occluded by the framing of $(V_X, V_Y) = (0, 0)$. These selected regions of the slave DG map can therefore legitimately occlude regions of the master map, if they have a higher g, which solves the problem caused by framing occlusions.

Figure 4.8: Inter image projection allows us to calculate which parts of the slave key image are occluded by the master image frame.

4.5 Post processing

In real world synthesis there will always be errors and missing information that needs to be contend with, by understanding what causes these errors and with our knowledge of a typical scheme there are several methods we can apply to improve the final output quality of the synthesis.

4.5.1 Removing orphan edges and alpha blending

If the layer segmentation does not exactly match the underlying image then, as illustrated in Fig. 4.9, shifting a layer results in the edges of an object being left behind.

These orphan edges are normally only a pixel or two wide but can cause very obvious rendering artefacts and can be distributed throughout the image (depending on the difference in disparity gradient on the object edge). The orphan edges can be

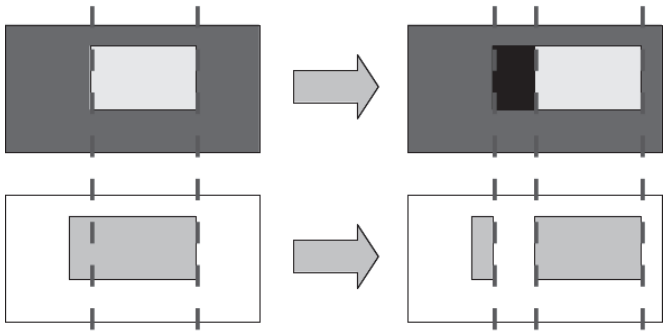
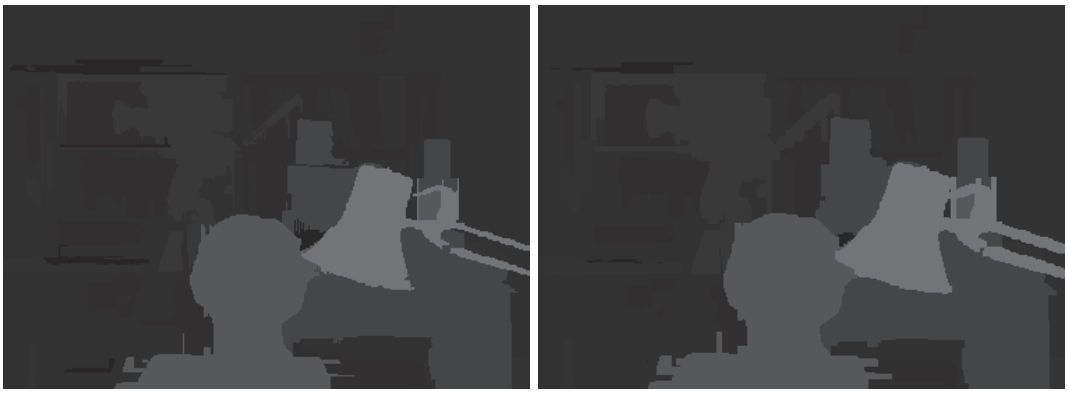


Figure 4.9: If the layer segmentation (top layer) does not match the underlying image (bottom layer) then prediction projection results in the edges of an object being left behind.

included in the correct layer if we pre-process the disparity map, enlarging each layer by extending the boundary into more distant layers by two pixels, as seen in Fig. 4.10. An additional benefit of this procedure is that any small holes or thin intrusion into layers are also absorbed, which generally improves the modelling of a typical scene. Layer extension solves the problems caused by orphan edges but introduces a different



(a) Original (b) Extended by 2 pixels

Figure 4.10: Each layer of the d map has been extended occluding pixels on lower layers only.

error, if the extension goes beyond the true layer boundary it leads to a halo of pixels round a foreground object that should be assigned to a lower layer causing an unsightly visual artefact.

As these errors will be on the edges of layers rather than distributed through the image they are easier to predict, additionally they are much easier to deal with via a technique called alpha blending or coherence matting [11]. We allow a degree of

transparency for each pixel in the layer between 0 (completely transparent) and 1 (completely opaque). We model the layers separately so for each pixel we can sum up all the pixels in proportion to their alpha transparency. If all pixels had a transparency of 0.8, a pixel would consist of 80% the top layer then 16% of the next layer (0.8 times the remaining 0.2) and the remaining 4% from the final background layer. If there are no layers underneath the alpha transparency of a layer pixel will always be 1. Alpha blending mitigates the haloing effect and has the added benefit of smoothing any jagged layer edges.

It is important that the blending is done with true in-line blending rather than just blurring the edges to avoid adding unwanted inaccuracies and artefacts. The first stage is to generate a alpha blending map for each layer. We use a linear blending profile,

$$\mathcal{A}(p,g_l) = \begin{cases} \dfrac{p_l}{p_{\max}+1} & \text{if } p_l \leq p_{\max}; \\ 1 & \text{otherwise.} \end{cases} \tag{4.8}$$

where p is the current pixel position, $p_{\max}$ is the number of extended pixels for the layer and p_l is minimum distance (in pixels) of a pixel to the edge of its layer, g_l, so

$$p_l = \min_{o \in \mathbb{E}(g_l)} \left(||p - o||_2 \right), \tag{4.9}$$

where o is part of the set $\mathbb{E}(g_l)$ of pixel positions around the edge of g_l and $||\cdot||_2$ is the ℓ_2 norm. If the underlying layer is not explicitly known it is interpolated from the surrounding geometry. This blending layer is used to calculate an alternative for the pixel in question which is then blended with the top level pixel value. Figure 4.11 shows the improvements using this combined extend/blend method. Orphan edges are removed and the edges of the foreground object are smoother and more natural looking without any loss of clarity or sharpness for the rest of the image.

The edge blending is projected along with the image to maintain a sub-pixel edge profile along with necessary scaling. As can be seen in Fig. 4.12 the blending scales as the objects grow maintaining a smooth curve which alleviates some of the stepping issues that become apparent with un-blended scaled regions.

(a) Original (b) Enhanced (c) Diff

Figure 4.11: By extending the d map by 2 pixels and then alpha blending by the same amount the orphan edge effects seen in (a) can be removed (b). The orphan edges can clearly be seen in the exaggerated diff map (c).

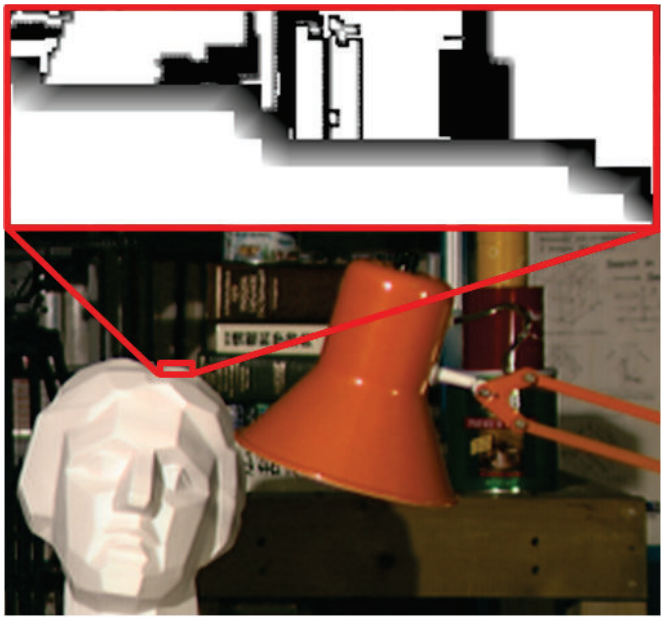


Figure 4.12: An example of scaled blending when moving the camera forwards in V_Z where the alpha transparancy of a layer is between 0 (black) and 1 (white).

4.5.2 Hole filling

As Fig. 4.4 shows, when the DG map is projected onto a synthesised view there are regions that are not covered. For these regions using a similar technique to that used in [80, 81], pixels are in-filled based on the most prevalent surrounding DG value, g, so for all available layers, g_l,

$$g_p = \underset{g}{\operatorname{argmax}} \left(B(p, g_l) \right), \tag{4.10}$$

where $B(p, g_l)$ is the number of pixel assigned to layer g_l bordering pixel p. If there are multiple possible values for g_p (same border value) the lowest (most distant) is chosen so

$$g_p = \min \left(\underset{g}{\operatorname{argmax}} \left(B(p, g_l) \right) \right). \tag{4.11}$$

For the example hole shown in Fig. 4.13 pixels (i) - (iii) will all be assigned to layer $g = 4$, (v) to $g = 7$ and the contested (iv) will be assigned to $g = 4$ as the lower g value.

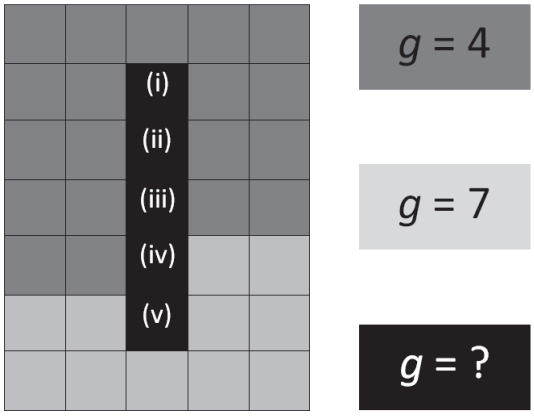
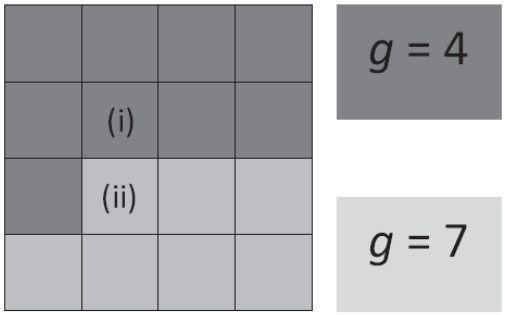


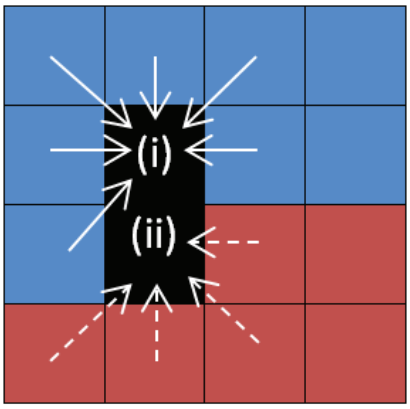
Figure 4.13: A region of the image where pixels are either assigned to $g = 4$, $g = 7$ or are an unassigned hole (i) - (v).

In most cases there will be at least one image that we can use for Plenoptic sampling, Sec. 4.4.1, directly synthesising the output from adjacent images based on the underlying DG value. However in crowded scenes dis-occlusions can reveal unique parts of the scene, in this instance we in-fill using a blended mixture of surrounding pixels, but only from the same layer. For example in Fig. 4.14 the pixel (i) will only be esti-

mated using other pixels from the layer $g = 4$ (solid arrows) and the pixel (ii) will be constructed from the layer $g = 7$ (dashed arrows).



(a) DG map



(b) Image

Figure 4.14: The DG map (a) corresponds to the image (b). Any holes in the image are infilled using pixels from the same layer eg. pixel (i) from layer $g = 4$.

4.6 Evaluation

For our evaluation we used the datasets [74, 75] shown in Table 4.1. The key images were segmented using the Mean Shift (MS) algorithm [69, 76, 82]. These sources are provided with GT DG maps with a granular resolution of $\frac{1}{16}^{th}$ of a pixel/ΔV_X for all cases except for Barn1 which has a granular resolution of $\frac{1}{32}^{th}$ of a pixel/ΔV_X. The 8-bit RGB images used lead to a calculated maximum possible image disparity measure $\tilde{I}$ for the images in the dataset. We used the 'leave q out' method of evaluation in which only every $(q+1)^{th}$ image is included in the input image set. These are used to synthesize one of the omitted images for which the ground truth is known. In all cases

Table 4.1: This table lists the datasets [74, 75] that were used in our evaluation.

Dataset	Image resolution	Image №	$\tilde{I}$
Teddy	450×375	9	255
Cones	450×375	9	255
Barn1	432×381	7	255
Sawtooth	432×380	7	255
Animal1	432×380	12	255

we use the first and last of the input images images as the key images of the EPI source and an infilling algorithm was used to fill any holes with the lowest adjacent disparity as described in Sec. 4.5.2.

We have measured the similarity of the synthesised images against the originals using a Peak Signal to Noise Ratio (PSNR) measure

$$PSNR(I_m) = 10 \cdot \log_{10}\left(\frac{\tilde{I}^2}{MSE(I_m))}\right) \tag{4.12}$$

$$MSE(I_m) = \frac{1}{I \times J}\sum_{i=0}^{I-1}\sum_{j=0}^{J-1}\left|I_m(i,j) - \bar{I}_m(i,j)\right|^2, \tag{4.13}$$

where Mean Squared Error (MSE) is the squared pixel difference between the original image, I_m, and the synthesised image, $\bar{I}_m$, for image position m. $\tilde{I}$ is the maximum pixel value possible for the scene and I and J are the image width and height, as detailed in Table 4.1. The final value we use is the mean of all non key images,

$$PSNR = \frac{\sum_{m=1}^{M-1} PSNR(I_m) \cdot \Xi_m}{\sum_{m=1}^{M-1} \Xi_m}, \tag{4.14}$$

where Ξ is the key mask such that

$$\Xi_m = \begin{cases} 1 & \text{if } I_m \text{ is not a key image;} \\ 0 & \text{if } I_m \text{ is a key image.} \end{cases} \tag{4.15}$$

4.6.1 Validation of the layer model

Plenoptic theory suggests that by choosing the appropriate number of layers we can have alias free rendering, and that no further improvement will be gained by adding extra geometric information. We validate this analysis and the effectiveness of our algorithm in Figure 4.15 which shows the variation of PSNR with the number of layers averaged over all the evaluation datasets. This demonstrates that the gap between our algorithm and rendering based on the knowledge of the GT geometry is only 0.25 dB. It also shows that the layer-based representation incurs no loss in performance when compared to the rendering based on complete geometry.

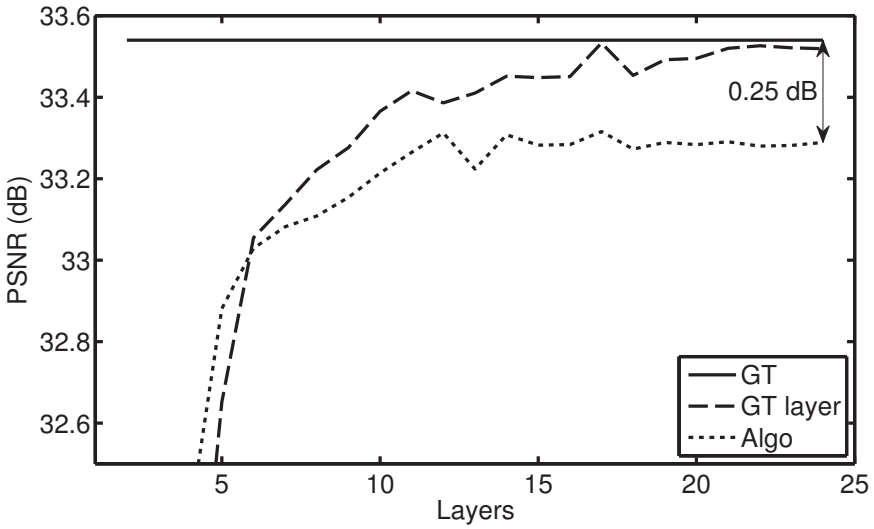


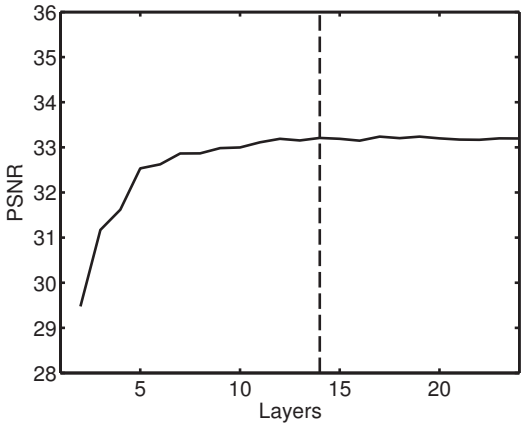
Figure 4.15: The horizontal line shows the best average possible performance using the raw ground truth DG map. The dashed line shows the average effect of applying the layer model to the raw ground truth (with no segmentation). The dotted line is our average algorithm result when the layer model is applied to our own calculated DG map (with segmentation). All three results are obtained by averaging over all the datasets.

Specifically, the solid horizontal line represents the best possible rendering result using the provided rawGT DG map, which provides full and accurate pixel based geometric information. The dashed line shows the effect of applying the layer model to this data by calculating the best layer positions and assigning all the pixels to the closest layer. As the number of layers used increases so does the quality of the output until the improvement plateaus with no further improvement from using additional layers.

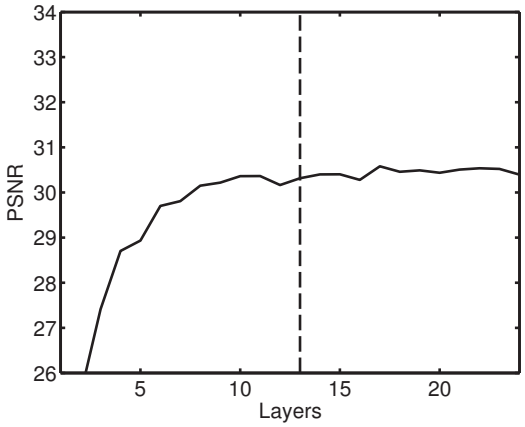
Importantly this plateau point is indistinguishable from the raw GT result showing that there is no inherent loss in quality if a sufficient number of layers is used. Finally the dotted line shows the result of our layer based DG extraction and rendering algorithm, which has only a 0.25 dB drop from the best possible performance. Part of this drop is due to the use of segments, as discussed in Sec. 3.5.1, and the remainder due to minor assignment errors. It is interesting to note that whereas the performance of single-pixel segments did not plateau when estimating the depth map in Figs. 3.24-3.27, this is not the case for the corresponding image rendering performance shown in the dashed line in Fig. 4.15.

4.6.2 Validation of the minimum layer constraint

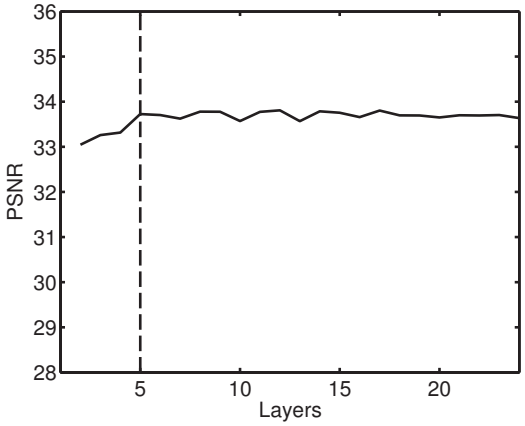
In Sec. 2.2.1 we discuss the prediction that Plenoptic theory makes in regard to the Minimum Sampling Criterion (MSC) of a scene based on $Z_{\min}$ and $Z_{\max}$ within a scene. This leads to the formula for $L_{\min}$ (2.10) the minimum number of layers required for high quality rendering. Although many of the assumptions of Plenoptic theory are not valid for a real world case, this requirement for a minimum number of layers is still a good guideline. If we look at the PSNR vs layer curve for our test datasets, Figs 4.16(a)-(e), in each case we can see that the curve has plateaued by the number of layers predicted by $L_{\min}$, shown as a vertical dashed line. In some cases, eg. Fig. 4.16(a) and Fig. 4.16(b), there is an initial sharp increase in synthesis quality as more layers are used, followed by a diminishing increase in quality, followed by the curve plateauing just before the calculated $L_{\min}$. In the case of Fig. 4.16(d) and Fig. 4.16(e), datasets that are highly clustered in Z, our non-uniformly spaced layer allocation, as described in Sec. 3.2.4, is effective at taking advantage of the empty regions in Z so the curve plateaus significantly before the $L_{\min}$ point. However the $L_{\min}$ point is still valid as a required number of layers. In the case of Fig. 4.16(c) because the scene has a very small ΔZ the curve increases sharply and plateaus exactly at the calculated $L_{\min}$.



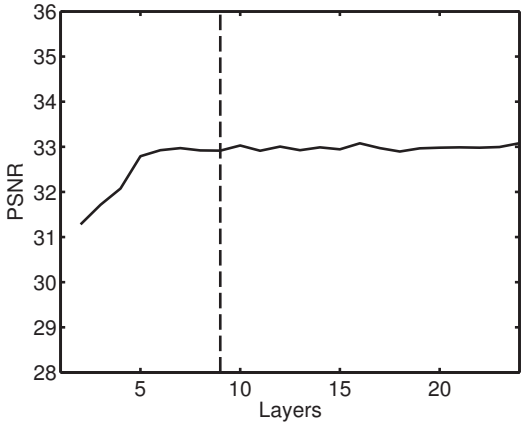
(a) Teddy sequence, $L_{\min} = 14$.



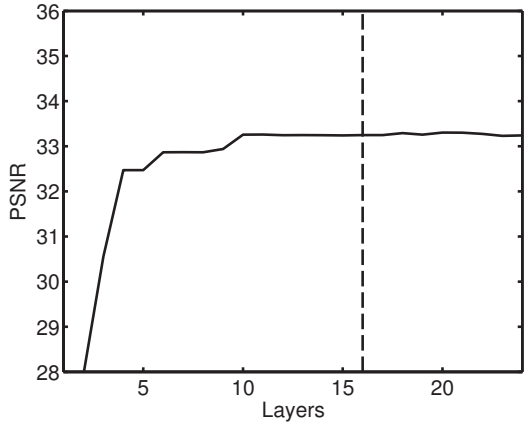
(b) Cones sequence, $L_{\min} = 13$.



(c) Barn1 sequence, $L_{\min} = 5$.



(d) Sawtooth sequence, $L_{\min} = 9$.



(e) Animal1 sequence, $L_{\min} = 16$.

Figure 4.16: The solid line is our average algorithm result when the layer model is applied to our own calculated DG map (with segmentation). This curve is calculated by averaging over all synthesised frames for the dataset Teddy. The average $L_{\min}$ based on the MSC is shown by the vertical dashed line.

Table 4.2: This table contains the comparison results between our 1st stage algorithm Sec. 3.2.3 and an alternative stereo-matching method [77] and the result of applying our 2nd Stage algorithm with and without variably spaced layers using Lloyd-Max. All results are from the Teddy dataset [74] with the same parameters and final rendering algorithm.

Method	PSNR (dB)
1st stage only (no Lloyd-Max)	32.43
Alternative method [77] (no Lloyd-Max)	32.65
Alternative method [77] + 2nd stage (14 layers)	33.04
1st + 2nd Stage (no Lloyd-Max)	33.20
1st + 2nd Stage (14 layers)	33.25

4.6.3 Comparison with alternative algorithms

Table 4.2 includes the results obtained when using an alternative pixel-based algorithm [77], [78] for which code was available. The stereo-matching performance of this algorithm on standard test sets is very high (94.5% of pixels within ±0.5 pixel disparity error [74]) although slightly worse that the current state-of-the-art, [83], (98% within ±0.5 pixel disparity error). Using only the 1st stage of our algorithm from Sec. 3.2.2 (row 1 of the table) results in a lower performance than this alternative algorithm (row 2), primarily because of a small number of wrongly assigned segments. Applying the 2nd stage of our algorithm from Sec. 3.2.5 improves the performance of both the alternative method (row 3) and our method (row 5). The disparity gradient histogram is here generated using either [77] or our 1st stage method, the layers are assigned using the Lloyd-Max algorithm from Sec. 3.2.4 and the number of layers is 14 as indicated by the minimum sampling criterion, $L_{\min}$, from (2.10).

Even though the raw performance of our 1st stage method is worse than that of [77], its disparity gradient estimates have a lower median error; this results in more accurate layer depth values and a slight increase in overall performance when the 2nd stage of our algorithm is applied (row 5 versus row 3). Row 4 of the table shows the results of using the full depth resolution (48 layers) in both stages of our algorithm. We note that not only does this require much more computation, but the performance is actually slightly degraded by 0.05 dB. Figure 4.17 shows that the performance of the layered

method is equal to or better than the non layered method for most layers, especially in the sweet spot around the MSC.

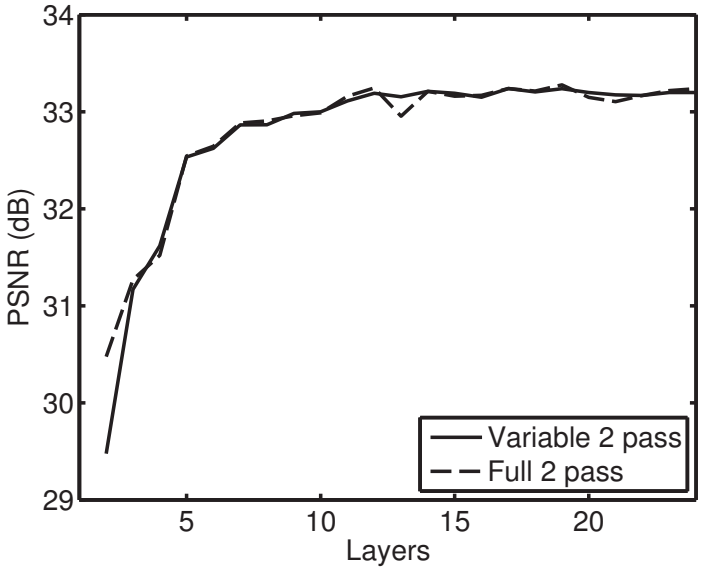


Figure 4.17: A graph showing the rendering quality for different number of layers with layer selection before or after the 2nd stage.

In Sec. 3.5.1 we saw that different segmentation methods, MS and Graph Cut (GC), gave very similar results but there was a small advantage to using the MS segmentation. However the small increase in predicted quality for MS segmentation shown in Figure 3.28 is not evident when the two segmentation schemes are used for rendering, as shown in Fig. 4.18 both give similar results.

4.6.4 Distance from key image

Figure 4.19 shows the synthesised outputs from $V_X = 1$ to $V_X = 7$ for the Teddy sequence, for 9 and 18 layers. The two key images are at $V_X = 0$ and $V_X = 8$. As the graph shows in both cases as the synthesis moves further away from the key images the quality drops, this is due to inconsistencies in the model having more of an effect the further the model is projected. Comparing the two curves we can see that middle section, furthest from the key images, improves the most indicating that additional geometric information is needed when synthesising views that are far away from the key images. Adding another key image at $V_X = 4$ would improve performance in this

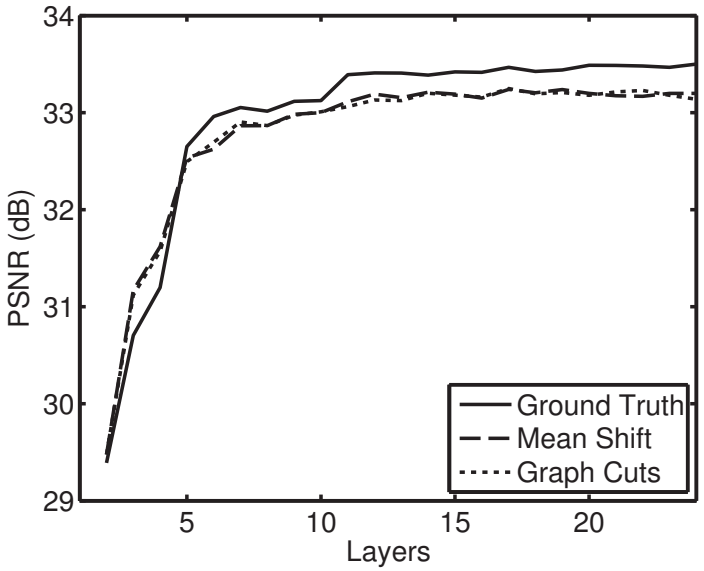


Figure 4.18: The rendering results from applying the different segmentation methods to a variably spacing layer scheme.

region but the increase in quality over the whole dataset is low and it adds a significant calculation penalty.

4.6.5 Algorithm breakdown

There are several major separable elements to the algorithm, the breakdown of the geometric calculation is shown in Fig. 4.20(a) for the Teddy sequence. With uniformly spaced layers (dotted line) the performance improves slowly with the number of layers and a very large number is required to reach the performance limit. The PSNR can be increased (dashed line) by incorporating layer extension (Sec. 4.5.1), and disparity gradient flattening (Sec. 3.4.2). With these improvements, the use of uniform layer spacing (dashed line) comes close to its limiting performance when using the number of layers, $L_{\min}$, predicted by Plenoptic theory and shown in Fig. 4.20(a) as the vertical dashed line at $L_{\min} = 14$. As noted in Sec. 3.2.4, the assumptions of Plenoptic theory are not fully met in practice and increasing the number of layers beyond $L_{\min}$ gives an additional performance improvement when using uniformly spaced layers. By using non-uniform layer spacing in our algorithm (solid line), we fully reach limiting performance with $L_{\min}$ layers and obtain significant performance improvement when using

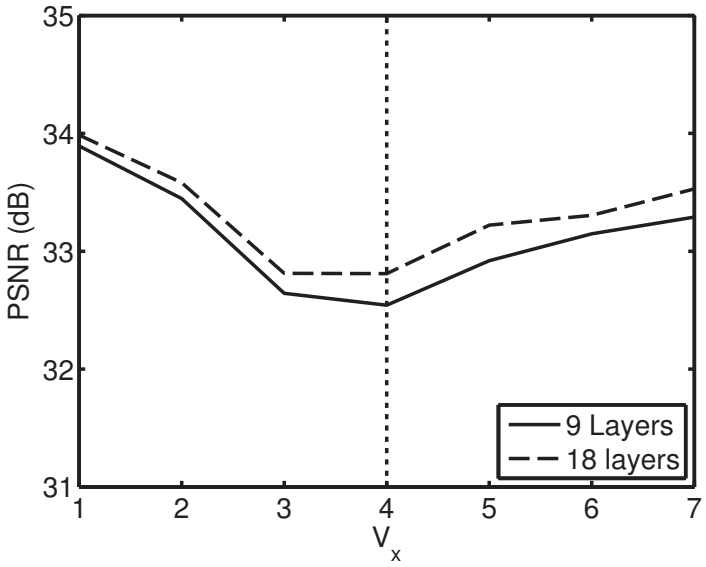


Figure 4.19: This graph shows the individual rendered "miss one out" results from $V_X = 1$ to $V_X = 7$ from the Teddy sequence, with 9 layers (solid line) and 18 layers (dashed line). In this example the key images are at $V_X = 0$ and $V_X = 8$, with the key image at $V_X = 0$ used as the master to the left of the vertical dotted line inclusive.

fewer layers than this.

The corresponding graph for the Cones sequence is shown in Fig. 4.20(b) where we see that the relationship between the three curves is very similar. The use of non-uniform layer spacing again provides a clear benefit although the improvement is less than with the Teddy sequence because the objects in the Cones sequence are more uniformly spread in depth. We note that $L_{\min}$ again indicates the number of layers required to reach limiting performance.

We can also breakdown the improvements in the results due to various elements within the synthesis, as shown in Fig. 4.21(a) for the Teddy sequence. The basic rendering method (dotted), with fixed pixel interpolation and no post-processing, can be improved by using probabilistic interpolation (dashed line), as described in Sec. 4.4.1. As well as smoothing the results it gives a substantial improvement in overall quality especially when few layers are used. Further improvements can be made across the board by using alpha blending (solid line) to minimise the errors on the edges of layers (see Sec. 4.5.1). Very similar effects may be seen in Fig. 4.21(b) for the Cones sequence although the differences are slightly greater.

Finally we note that on a desktop PC the total time to read in the input frames, extract the layers and synthesise an output image is 2.8 seconds, 0.6 seconds of which is the third party segmentation algorithm and 0.2 seconds is the time to synthesise each output image.

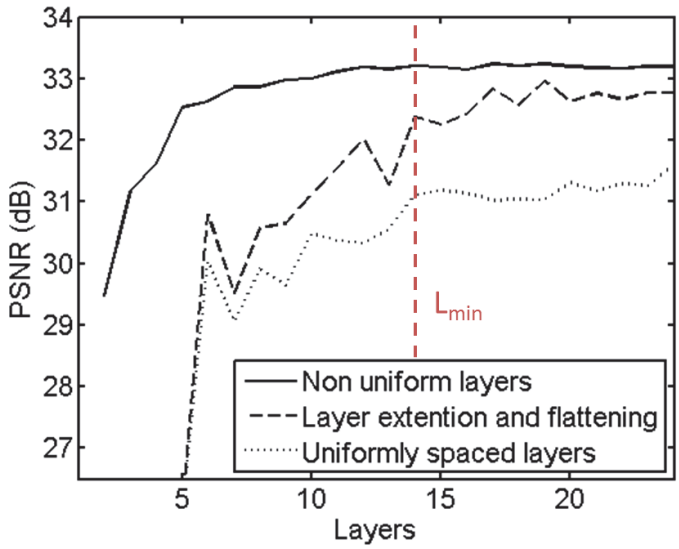
4.6.6 Output examples

In Fig. 4.22(a) we can see an example output of the algorithm from the Teddy sequence. With a PSNR of 33.9 dB and no major visual artefacts the rendering quality is very high with a definite photo-realistic feel. Looking at the luminance error map, Fig. 4.22(b), for the image we can see that 86% of the image has an error of one or less, the overall mean error is 1.004 (for a full scale of 255) and that the larger errors are only to be found on the edges of segments in thin bands. These edge errors are reduced due to the layer extension and alpha blending.

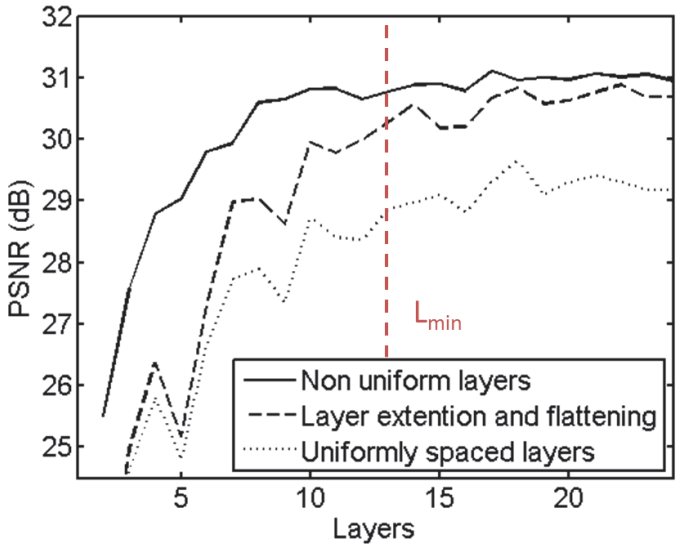
4.7 Conclusions

In this chapter we have presented a novel layer based rendering algorithm for Image Based Rendering (IBR). The rendering is improved by using a probabilistic interpolation approach and by an effective use of key images in a scalable master-slave configuration. Numerical results demonstrate that the algorithm is fast and yet is only 0.25 dB away from the ideal performance achieved with the ground-truth knowledge of the 3D geometry of the scene of interest. We have shown that our algorithm performs well in comparison with an alternative method.

We have also shown that the Plenoptic theoretical framework is applicable to real world cases since a layer based model does not lead to any loss in output quality and the number of layers required is correctly predicted by the theory. This indicates that despite several assumptions of Plenoptic theory not being satisfied in real world cases it is still an effective guide for producing high quality synthesised outputs.

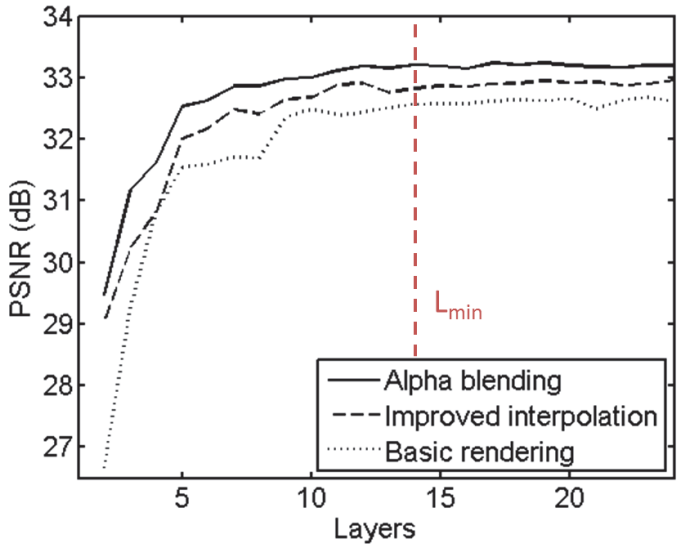


(a) Teddy $L_{\min} = 14$

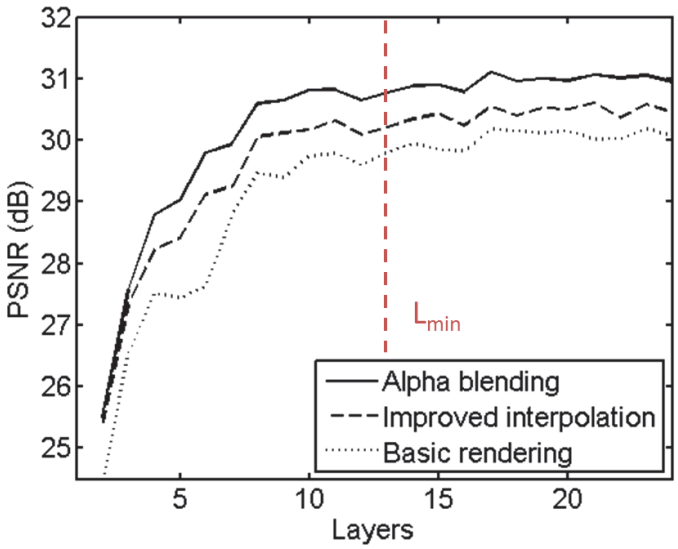


(b) Cones $L_{\min} = 13$

Figure 4.20: Showing the improvements in the algorithm results by using uniformly spaced layers (dotted), uniformly spaced layers with extension and layer flattening (dashed) and finally the best layer model with all enhancements and non-uniformly spaced layers. Results are for the Teddy sequence. The vertical line shows the calculated $L_{\min}$.



(a) Teddy $L_{\min} = 14$



(b) Cones $L_{\min} = 13$

Figure 4.21: Rendering improvements broken down into the basic rendering (dotted), improved interpolation (dashed) and the final alpha blended rendering (solid). Results are for the Teddy sequence. The vertical line shows the calculated $L_{\min}$.

(a) Output (b) Error

Figure 4.22: In (a) is an example rendered "miss one out" output for $V_X = 1$ from the Teddy sequence with a PSNR of 33.9 dB, with 18 layers. In (b) is an exaggerated difference error map (error $\times$ 10) for the image, with an average error of 1.004.

Chapter 5

Arbitrary virtual camera positions and rotation

5.1 Introduction

Previously we have described our Image Based Rendering (IBR) algorithm for the camera Epipolar Planar Image (EPI) line case and shown how it can be extended to encompass a plane of cameras, using the extra information that this provides to calculate more accurate geometry. This extra information can also allow us to relax restrictions on the output camera position allowing us greater freedom for synthesis. In this chapter we present three extensions to model multiple camera planes and allow even greater freedom in our synthesised camera position. The first extension removes the restriction that all input cameras must lie in a single plane and shows how the outputs from multiple camera planes can be combined. We also allow additional degrees of freedom for virtual camera rotation allowing camera planes at different angles.

A second extension describes how we remove the restriction that the viewpoint of a synthesised image must lie on the camera plane of the input images and allow the virtual camera to move away from. This gives greater freedom in our synthesis and permits the generation of smooth.

In the third extension, we relax our requirement that the layers be fronto-parallel

and show that the use of angled layers can result in improved synthesis quality. This is important for modelling multiple input camera arrays, using angled planes allows us the bridge the gap between the two separate geometry models with fewer discontinuities. This results in fewer artifacts when both models are combined to perform the view synthesis.

5.2 Multi-planar camera arrays

In previous chapters, we have constrained the input camera positions to lie on a single line or plane. Here we show that our approach can be extended to deal with the more general form of multiple planes as illustrated in Fig. 5.1.

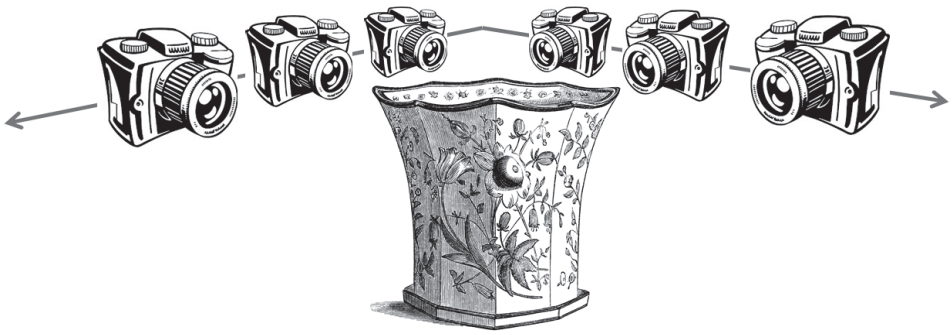


Figure 5.1: More of a scene can be viewed by allowing multiple planes of input cameras.

Relaxing our assumptions and restrictions to allow this means that we need our system to model camera rotation (see Sec. 5.2.1); projections between the planes (see Sec. 5.2.2); the new inter-plane occlusion ordering, (see Sec. 5.2.3); and merging the two models into a high quality output, (see Sec. 5.2.4).

A two dimensional (2D) example is shown in Fig. 5.2 showing a camera setup with two planes. Each model can be treated separately as a single-plane system, so the geometry model can be calculated as previously discussed, with its own (V_X, V_Y, V_Z) coordinate system and parallel layers. To differentiate the plane models we will be using the superscript symbols $^{(1)}$ and $^{(2)}$. The two models will intersect at $(V_X^{(1)}, V_Z^{(1)}) = (0, 0)$ with an angle between the planes of ϕ.

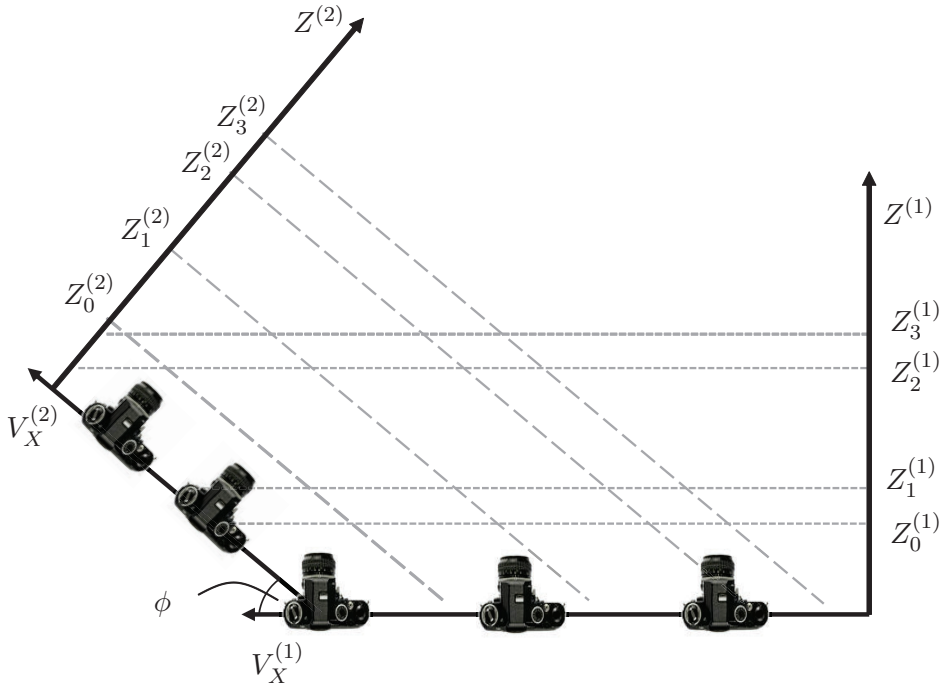


Figure 5.2: Top down view of a multi plane system with the two planes, along $V_X^{(1)}$ and $V_X^{(2)}$, intersecting at $V_X^{(1)} = 0$ at an angle of ϕ.

5.2.1 Camera rotation

The first enhancement to our system is modelling and performing camera rotations, allowing us a further degree of freedom for the output synthesis position. It is important to note that no geometry information is required for a camera rotation, as long as the camera position remains fixed in (V_X, V_Y, V_Z) in that the same light rays will pass through the camera position, so the light ray intersection with the camera plane in (i, j) will vary only with the Field of View (FOV), the focal length$f$ and the camera pose. We can construct a camera rotation transform matrix [84] and apply it to every pixel allowing us to model camera rotations. So pixel mapping for a point (i, j) to its rotated position (i', j') can be described by

$$\begin{pmatrix} i' \\ j' \\ 1 \end{pmatrix} = \mathbf{K}_2\mathbf{R}\mathbf{K}_1^{-1} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix} \tag{5.1}$$

where the camera matrix, $\mathbf{K}$, is defined as

$$\mathbf{K} = \begin{pmatrix} f_i & 0 & \bar{\imath} \\ 0 & f_j & \bar{\jmath} \\ 0 & 0 & 1 \end{pmatrix} \tag{5.2}$$

where f_i and f_j are the focal lengths in the i and j dimensions and $(\bar{\imath}, \bar{\jmath})$ is the optical centre of the image. The matrix entry k_{12} is always zero because we assume there is no pixel skew. We also assume for our rectified images that $f_i = f_j = f$ and $\mathbf{K}_1 = \mathbf{K}_2$.

The rotation matrix, $\mathbf{R}$, is in the Rodrigues form [85] so

$$\mathbf{R}(\widehat{\boldsymbol{\vartheta}}, \phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \widetilde{\vartheta} \sin\phi + \widetilde{\vartheta}^2 (1 - \cos\phi), \tag{5.3}$$

Where ϕ is the rotation angle about the axis specified by unit vector $\widehat{\boldsymbol{\vartheta}}$ where

$$\widehat{\boldsymbol{\vartheta}} = (\vartheta_X, \vartheta_Y, \vartheta_Z) \tag{5.4}$$

and $\widetilde{\vartheta}$ denotes the antisymmetric matrix where

$$\widetilde{\vartheta} = \begin{pmatrix} 0 & -\vartheta_Z & \vartheta_Y \\ \vartheta_Z & 0 & -\vartheta_X \\ -\vartheta_Y & \vartheta_X & 0 \end{pmatrix}. \tag{5.5}$$

In our case we will only ever have rotation about the Y axis so our fixed unit vector $\dot{\boldsymbol{\vartheta}} = (0, 1, 0)$ leads to a simplified rotation matrix

$$\mathbf{R}(\dot{\boldsymbol{\vartheta}}, \phi) = \begin{pmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix}. \tag{5.6}$$

5.2.2 Connecting the planes

As Fig. 5.2 shows the two camera planes intersect at $V_X^{(1)} = 0$ and both have a camera used as a key image at this position. As we previously discussed the camera rotation is independent of geometry. Therefore if we apply the camera rotation to the different key images we can achieve a direct mapping between the two plane models. This allows us to project from any camera in one plane to another one in the other plane using the pixel mapping

$$\begin{pmatrix} wi' \\ wj' \\ w \end{pmatrix} = \begin{pmatrix} 1 & 0 & g^{(2)}V_m^{(2)} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{K}_2\mathbf{R}\mathbf{K}_1^{-1} \begin{pmatrix} 1 & 0 & g^{(1)}V_m^{(1)} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}. \tag{5.7}$$

Any further motion to the final synthesis position can be applied on top of this inter-camera mapping projection.

5.2.3 Occlusion ordering between planes

In Sec. 2.2.2 we discussed one of the benefits of our model being a fixed and predictable layer occlusion order. When using multiple camera planes the occlusion ordering is no longer fixed but it is still predictable. Previously with our fronto-parallel layers (Sec. 2.2.2) and even with our angled planes (Sec. 5.4) the layers never crossed so the occlusion ordering was consistent. As the layers for the two plane models will be angled relative to each other the layers will intersect and the occlusion ordering will alter, so the occlusion ordering is depended on the layer Disparity Gradient (DG) and the position in the image plane. Fig. 5.3 shows a 2D example with two planes, $X^{(1)}$ and $X^{(1)}$, and two layers, $Z_l^{(1)}$ and $Z_l^{(2)}$. For cameras along $X^{(1)}$ in region (a) $Z_l^{(1)}$ will occlude $Z_l^{(2)}$ and in region (b) $Z_l^{(2)}$ will occlude $Z_l^{(1)}$. We can calculate the angle, σ, of this intersection relative to the optical axis as

$$\sigma = \tan^{-1}\left(\frac{\Delta V_X}{Z_l^{(1)}} + \frac{Z_l^{(2)}}{\sin\phi\tan\phi} - \cot\phi\right) \tag{5.8}$$

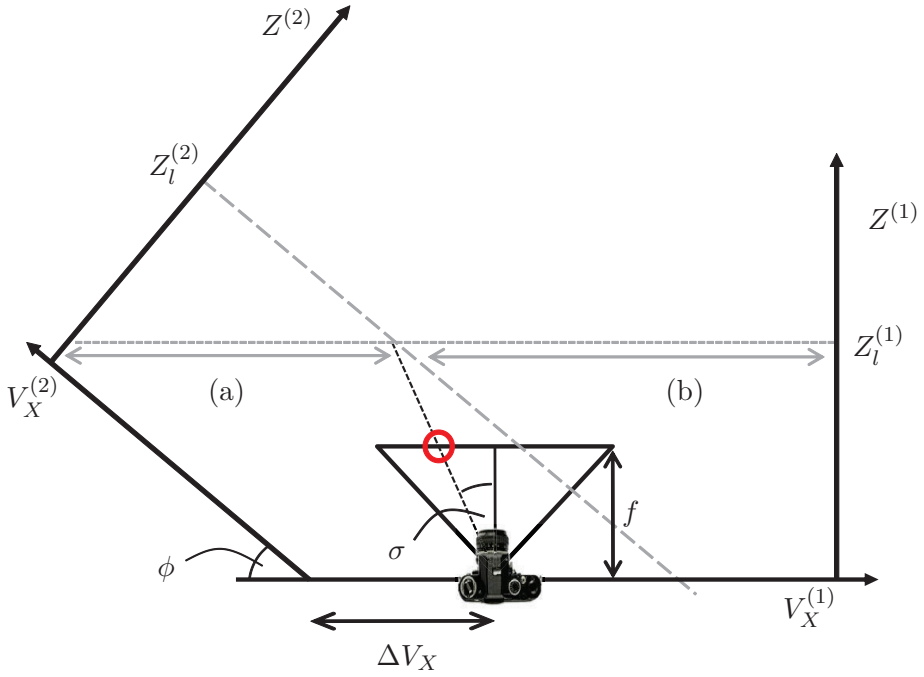


Figure 5.3: Top down view of a multi-plane layer occlusion. In region (a) $Z_l^{(1)}$ will occlude and in region (b) $Z_l^{(2)}$ will occlude. The triangle denotes the image plane and FOV for the camera, and the circle shows the position of this occlusion switchover.

where ΔV_X is the distance of the camera from $V_X = 0$ and ϕ is the plane intersection angle. The position of this switchover in the image plane, Υ, is defined as

$$\Upsilon = f \tan \sigma \tag{5.9}$$

where f is the focal length of the camera. This expands out to,

$$\Upsilon(\Delta V_X, \phi, l^{(1)}, l^{(2)}) = f \left(\frac{\Delta V_X}{Z_l^{(1)}} + \frac{Z_l^{(2)}}{\sin \phi \tan \phi} - \cot \phi \right) \tag{5.10}$$

where $l^{(1)}$ is a layer in the first camera plane and $l^{(2)}$ is a layer in second camera plane. Using the equation for DG (2.7) we can convert this into the form,

$$\Upsilon(\Delta V_X, \phi, l^{(1)}, l^{(2)}) = \Delta V_X g_l^{(1)} + \frac{f^2}{g_l^{(2)} \sin \phi \tan \phi} - f \cot \phi. \tag{5.11}$$

This equation allows us to pre-calculate the occlusion ordering for all the layers in

both planes quickly before we start the pixel level interpolation.

5.2.4 Merging results

Now that we can connect both models and have a consistent occlusion ordering we can use both sides simultaneously. We use an extension of the master-slave approach: we infill any holes and regions in the synthesis that are not visible from one model using the other model. In addition, if available, we can replace low confidence segment geometry in one model with a high confidence segment geometry from the other. This is possible because of the shared key image allowing easy comparison between the segments of each model. As before the master is set based on the proximity to the synthesised result.

5.2.5 Simulation results

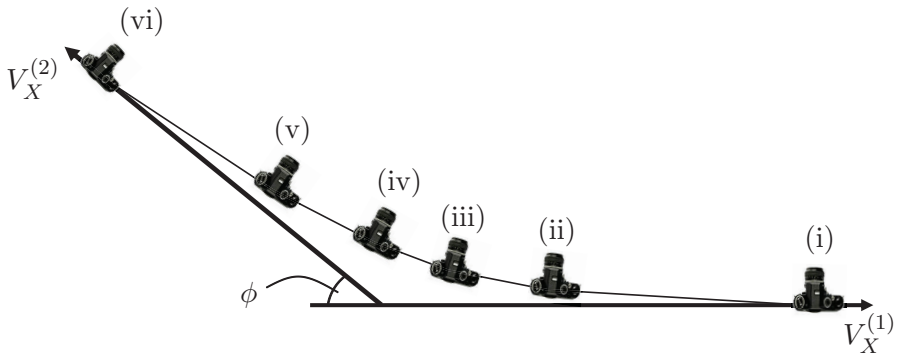


Figure 5.4: The six virtual camera positions for the synthesis results shown in Fig. 5.5, $\phi = 30°$. The camera positions are detailed in Table 5.1.

To demonstrate our ability to transition smoothly between two different plane models we have synthesised several points along a curve between the two camera planes using a synthetic sequence. The camera positions along the curve are illustrated in Fig. 5.4, the details in Table 5.1 and the resultant synthesis is shown in Fig. 5.5.

Starting on the camera plane at (i) near one end of the input sequence we follow the curve, moving towards $V_X^{(1)} = 0$, moving into the scene and starting to rotate from one camera plane to the other. At (ii) the first camera plane is still being used as the master and the movement into the scene and the rotation is slight but by (iii) it is

Table 5.1: This table lists the camera positions used for synthesising the results shown in Fig. 5.5.

Camera	Plane	V_X	V_Z	Rotation
(i)	1	7	0	0°
(ii)	1	3	1	6°
(iii)	1	1	3	15°
(iv)	2	1	3	-15°
(v)	2	3	1	-6°
(vi)	2	7	0	0°

more pronounced, the rotation is half way between the two planes so there is an easy transition to (iv) transitioning to using the second camera plane as the master. The process is revered through (v) until we arrive at the far extent of the second camera plane, (vi). Views (i) and (iv) in Fig. 5.5 are synthesised on each of the two camera planes and the spatial and angular distance between them is very noticable, especially in the background segments. Because the relative angles are similar there is no jarring discontinuity between (iii)(iv) as the system smoothly transitions from one to the next. The synthesis quality is high throughout the transition, despite moving between the two planes and moving out of the camera plane.

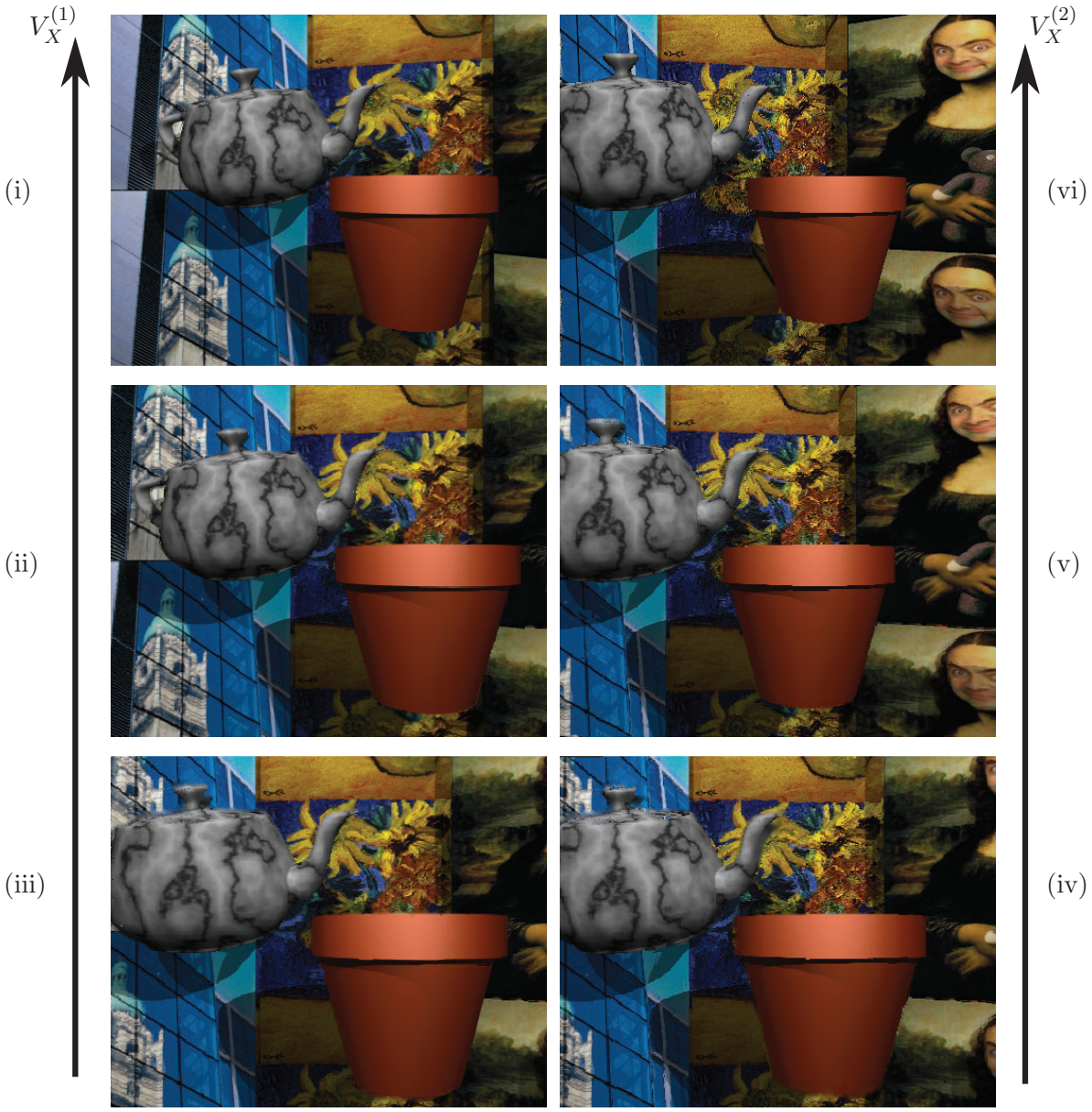


Figure 5.5: The synthesis results for the camera positions detailed in Fig. 5.4 moving between two camera planes. (i) (vi) lie on their respective camera planes with no rotation, (ii) (v) are moved slightly into the scene with a small rotation and (iii) (iv) have moved significantly into the scene with a large rotation.

5.3 Out-of-plane camera positioning

We have shown in Chapter 3 that our algorithm scales from camera lines to camera planes and that the extra information available in for example a Lightfield sequence such as Tsukuba, as illustrated in Fig. 3.2, enables us to improve the layer allocation (see Sec. 3.3). This extra information can also allow us to generalise the output position of the synthesis, allowing the synthesis of images from viewpoints out of the input image plane [86].

5.3.1 Alternative camera paths

Previously we have considered the case of camera movement only within the camera plane as shown in Fig. 5.6(a) which shows a top-down view of the camera motion and four scene points. This results in the linear EPI lines, shown in Fig. 5.6(b), whose gradient depends only on the layer position Z_l.

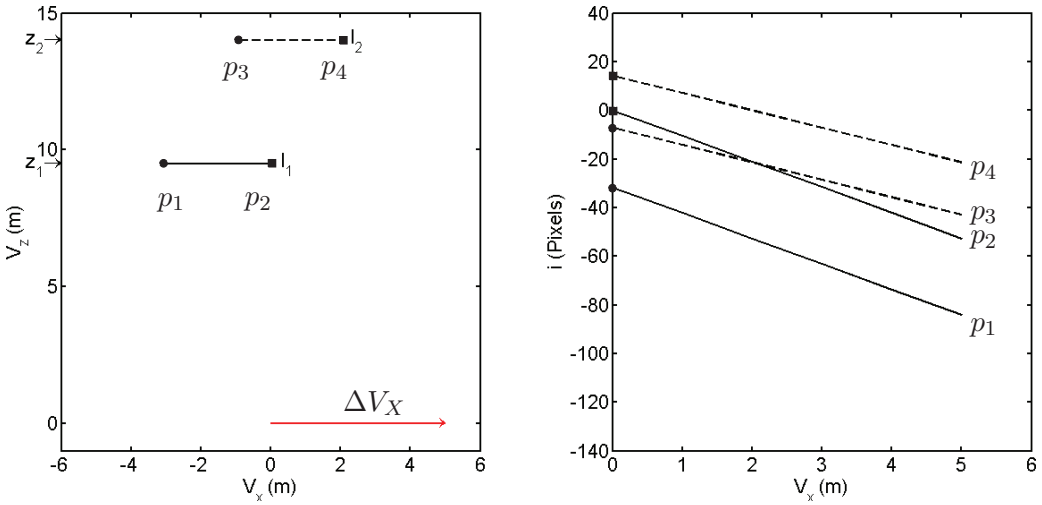


Figure 5.6: In (a) we show a top down view of a simple scene with points, p_1 and p_2 on one layer l_1 and points p_3 and p_4 on another layer l_2. In (b) we show the locus of these points in the camera plane as EPI lines against movement of the camera (the arrow in (a)) in V_X.

In this case the EPI mapping between two images for the pixel position $(i, j) \underset{g}{\mapsto} (i', j')$ with a DG g is described by

$$\begin{pmatrix} i' \\ j' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & gV_m \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}, \tag{5.12}$$

where V_m is the camera motion between the synthesised and key images. The shift in i is only dependent on V_m and g and there is no shift in j.

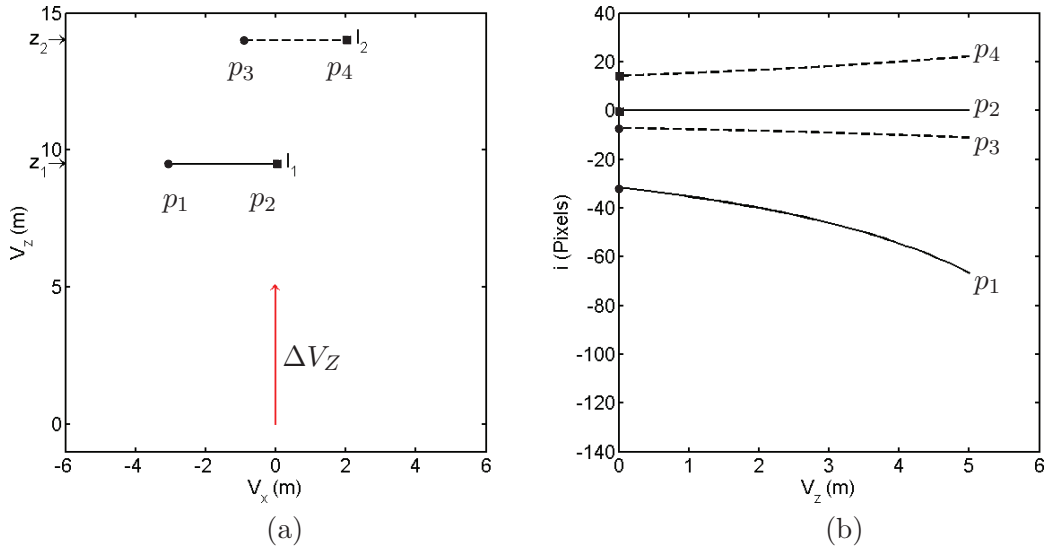


Figure 5.7: In (a) we show a top down view of a simple scene with points, p_1 and p_2 on one layer l_1 and points p_3 and p_4 on another layer l_2. In (b) we show the locus of these points in the camera plane as EPI lines against movement of the camera (the arrow in (a)) in V_Z.

Movement outside the camera plane has different and more complex effects, Fig. 5.7 illustrates the effects of movement in V_Z. In this instance the EPI line gradient is dependent on two factors, the depth of the layer relative to the current camera position (which will change over time) and the value of i. As shown in Fig. 5.8 the position of a point that lies along the optical axis p_1 is unaffected by movement in V_Z, whereas another point p_1 will shift depending on its initial distance from the optical axis. This results in a difference from previous cases is that the gradient is not constant.

This results in the EPI lines moving away with increasing gradient from the optical axis, in this case the EPI mapping is described by

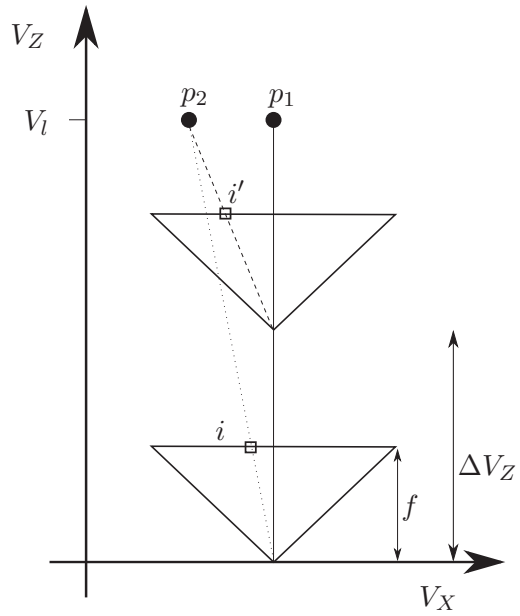


Figure 5.8: Top down view of the camera plane with movement in V_Z, indicating the shift in the intersection of a point from i to i'. If the point is along the optical axis, p_1, there will be no change as the camera moves in V_Z. If the point lies off the optical axis, p_2, the pixel position will shift.

$$\begin{pmatrix} i' \\ j' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{Z_l}{Z_l - \Delta V_Z} & 0 & 0 \\ 0 & \frac{Z_l}{Z_l - \Delta V_Z} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}, \tag{5.13}$$

using the identity (2.7) this can be converted into a DG form

$$\begin{pmatrix} i' \\ j' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{f}{f - g_l \Delta V_Z} & 0 & 0 \\ 0 & \frac{f}{f - g_l \Delta V_Z} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}, \tag{5.14}$$

where ΔV_Z is the distance moved into the scene, f is the focal length, Z_l is the depth layer and g_l is the disparity gradient layer of the point.

The important difference is that the layers are no longer rigid as movement of the camera in V_Z translates into movement in (i, j) for a point based both on its DG value and on its position within the image.

These EPI line predictions can be combined to produce complex behavior such as

that shown in Fig. 5.9 where the camera moves in both V_X and V_Z. Despite the shift in V_Z, when V_Z returns to 0 the EPI lines return to the linear loci that were shown in Fig. 5.6, demonstrating the separable nature of this shift.

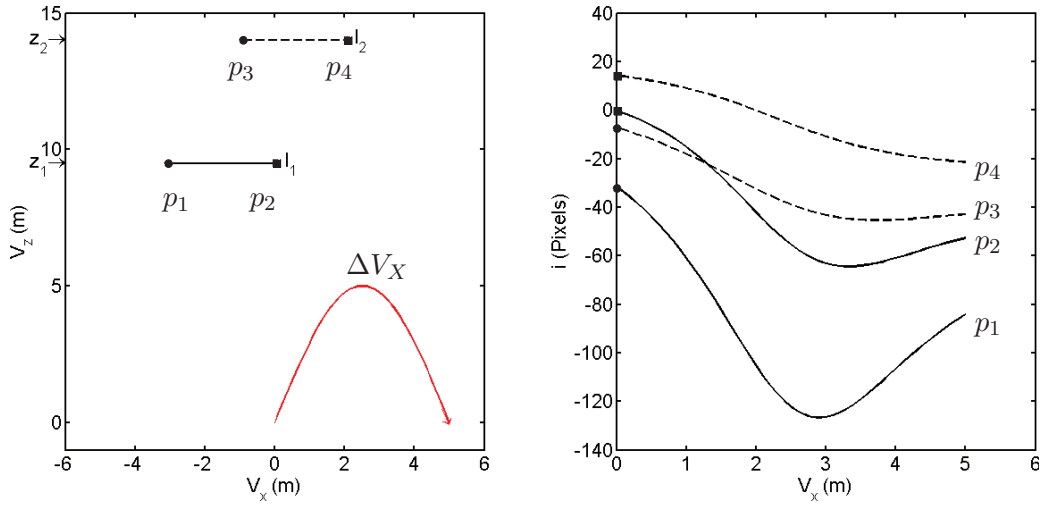


Figure 5.9: In (a) we show a top down view of a simple scene with points, p_1 and p_2 on one layer l_1 and points p_3 and p_4 on another layer l_2. In (b) we show the locus of these points in the camera plane as EPI lines against movement of the camera (the arrow in (a)) in V_X and V_Z.

This EPI mapping can be described by a combination of the previous two mappings (5.12, 5.14) to give

$$\begin{pmatrix} i' \\ j' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{f}{f - g_l \Delta V_Z} & 0 & 0 \\ 0 & \frac{f}{f - g_l \Delta V_Z} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & gV_m \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}. \tag{5.15}$$

5.3.2 Pixel scaling

As discussed in Sec. 5.3.1 the main problem when moving out of the image camera plane and along the Z axis is that pixel shifts within a layer are not consistent. This is demonstrated in the 1D example shown in Fig. 5.10 where we show the canonical approach in which the centres of the pixels in the original image are projected on the synthesised image. Because the shifts have a sub-pixel precision the projection point will not lie exactly on a pixel centre so the pixel assignment is made to the nearest

pixel, leaving us with a one-one pixel mapping, apart from pixels that are occluded by other pixel or the FOV framing, between the original and synthesised image. As a camera moves into the scene, objects become closer and therefore bigger in view and so more pixels are required, so if we maintain a one to one mapping we will only sparsely cover the output image, leaving gaps. As well as gaps between pixels we do not retain the sub-pixel positions so their relative shape has been lost.

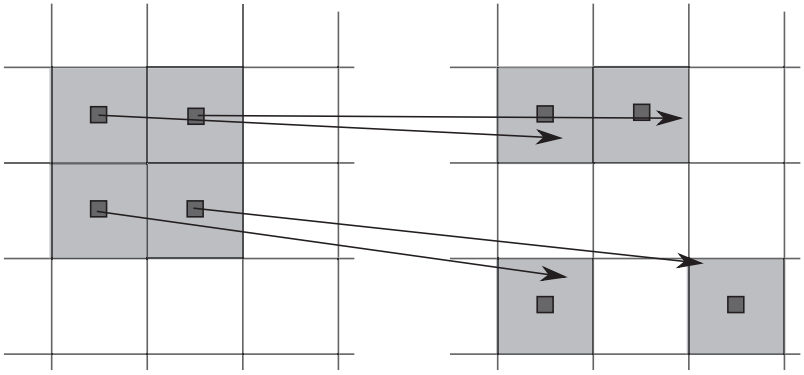


Figure 5.10: Pixel projection assignment showing the sub-pixel precision projection points (arrows) and the rounded pixel assignment points. After the projection there is now a gap between the four pixel cluster and their relative shape has been lost.

A real world example of this can be seen in Fig. 5.11 where we have projected a DG map forward along V_Z and performed no infilling. The periodic cracks within the layer, the black lines, can clearly be seen. The frequency of the cracks is noticeably higher in the foreground regions as the higher disparity means that the pixels will move past the pixel rounding boundaries more frequently. As Fig. 5.12 shows, the resultant synthesis is filled with cracks. When the cracks are in a background layer, Fig. 5.13(i), there is empty space like any other dis-occlusion so there is the possibility for post synthesis infilling as described in Sec. 4.5.2. However when the cracks are in foreground layers they will no longer occlude over layers effectively as regions of the underlying layer will peep through, as shown in Fig. 5.13(ii). Because of these underlying layers filling the cracks (when they should be occluded) our previous scheme of hole filling un-assigned pixels is no longer sufficient, these holes in the foreground layer cannot be detected in such a manner so the infilling techniques will be ineffective.

Our novel solution to this problem is to treat the pixels as squares rather than

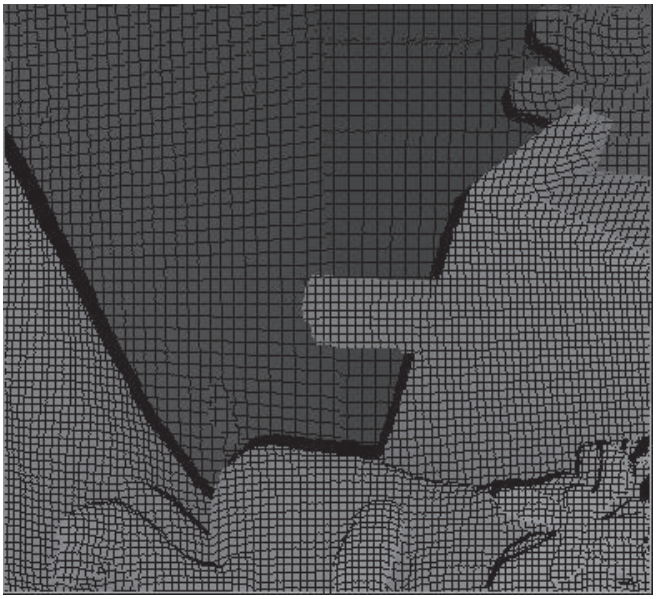


Figure 5.11: Projecting the DG map for a shift in V_Z, with no hole filling.

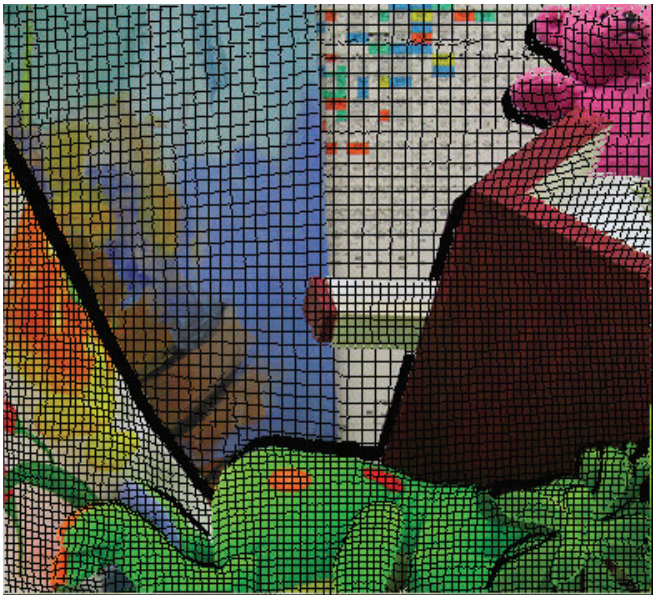


Figure 5.12: Synthesising a new view after a shift in V_Z, with no hole filling.

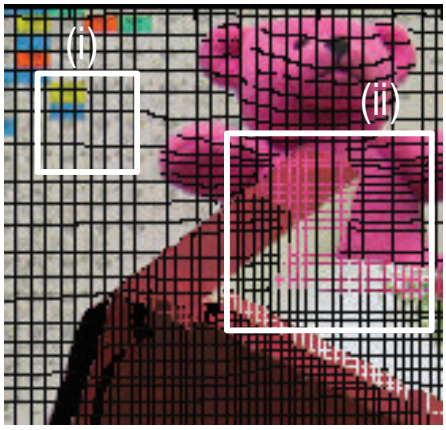


Figure 5.13: Zoomed in region from a new view after a shift in V_Z, with no hole filling.

points, allowing us to project the corners separately which allows pixel scaling. This is demonstrated in Fig. 5.14.

By projecting the pixel corners, as shown in Fig. 5.14, and assigning pixels to anything that lies under the projected pixel area we keep the pixel shape, maintain the sub-pixel information and will never have cracks. This is because adjacent pixels share pixel corner points so the projected pixel areas will always be connected in the output and every pixel within the layer will be covered. When moving into the scene this assignment will lead to a one-to-many pixel mapping as an input pixel may be assigned to more than one output pixel. When moving away from the scene the opposite happens, there is a many to one mapping problem. Our pixel corner projection and inter pixel interpolation deals with both of these issues.

5.3.3 Real world example

Fig. 5.15 shows the results of changing V_Z for the output image, with increasing V_Z from left to right. Note that this is not a zoom but rather a true movement into the scene with resulting occlusions by foreground objects. The layer and position dependent scaling and warping can clearly be seen in the different relative sizes of objects within the scene as you move from left to right, foreground objects drastically change size while the background is largely unaffected. It should be noted that even with a large amount of movement into the scene the output quality is still maintained.

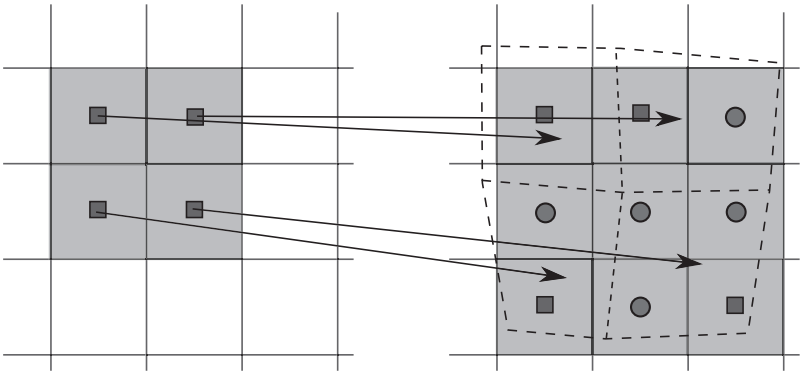


Figure 5.14: Pixel projection assignment showing the pixel centre projections (solid arrows) and corner projections (dotted lines). The shaded region show the pixel assignment areas, squares are used to denote the original pixels and circles the extra pixels cause by the pixel scaling. By using these sub-pixel precise areas as a guide to pixel assignment we maintain the pixel position shape and leave no gaps.

Figure 5.15: These images show the results of moving the position of the output viewpoint in V_Z as well as V_X or V_Y. V_Z increases left to right.

5.4 Angled layers

One of the key assumptions of Plenoptic theory is that the scene can be modelled as a set of fronto-parallel planes. We have shown in Sec. 3.5 that although there are some errors due to inconsistencies between this assumption and reality, it nevertheless is able to achieve good results over the sequences tested. However as sequences diverge from the fronto-parallel assumption these errors will increase. By relaxing the flat layer constraint for a restricted number of carefully chosen angled planes we can improve performance and model a greater variety of scenes without violating any of our other assumptions.

5.4.1 Angled layer model

The important constraint we need to adhere to is that any adjustment to a segment layer angle is independent of its surrounding segments and that the layout occlusion ordering is preserved. In Fig. 5.16 two layers (solid lines) are shown, g_l and g_{l-1}.

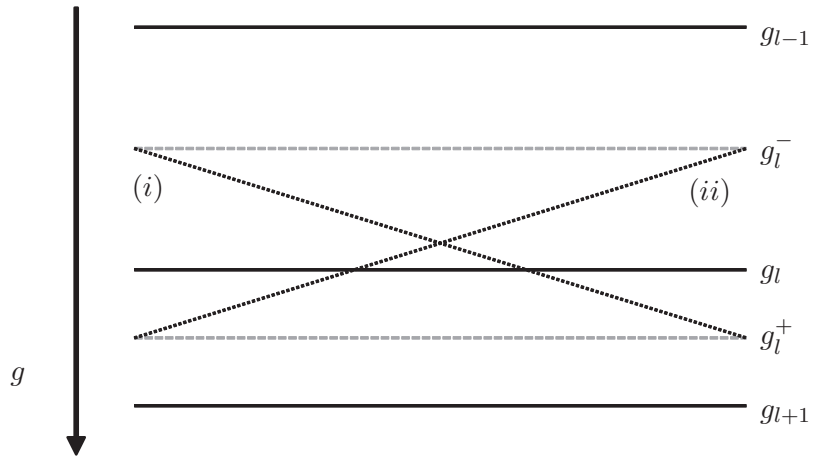


Figure 5.16: A diagram showing the layer (solid lines) g_l, the preceeding layer g_{l-1} and the following layer g_{l+1}. The assignment limits (dashed lines) g_l^+, g_l^-, and the two alternative angled layers (dotted lines) (i) and (ii).

Disparities will be assigned to the closest layer, so the assignment boundary for each layer, g_l, can been defined as an upper bound $g_l^+$where

$$g_l^+ = \left(\frac{g_{l+1} - g_l}{2} \right) \tag{5.16}$$

and a lower bound g_l^- where

$$g_l^- = \left(\frac{g_l - g_{l-1}}{2} \right) \tag{5.17}$$

These are indicated in Fig. 5.16 by the dashed lines. For each layer we now also allow two angled layers, Fig 5.16(i) and (ii), each layer incurs a fixed calculation cost and as we will explain in Sec. 5.5 there are rapidly diminishing returns from using more angle possibilities. Even with only two angle possibilities significant improvements are obtained. Each layer is defined as going from one assignment boundary to the next over the entire width of the segment.

The first angled layer, Fig 5.16(i), has the DG value ϱ given by

$$\varrho(S_n,\, i,\, g_l) = g_l^+ - \left(\frac{\left(g_l^+ - g_l^- \right) \left(i - \min\limits_i(S_n) \right)}{\max\limits_i(S_n) - \min\limits_i(S_n)} \right) \tag{5.18}$$

where $\max\limits_i(S_n)$ is the largest and $\max\limits_i(S_n)$ is the lowest i value in segment S_n.

The angled plane for the alternative angle, Fig 5.16(ii), has the DG value $\bar{\varrho}$ where

$$\bar{\varrho}(S_n,\, i_k^{(n)},\, g_l) = g_l^- + \left(\frac{\left(g_l^+ - g_l^- \right) \left(i - \min\limits_i(S_n) \right)}{\max\limits_i(S_n) - \min\limits_i(S_n)} \right). \tag{5.19}$$

5.4.2 Assigning angled layers

We can use the methods described in Sec. 3.2.5 to test the original fronto-parallel layer assignment against the two angled potential layer assignments and choose the best match :

$$\hat{\varrho}_n = \underset{g}{\operatorname{argmax}} \left(\; \bar{\epsilon}(S_n, \hat{g}_n), \breve{\epsilon}(S_n, \varrho), \breve{\epsilon}(S_n, \bar{\varrho}) \; \right), \tag{5.20}$$

where the confidence measure now becomes

$$\breve{\epsilon}\,(S_n,\varrho) = \frac{M\left(\sum_{k=0}^{K_n-1} O_k^{(n)}\right)\log\left(\sum_{k=0}^{K_n-1} O_k^{(n)}\right)}{\sum_{k=0}^{K_n-1}\sum_{m=1}^{M-1} O_k^{(n)}\left|I_0\left(i_k^{(n)},j_k^{(n)}\right) - I_m\left(i_k^{(n)}+\varrho(S_n,i_k^{(n)},g_l)V_m,j_k^{(n)}\right)\right|}, \quad (5.21)$$

here K_n is the total number of pixels within the segment S_n which is being evaluated over M images. I_0 is the current key image and I_m is the target image. $\varrho(S_n,\, i_k^{(n)},\, g_l)$ is the proposed pixel dependent DG and V_m is the V_x position of image m so the $\breve{\epsilon}\,(S_n,\varrho)$ value is a sum over all available images. As before to account for occlusions we use the visibility mask $O_k^{(n)}$ where

$$O_k^{(n)} = \begin{cases} 1 & \text{if } I_m(i_k^{(n)}+\varrho(S_n,i_k^{(n)},g_l)V_m,j_k^{(n)}) \text{ is visible;} \\ 0 & \text{if } I_m(i_k^{(n)}+\varrho(S_n,i_k^{(n)},g_l)V_m,j_k^{(n)}) \text{ is occluded.} \end{cases} \quad (5.22)$$

By constructing the angled layers this way, we can still easily calculate layer assignments without violating any of our other constraints or assumptions. The effectiveness of using angled planes will be evaluated in Sec. 5.5.

5.5 Numerical simulations

For our evaluation we used the sequences [74, 75] shown in Table 5.2. The key images were segmented using the Mean Shift (MS) algorithm [69, 76, 82]. These sources are provided with Ground Truth (GT) DG maps with a granular resolution of $\frac{1}{16}$ pixel/ΔV_X for all cases except for Barn1 which has a granular resolution of $\frac{1}{32}$ pixel/ΔV_X. This results in the calculated maximum possible image disparity measure $\widetilde{DG}$ for the GT DG maps.

The performance of the angled planes can be assessed by studying the error when the layer model is used to estimate the DG map against the 255 layer GT results provided with the sequences.

Fig. 5.17 shows the error in estimating the DG map using our angled planes method

Table 5.2: This table lists the sequences [74, 75] that were used in our evaluation of the use of angled layers.

Sequence	Image resolution	Number of images	$\widetilde{DG}$
Teddy	450×375	9	16
Cones	450×375	9	16
Barn1	432×381	7	8
Sawtooth	432×380	7	16

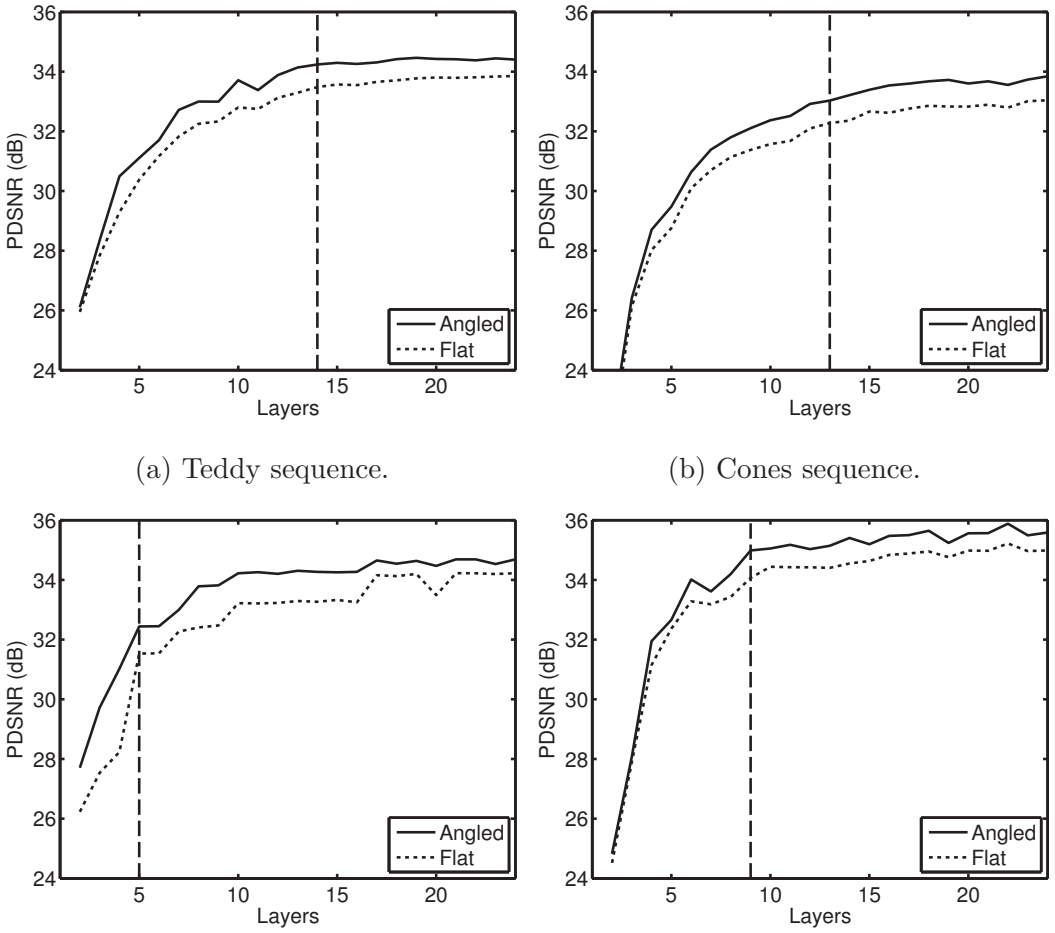
against the GT map for each of the sequences. We have measured the similarity of the two DG maps using a Peak Disparity Signal to Noise Ratio (PDSNR) measure

$$PDSNR = 10 \cdot \log_{10} \left(\frac{\widetilde{DG}^2}{MSE} \right) \tag{5.23}$$

where Mean Squared Error (MSE) is the squared pixel difference between the GT DG map, DGM_{GT}, and the layer model DG map, DGM_M, we are investigating giving us

$$MSE = \tfrac{1}{I \cdot J} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} \left| DGM_{GT}(i,j) - DGM_M(i,j) \right|^2, \tag{5.24}$$

where $\widetilde{DG}$ is the maximum disparity value possible for the scene and I and J are the image width and height, as detailed in Table 5.2.



(a) Teddy sequence.

(b) Cones sequence.

(c) Barn1 sequence.

(d) Sawtooth sequence.

Figure 5.17: The assignment error from applying the angled (solid) or flat (dotted) layer models to the DG GT map. The calculated $L_{\min}$ for each sequence is shown by the vertical dotted line.

In all cases using angled layers causes a significant increase in performance, initially there is only a small increase but this grows bigger as more layers are used, until the improvement peaks during the plateau stage. This increase is particularly evident in the highly angled Barn sequence, Fig. 5.17(iii) and less so in the relatively flat Cones sequence, Fig. 5.17(ii).

Fig. 5.18 shows the percentage of segments that have been assigned to an angled layer against the total number of available layers. Initially only a few, 16%, of layers are assigned this way due to the large angle only matching a few segments. However

with a few more layers the percentage of assigned layers rapidly climbs to over 85 %, the remaining segments are of fronto-parallel regions that will never be assigned to angled layers. This shows that the majority of segment assignments can be improved with angled layers.

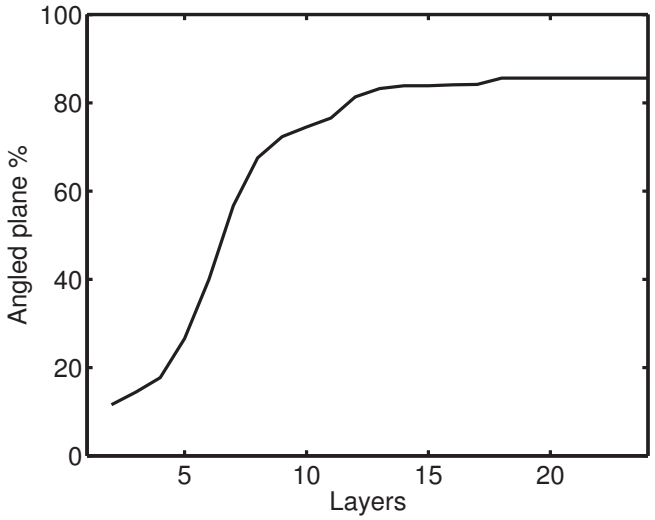


Figure 5.18: A graph showing what percentage of the DG map is constructed using angled planes against the number of layers in the model for the Teddy sequence.

We have also evaluated our angled layers method against real world data. As before we used the 'leave q out' method of evaluation in which only every $(q+1)^{th}$ image is included in the input image set. These are used to synthesize one of the omitted images for which the ground truth is known. In all cases we use two key images at either end of the EPI source and an infilling algorithm was used to fill any holes with the lowest adjacent disparity as described in Sec. 4.5.2.

We have used the Teddy sequence, as shown in Table 5.2 and evaluated the results using the Peak Signal to Noise Ratio (PSNR) measure (4.12). The image synthesis results in Fig. 5.19 show similar characteristics to the previous DG map results in Fig. 5.17, with a small increase in quality for very low and very high numbers of layers and a significant increase in quality before and around the $L_{\min}$ point. This is the most important region to improve performance.

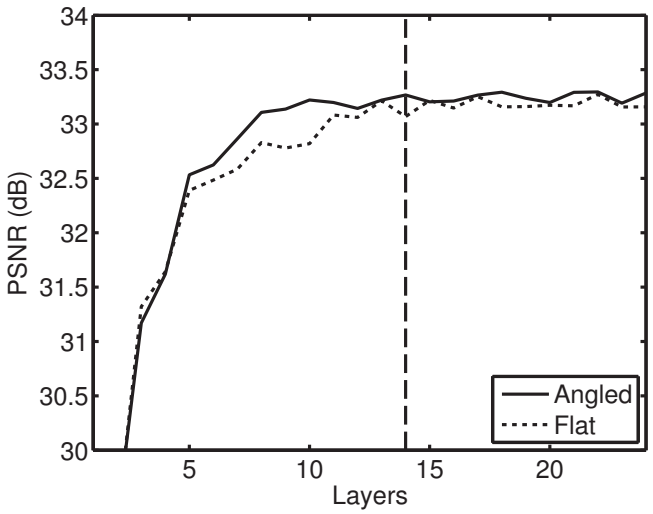


Figure 5.19: Comparing the rendering quality of the angled (solid) against the flat (dotted) layer models on real world data, Teddy sequence. The vertical dashed line represents the $L_{\mathbf{min}} = 14$ for the dataset.

5.6 Conclusions

We have shown how our algorithm is flexible enough to allow modelling multiple planes and that as well as giving us greater freedom relaxing our assumptions can lead to higher quality output synthesis. To allow this we have extended our algorithm to accurately model more complex scenes by allowing camera rotation, angled planes and virtual camera synthesis positions out of the input camera plane. The numerical simulations show that even for our existing single plane scenes the benefits are still apparent, in particular the angled planes lead to a perceived smoother more realistic motion when synthesising multiple consecutive views. We have detailed exactly what changes needed to be made to our algorithm and why this does not violate any of the conditions that allow us the benefits of using Plenoptic theory as a guide.

Chapter 6

Conclusions

6.1 Summary of thesis achievements

This thesis has been concerned with an Image Based Rendering (IBR) approach to view synthesis: the generation of arbitrary new views of a scene from a set of existing views. IBR is an attractive method for view synthesis as it can give near photo-realistic results with low complexity and limited resources. By considering the scene in terms of light rays emanating from the scene rather than the objects themselves, we can frame the situation in terms of a traditional sampling and interpolation problem where new images are generated by interpolating between existing images by sampling the light rays. This is important because it gives us a theoretical framework to understand the tradeoff between geometric completeness and the number of images necessary to maintain a consistent quality.

Plenoptic theory shows that, provided certain assumptions are met, alias-free rendering can be achieved by a layer-based model of the scene geometry in which the layers are spaced uniformly and by using a number of layers that exceeds the minimum, $L_{\min}$, given in (2.10). In practice however, these assumptions, which include the absence of occlusions, an infinite field of view and a perfect low-pass filter may not hold true. The further you diverge from these assumptions the more aliasing is inevitable.

We have presented a novel layer based algorithm for IBR. Our method uses Plenoptic sampling theory to infer the right amount of geometric information required for

artefact-free rendering. Moreover it takes advantage of the knowledge of the typical structure of multiview data in order to perform a fast occlusion-aware non uniformly spaced layer extraction. The rendering is improved by using a probabilistic interpolation approach and by an effective use of key images in a scalable master-slave configuration. Numerical results demonstrate that the algorithm is fast and yet is only 0.24 dB away from the ideal performance achieved with the ground-truth knowledge of the 3D geometry of the scene of interest. We have shown that the Plenoptic framework is applicable for real world cases and that a layer based model does not lead to any loss in output quality.

We have also shown that the Plenoptic theoretical framework is applicable to real world cases since a layer based model does not lead to any loss in output quality and the number of layers required is correctly predicted by the theory. This indicates that despite several assumptions of Plenoptic theory not being valid in real world cases it is still an effective guide for producing high quality synthesised outputs.

Specifically we have shown that our novel non-uniformly spaced layer placement model gives a major improvement in quality and robustness over the uniform spacing layer model. Moreover, our layer extraction algorithm is independent of the segmentation method used. We have also shown that many mis-assignments or inconsistencies can be solved by smoothing and splitting the segments. The rendering is improved by using a probabilistic interpolation approach and by an effective use of key images in a scalable master-slave configuration. We have shown that our algorithm performs well in comparison with an alternative method.

Finally we have demonstrated the flexibility of our system by showing how it can be extended to model more general cases of synthesis and how relaxing our assumptions can lead to higher quality output synthesis. Modelling angled planes and allowing multi-planar geometry allows us to model scenes more accurately, improving rendering quality, while maintaining all the advantages that using Plenoptic theory as a guide bestows.

6.2 Future research

In conclusion we will present some possible directions for future research.

6.2.1 Depth and image camera fusion

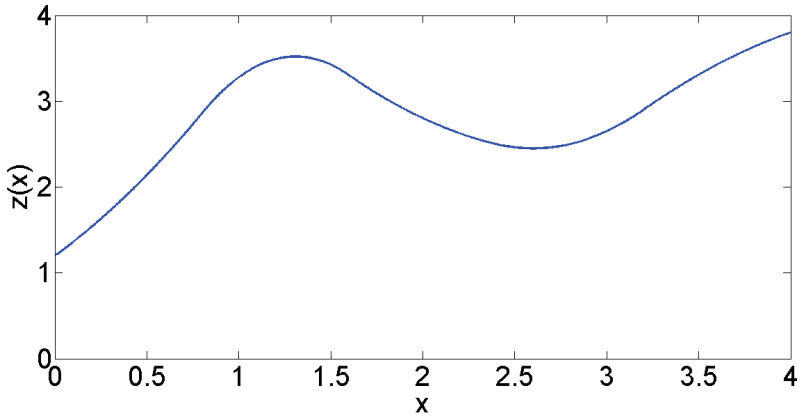
We have discussed the importance of depth based geometry for reducing the number of images required for IBR, there are many different methods that have been proposed for multi-view stereo vision algorithms, for example [71, 87, 88], and in Chapter 3 we described our approach to calculating this information from the input images. An alternative approach is using a dedicated depth sensing camera, traditionally these have often been expensive and of low resolution, however with advances in the area of Structured Light (SL) [75,89–91], Single Depth Single Colour (SD-SC) [92,93], Multiple Depth Multiple Colour (MD-MC) systems [94–96], and increasing commoditization they have broken out of their niche and mass-market alternatives, such as the Microsoft Xbox Kinect, are available. These provide cheap, accurate depth sensing with a higher resolution and the additional benefit of providing both the depth image and a matching image source, this pairing is often referred to as a Red, Green, Blue and Depth (RGB-D) image. This has encouraged the investigation into hybrid depth and image based schemes, often referred to as Depth Image Based Rendering (DIBR). This approach can easily be modified to use (possibly low resolution) depth information from a depth-camera as an additional input. In this way we can combine the real time, accurate but potentially incomplete and low resolution depth-map information with our existing slower but complete and high resolution image based methods to produce a faster, more accurate and higher resolution result.

6.2.1.1 Depth image based rendering

In previous work it has been shown for synthetic, [97], and real world data, [14], that the Minimum Sampling Criterion (MSC) holds true for these large sample cases (50+ images and depth maps), when dealing with scenes with a smooth, continuous non-occluding surfaces and a known geometry. Simple easily measurable geometry was

used to aid in the measurement and calibration. Using depth cameras more complex curve geometries could be constructed and sampled, Fig. 6.1(a), and the resultant depth map can be converted into a surface curve, as shown in Fig. 6.1(b). By using a depth camera rather than stereo matching methods we can quickly measure hundreds of separate depth maps for a scene that is not conducive for traditional stereo matching, the acquisition rig for capturing simultaneous RGB-D images is shown in 6.2.

(a) Depth image.



(b) Surface depth

Figure 6.1: The inverse depth map (brighter is closer) of a curved plane captured from a depth camera is shown in (a) and the corresponding surface curve extracted in (b).

This method can be extended to expand our own work on cluttered occluding scenes, by using commodity depth sensing device we can take a series of RGB-D images in an

Figure 6.2: Acquisition rig for capturing high resolution simultaneous RGB-D images along an EPI line.

image plane. In the example shown in Fig. 6.3 we use a RGB-D camera and a fixed movement rig to take a 10×10 grid of images.

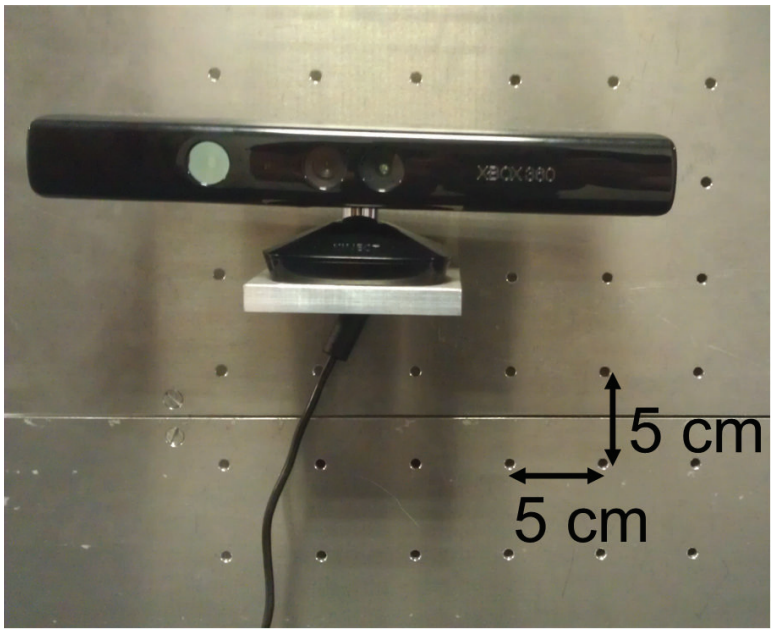


Figure 6.3: Camera plane RGB-D acquisition rig.

An example image output from this setup is shown in Fig. 6.4(a) and the matching depth map is shown in Fig. 6.4(b), any holes in this depth map are infilled using the

techniques described in Sec. 4.5.2.

(a) Colour image

(b) Depth image

Figure 6.4: The DG map (a) corresponds to the image (b). Any holes in the image are infilled using pixels from the same layer.

Using a depth camera allows us to generate a Lightfield with 100 accurate RGB-D images which gives us greater flexibility in our investigations.

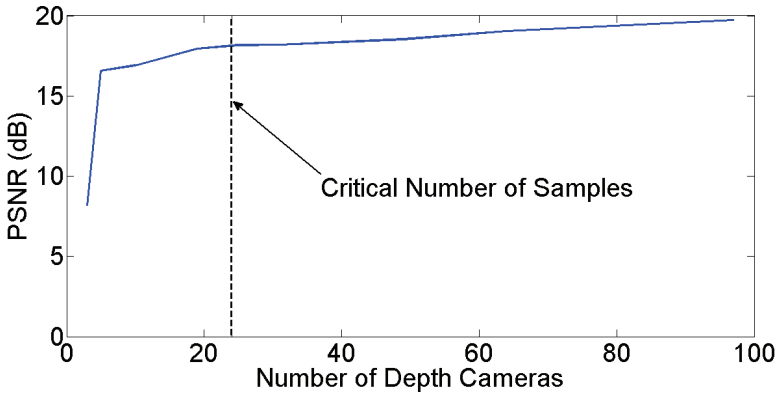
6.2.1.2 Provisional results

In [19] we start to investigate these new possibilities, using a similar frame-work to the plenoptic function, called the $Pantelic function$, [14], where the multi-view depth images represent samples of the Pantelic function. Using this function an initial analysis of multi-view depth images can be made. The preliminary results in Fig. 6.5(a) for the single surface case, as shown in Fig. 6.1, shows a required minimum number of depth maps in a similar fashion to Plenoptic sampling. This finding is also true of the more complex case with multiple occluding surfaces, Fig. 6.5(a), for the case described in Fig.6.4. These results are intriguing but several open questions remain : How many depth cameras are required to describe the scene geometry and can this be adapted to account for the distribution of objects similarly to Sec. 3.2.4? What is the relationship between the required number of depth cameras and colour cameras? Can extra information in one compensate for insufficient information in another? What effect the complexity of the scene at a micro level (within the scene objects) have on this relationship?

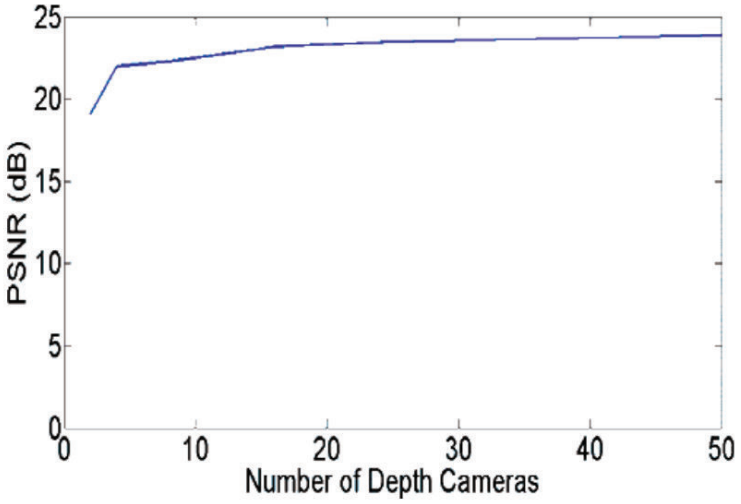
6.2.1.3 Improving depth map accuracy

One method for active depth sensing is to project a known pattern onto a scene, often in the Infra Red (IR) spectrum to avoid interference from visible light, and example is shown in Fig. 6.6(a). As the pattern is known and the separation between the pattern projector and the receiving camera is fixed, this pattern can be used to calculate the depth map for the scene, Fig. 6.6(b). Although this is a fast and on the whole accurate method for measuring the scene depths it does have some issues.

If a region is saturated by IR light, due to outside sources or its proximity to the projector, or there is no visible pattern in a region this method fails. Moreover if there are occluding objects then there will consistently be 'shadowed' regions, (as shown by the pure white regions of Fig. 6.6(b)). An additional issue is that the depth is only measured at these points which are sparsely scattered around the scene, the resulting depth map needs to be interpolated. Unfortunately these problems tend to occur at and



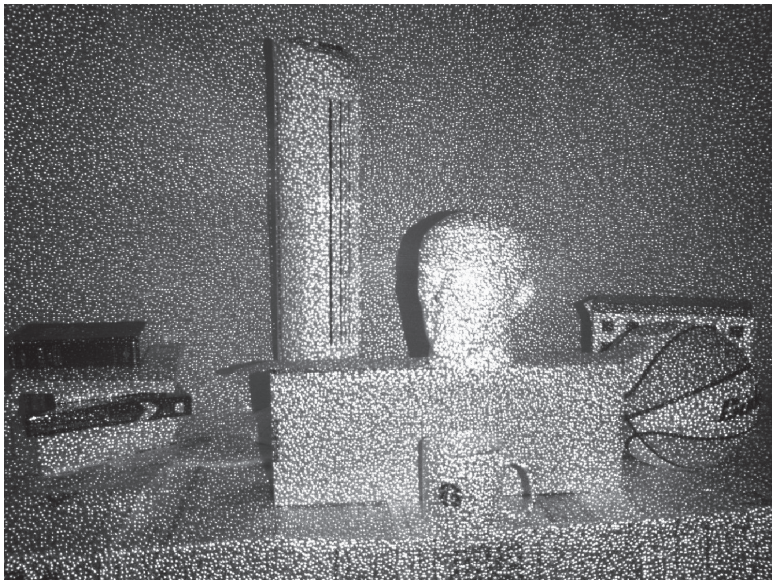
(a) Colour image



(b) Depth image

Figure 6.5: The DG map (a) corresponds to the image (b). Any holes in the image are infilled using pixels from the same laye.

around the edges of scene objects which is also where IBR is most vulnerable to geometry based errors. Some work has been done into investigating running multiple depth camera working together to cover these blind spots, using existing depth map fusion techniques, [98], or moving the camera and combining the results, such as Simultaneous Location And Mapping (SLAM), [99,100], or super-resolution techniques [101–103]. However there are several interesting possible avenues of exploration: Could the raw data be fused with our existing algorithm to produce a hybrid system with increased speed and robustness? There are various methods for reconstructing shape and shade from sparse data, could these be applied to improve the edge performance of the depth map extraction? Could the raw IR data and the image data be combined to improve



(a) IR image

(b) Depth image

Figure 6.6: The raw IR view of the projected points is shown in (a) the reconstructed depth estimate (brighter is closer) is shown in (b) with the holes shown in white.

the output depth map robustness?

6.2.2 Unconstrained camera positions

6.2.2.1 Mobile applications

With our focus on a fast, robust algorithm with high quality outputs while minimising the amount of calculated geometry it seems a natural step to implement our algorithm on a mobile platform. Especially with the aid of some dedicated depth sensing hardware and the increasing sophistication and power of modern mobile devices there are great possibilities in this area. One area of particular interest is single sensor compressed sensing, for example the Compressive Depth Acquisition Camera (CODAC) [104, 105], which would allow us to use smaller cheaper sensors more suitable for a mobile phone.

6.2.2.2 Extending to the complete Lightfield case

We have already shown how our algorithm can be extended to a multi-plane version, it is possible to extend this to the complete Lightfield case where the entire scene would be enclosed. This would greatly extend the flexibility of the output position giving a truly unconstrained synthesis position. This omni-directional capture and synthesis is very applicable to the world of sports, where there are already cameras all the way round a pitch and allowing a viewer to adjust their viewing position and direction is a popular area of research [106–109]. There would be some adjustment necessary to take advantage of some of the features of such a set-up, such as the flat pitch perpendicular to the players and crowd, but our algorithm is flexible enough for these changes to be made to the internal geometric model. Alternatively this approach is just as applicable to smaller scenes, potentially utilising user-generated content from smart-phones or Closed Circuit TV (CCTV), allowing people to create a much more comprehensive and immersible record.

Bibliography

[1] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Trans. Pattern Anal and Machine Intell*, vol. 16, no. 2, pp. 150–162, Feb. 1994.

[2] S. Yaguchi and H. Saito, "Arbitrary viewpoint video synthesis from multiple uncalibrated cameras," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 1, pp. 430–439, Feb. 2004.

[3] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image Based Rendering*. Springer, 2007.

[4] C. Zhang and T. Chen, "A survey on image-based rendering – representation, sampling and compression," *Signal Processing: Image Communication*, vol. 19, no. 1, pp. 1–28, 2004.

[5] M. Tanimoto, "FTV: Free-viewpoint television," *Signal Processing: Image Communication*, vol. 27, no. 6, pp. 555–570, 2012.

[6] A. Lumsdaine and T. Georgiev, "Full resolution lightfield rendering," *Indiana University and Adobe Systems, Tech. Rep*, 2008.

[7] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, "Plenoptic sampling," in *Proc. Intl. Conf. on Comp. Graphics and Interactive Techniques*. ACM Press, 2000, pp. 307–318.

[8] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," in *Proc. SIGGRAPH*, Los Angeles, 1995, pp. 39–46.

[9] M. Do, D. Marchand-Maillet, and M. Vetterli, "On the bandwidth of the plenoptic function," *IEEE Trans. Image Proc.*, vol. 21, no. 2, pp. 708–717, Feb. 2012.

[10] J. Shade, S. Gortler, L.-w. He, and R. Szeliski, "Layered depth images," in *Proc. SIGGRAPH*, New York, 1998, pp. 231–242.

[11] H.-Y. Shum, J. Sun, S. Yamazaki, Y. Li, and C.-K. Tang, "Pop-up light field: An interactive image-based modeling and rendering system," *ACM Trans. Graph.*, vol. 23, no. 2, pp. 143–162, 2004.

[12] Y. Li, X. Tong, C.-K. Tang, and H.-Y. Shum, "Rendering driven depth reconstruction," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Processing*, vol. 4, Apr. 2003, p. 780.

[13] A. Chebira, P. Dragotti, L. Sbaiz, and M. Vetterli, "Sampling and interpolation of the plenoptic function," *Proc. Intl. Conf. Image Processing*, 2003.

[14] C. Gilliam, M. Brookes, and P. L. Dragotti, "Image-based rendering and the sampling of the plenoptic function," in *Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering*, F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, Eds. Wiley, 2013, ch. 12, pp. 231–248.

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

[16] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH*, New York, 1996, pp. 31–42.

[17] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. SIGGRAPH*, New York, 1996, pp. 43–54.

[18] J. Pearson, M. Brookes, and P. L. Dragotti, "Plenoptic layer-based modelling for image based rendering," in *IEEE Trans. on Image Processing*, vol. Special Issue on 3D video, 2013.

[19] C. Gilliam, J. Pearson, M. Brookes, and P. L. Dragotti, "Image based rendering with depth cameras: How many are needed?" in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Processing*, 2012.

[20] J. Pearson, P.-L. Dragotti, and M. Brookes, "Accurate non-iterative depth layer extraction algorithm for image based rendering," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Processing*, May 2011, pp. 901–904.

[21] E. H. Adelson and J. Bergen, "The plenoptic function and the elements of early vision," *Computational Models of Visual Processing*, pp. 3–20, 1991.

[22] A. Criminisi, S. B. Kang, R. Swaminathan, R. Szeliski, and P. Anandan, "Extracting layers and analyzing their specular properties using epipolar-plane-image analysis," *Comp. Vis. Image Underst.*, vol. 97, no. 1, pp. 51–85, Jan. 2005.

[23] J. Berent and P. L. Dragotti, "Plenoptic manifolds," *IEEE Signal Processing Mag.*, vol. 24, no. 6, pp. 34–44, 2007.

[24] K. Li, S. Wang, M. Yuan, and N. Chen, "Scale invariant control points based stereo matching for dynamic programming," in *Proc. Intl. Conf. Elec. Meas. Inst.*, 2009, pp. 769–774.

[25] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, Jun. 2005, pp. 807–814.

[26] ——, "Improvements in real-time correlation-based stereo vision," in *Proc. IEEE Stereo and Multi-Baseline Vision*, 2001, pp. 141–148.

[27] O. Gangwal and R.-P. Berretty, "Depth map post-processing for 3D-TV," in *Int. Conf. Consumer Electronics*, Jan. 2009, pp. 1–2.

[28] S.-Y. Kim, E.-K. Lee, and Y.-S. Ho, "Generation of ROI enhanced depth maps using stereoscopic cameras and a depth camera," *IEEE Trans. Broadcasting*, vol. 54, no. 4, pp. 732–740, Dec. 2008.

[29] S. Ince, E. Martinian, S. Yea, and A. Vetro, "Depth estimation for view synthesis in multiview video coding," in *Proc. 3DTV Conference*, May 2007, pp. 1–4.

[30] H. Hirschmuller, "Real-time correlation-based stereo vision with reduced border errors," in *Proc. Journal Comp. Vision*, vol. 47, no. 1, 2002, pp. 229–247.

[31] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. SIGGRAPH.* New York, NY, USA: ACM, 2004, pp. 600–608.

[32] G. Zhang, J. Jia, T.-T. Wong, and H. Bao, "Recovering consistent video depth maps via bundle optimization," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 23-28 2008, pp. 1 –8.

[33] ——, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Anal and Machine Intell*, vol. 31, no. 6, pp. 974–988, 2009.

[34] H. Tao and H. Sawhney, "Global matching criterion and color segmentation based stereo," in *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.*, 2000, pp. 246–253.

[35] Z. Liu, Z. Han, Q. Ye, and J. Jiao, "A new segment-based algorithm for stereo matching," in *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*, 2009, pp. 999–1003.

[36] Z. Liu, Q. Ye, L. Ke, and J. Jiao, "A progressive region-merging algorithm for stereo matching," in *Information, Computing and Telecommunication, 2009. YC-ICT '09. IEEE Youth Conference on*, 2009, pp. 142 –145.

[37] Z. Zhai, Y. Lu, and H. Zhao, "Stereo matching for larger disparity range using gradient information and adjacent segments cooperative optimization," in *Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on*, 20-22 2008, pp. 526 –530.

[38] C. L. Zitnick and S. B. Kang, "Stereo for image-based rendering using image over-segmentation," *Int. J. Comput. Vision*, vol. 75, no. 1, pp. 49–65, Oct. 2007. [Online]. Available: http://dx.doi.org/10.1007/s11263-006-0018-8

[39] S. Chan, H.-Y. Shum, and K.-T. Ng, "Image-based rendering and synthesis," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 22–33, Nov. 2007.

[40] S.-C. Chan, Z.-F. Gan, K.-T. Ng, K.-L. Ho, and H.-Y. Shum, "An object-based approach to image/video-based synthesis and processing for 3-D and multiview televisions," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 821–831, 2009.

[41] M. P. Kumar, P. H. S. Torr, and A. Zisserman, "Learning layered motion segmentations of video," in *Proc. Intl. Conf. Computer Vision*, 2005.

[42] X. Huang, L. Wang, J. Huang, D. Li, and M. Zhang, "A depth extraction method based on motion and geometry for 2d to 3d conversion," in *Third International Symposium on Intelligent Information Technology Application*, vol. 3, 2009, pp. 294–298.

[43] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," *Signal Processing: Image Communication*, vol. 24, pp. 65–72, 2009.

[44] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Distance dependent depth filtering in 3D warping for 3DTV," in *Proc. IEEE Workshop on Multimedia Signal Processing*, Oct. 2007, pp. 312–315.

[45] L. Zhang and W. Tam, "Stereoscopic image generation based on depth images for 3D TV," *Broadcasting, IEEE Transactions on*, vol. 51, no. 2, pp. 191–199, Jun. 2005.

[46] M. Do, Q. Nguyen, H. Nguyen, D. Kubacki, and S. Patel, "Immersive visual communication," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 58–66, Jan. 2011.

[47] D. Min, J. Lu, and M. Do, "Depth video enhancement based on weighted mode filtering," *IEEE Trans. Image Processing*, vol. 21, no. 3, pp. 1176–1190, 2012.

[48] K. Takahashi, "Theoretical analysis of view interpolation with inaccurate depth information," *IEEE Trans. Image Processing*, vol. 21, no. 2, pp. 718–732, Feb. 2012.

[49] S.-T. Na, K.-J. Oh, and Y.-S. Ho, "Joint coding of multi-view video and corresponding depth map," in *Proc. Intl. Conf. Image Processing*, vol. 15, Oct. 2008, pp. 2468–2471.

[50] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video," in *Proc. Picture Coding Symposium*, May 2009, pp. 1–4.

[51] S. Yamashita, N. Katoh, Y. Sasaki, Y. Akita, H. Chikata, and K. Yano, "Hole filling: a novel delay reduction technique using selector logic," in *Proc. IEEE Custom Integrated Circuits Conference*, May 1998, pp. 291–294.

[52] M. Germann, A. Hornung, R. Keiser, R. Ziegler, S. Wrmlin, and M. Gross, "Articulated billboards for video-based rendering," *Computer Graphics Forum*, vol. 29, no. 2, pp. 585–594, 2010. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2009.01628.x

[53] H. Nguyen and M. Do, "Error analysis for image-based rendering with depth information," *IEEE Trans. Image Proc.*, vol. 18, no. 4, pp. 703 –716, 2009.

[54] M. Tanimoto, "FTV (free viewpoint television) creating ray-based image engineering," in *Proc. Intl. Conf. Image Processing*, vol. 2, Sep. 2005, pp. 25–8.

[55] M. Pourazad, P. Nasiopoulos, and R. Ward, "A new prediction structure for multiview video coding," in *International Conference on Digital Signal Processing*, vol. 16, Jul. 2009, pp. 1–5.

[56] A. da Cunha, M. Do, and M. Vetterli, "On the information rates of the plenoptic function," *Information Theory, IEEE Transactions on*, vol. 56, no. 3, pp. 1306–1321, Mar. 2010.

[57] K. Muller, P. Merkle, and T. Wiegand, "3-d video representation using depth maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643 –656, 2011.

[58] A. Gelman, P.-L. Dragotti, and V. Velisavljevic, "Interactive multiview image coding," in *IEEE Int. Conf. on Img. Proc. (ICIP)*, 2011, pp. 601–604.

[59] K.-J. Oh, A. Vetro, and Y.-S. Ho, "Depth coding using a boundary reconstruction filter for 3-d video systems," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 3, pp. 350 –359, 2011.

[60] M. Maitre, Y. Shinagawa, and M. Do, "Wavelet-based joint estimation and encoding of depth-image-based representations for free-viewpoint rendering," *Image Processing, IEEE Transactions on*, vol. 17, no. 6, pp. 946 –957, Jun. 2008.

[61] A. Gelman, P.-L. Dragotti, and V. Velisavljevic, "Multiview image coding using depth layers and an optimized bit allocation," *IEEE Trans. on Img. Proc.*, vol. 21, no. 9, pp. 4092–4105, 2012.

[62] A. Gelman, J. Onativia, and P.-L. Dragotti, "A fast layer-based multiview image coding algorithm," in *Proc. of the European Sig. Proc. Conf. (EUSIPCO)*, 2012, pp. 1224–1228.

[63] X. Tong, J. Chai, and H.-Y. Shum, "Layered lumigraph with lod control," *The Journal of Visualization and Computer Animation*, vol. 13, no. 4, pp. 249–261, 2002. [Online]. Available: http://dx.doi.org/10.1002/vis.293

[64] R. I. Hartley, "Theory and practice of projective rectification," *Intl J. Computer Vision*, vol. 35, no. 2, pp. 115–127, 1999.

[65] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. European Conf. Computer Vision*, 2006, pp. 430–443.

[66] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Imaging Understanding Workshop*, 1981, pp. 121–130.

[67] J. Bouguet. (2000) Opencv. Intel Corporation. [Online]. Available: http://opencv.org/

[68] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[69] C. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," in *Proc. Int. Conf. on Pattern Recognition*, vol. 4, 2002, pp. 150–155.

[70] M. Maitre, Y. Shinagawa, and M. Do, "Symmetric multi-view stereo reconstruction from planar camera arrays," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, Jun. 2008, pp. 1 –8.

[71] H. Hirschmuller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *IEEE Trans. Pattern Anal and Machine Intell*, vol. 31, no. 9, pp. 1582–1599, 2009.

[72] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[73] J. Berent, P. L. Dragotti, and M. Brookes, "Adaptive layer extraction for image based rendering," in *International Workshop on Multimedia Signal Processing*, 2009.

[74] Middlebury. (2003) Teddy stereo dataset. Website. Middlebury. [Online]. Available: http://vision.middlebury.edu/stereo/data/

[75] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, Jun. 2003, pp. 195–202.

[76] C. M. Christoudias and B. Georgescu. (2002) Website. Rutgers University. [Online]. Available: http://coewww.rutgers.edu/riul/research/code/EDISON/index.html

[77] V. Kolmogorov, R. Zabih, and S. Gortler, "Generalized multi-camera scene reconstruction using graph cuts," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003, pp. 501–516.

[78] V. Kolmogorov. (2012) MATCH - stereo matching algorithm. [Online]. Available: http://pub.ist.ac.at/~vnk/software.html

[79] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, Sep. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77

[80] S. Zinger, L. Do, and P. de With, "Free-viewpoint depth image based rendering," *Journal of Vis. Com. and Img. Rep.*, vol. 21, no. 56, pp. 533 – 541, 2010.

[81] W.-Y. Chen, Y.-L. Chang, S.-F. Lin, L.-F. Ding, and L.-G. Chen, "Efficient depth image based rendering with edge dependent depth filter and interpolation," in *ICME*, 2005, pp. 1314–1317.

[82] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal and Machine Intell*, vol. 24, no. 5, pp. 603–619, May 2002.

[83] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," *GPUCV*, 2011.

[84] R. Hartley and A. Zisserman, "The projective camera," in *Multiple View Geometry in computer vision*, 5th ed. Cambridge University Press, 2000, ch. 6.2, p. 161.

[85] ——, "Representations of rotation matrices," in *Multiple View Geometry in computer vision*, 5th ed. Cambridge University Press, 2000, ch. Appendix 4.3.

[86] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 425–432. [Online]. Available: http://doi.acm.org/10.1145/383259.383309

[87] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," *Signal Processing: Image Communication*, vol. 24, no. 1, pp. 65–72, 2009.

[88] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, New York, NY, 2006, pp. 519–528.

[89] P. Niu, X. He, and A. Wong, "Dense depth map acquisition by hierarchic structured light," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1, 2002, pp. 165–171.

[90] H. Kawasaki, R. Furukawa, R. Sagawa, and Y. Yagi, "Dynamic scene shape reconstruction using a single structured light pattern," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1–8.

[91] K. Sakashita, Y. Yagi, R. Sagawa, R. Furukawa, and H. Kawasaki, "A system for capturing textured 3d shapes based on one-shot grid pattern with multi-band camera and infrared projector," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, 2011, pp. 49–56.

[92] F. de Sorbier, Y. Uematsu, and H. Saito, "Depth camera based system for autostereoscopic displays," in *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, 2010, pp. 361–366.

[93] H. Saito, "Computer vision for 3DTV and augmented reality," in *Int. Symp. on Ubiquitous Virtual Reality*, 2011, pp. 5–8.

[94] Y. S. Kang and Y. S. Ho, "High-quality multi-view depth generation using multiple color and depth cameras," in *IEEE Int. Conf. Multimedia and Expo*, 2010, pp. 1405–1410.

[95] A. Maimone and H. Fuchs, "Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras," in *IEEE Symp. Mixed and Augmented Reality*, 2011.

[96] A. D. Wilson and H. Benko, "Combining multiple depth cameras and projectors for interactions on, above and between surfaces," in *Proc. ACM symp. User interface software and technology*, 2010, pp. 273–282.

[97] C. Gilliam, P. L. Dragotti, and M. Brookes, "A closed-form expression for the bandwidth of the plenoptic function under finite field of view constraints," in *Proc. Intl. Conf. Image Processing*, Hong Kong, Sep. 2010, pp. 3965–3968.

[98] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nister, and M. Pollefeys, "Real-time visibility-based fusion of depth maps," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Oct. 2007, pp. 1–8.

[99] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," *Robotics & Automation Magazine, IEEE*, vol. 13, pp. 108–117, 2006.

[100] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics & Automation Magazine*, vol. 13, pp. 99–110, 2006.

[101] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Trans. Image Proc.*, vol. 13, no. 10, pp. 1327–1344, Oct. 2004.

[102] L. Baboulaz and P. Dragotti, "Exact feature extraction using finite rate of innovation principles with an application to image super-resolution," *IEEE Trans. Image Proc.*, vol. 18(2), pp. 281–298, 2009.

[103] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Mag.*, vol. 20, no. 3, pp. 21–36, May 2003.

[104] A. Kirmani, A. Colaço, F. N. C. Wong, and V. K. Goyal, "Exploiting sparsity in time-of-flight range acquisition using a single time-resolved sensor," *Opt. Express*, vol. 19, no. 22, pp. 21 485–21 507, 2011.

[105] ——, "CODAC: A compressive depth acquisition camera framework," in *Proc. IEEE Int. Conf. Acoustics., Speech, and Signal Processing*, Kyoto, Japan, 2012.

[106] O. Grau and G. Thomas. (2008) IVIEW:free-viewpoint video. BBC. [Online]. Available: http://www.bbc.co.uk/rd/projects/iview

[107] N. Inamoto and H. Saito, "Free viewpoint video synthesis and presentation of sporting events for mixed reality entertainment," in *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, ser. ACE '04. New York, NY, USA: ACM, 2004, pp. 42–50. [Online]. Available: http://doi.acm.org/10.1145/1067343.1067348

[108] J. Kilner, J. Starck, and A. Hilton, "A comparative study of free viewpoint video techniques for sports events," in *IET European Conference on Visual Media Production*. IET, 2006, pp. 87–96.

[109] M. Germann, T. Popa, R. Keiser, R. Ziegler, and M. Gross, "Novel-view synthesis of outdoor sport events using an adaptive view-dependent geometry," in *Computer Graphics Forum*, vol. 31, no. 2pt1. Wiley Online Library, 2012, pp. 325–333.

[110] R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*, 5th ed. Cambridge University Press, 2000.