SYNTHETIC APERTURE IMAGING
USING DENSE CAMERA ARRAYS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER
SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Vaibhav Vaish
April 2007

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Marc Levoy) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Richard Szeliski)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Mark Horowitz)

Approved for the University Committee on Graduate Studies.

न चोरहार्यं न च राजहार्यं न भ्रातृभाज्यं न च भारकारि ।
व्यये कृते वर्धत एव नित्यं विद्याधनं सर्वधनप्रधानं ॥

– भामिनि, सभातरंगिणी

(What is the wealth ...)
Which cannot be stolen by a thief,
Nor be taken away by a king,
Nor be partitioned amongst brothers,
Nor be a burden on the body,
Which on expending grows truly ...
(It) The wealth of knowledge
Is the greatest of all wealths.

-Sanskrit couplet from Bhamini's Sabhatarangini.

Abstract

Synthetic aperture imaging is a technique that projects images of a scene from different views on to a virtual focal surface, enabling us to “see through” occlusions in the scene. Using a 100-camera array we have used synthetic aperture imaging to view objects concealed behind dense foliage and to track a person moving through a crowd. The ability to see through occlusions makes synthetic aperture imaging a potentially powerful tool for surveillance.

This work makes two contributions. First, we characterize the image warps required for synthetic aperture imaging using projective geometry. This analysis leads to a robust camera calibration procedure for synthetic aperture imaging. Our analysis also shows the relation between the geometric complexity of the image warps and camera/focal plane configurations. In particular, we show that we can vary the focus through families of frontoparallel and tilted focal planes by simply shifting and adding the camera images. As image shifts are relatively simple to realize in hardware, this leads to a real-time system, which we demonstrate for tracking a person moving through a crowd.

Second, we explore methods to achieve sharp focus at every pixel in the synthetic aperture image by reconstructing the 3D surfaces of the occluded objects we wish to see. We compare classical shape from stereo with shape from synthetic aperture focus, and describe variants of stereo that improves image contrast by deleting some of the light rays that are incident on the occluder.

Acknowledgments

When I decided to embark on a Ph.D., I hoped for a chance to have tremendous fun, to do some things I had not even dreamt about and emerge a wiser person. I would like to thank everyone I have interacted with during the last six years who helped me successfully achieve the first two. If I have not become any wiser, the credit goes to me alone. I would like to thank:

My adviser, Prof. Marc Levoy for his unbounded and infectious enthusiasm, for teaching me how to give better talks, for showing me it can be more important to seek simple and intuitive explanations than rigorously provably correct ones. Rick Szeliski for being an inspiring mentor since I first met him, for his unlimited patience and encouragement during my internship and Microsoft and rock solid support whenever I needed. Prof. Mark Horowitz for his helpful insights in all our research group meetings. Thanks to Prof. Sebastian Thrun and Prof. Bernd Girod for being on my oral examination committee.

I would like to thank each and every person in the Stanford Graphics Laboratory for making it a truly awesome place to be. Thanks to Heather and Ada for helping us retain our sanity during SIGGRAPH deadlines (and the rest of the year). Whenever I went into their office, I always come out happier. Thanks to John Gerth, Joe Little, Charlie Orgish and Kevin Colton for being the best system administrators in the world and going beyond the call of duty to accommodate all our eccentric hardware, software and group members. Thanks to all my office mates and colleagues: Bennett, Billy, Gaurav, Neel, Leslie, Sean, David, James, Niloy, Natasha, Erika, Tony, Doantam, Jeff, Ron, Merrie, Dan, Eddy, Emilio, Andrew, Mike, Ren, Kayvon, Augusto, Zhang, Manu, Pradeep. Thanks to Teresa Lynn for her help in helping me schedule my defense.

It was my privilege to work with Prof. Donald Greenberg - my academic grandfather, prior to coming to Stanford. Helping him teach CS 417 made me learn more about computer graphics than any class I could have taken. I'd like to thank everyone at Microsoft Research, Redmond who helped make my summer internship a fantastic experience: Rick Szeliski, Sing Bing Kang, Larry Zitnick, Brian Curless, Kamal Jain, Ashish Kapoor, Jan-Michael Frahm. Special thanks to all my friends from the Indian Institute of Technology, New Delhi for being like a family to me in USA.

Finally, I would like to thank my family: my parents and my brother for giving me the gift of education, the freedom to seek it howsoever I chose and their unflagging support in all my endeavors. This dissertation would never have been possible without the infinite love, care and support of my wife, Swati - my finest acquisition in life.

Contents

List of Tables

List of Figures

Chapter 1

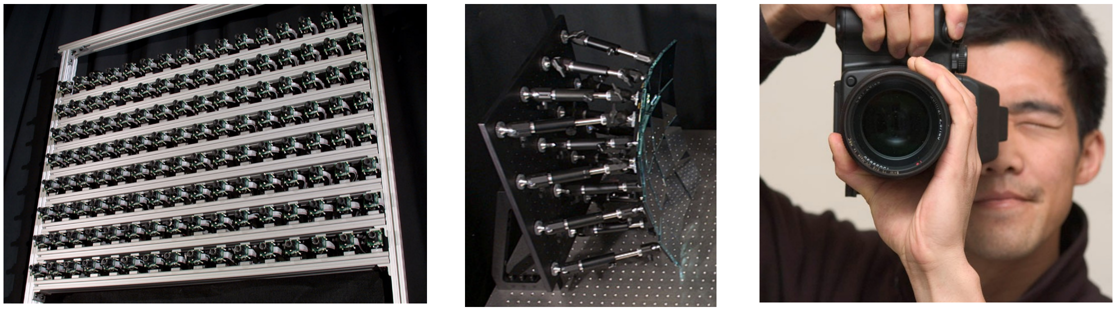
Introduction

The decreasing of cost digital image sensors has made is feasible to build large multi-camera arrays. Camera arrays are useful for several applications in computer vision, graphics and cinematography. The bullet-time scene in the *The Matrix* was created using an array of still cameras triggered at different times [27]. At Super Bowl 2001, an array of 30 synchronized video cameras was used to display views of the action from different viewpoints giving viewers a sense of flying through the scene [18]. Researchers in computer vision and graphics have built camera arrays for virtualized reality [19, 55], image-based rendering [50, 53], high performance imaging [48] and synthetic aperture imaging [43, 41]. This dissertation describes the use of large camera arrays for the latter application.

1.1 Synthetic aperture imaging

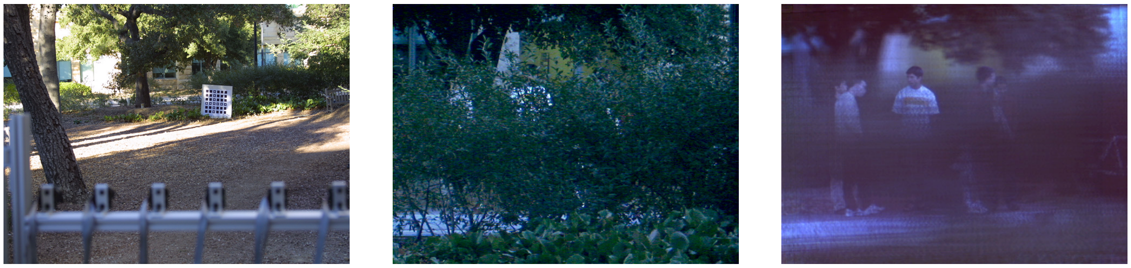
Synthetic aperture imaging is a technique that uses an array of cameras to simulate a giant virtual lens, enabling us to synthesize images which would be impossible to acquire with an ordinary camera lens. An example of synthetic aperture imaging is shown in Fig. (1.2). Some of the effects we can achieve with synthetic aperture imaging are:

1. **Seeing through occlusions:** A camera array simulates a lens with an aperture as large as the camera baseline (we have experimented with apertures



(a) Stanford Multi−camera array (b) Mirror array (c) Hand−held plenoptic camera

Figure 1.1: Some synthetic aperture imaging systems. (a) An array of 100 video cameras, also used for high-performance imaging [48]. (b) An array of planar mirrors. By aiming a high resolution camera at these mirrors, we obtain a virtual array of cameras which can be used for synthetic aperture imaging. This has also been used for synthetic aperture confocal imaging [25], dual photography [35] and symmetric photography [9]. (c) A hand-held camera modified by placing a microlens array on the camera sensor [30]. With such a camera, one can vary the focus after acquisition using the same procedure as in synthetic aperture imaging using a camera array.



(a) Cameras aimed at bushes (b) View from one camera (c) Synthetic aperture image

Figure 1.2: Example of synthetic aperture imaging. (a) An array of 45 cameras was aimed at dense bushes. (b) Image from one camera with students standing behind the bushes. It is almost impossible to make out the students from any single camera image. (c) We can combine images from all 45 cameras to automatically construct a synthetic aperture image in which the students behind the bushes are clearly visible.

up to 3m wide). Such a huge aperture leads to a proportionately shallow depth of field. This can be used to "see through" occlusions like dense foliage. When we *synthetically "focus"* this virtual lens on objects behind occlusions, the defocus blur of the foreground occluders is so large that they effectively disappear, enabling us to see the objects behind. This is illustrated in Fig. (1.2). The ability to see through dense occlusions make synthetic aperture imaging a potentially powerful tool for surveillance.

2. **Improving Signal-to-noise Ratio:** Each pixel of a synthetic aperture image is computed by averaging pixels from some or all of the cameras in the array. This improves the signal-to-noise ratio by a factor of $\sqrt{N}$ where N is the number of pixels averaged to compute a pixel in the synthetic aperture view. In our experiments, N varies from 30 to 100. The reduction noise helps preserve detail and contrast even in low-light conditions. The synthetic aperture image of Fig. (1.2c) was acquired close to sunset; since 45 images were averaged to produce it we can see enough detail.

3. **Per-pixel focus control:** Using a conventional camera lens, a photographer has to choose a single focal depth and depth of field (determined by the lens aperture) for the entire image. In a synthetic aperture image, we can choose the focal depth and aperture for each pixel independently of the others. This can be used to *compute* images where every pixel is in sharp focus, instead of having to trade-off the amount of light captured by stopping down the aperture to increase the depth of field. Note that synthetic aperture imaging allows us to select the focus for each pixel *after* capturing the images.

4. **Improving depth from focus/defocus:** Reconstructing the 3D shape of a scene using focus/defocus cues is limited in accuracy by the size of the lens aperture [32]. This limitation can be overcome using depth from "synthetic

aperture focus" since the latter uses aperture much larger than that of conventional camera lenses.

To understand how synthetic aperture imaging works, it is helpful to consider an analogy. A convex lens in a digital camera focuses a plane in the scene on to the sensor plane points not in the plane of focus are blurred. This process is illustrated in Fig. (1.3). Rays starting from a point on the plane in focus after refraction converge to a point on the sensor plane. Rays starting from a point which is not on the plane of focus converge to a point in front of or behind the sensor plane. These rays intersect the sensor plane over an area known as the *circle of confusion*, resulting in a blurred or out of focus image [22]. The greater the size of the lens aperture, the larger the circle of confusion and amount of misfocus.

We can use a camera array to simulate the defocus blur of a convex lens. To "focus" on a plane, we *backproject* the images from each camera on to the chosen plane and compute their average. This projection of images is accomplished via image warps called homographies, described in Chapter 3. In the resulting image, objects close to the chosen plane of focus will appear sharp, whereas objects away from this plane will appear blurry being smeared out due to parallax (Fig. 1.4) The camera array functions as a synthetic lens aperture, hence the term synthetic aperture imaging.

The size of the synthetic aperture spanned by the cameras (typically 1-3m) is much larger than a normal lens aperture. Thus, the defocus blur in synthetic aperture images will be far more pronounced. When the synthetic aperture is much wider than the occluding objects in front of the plane of focus, these occluders become so blurred that they effectively disappear, enabling us to see the objects concealed behind them. To see the students behind the bushes in Fig. (1.2), a 2m wide synthetic aperture was used.

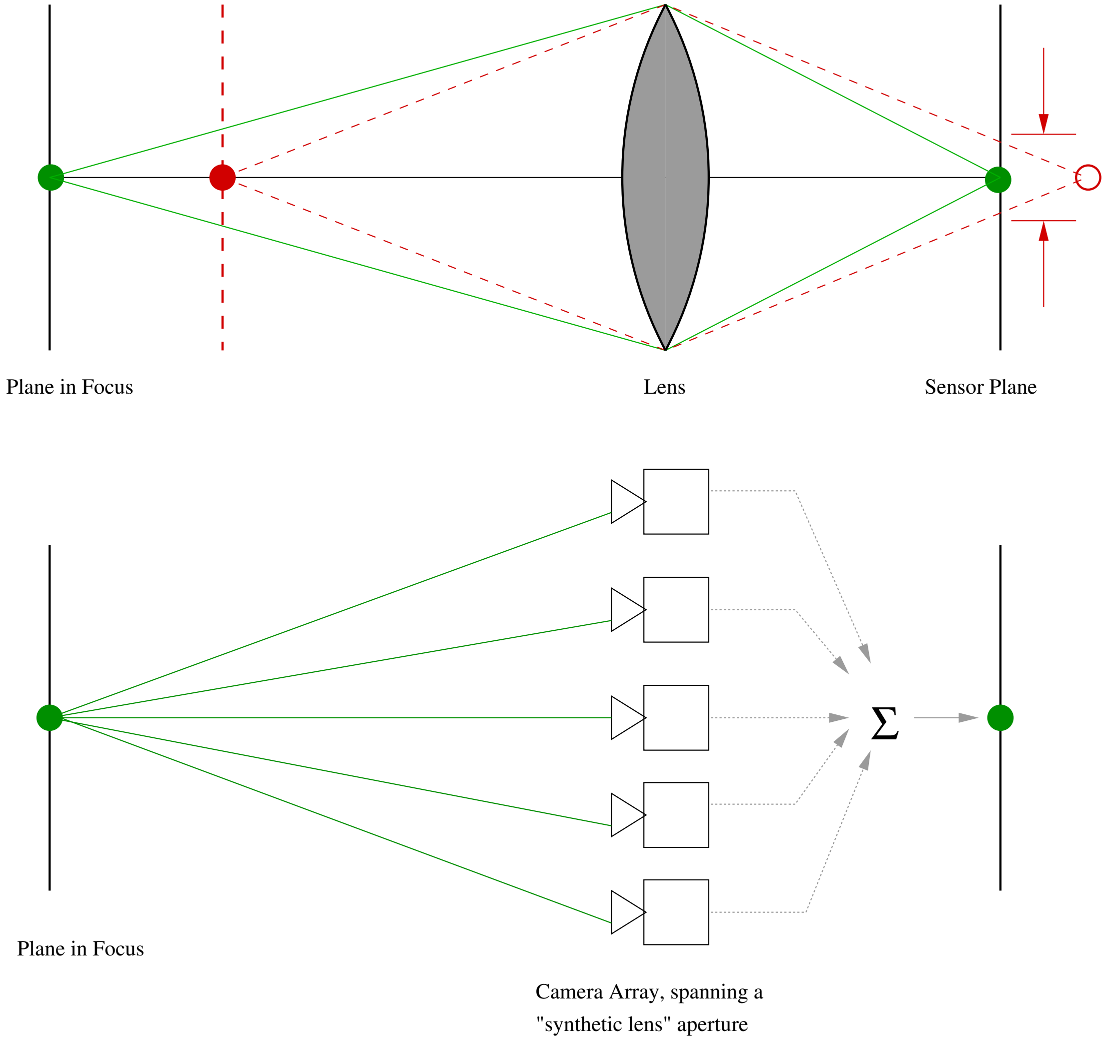


Figure 1.3: Explanation of synthetic aperture imaging via geometric optics. Top: In a convex lens, rays from the green point on the plane of focus after refraction converge to a single point on the imaging plane, forming a sharp image. Rays from the red point, which is not in the plane of focus form a *circle of confusion* on the image plane, resulting in a blurred image. Bottom: A camera array is analogous to a “synthetic” lens aperture, each camera being a sample point on a virtual lens. We synthetically focus the camera array by choosing a plane of focus, and adding up all the rays corresponding to each point on the chosen plane to get a pixel in “synthetic aperture” image. This is the method used to create the right image in Fig. 1.2.

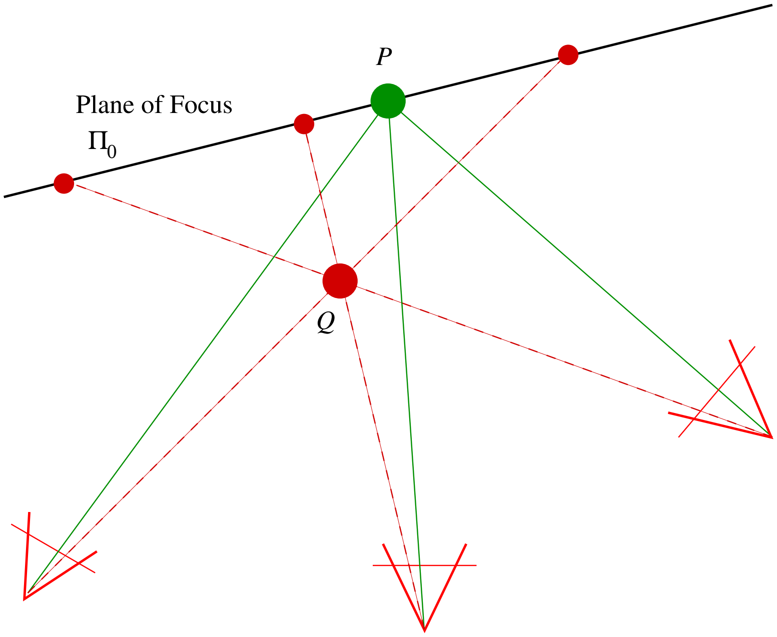


Figure 1.4: Focusing with homographies. To construct a synthetic aperture image focused on a plane Π_0, we need to *backproject* the images from each camera on to Π_0 and compute their average. The backprojection is done by applying image warps known as homographies. Pixels from different cameras that correspond to a point P on Π_0 will project to the same location, resulting in a sharp image of P. Pixels corresponding to a point Q not on Π_0 will project to different locations on Π_0 (this is called parallax or disparity) resulting in a blurred image of Q.

1.2 Differences with Geometric Optics

A convex lens forms a focused image at the speed of light. However, synthetic aperture focusing is a computational process. To focus on a given plane, we need to compute and apply the suitable warps (2D spatial transforms) to the camera images. These warps will depend on the camera arrangement and choice of focal plane. To compute these warps, we would need to measure the camera arrangement via a calibration procedure. This raises the question: what camera parameters do we need to calibrate to compute the required image warps for focusing on a given plane ? Moreover, what is the minimal amount of additional computation required to move the focus to a different plane ?

Secondly, the focal surface of a lens is determined by geometric optics. An ideal convex lens has planar focal surfaces; points not on the focal plane appear misfocused in the image it creates. The amount of misfocus depends on the size of the lens aperture and distance from the focal plane, this is known as *depth of field* [22]. However, there is no such constraint on synthetic aperture imaging: we can focus on a surface of arbitrary shape. We simply need be able to project the camera images on to the desired surface to compute a synthetic aperture image. Ideally, we would like the focal surface to match the surface of the object we wish to view, so that every pixel is in good focus. To do so, we need to estimate the surfaces of objects behind dense occlusion as accurately as possible.

Thirdly, a lens integrates *all* the rays across its aperture to produce a focused image. In synthetic aperture focusing, this is not a constraint. If we were to ignore the rays incident on occluding objects in the computation of a synthetic aperture image, we would eliminate the blurred image of the occluders superimposed on the objects in focus. This would yield a clearer, higher-contrast image of the objects we wish to focus on. This raises the question: how well can we separate the rays incident on the objects in focus from the rays incident on the occluders in front ?

Finally, synthetic aperture images are created from a *discretely* sampled light field. A synthetic aperture is a sum of finitely many rays, whereas a focused image created by a convex lens is an integral of a continuous set of rays. If the light field is not bandlimited (and it typically is not), the synthetic aperture image will contain aliasing (which it typically does). A detailed signal-theoretic analysis of synthetic aperture imagery may be found in [29].

1.3 Contributions

In addressing the questions raised above, this thesis makes the following contributions:

1. *Geometric analysis for focusing on planes*: We characterize the space of image warps required to focus on arbitrary planes using projective geometry. Our analysis leads to:

   - A robust calibration procedure for synthetic aperture imaging [43]. We show that synthetic aperture focusing does not require a full metric calibration of the cameras; it is sufficient to match images of four points on a plane and two points not on the plane across all cameras.
   - A classification of camera arrangements and focal planes corresponding to different families of image warps [41]. We describe all configuration where focusing can be achieved by affine (or simpler) image warps. We show there are families of frontoparallel planes, as well families of tilted planes through which we can vary the focus by simply shifting the images. Since image shifts are relatively easy to realize in hardware, this leads to a real-time focusing system. We demonstrate the use of this to track a person moving through a crowd.

2. *Reconstruction of occluded surfaces for focusing* [42]: We explore different methods to reconstruct occluded surfaces using synthetic apertures. We present a comparison of shape from stereo with shape from synthetic aperture focus. We show that while shape from stereo can reconstruct surfaces with weaker texture than shape from synthetic aperture focus, the latter is more tolerant of occlusions. We also describe novel variants of stereo that improve image contrast by eliminating rays incident on foreground occluders.

Although synthetic aperture imaging is the primary motivation for our research, some of our work is is applicable to other applications of camera and projector arrays. Our camera calibration procedure based on plane + parallax has also been used to calibrate projector arrays [25]. Based on our calibration, Wilburn et al [48] have developed an optic flow based approach to view interpolation in space and time. Our method for focusing on tilted planes using shifts can be used extend the Fourier refocusing technique of [29] to tilted focal planes with view cameras. We will describe these connections in more detail in the subsequent chapters.

Chapter 2

Related Work

Implementing a synthetic aperture imaging system draws upon work in several different areas such as design of the imaging systems (camera hardware), analysis of image warps required for focusing (multiple view geometry), determining general focal surfaces (3D reconstruction from images in computer vision). This chapter reviews relevant work in these areas.

2.1 Camera Arrays

The decreasing cost of digital image sensors has made it feasible to build large arrays of cameras. In recent years, a number of camera arrays have been built for different applications including high performance imaging, image and video-based rendering, shape and motion capture and virtual cinematography. We believe that the continuing reduction in cost and the wide range of applications will lead to increased availability of large camera arrays. Camera arrays can be classified into two threads based on the arrangement of cameras.

Single Center of Projection: For high performance imaging, the cameras are tightly packed together to approximate a single center of projection. The goal is to acquire improved imagery by combining images from different cameras (which are aligned to a common perspective). Baker et al have developed a system of up to 24 cameras with abutting fields of view whose images are mosaiced yielding video with higher pixel resolution than the individual cameras [2]; their array is now available commercially [14]. A larger scale implementation of this was developed by Wilburn et al using 96 cameras to acquire 7 megapixel video; they also increase the dynamic range by trading off field the field of view and using different exposures across their cameras [48]. In separate work, they used 52 cameras with overlapping fields of view with temporally staggered frames to achieve high speed video [47].

Multiple Centers of Projection: When the cameras are spread over a wide baseline to capture the scene from many different points of view, they sample the *light field*, the set of all light rays intersecting the scene [26]. Images from these viewpoints can be used to synthesize views of the scene from novel viewpoints by resampling the acquired images [26, 11] or by reconstructing the scene geometry and reprojecting it into the desired viewpoint [55]. Wilburn et al have demonstrated view interpolation in across viewpoints in space and time using their camera array [48]. Cycling through the images - without doing any view interpolation at all - is a powerful visual effect which can give the viewer a sense of flying through the scene. Examples of this include the bullet-time scenes in *The Matrix*, created using an array of 120 still cameras [27] and the Eye Vision technology used during the broadcast of Super Bowl 2001 [18].

For seeing through occlusions using synthetic aperture imaging we would like to have an acquisition system which has:

1. A large number of cameras to get as dense a sampling as possible of rays passing between the occluders.

2. A flexible aperture that can be expanded till it is larger than the size of occluders in the scene.

3. The ability to compute image warps required for synthetic aperture focusing at video rates for dynamic scenes.

We were fortunate to have access to the Stanford Multi-Camera array built by Wilburn [48], which is unique amongst available camera arrays in having all these properties. The array offers up to 100 low-cost video cameras of imaging, flexible apertures from 0.5m to 3m, narrow and wide angle lenses for long and short range imaging. Per-camera FPGAs enable the computation required for real-time synthetic aperture video.

2.2 Finite Aperture Imaging

Cameras are often modeled as pinhole devices in computer vision and graphics, neglecting the size of the lens aperture. In practice, the effect of a finite lens aperture is to capture more light than a pinhole device at the cost of reducing the depth of field i.e. causing a depth-dependent defocus blur. This results in a depth cue that is used for reconstructing 3D shape by depth from defocus [7] and depth from focus algorithms [23]. One can also reconstruct shape by observing changes in the images on varying the lens aperture (which controls the amount of defocus), this approach is known as *confocal stereo* [13]. The finite aperture of a lens can be used to see beyond occlusions, however the limited aperture of the lens greatly restricts the size of occluders one can see around [8].

There are two major differences between focused images from a lens and synthetic aperture imaging. First, the synthetic apertures spanned by a camera array are much larger than those of an ordinary lens. This means that all the aperture effects described above are more pronounced with synthetic apertures. A larger aperture leads to a shallower depth of field which increases the depth resolution of shape from focus algorithms. To effectively see beyond occluders, we require an aperture larger than the size of the occluders. This is much easier to accomplish with synthetic apertures.

Secondly, a camera array acquires more information about a scene than a lens. A camera array captures a 4D light field (a 2D array of 2D images) whereas a lens is constrained to acquire a 3D focal stack (set of 2D images focused over a range of depths). This makes it possible, for example, to ignore rays incident on occluders in the construction of a synthetic aperture image which is not possible with single lens apertures.

The idea of averaging images from adjacent cameras in an array to simulate a synthetic aperture was observed by Levoy and Hanrahan [26]. This was used to dynamically control the focus and aperture of images synthesized from light fields by Isaksen et al [17]. Schechner and Kiryati independently observed the analogy between lens apertures and camera baselines in their comparison of depth from defocus with depth from stereo [32].

A technique similar to synthetic aperture imaging is *synthetic aperture radar* [28]. The idea is to combine measurements from a moving radar antenna (such as on an aeroplane) to effectively synthesize a much larger antenna. The distance the aeroplane travels to synthesize the larger antenna is called the synthetic aperture. A larger antenna can greatly improve the azimuthal resolution of the radar imagery. One difference between synthetic aperture imaging and synthetic aperture radar is that the latter uses the phase and frequencies of the reflected signal (it is based on the Doppler effect) while the former requires only the amplitude (recorded as pixel intensities).

2.3 Multiple View Geometry

The goal of multiple view geometry in computer vision is to study the relations between images of a scene from different viewpoints, this is based on projective geometry [12]. These relations are used in structure from motion, self-calibration, matching image features across viewpoints and reconstructing 3D shape. One representation that greatly simplifies multi-view geometric relations is *plane + parallax* developed by Irani and Anandan [16, 15]. In this approach, images from all viewpoints are projected on to a common plane as shown in Fig. 1.4. The location of a point in the scene is described in terms of the parallax between its images in different views as measured on the reference plane.

Our geometric analysis of synthetic aperture imaging is based entirely on plane + parallax. The parallax caused by a point not on the reference plane is the analog of the circle of confusion in photography. By characterizing the parallax caused by a points on a given plane, we can compute the image warps required to move the plane of focus from the reference plane to the given plane (Chapter 3). We also use parallax measurements for calibrating the camera positions (Chapter 4).

Most of the work in computer vision on plane + parallax is directed towards images acquired from a single moving camera (whose motion may be unknown). For synthetic aperture imaging, we have an array of cameras with fixed positions which can be calibrated a priori, which leads to further simplification. For example, Zelnik-Manor and Irani have shown that *planar homologies* (the image warps required to change the plane of focus) are constrained to lie in a 4D space, which requires calibration of 4 basis vectors [52, 51]. In Chapter 3, we show that for fixed camera positions the homology parameters actually lie in a 1D space which simplifies the calibration and makes it possible to change the focus by simply scaling the basis vector (we call this basis vector the μ-vector).

2.4 3D Shape Reconstruction

Constructing a synthetic aperture image where each pixel is in sharp focus requires finding the correct focal depth for every pixel. This amounts to reconstructing the 3D surface of the object(s) we desire to focus on. The reconstruction of 3D surfaces from their 2D images is a challenging problem in computer vision which has inspired decades of research; a recent survey of multi-view reconstruction algorithms may be found in [33]. Here we are interested in the use of 3D reconstruction to improve the performance of synthetic aperture imaging.

One approach is to reconstruct 3D shape is to use focus cues from the imaging lens. Shape from focus methods take as input a stack of images focused over a range of depths, and determine for each pixel the depth at which it is in best focus. The criterion for determining when a pixel is in good focus is some form of a high-pass filter, a comparison of several focusing criteria may be found in [23]. We can use the same approach for synthetic aperture images; in this case the focus cue comes from a *virtual lens* simulated by an array of cameras.

Shape from stereo methods work by matching pixels across different views using some measure of photo-consistency and finding depth via triangulation [31]. In their comparison of shape from stereo versus focus, Schechner and Kiryati observed that shape from focus (using a single camera lens) ought to be more tolerant of occlusions than stereo while stereo ought to have finer depth resolution due to larger baselines. Using a synthetic aperture baseline has the potential of giving the best of both worlds: robustness to occlusion and finer depth resolution.

There are three differences between our work on reconstructing focal surfaces for synthetic aperture imagery and prior work on 3D reconstruction. First, we use many more views (typically 80 to 100) spanning a wider wider synthetic aperture than most previous methods. Second, since we use synthetic aperture imaging to see through occlusions, we are interested in reconstructing surfaces that are *occluded* in a significant fraction of the input images. Thus, we expect several outliers amongst pixels amongst the pixels to be matched across views. This is a harder problem than classical multi-baseline stereo [31] which assumes no outliers. Finally, we do not reconstruct the occluders explicitly, but rely on the large number of views and robust matching functions to obtain a good reconstruction. We defer a more detailed discussion of the differences between our approach and existing methods to Chapter 5, where we define the problem in terms of reconstructing general focal surface and shaped apertures.

Chapter 3

Geometric Foundations

This chapter describes the geometry underlying the computation of synthetic aperture images which are focused on a plane. There are three important reasons for doing this analysis. First, it will let us characterize the image warps or homographies required to focus on to any plane in terms of the camera layout and chosen focal plane. This tells us precisely the minimal set of camera parameters we need to estimate, and also leads to a robust calibration procedure. Secondly, it shows that there are combinations of camera arrangements and focal planes on which one can focus using image warps geometrically simpler and computationally less expensive than homographies. These combinations are desirable for real-time focusing where image warps need to be implemented in per-camera hardware, and simpler warps like image shifts are easier to map to the hardware. Thirdly, given the ability to focus on any plane, we can sweep the plane of focus through the scene being imaged to try and determine the best focal depth for each pixel.

We begin by describing how to focus on an arbitrary plane using image warps called homographies. Next, we will show how to vary the focus from one plane to another using image warps called *planar homologies*, which are special cases of homographies. We will describe all camera/focal plane configurations for which the homologies are affine transforms or simpler, and outline some of their applications. Based on this analysis, we will describe our method for camera calibration and real-time focusing in the following chapter.

3.1 Focusing on a Plane

The process of focusing on a plane is depicted in Fig. (1.4). We need to back project the image from each camera (as if it were a projector) on to the desired plane. For perspective cameras, the mapping from the camera's imaging plane to another plane in the world is given by an image warp called homography (also known as collineation or perspective warp) [12]. When points are expressed in homogeneous coordinates, a homography is an invertible linear transform described by a 3×3 matrix.

Suppose we have $N + 1$ cameras with centers $C_0, \ldots, C_N$. Let I_i denote the image from camera C_i and H_i the homography required to project I_i on the desired focal plane. Mathematically, the synthetic aperture image is given by:

$$I_{SA} = \frac{1}{N+1} \sum_{i=0}^{N} H_i \circ I_i \tag{3.1}$$

where I_{SA} is the synthetic aperture image and $H_i \circ I_i$ is the projection of image I_i on to the focal plane. Fig. (3.1) shows the construction of a synthetic aperture image using two cameras. Pixels that correspond to points on the focal plane - such as point P in Fig. (1.4), or points on the calibration grid in Fig. (3.1) - are mapped to the same location by the homographies, *reinforcing* each other in the synthetic aperture image. Pixels that correspond to points not on the focal plane - such as point Q in Fig. (1.4), or the computer monitors in Fig. (3.1) - project to different locations on the focal plane, causing them to be smeared out or blurry in the synthetic aperture image.

The homography required for each image depends on the camera parameters and the focal plane. If we wish to change the focal plane, we need to apply different homographies. This requires substantial computation and may be difficult to achieve in real-time. In the next section, we describe an alternative to approach to changing

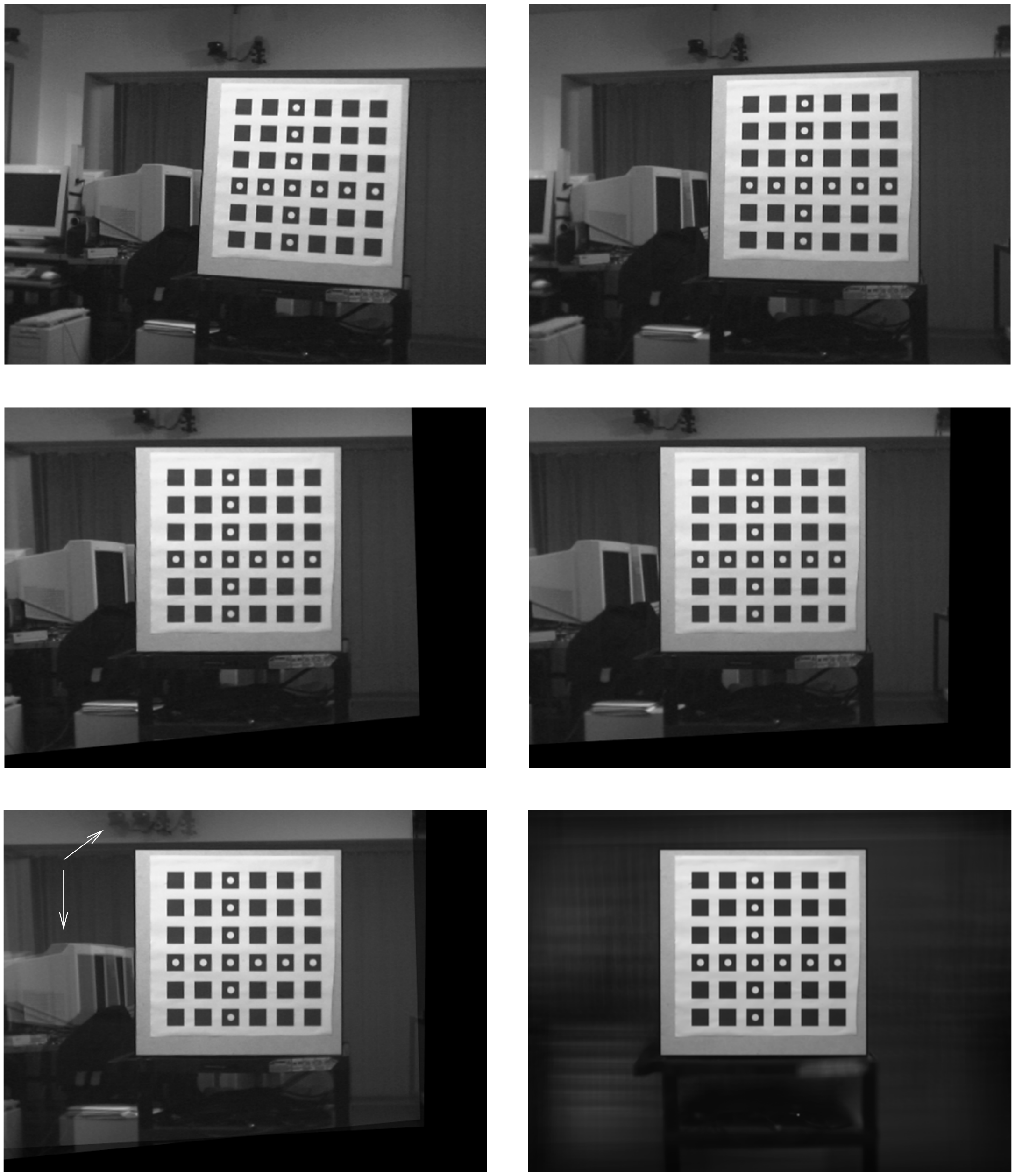


Figure 3.1: Focusing on the plane of a calibration grid with homographies. Top row: images from two cameras of a planar calibration grid. Middle: The result of back projecting the images on to the plane of the calibration grid via homographies. Bottom left: The average of the two images from the middle row. The calibration grid is in good focus but the monitors and cameras on the wall above exhibit ghosting due to parallax. Bottom right: When we add the backprojected images from 88 cameras, the grid remains in focus and everything else is completely blurred out.

the focal plane which, in some cases, avoids the need for full homographies.

3.2 Refocusing with Homologies

Consider an array of cameras with centers $C_0, \ldots, C_N$ whose images have been backprojected on to some initial focal plane Π_0. We call this initial focal plane the *reference plane*. If we compute the average of these projected images, we get an image focused on the reference plane. Suppose we wish to focus on a different plane, Π. One could do so by applying different homographies to the original camera images, as described in the previous section. Another way would be to *reproject* each of these projected images from the reference plane on to Π through center of projection C_i (Fig. 3.2). This reprojection is called a planar homology and can be described by a 3×3 matrix. Homologies are geometrically simpler to analyze and in many cases, facilitate faster computation of synthetic aperture images than would be possible using homographies. This section describes the homology matrices for varying the focal plane in terms of the camera geometry and focal planes.

We begin our study of homologies by defining the coordinate systems we will use. Assume that there is a given 2D coordinate system on the reference plane. The pixels of the projected images I'_i are specified in this reference coordinate system. We also need to pick a coordinate system on the new focal plane Π. To do so, we pick a reference camera - say C_0. A point P on Π will be assigned the coordinates of its projection p_0 on the reference plane through C_0 (Fig. 3.2). Thus, we are projecting the coordinate system of the reference plane on to the new focal plane Π through center of projection C_0.

Let G_i be the homology required to reproject I'_i on to $\Pi, 1 \leq i \leq N$ ($G_0 = I$, the identity matrix). G_i maps point p_i on the reference plane to the point P on Π (Fig. 3.2). Since P has the same coordinates as p_0, we may write $G_i p_i \cong p_0$ where G_i denotes the homology matrix, points and lines on the reference plane are represented in homogeneous coordinates and $\cong$ denotes equality up to a scale. For the ensuing

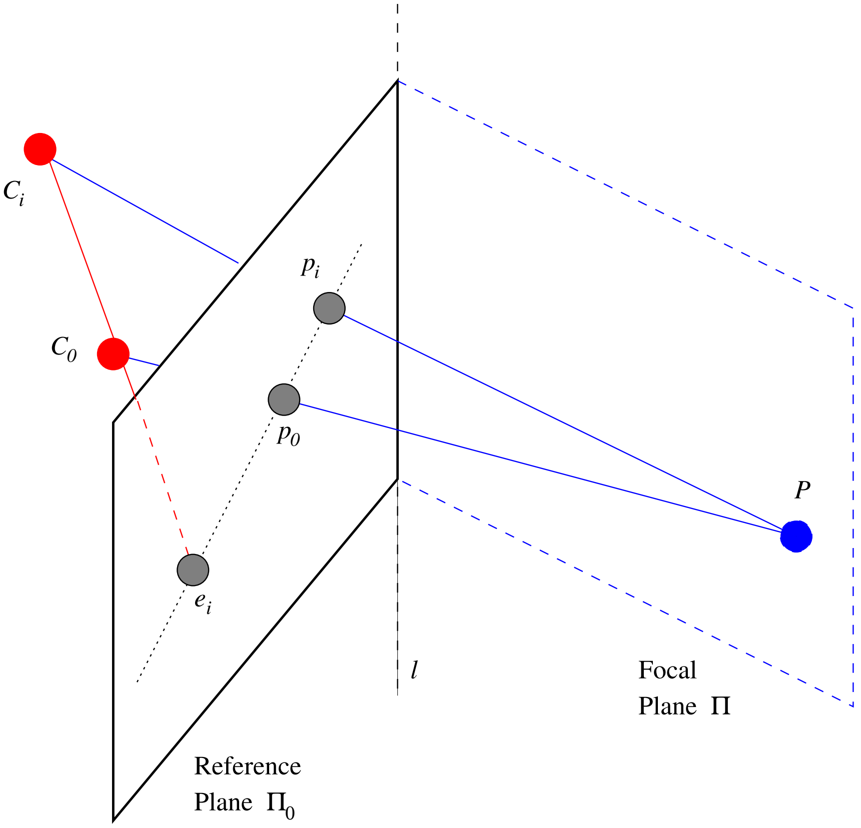


Figure 3.2: Refocusing with homologies. C_0 is the center of the reference camera, and C_i is the center of another camera in the array. The line joining the camera centers C_0, C_i intersects the reference plane at a point called the epipole, which we denote by e_i. Images from all the cameras have been backprojected on the reference plane Π_0. If we compute the average of these backprojected images, we get a synthetic aperture image focused on the reference plane. To change the focus from reference plane Π_0 to plane Π, we need to *reproject* the image of camera C_i from the reference plane Π_0 on to the focal plane Π. This projection is called a *planar homology*. It can be describe in terms of the l, the line of intersection of Π_0 and Π and e_i, the epipole.

analysis, it will be simpler to work with the inverse homologies $K_i = G_i^{-1}$, i.e. K_i projects points on Π on to the reference plane through center C_i and $K_i p_0 \cong p_i$.

We now proceed to characterize the 3×3 homology matrices for changing the focal plane. From projective geometry [12, 38] the homology K_i can be written as

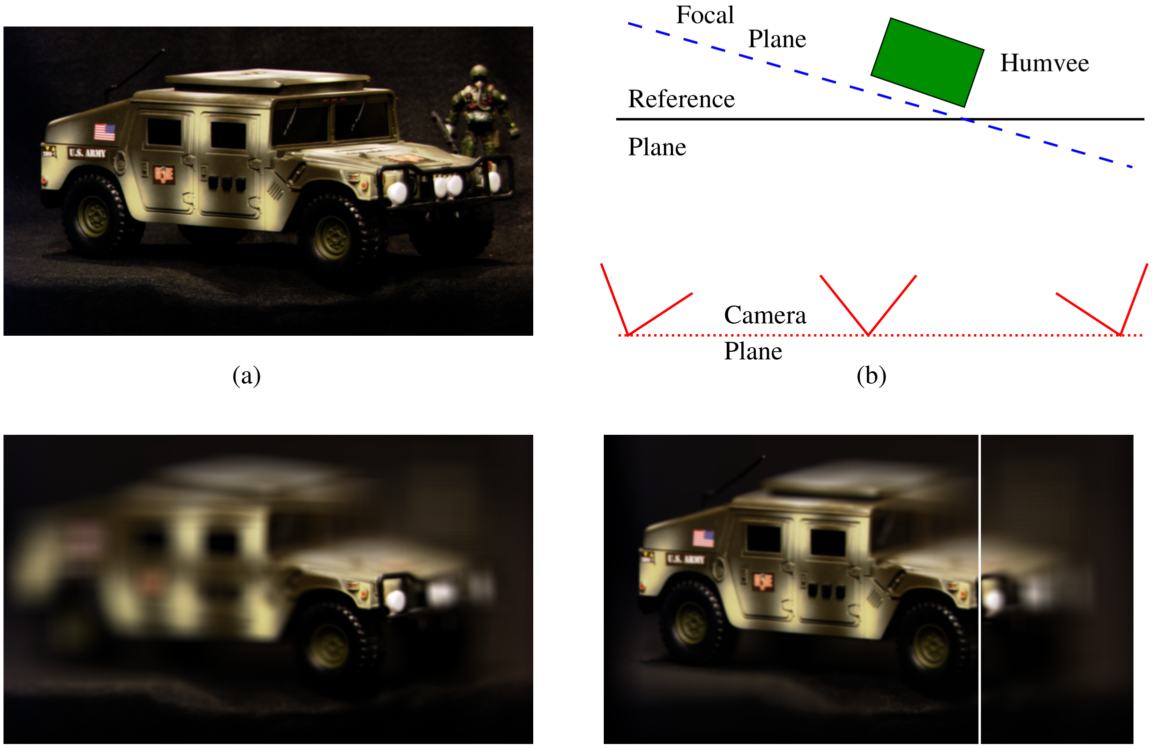
$$K_i = I + \mu_i e_i l^T \tag{3.2}$$

Here I is the identity, e_i the epipole associated with cameras C_0, C_i and the reference plane, l the line of intersection of the reference plane and Π, and μ_i is a scalar. Geometrically, varying μ_i while keeping e_i, l fixed corresponds to rotating the focal plane about axis l and moving p_i along the epipolar line through e_i and p_0. Choosing a value for μ_i amounts to choosing a focal plane (through l) for camera C_i.

Suppose we are given the epipoles $e_i, 1 \leq i \leq N$. We wish to characterize $\mu_1, \ldots, \mu_N$ which correspond to the same choice of focal plane Π. Let us call $[\mu_1 \ldots \mu_N]$ the μ-vector for homologies induced by a focal plane Π. The following result helps us characterize which vectors of $\mathbb{R}^N$ are μ-vectors for some focal plane.

μ-vector theorem. *Given epipoles $e_1, \ldots, e_N$ on a reference plane as defined above, the μ-vectors for all focal planes lie in a 1-D space, i.e. μ-vectors for any two planes are equal up to scale.*

We defer the proof to the end of the chapter. An important point to note is that in homogeneous coordinates, points and lines can be scaled arbitrarily. The μ-vector for a focal plane will change if we change the scale of any of the epipoles or the line l. It is assumed in the theorem that we have chosen a fixed scale for each epipole e_i, this scale could be arbitrarily chosen but must be the same for all homologies we compute. If we change the scale for any of the epipoles, we will change the 1D space the μ-vectors lie in. However, as long as the scales are fixed they will still lie in a 1D space.



(a)

(b)

(c)

(d)

Figure 3.3: Rotating the plane of focus using homologies. (a) An image from a 16×16 image light field of a humvee at an angle to the plane of cameras. (b) Scene layout. (c) Synthetic aperture image focused on the reference plane. (d) Using homologies, we can rotate the focal plane about the vertical white line until it coincides with the humvee surface.

At this point, we have a complete description of the homologies required to change the focus from the reference plane to any other plane. To vary the focus, we will need to calibrate homographies for projecting the camera images on to some chosen reference plane, the epipoles e_i and the space of μ-vectors. With this information a user can focus on any plane by starting from the reference plane, specifying a line of intersection l on the reference plane and rotating the focal plane about l by scaling the μ-vector. An example of this process is shown in Fig. (3.3).

3.3 Simpler Configurations

We now enumerate the camera/focal plane configurations in which the homologies are not projective warps, but affine transforms or simpler. These configurations are interesting for two reasons. First, we save computation by not having to perform a full homography per image to change the focal plane. Second, it can be easier to calibrate the required parameters. For each of these configurations, the homologies lie in a proper subgroup of the group of planar homologies. We will assume we have established an affine coordinate system on the reference plane (one way to do so is to choose the plane of a calibration grid as the reference plane).

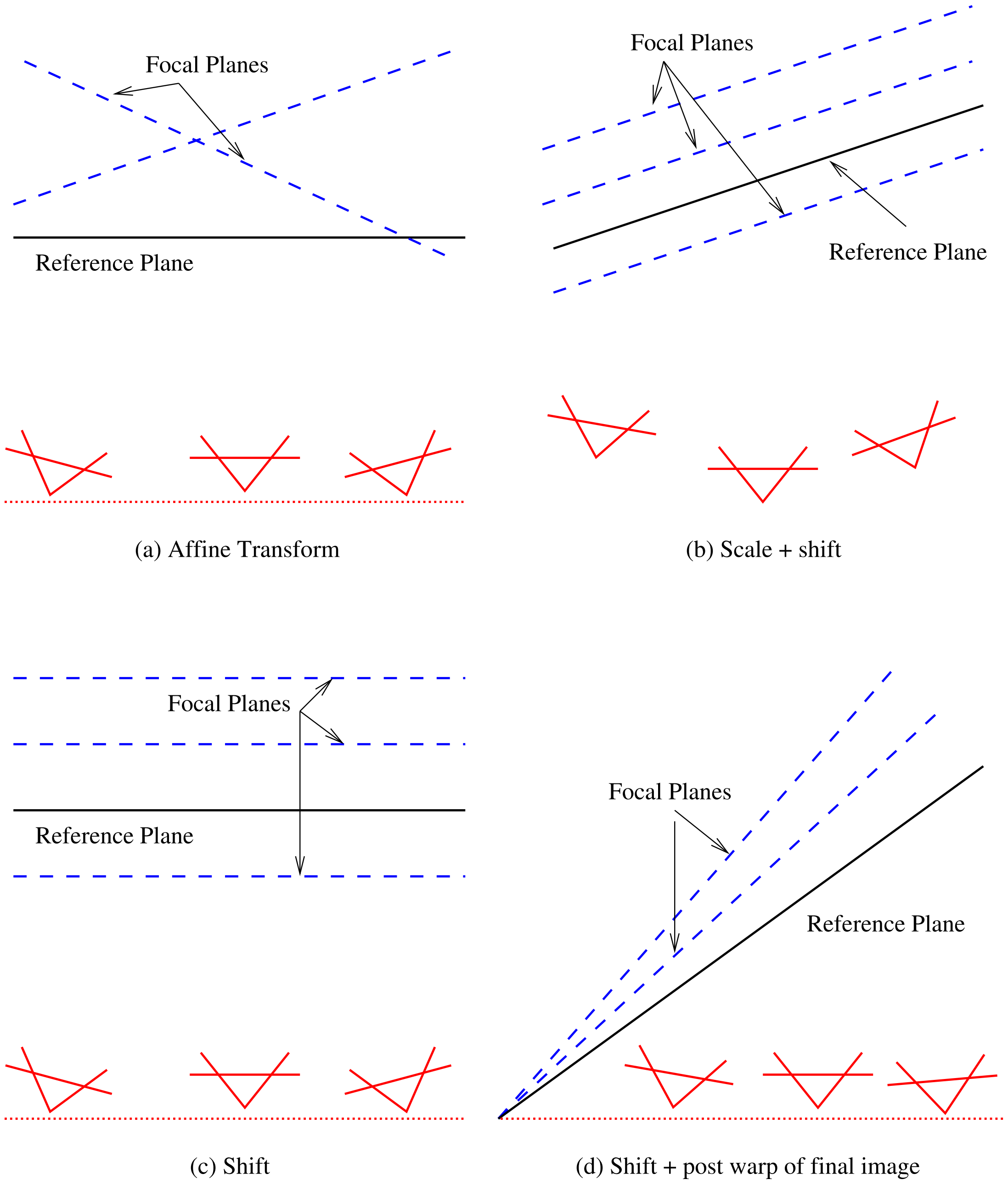
Affine transforms: When the camera centers lie on a plane, and the reference plane is parallel to this plane, the epipoles $e_i = [e_i^{(x)}\ e_i^{(y)}\ 0]^T$ are points at infinity. The bottom row of the homology matrix $I + \mu e_i l^T$ becomes [0 0 1]. Hence, homologies for any focal plane are affine transforms. Note that this holds for arbitrary focal planes. Fig. (3.3) shows an instance of this configuration.

Scale and translation: When the focal plane and reference plane are parallel, their intersection is the line at infinity $l = [0\ 0\ 1]^T$ on the reference plane. Writing the epipoles as $e_i = [e_i^{(x)}\ e_i^{(y)}\ e_i^{(z)}]^T$, the homologies are of the form:

$$K_i = I + \mu e_i l^T = \begin{bmatrix} 1 & 0 & \mu e_i^{(x)} \\ 0 & 1 & \mu e_i^{(y)} \\ 0 & 0 & 1 + \mu e_i^{(z)} \end{bmatrix}$$

This is just a scale followed by a shift. Thus, if we wish to vary the focus through a family of planes parallel to the reference plane, we need to just scale and shift the images before computing their average. Note that this holds for arbitrary camera positions. This configuration is illustrated in Fig. (3.4b)

Translation: When both the preceding conditions hold - the cameras are on a plane and the reference plane and focal planes are parallel to the camera plane, the



(a) Affine Transform

(b) Scale + shift

(c) Shift

(d) Shift + post warp of final image

Figure 3.4: Taxonomy of homologies and the corresponding camera/focal plane configurations.

epipoles e_i and the line of intersection l are at infinity. The homologies reduce to translations:

$$K_i = I + \mu \begin{bmatrix} e_i^{(x)} \\ e_i^{(y)} \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \mu e_i^{(x)} \\ 0 & 1 & \mu e_i^{(y)} \\ 0 & 0 & 1 \end{bmatrix}$$

and we can vary the focus by simply translating the images. This configuration is illustrated in Fig. (3.4c)

Translation + warp of final image (Scheimpflug): There is one additional configuration in which the homologies can be reduced to translations. Consider the case when the cameras lie on a plane, and the camera plane, reference plane and desired focal plane intersect in the same line l (Fig. 3.4d). Thus, the epipoles lie on the line of intersection l. Such a homology is considered degenerate and is called an *elation*.

To reduce the elations to image translations, we can redefine the coordinate system on the reference plane so that l is the line at infinity, i.e. $l \cong [0\ 0\ 1]^T$. This becomes identical to the previous case, and the homologies are again reduced to shifts. This is not surprising since from a projective geometer's viewpoint, Fig. (3.4c) and Fig. (3.4d) are equivalent configurations.

There are two consequences of changing the coordinate system on the reference plane. First, this changes the homographies required to project the original camera images on to the reference plane. This is not a significant change in terms of the computation: we would need to apply homographies to project camera images on to the reference anyway. However, it does mean that we need to calibrate the homographies for the new coordinate system (or estimate them in some other way). Second, when we shift and add the images to change the focal plane, the resulting synthetic aperture image will be computed in the new coordinate system. This results in an image that looks skewed to a human observer since it corresponds to an

Cameras	**Reference Plane**	**Focal Planes**	**Homologies**
Lie on plane	Parallel to camera plane	Arbitrary	Affine transforms
Arbitrary	Arbitrary	Parallel to reference plane	Scale + translation
Lie on plane	Parallel to camera plane	Parallel to camera plane	Translation
Lie on plane	Arbitrary	Intersect reference plane and camera plane in same line	Translation + homography to final image

Table 3.1: Summary of the configurations in which the homologies are affine transforms or simpler, in terms of the conditions on the camera layout, reference plane and focal planes. These are illustrated in Fig (3.4).

unusual off-axis projection. We can correct this by applying a homography to the synthetic aperture image that restores it to the original coordinate system. Note that this homography needs to be applied only to the synthetic aperture image, the per-camera computational cost does not change.

This configuration (Fig 3.4d) is analogous to the Scheimpflug condition in photography for focusing with a view camera on tilted planes: that the lens plane, sensor plane and focal plane must intersect in a common line [46]. Thus, we refer to it as the Scheimpflug configuration.

It is easy to show that these are the only configurations in which the homologies for refocusing are affine transforms or simpler. Moreover, any configuration in which the homologies reduce to translations can be shown to be a Scheimpflug configuration. The cases discussed above are summarized in Table (3.1) and illustrated in Fig. (3.4).

3.3.1 Applications

The configurations above are useful in a variety of applications. A common feature amongst these is the requirement to first project the camera images on to a reference plane to take advantage to the resulting geometric simplicity. The images projected on to a reference plane form what is called a *keystone corrected* light field [26], in computer vision this is also called *rectification* [12]. Many applications (calibration, computing synthetic aperture images, reconstructing focal surfaces) require these rectified images as input.

Reconstructing 3D surfaces from images (which could then be used as focal surfaces) is implemented by sweeping the focus over a family of parallel planes and determining the most likely depth for each pixel [4]. This corresponds to the scale + translation configuration (Fig. 3.4b), which means the plane sweep requires only similarity transforms if we use the rectified images. This configuration was first observed by Szeliski and Golland [37].

The frontoparallel configuration (Fig. 3.4c) is the simplest and most extensively studied. We show how it leads to a simple and robust algorithm for camera calibration in the next chapter. The key advantage of this configuration for calibration is that we *do not need to calibrate the μ-vectors* which makes it possible to solve for the calibration parameters using *linear* optimization. Building on this, Wilburn et al have developed an optical flow procedure for view interpolation in space and time [48]. The image shifts required for changing the focus are relatively simple to realize in per-camera hardware which leads to the development of a real-time synthetic aperture focusing system we describe in the next chapter.

We can also vary the focus using just image shifts for tilted planes in the Scheimpflug configuration (Fig. 3.4d) by redefining the coordinate system on the reference plane. This lets us use our real-time focusing system to focus on tilted planes without any modification other than a final homography of the synthetic aperture image. Since this homography is applied to only one image (rather than every camera image), there is no problem in doing this in real-time.

It has been shown by Ng that focusing by shifting and adding the images corresponds to a 4D shear of the light field, which corresponds to slicing in the Fourier domain [29]. This leads to insights on the signal-theoretic limits on refocusing as well as computational speed-ups. They restricted their analysis to the frontoparallel focal planes. The Scheimpflug configuration shows how their work could be extended to view cameras with tilted focal planes.

3.4 Proof of μ-vector theorem

μ-vector theorem. *Given epipoles $e_1, \ldots, e_N$ on a reference plane as defined above, the μ-vectors for all focal planes lie in a 1-D space, i.e. μ-vectors for any two planes are equal up to scale.*

To show the μ-vectors live in a 1D space, it suffices to show that the ratio $\mu_i/\mu_j, 1 \leq i < j \leq N$ is independent of the plane Π.

Without loss of generality, we may take $i = 1, j = 2$. Let P be a point on Π, and p_0, p_1, p_2 be its projections on to the reference plane through centers of projection C_0, C_1, C_2 (Fig. 3.5). Let e_{12} be the epipole associated with camera centers C_1, C_2 lying on the reference plane. The epipoles e_1, e_2, e_{12} lie on the line of intersection of the reference plane and the plane of camera centers C_0, C_1, C_2 by Desargues' theorem [5]. By epipolar geometry, e_{12}, p_1, p_2 are collinear. We will use $|a\ b\ c|$ to denote the determinant of the matrix whose columns are the 3-vectors a, b, c. Note that $|a\ b\ c| = 0$ if a, b, c are collinear. We have

$$
\begin{aligned}
p_1 \cong K_1 p_0 &= p_0 + \mu_1 e_1 (l^T p_0) && (3.3)\\
p_2 \cong K_2 p_0 &= p_0 + \mu_2 e_2 (l^T p_0) && (3.4)\\
|e_{12}\ e_1\ e_2| &= 0 && (3.5)\\
|e_{12}\ p_1\ p_2| &= 0 && (3.6)
\end{aligned}
$$

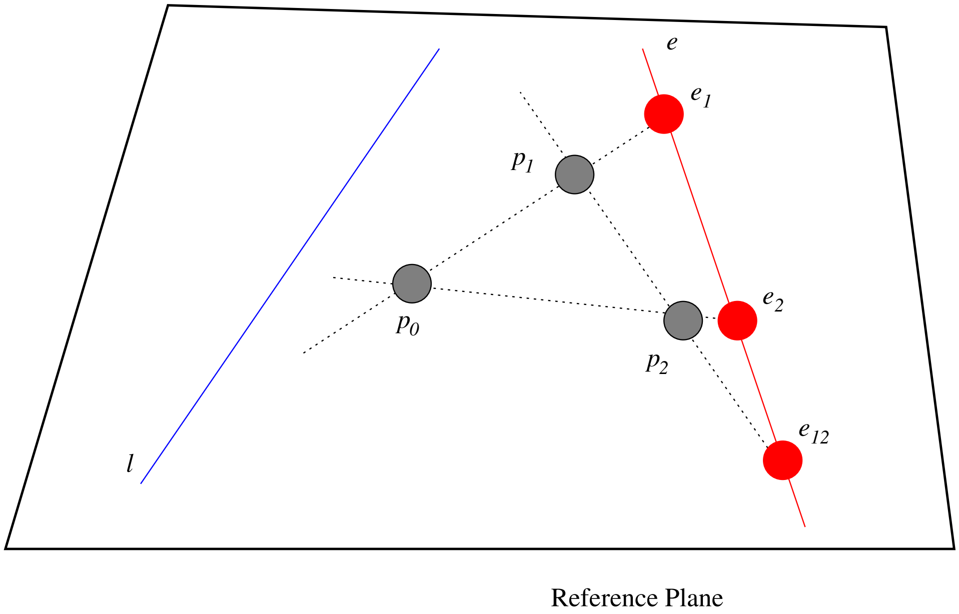


Figure 3.5: Proof of the μ-vector theorem. p_0, p_1, p_2 are images of a point P on the focal plane Π in cameras C_0, C_1, C_2 (see Fig 1b). e_1, e_2, e_{12} are the epipoles. The plane through camera centers C_0, C_1, C_2 intersects the reference plane in a line e containing these epipoles. l is the intersection of the focal plane with the reference plane.

where the first two equations follow from the formula for homologies (3.2) and the last two from collinearity. If we substitute (3.3) and (3.4) in (3.6), expand using properties of determinants and use (3.5), we get

$$\begin{aligned} 0 &= |e_{12}\ p_1\ p_2| \\ &= |e_{12}\ p_0 + \mu_1 e_1 (l^T p_0)\ p_0 + \mu_2 e_2 (l^T p_0)| \\ &= |e_{12}\ \mu_1 e_1 (l^T p_0)\ p_0| + |e_{12}\ p_0\ \mu_2 e_2 (l^T p_0)| \\ &= \mu_1 |e_{12}\ e_1\ p_0| + \mu_2 |e_{12}\ p_0\ e_2| \\ &= \mu_1 |e_{12}\ e_1\ p_0| - \mu_2 |e_{12}\ e_2\ p_0| \end{aligned}$$

This yields

$$\mu_1/\mu_2 = |e_{12}\ e_2\ p_0| / |e_{12}\ e_1\ p_0|$$

The right hand side does not depend on the plane Π (we can also show it does not depend on p_0). This completes the proof.

Chapter 4

Synthetic Aperture Focusing System

Having understood the geometry underlying the computation of synthetic aperture images, we can now describe the implementation of a synthetic aperture focusing system. This chapter describes our method of camera calibration and real-time focusing system. Both of these rely on varying the focus by shifting the camera images (Fig. 3.4c,d). The key advantage of the calibration method is that it does not require a full metric calibration of the cameras; the homographies for projection on a reference plane and relative camera positions suffice. We show these can be computed using straightforward linear optimization, which avoids local minima that can occur in a metric calibration. The key strength of the real-time focusing system is the ability to map the computation to relatively inexpensive per-camera hardware. One limitation common to both the calibration and the real-time system is that the cameras are constrained to lie on a plane.

We will first describe our calibration procedure and compare it with metric calibration. We will show how we can compute the image shifts required to focus on frontoparallel focal planes. Finally, we will describe how the image warps can be mapped to per-camera hardware leading to real-time focusing with synthetic aperture video. The calibration procedure was first described in [43] and the real-time system in [41].

4.1 Camera Calibration

The calibration procedure consists of two steps: estimating the homographies for backprojection on to the reference plane, then estimating the relative camera positions (epipoles) from parallax measurements. Once we have the camera positions we will use them to focus on different focal planes.

4.1.1 Finding Homographies

To compute homographies for projection on to the reference plane, we place a planar calibration grid (such as the one in Fig. 3.1) in the scene. The plane of the calibration grid is chosen to be the reference plane. The calibration grid defines a 2D Euclidean coordinate system on the reference plane. We locate the positions of the corners in the grid using an automatic feature detector [40]. By matching the positions of the grid corners in the camera images to their known coordinates on the calibration grid, we can compute the homographies to project the camera images on the reference plane. A minimum of four points on the grid are required, in practice we use about 100-140 points and solve for the homography using linear least squares [12].

4.1.2 Camera Positions from Parallax

Having projected the images, let us see what is required to vary the focal plane. We will restrict ourselves to the frontoparallel configuration (Fig. 3.4c). Consider a set of cameras whose centers $C_0, \ldots, C_N$ lie on a plane and whose images have been projected on to a plane parallel to the reference plane. A point P not on the reference plane will have images p_0, p_i in cameras C_0, C_i (Fig. 4.1). The *parallax* of P between the two cameras is given by $\Delta p_i = p_i - p_0$. To focus on the plane Π through P, we need to shift the image from camera C_i by an amount $-\Delta p_i$ to eliminate the parallax. Our goal is to:

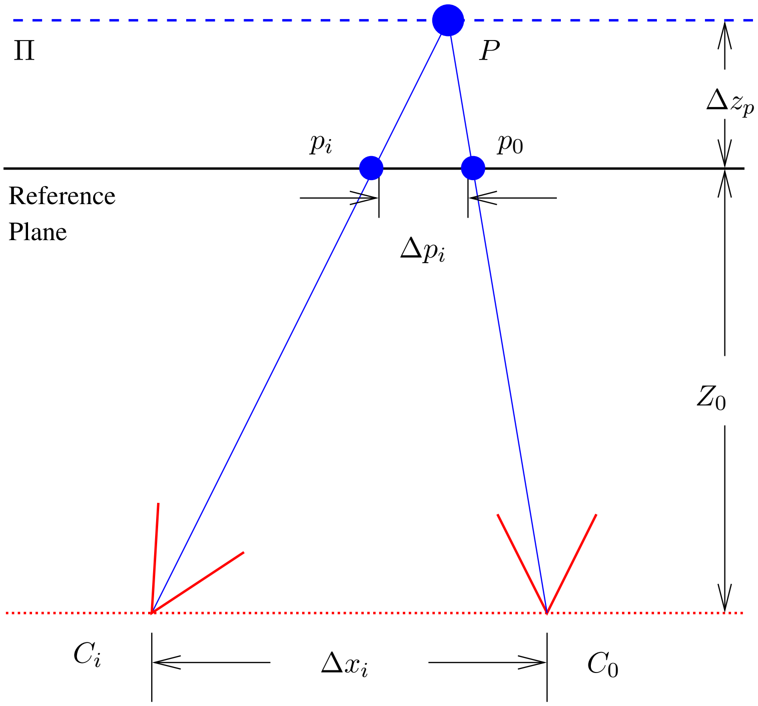


Figure 4.1: Calibration with frontoparallel planes. A point P not on the reference plane has distinct image p_0, p_i in cameras C_0, C_i. The parallax between these two depends on the relative camera displacement Δx_i and the relative depth of P

1. Estimate the relative camera positions using parallax measurements.
2. Use the estimated camera positions to compute the parallax for frontoparallel focal planes at arbitrary depths.

Note that the parallax is constant for all points on the plane Π. Thus, to focus on a frontoparallel plane we need to know the parallax induced by it across every pair of cameras. From similar triangles, we can show the parallax is given by

$$\Delta p_i = \Delta x_i \frac{\Delta z_p}{\Delta z_p + Z_0} \tag{4.1}$$

The parallax is product of the relative camera displacement Δx_i and the relative depth of P, $d_P = \frac{\Delta z_p}{\Delta z_p + Z_0}$. The first factor depends only on the camera geometry, the second only on the depth of P. To estimate the camera positions, we need to factorize the parallax measurements. If we measured the parallax of P with respect to C_0 in all the cameras, we would get the camera positions relative to C_0 up to an unknown scale (given by the relative depth of P).

Let us extend this to all cameras and multiple scene points. We designate camera C_0 as the reference view with respect to which we will measure parallax. Suppose we observe m points $P_1, \ldots, P_m$ with relative depths $d_1, \ldots, d_m$. In practice, these are obtained by moving the calibration grid to different (unknown) positions and detecting the corners in its images. We will denote by Δp_{ij} the parallax of point P_j in camera C_i. We can factorize the matrix of parallax measurements:

$$\begin{bmatrix} \Delta p_{11} & \ldots & \Delta p_{1m} \\ \vdots & \ddots & \vdots \\ \Delta p_{N1} & \ldots & \Delta p_{Nm} \end{bmatrix} = \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_N \end{bmatrix} \begin{bmatrix} d_1 & \ldots & d_m \end{bmatrix} \qquad \text{or}$$

$$\mathbf{\Delta P} = \mathbf{\Delta X}\ \mathbf{D} \tag{4.2}$$

Here $\mathbf{\Delta P}, \mathbf{\Delta X}, \mathbf{D}$ denote the matrix of parallax measurements, (column) vector of camera displacements with respect to the reference camera and (row) vector of relative depths respectively. Thus, parallax measurements form a rank-1 matrix. We can find the nearest rank-1 approximation to the observed parallax matrix $\mathbf{\Delta P}$ using singular value decomposition. This yields $\mathbf{\Delta X}$ the camera positions relative to C_0 up to an unknown scale. These are sufficient to compute the shifts required to focus on any frontoparallel plane, as described below.

4.1.3 Focusing on Frontoparallel Planes

Suppose we wish to focus on a frontoparallel plane that causes parallax Δp between cameras C_0 and C_1. We can compute the relative depth d of the desired focal plane from the relation (4.1):

$$\Delta p = d \Delta x_1$$

Having obtained d, we can compute the parallax caused by the focal plane in camera C_i with respect to camera C_0 using Eq. (4.1).

Crucially, this approach works correctly even if the relative camera positions Δx_i are known only up to scale. A scale change in camera positions $\mathbf{\Delta X}$ causes a corresponding scale change in the relative depths $\mathbf{D}$, but there is no change in parallax. The key idea is to specify the depth of a frontoparallel focal plane by specifying the parallax it causes between two cameras (which we chose to be C_0, C_1).

4.1.4 Focusing on Tilted Planes

We can also use the relative camera positions Δx_i to compute the homologies required to focus on arbitrary planes. As described in the previous chapter, we need to know the epipoles e_i and scalars μ_i for each camera. If we take C_0 to be the reference camera, these parameters for the camera C_i with position Δx_i are given by

$$\begin{aligned} e_i &= \begin{bmatrix} \Delta x_i \\ 0 \end{bmatrix} \\ \mu_i &= 1 \end{aligned}$$

These can be used to compute the homology to focus on any focal plane. The synthetic aperture image on a tilted focal plane in Fig. (3.3d) was computed by this method.

4.1.5 Minimal Calibration Requirements

We have described a calibration procedure for computing synthetic aperture images when the camera centers lie on a plane and the reference plane is chosen to be parallel to the plane of camera centers. This requires that we observe

1. At least four points on the reference plane, of which no three are collinear.

This is to estimate the homographies for projecting the camera images on to the reference plane.

2. At least one point not on the reference plane. This is to estimate the relative camera positions using Eq. (4.2).

With this calibration, we can focus on *both* frontoparallel and tilted planes as described above. If the camera centers do not lie on a plane, we would need to observe at least *two* points not on the reference plane to estimate the epipoles and μ-vector required for synthetic aperture focusing. These can be estimated using standard projective calibration techniques [12].

4.2 Experiments

We used our method for calibration and focusing to see through dense foliage outdoors using an array of 45 cameras arranged in a 3×15 grid (Fig. 4.2). The cameras were within 5cm of being perfectly planar. We used narrow angle lenses whose field of view was approximately 4.5^o. The image resolution was 640x480.

To calibrate the array, we used an 85cm × 85cm calibration grid. The reference plane was set by placing the grid 28m in front of the array, kept approximately parallel to the plane of cameras. To measure parallax of points not on the reference plane, we moved the calibration grid to six different positions ranging from 28m to 33m from the array. These gave us parallax measurements for 864 points. We measured parallax with respect to the central camera, and computed relative camera positions using rank-1 factorization (Eq. 4.2). The RMS error between the rank-1 factorization and the observed parallax measurements was 0.3 pixels (per parallax observation), which suggests our approximation of a planar camera array and parallel reference plane were fairly accurate.

	Parallax Calibration	**Metric Calibration**
Camera Layout	Planar	Arbitrary
Optimization	Linear	Non-linear, initial estimate required
Parameters calibrated	Reference plane homographies and relative camera positions up to scale	All (intrinsics and pose)

Table 4.1: Comparison of parallax-based and metric calibration.

One challenge for calibration in this experiment is that we can take calibration measurements (specifically, measurements of parallax between points descried in Eq. (4.2)) only in front of the bushes (Fig. 4.2). However, we seek to focus *behind* the bushes - i.e. at depths where the cameras cannot observe the calibration grid. Any error in the calibration of relative camera positions $\mathbf{\Delta X}$ will get magnified as we focus further and further behind the calibration volume.

We have computed synthetic aperture images focused at depths from 28m to 45m from the camera array (Fig. 4.3, 4.4). We found we could get sharp focus on objects more than 10m behind the depths to which me moved the calibration grid. This suggests that the error in our calibration is small enough that it can be "extrapolated" beyond the calibration volume. For the setup of Fig. 4.2, we can focus at depths for which the parallax is about 2.5 times greater than the parallax measurements used for calibration.

4.2.1 Comparison with Metric Calibration

An alternative approach to is perform a metric calibration of the cameras. A metric calibration gives the mapping from camera pixels to the equation of the ray it corresponds to, and requires estimating about 10 parameters per camera (such as its focal length, principal point, pose). Our first approach to calibration was to implement a metric calibration system. Our implementation is based on a multi-camera

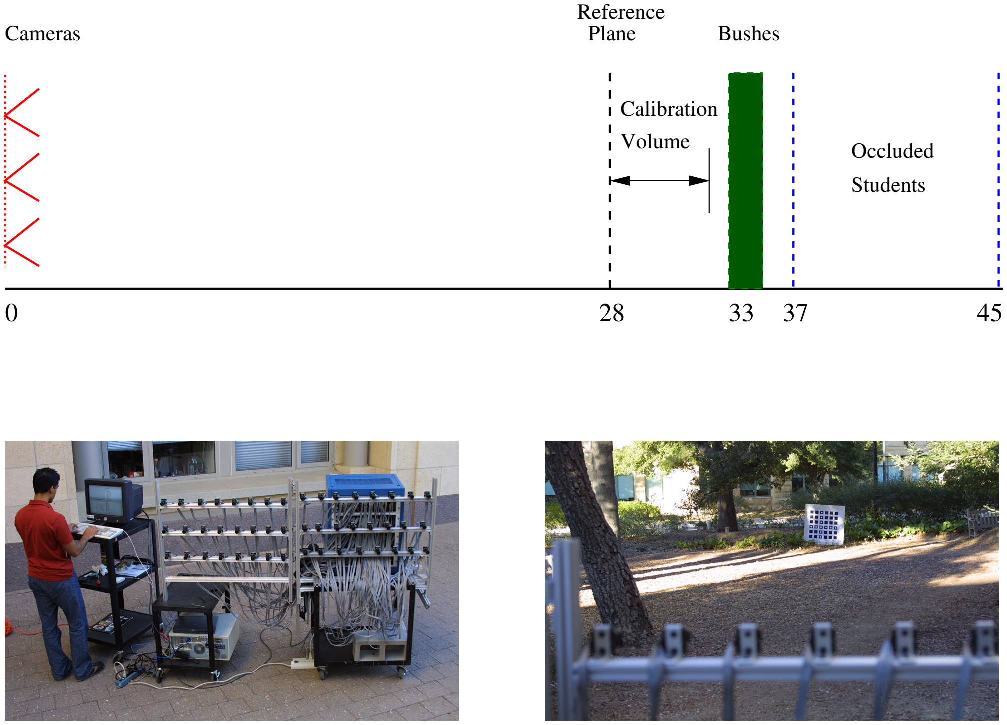
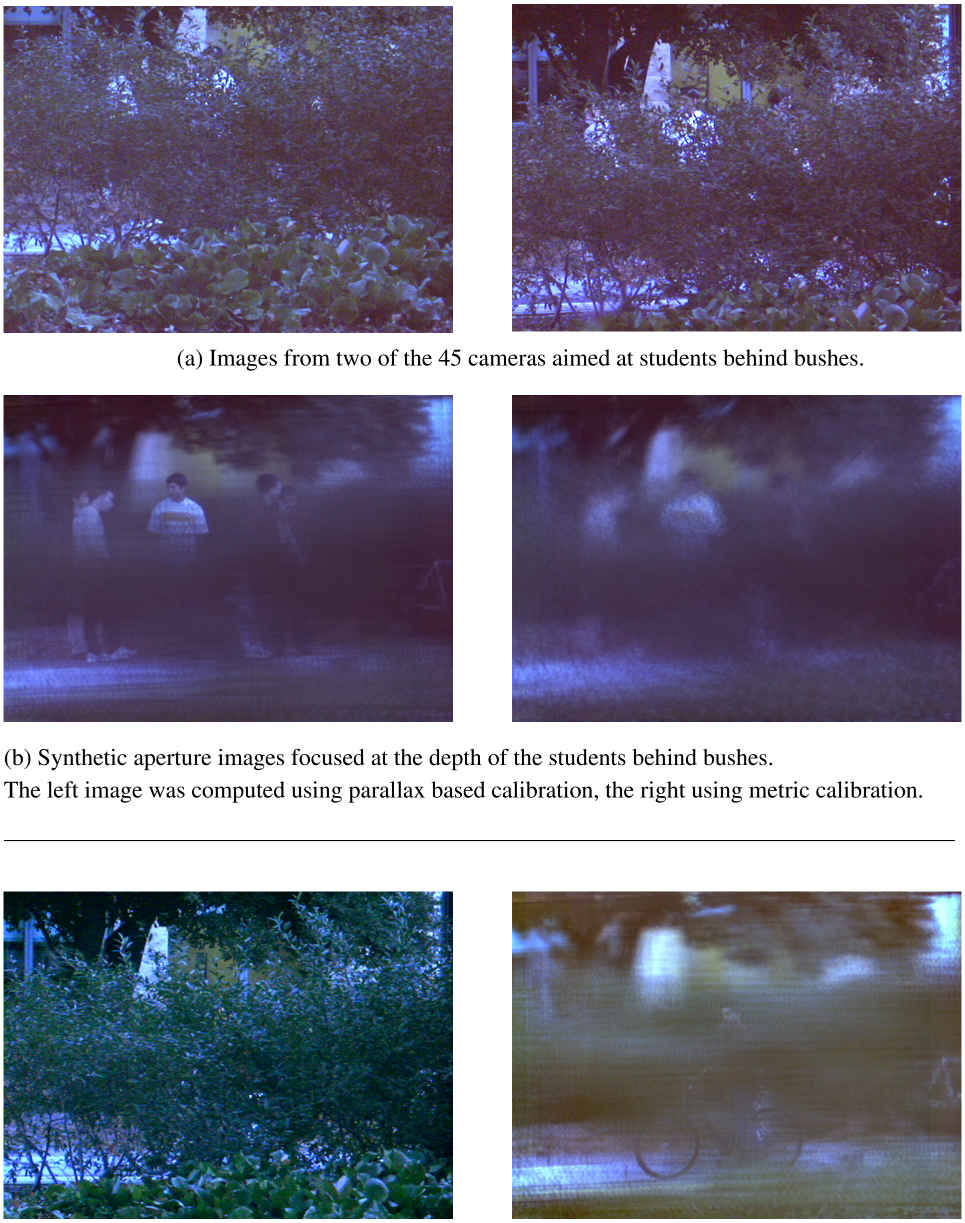


Figure 4.2: Outdoor synthetic aperture imaging setup. (a) Experimental setup showing scene layout and distances in meters from camera array. The calibration volume is the region over which the calibration grid was moved. (b) Our 45 camera array on a mobile cart. The array was controlled by a standalone 933 MHz Linux PC. (c) The view from behind the camera array showing our calibration target in front of the bushes. Our goal was to try and see people behind the bushes.

extension of Zhang's method [54]; details of this can be found in [39]. The metric calibration yields accurate results for lenses with a reasonably large field of view (for lenses with field of view 38^o, it is accurate to within 0.35 pixels) and we have used it successfully for synthetic aperture focusing in our laboratory.

However, for longer range outdoor imaging where it is desirable to use narrow angle lenses (such as the scenario of Fig. (4.2)) the metric calibration yields poor results. If we use metric calibration to compute synthetic aperture images, we get reasonable results when focusing in front of the bushes (which is the region where we took calibration measurements). We are unable to get a good focus on objects behind



(a) Images from two of the 45 cameras aimed at students behind bushes.

(b) Synthetic aperture images focused at the depth of the students behind bushes.
The left image was computed using parallax based calibration, the right using metric calibration.

(c) Left: image from one of the cameras of a cyclist behind the bushes. Right: synthetic aperture image focused on the cyclist. The cyclist is closer to the bushes than the students in the above example, which results in reduced visibility.

Figure 4.3: Results from the outdoor synthetic aperture imaging setup of Fig. (4.2) for students behind bushes and a cyclist behind the bushes. Note that the parallax based calibration yields much clearer results.

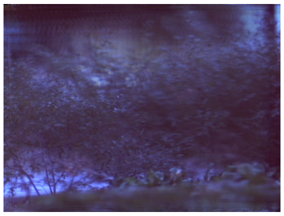
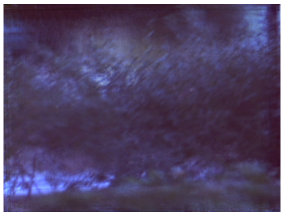
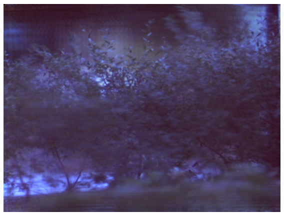
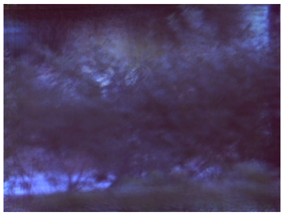
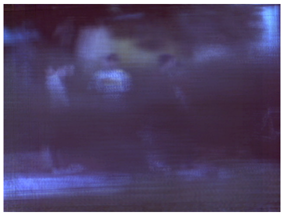
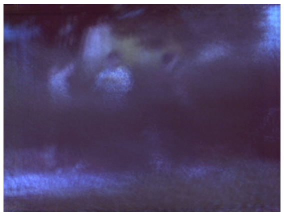
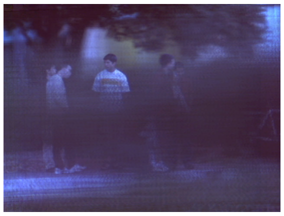
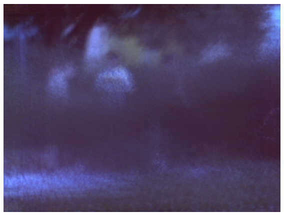
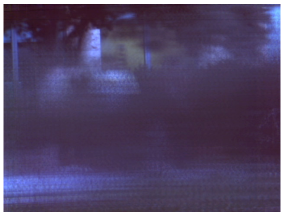
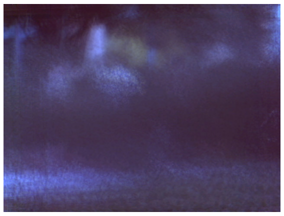
Depth of Focal Plane	Parallax calibration	Metric Calibration
In front of bushes		
On the bushes		
Behind the bushes		
On the students		
On the building		

Figure 4.4: Comparison of the synthetic aperture images computed using parallax based calibration and metric calibration for focal planes at different depths. The metric calibration gives reasonable focus in front of the bushes but does not yield a good focus behind the bushes.

the bushes (Fig. 4.3b, 4.4), which defeats our goal of seeing through the bushes. The failure of metric calibration forced us to study alternatives and led us to the parallax based calibration procedure described above.

Computing a metric calibration requires a nonlinear optimization about 450 parameters (assuming 45 cameras and 10 parameters per camera). This is initialized using Zhang's method [54] where parameters of each camera are estimated separately. Zhang's method cannot incorporate constraints like the camera centers lying on a plane. Estimating the camera pose and intrinsics becomes poorly conditioned as the field of view decreases. This leads to inaccurate initial estimates causing the nonlinear minimization to get stuck in a local minimum. In contrast, the parallax based calibration makes full use of the knowledge that the cameras lie on a plane. It requires only a linear optimization (a rank-1 factorization using SVD) so there is no danger of local minima.

There are three trade-offs made in the parallax-based method. First, we require the calibration grid be placed parallel to camera array to specify the reference plane. For narrow-angle lenses, an approximation by eyeballing is good enough. Second, we do not compute the intrinsics and orientation of each camera, these are left implicit in the homographies required for projection on to the reference plane. However, for computing synthetic aperture images, the homographies are adequate. Third, we have to give up the ability to specify the depth of the focal plane in metric units - we can only specify the depth implicitly in terms of parallax.

4.3 Real-time Focusing

We now discuss how to map the computation of synthetic aperture images to camera hardware to obtain real-time synthetic aperture video with interactive focus control. The challenge is to perform the necessary computation while keeping the per-camera cost hardware low. We can reduce the computation required for image warps by restricting the camera geometry or focal planes to the configurations of Fig. (3.4).

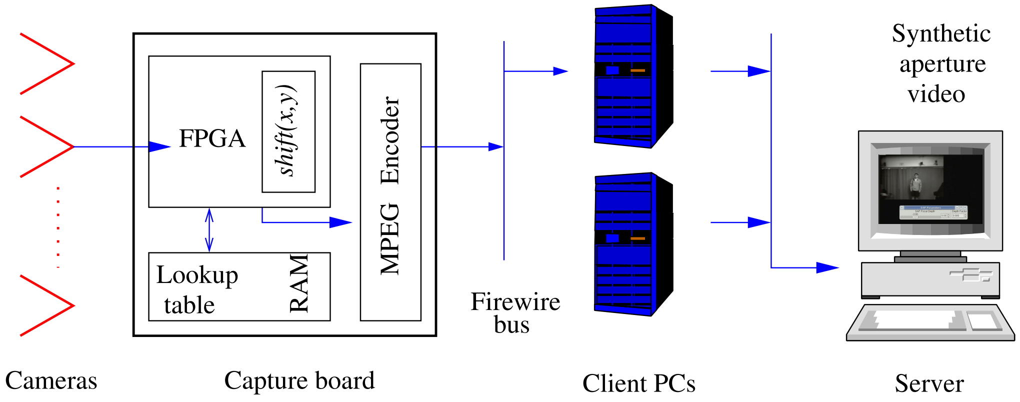


Figure 4.5: Real-time focusing pipeline. Frames from the camera are sent to a per-camera capture board where the homography for projection on to the reference plane is applied using a look-up table. The projected frames are shifted for the desired focal plane, MPEG encoded and transmitted over a firewire bus to a host PC. A host PC decodes streams from 15 cameras, computes their average and transmits the stream to the master PC. The master PC adds streams from two host PCs and displays synthetic aperture video.

This may be useful when the image warps are implemented in per-camera hardware since lesser computation required per camera could lead to simpler (and potentially less expensive) camera hardware.

If there are no restrictions on the cameras or focal planes, focusing on a plane requires applying one homography per camera. This requires three additions and two divisions per pixel of input. However, if we restrict ourselves to having the cameras on a plane parallel to the reference plane (Fig. 3.4a), we need to perform affine transforms to vary the focus. These require only two additions per pixel. A summary of the computation required for all configurations is presented in Table (4.2). In our implementation we restrict ourselves to the frontoparallel and Scheimpflug configurations (Fig. 3.4c,d) where we simply need to shift the images to change the focus.

Configuration	**Image transforms for refocusing**	**Computation**
Arbitrary camera positions and focal planes	Homographies	3 additions and 2 divisions per pixel
Cameras on plane parallel to reference plane (Fig. 3.4a)	Affine transforms	2 additions per pixel
Parallel focal planes (Fig. 3.4b)	Scale + translation	2 additions per pixel
Scheimpflug configurations (Fig. 3.4c,d)	Translation	2 additions *per camera*

Table 4.2: A comparison of the amount of computation required per camera per pixel for different configurations.

4.3.1 Experimental Setup

Our prototype consists 30 cameras from our array [48]. Each video stream is grayscale, runs at 30 frames/sec and has resolution 320 × 240. The processing pipeline is shown in Fig. (4.5). The video stream from each camera is sent to its capture board. The FPGA applies the initial homography required for projection onto the reference plane using a precomputed lookup table stored in RAM. For every pixel (x, y) in the warped image, the lookup table stores the corresponding pixel (truncated to integer coordinates) (x', y') in the incoming camera frame.

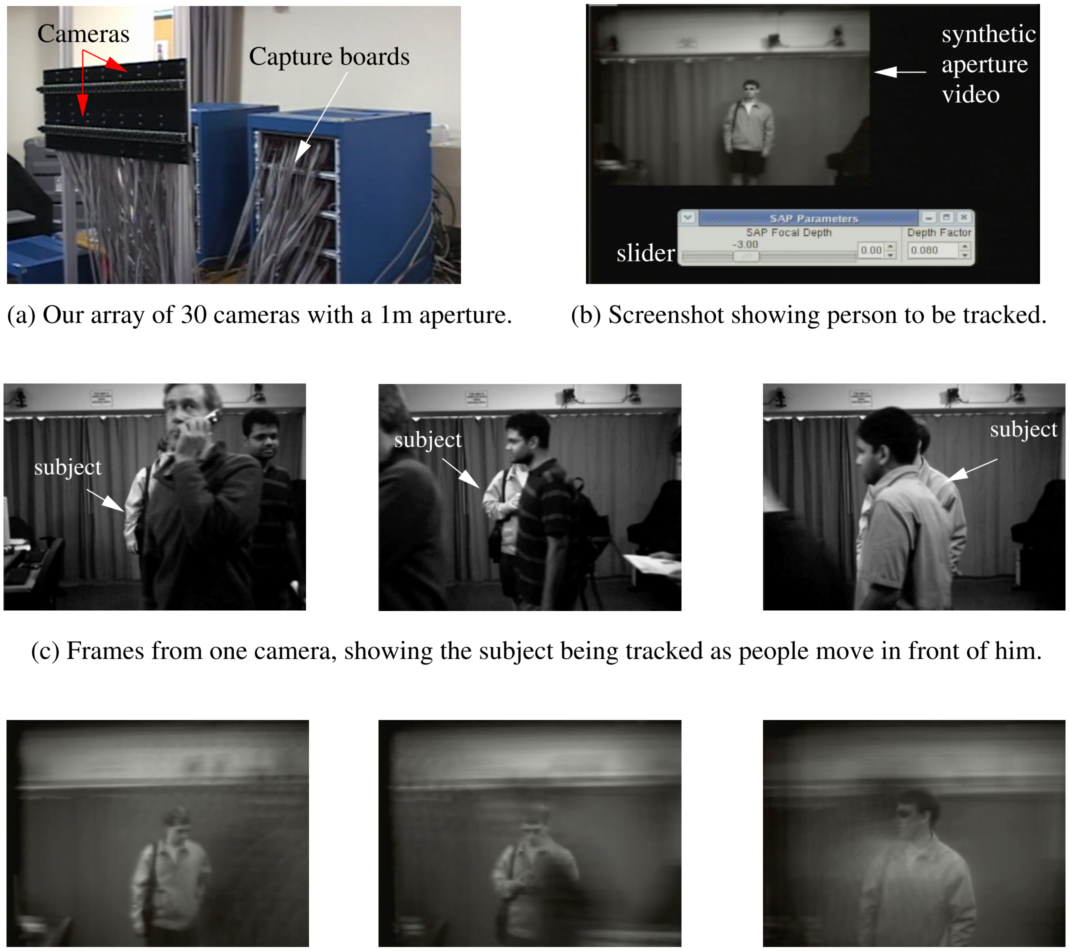
The warped frames are shifted, MPEG compressed and transmitted to the host PC. Each host PC receives streams from 15 cameras, which are decoded, added and sent over the network to a master PC. The master PC adds the streams from the hosts and displays the final synthetic aperture video. It can also perform a homography on the final image, as required for the Scheimpflug configuration. The user can move the focal plane back and forth with a slider which changes the shifts being performed in the FPGAs.

Since the homographies for projection on the reference plane do not change with the focal plane, we can precompute and cache them in a lookup table. This saves us the effort of having to compute homographies in the FPGAs which would require additions and divisions for every pixel, and may not be possible to fit in modest per-camera hardware. A lookup table is relatively easy to realize in memory and frees room in the FPGAs for the frame buffers. It can also let us apply arbitrary warps, such as lens distortion correction in conjunction with a homography. The trade-off is that this approach restricts the family of planes we can focus on.

4.3.2 Results

We show two examples of our system in action. The first scene consists of a person moving forward towards the cameras, with people walking in front of him (Fig. 4.6). By adjusting the focal plane with a slider as the subject moves, the user is able to keep him in focus while the people occluding him are blurred out.

The second scene consists of a suitcase placed at an angle to the camera array (Fig. 4.7). It is not possible to bring the entire suitcase into good focus using frontoparallel planes. Using tilted focal planes from a Scheimpflug configuration we can bring the entire suitcase into focus, while people in front of it are blurred out. (We urge the reader to view a video demonstrating the capabilities of our system in action at *http://graphics.stanford.edu/papers/shear-warp/a3diss.avi*).



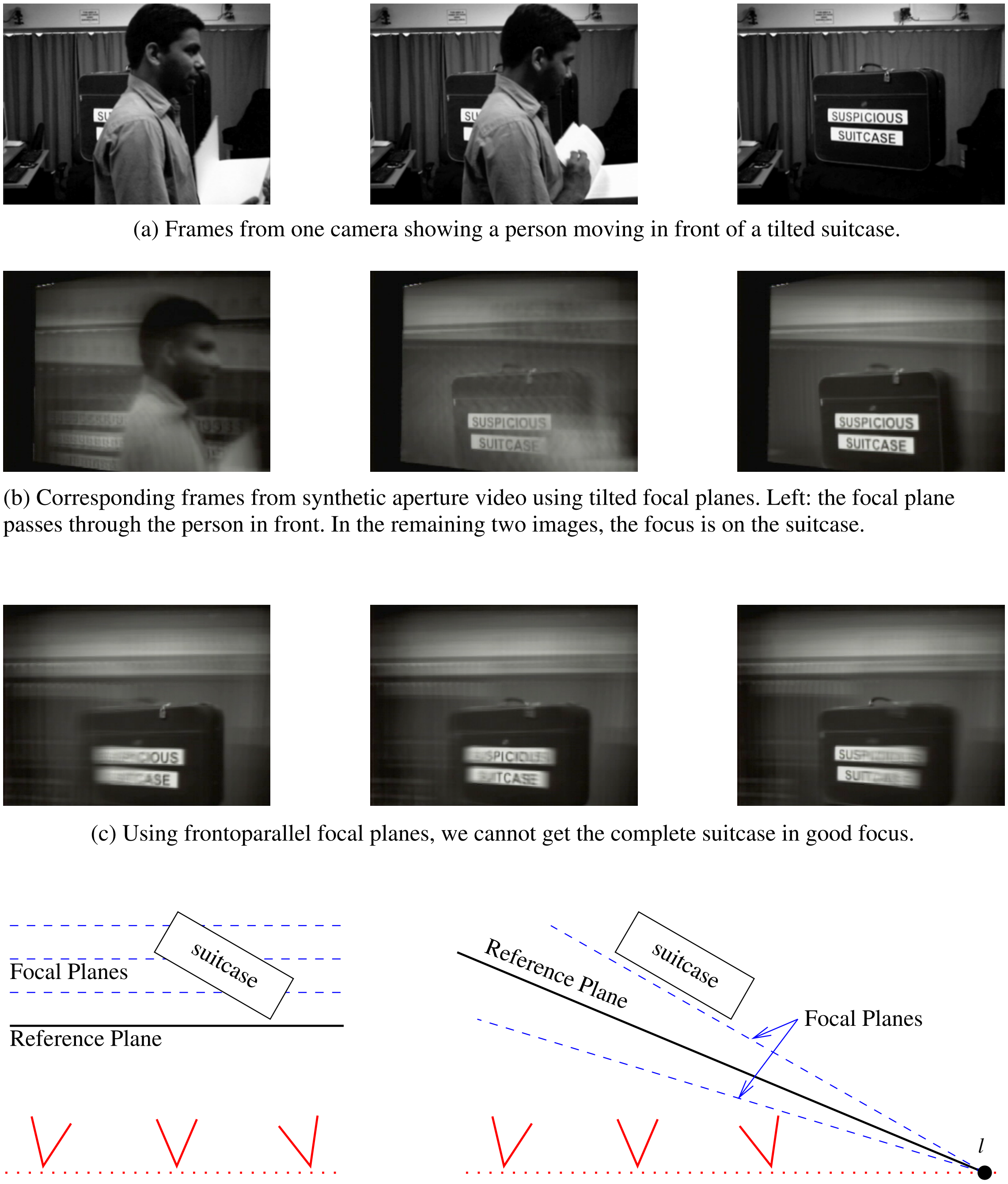
(a) Our array of 30 cameras with a 1m aperture.

(b) Screenshot showing person to be tracked.

(c) Frames from one camera, showing the subject being tracked as people move in front of him.

(d) Corresponding frames from the synthetic aperture video, keeping the subject in focus.

Figure 4.6: Real-time synthetic aperture video being used to track a person moving through a crowd.



(a) Frames from one camera showing a person moving in front of a tilted suitcase.

(b) Corresponding frames from synthetic aperture video using tilted focal planes. Left: the focal plane passes through the person in front. In the remaining two images, the focus is on the suitcase.

(c) Using frontoparallel focal planes, we cannot get the complete suitcase in good focus.

(d) Left: frontoparallel planes used in (c). Right: tilted planes in Scheimpflug configuration used in (b).

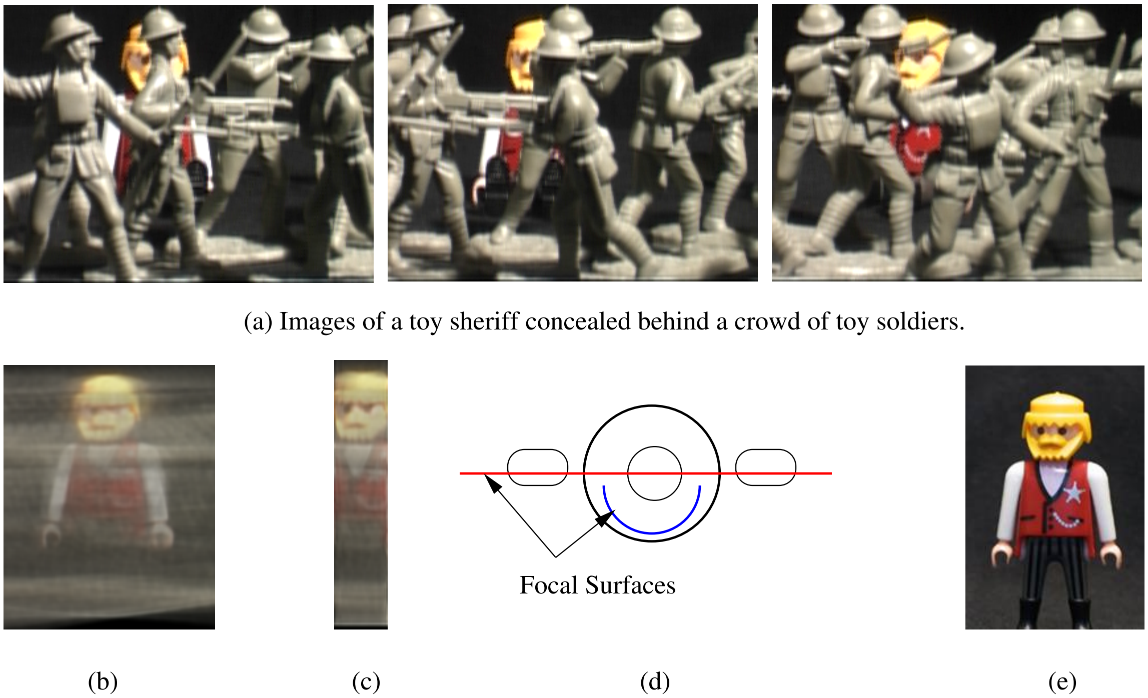
Figure 4.7: Focusing on tilted planes.

Chapter 5

Reconstructing Focal Surfaces

This chapter explores computation of synthetic aperture images where the focal depth and aperture for each pixel can be chosen independently of the remaining pixels. We have thus far restricted ourselves to synthetic aperture images that are focused on a *plane*, using *all* the cameras of the array. In these images points off the focal plane are blurred out. The large synthetic apertures we use result in an extremely shallow depth of field; this can be much smaller than the depth variation of the object (or objects) we wish to focus on. To view objects away from the focal plane - or parts of an extended object that extrude the focal plane - we must move the focal plane back and forth. If we used a focal surface that coincided with the surface of the object we wish to focus on, we would be able obtain a clearer view of the entire object in a single synthetic aperture image. An example of focusing on a curved surface that coincides with the object surface is shown in Fig. (5.1). In general, we would like to estimate the correct focal depth for each pixel in the synthetic aperture image and use the estimated depth map as our focal surface.

In addition to reconstructing per-pixel focal depths, we can also use *per-pixel apertures* to improve the contrast and clarity of synthetic aperture images. Suppose we wish to focus on an object which is concealed behind some occluders in the foreground. The resulting synthetic aperture image has a highly blurred (or defocused) image of the foreground occluders superimposed on the focused image of the desired


(a) Images of a toy sheriff concealed behind a crowd of toy soldiers.

(b)

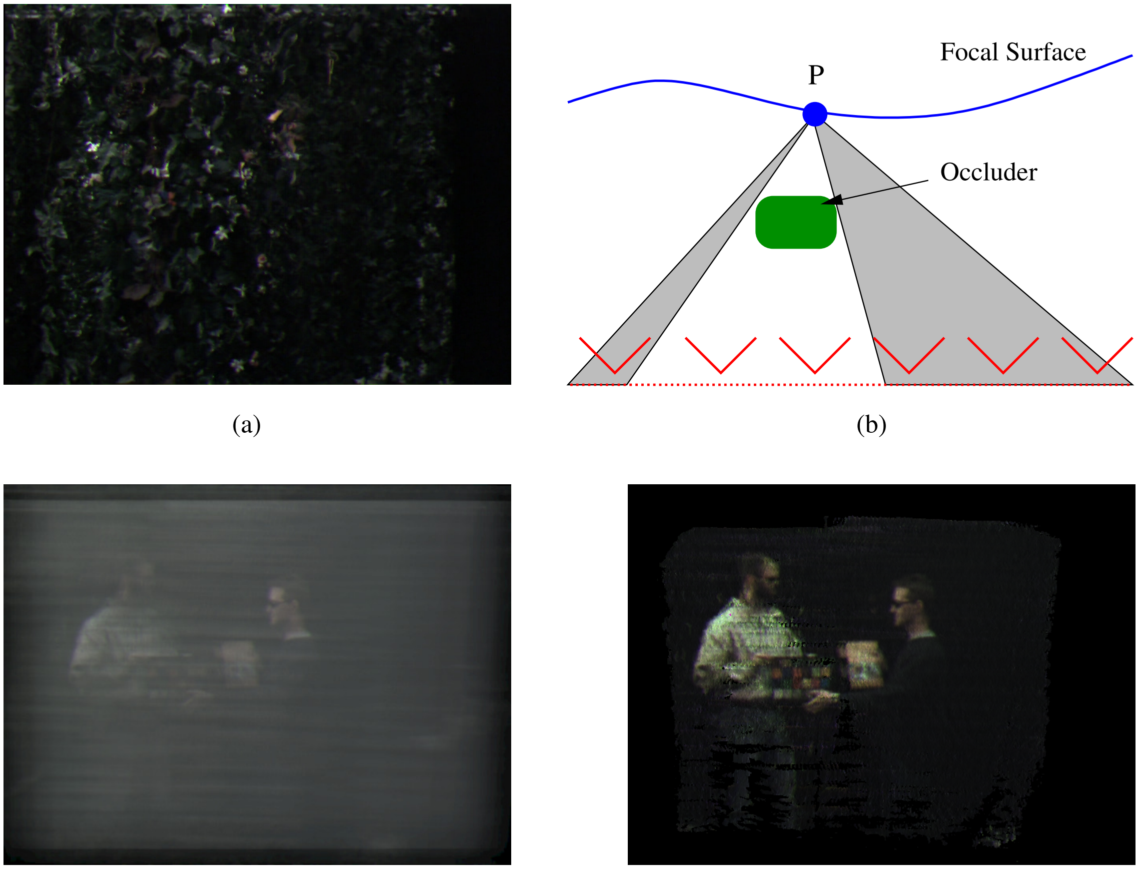
(c)

(d)

(e)

Figure 5.1: Focusing on a cylindrical surface. (a) 3 images from a scene consisting of a toy sheriff behind toy soldiers. A total of 60 camera positions in a ring around the scene were used. (b) Synthetic aperture image focused on a plane passing through the sheriff. (c) Synthetic aperture on a cylindrical focal surface circumscribing the sheriff. The face is in better focus than (b). (d) Cross section of the sheriff (in black) showing the focal surfaces used. The red focal plane was used to compute (b), the blue cylindrical focal surface for (c). (e) Unoccluded view of sheriff.

object. This reduces the contrast of the object in focus. Instead of integrating the all the rays from all the cameras (which is what we have done to compute all the synthetic aperture images so far), we could ignore the rays which are incident on the occluders and integrate only those rays which are incident on the object. Specifically, for each point on the focal surface, we integrate rays only from cameras in which that point is not occluded by a foreground object. By choosing a subset of camera rays for each point, we are effectively *shaping the aperture* for each pixel in the synthetic aperture so as to mask out the occluder. This greatly improves the contrast, as illustrated in Fig. (5.2). To achieve this, we need to be able to estimate



(a)

(b)

(c)

(d)

Figure 5.2: Using per-pixel shaped apertures to mask out occluders. (a) Image from one of 90 cameras of two students behind dense foliage. (b) Illustration of shaped apertures. For a point P on the focal surface we use rays only from those cameras in which it is visible. Using a subset of cameras amounts to shaping the aperture for each point. (c) Synthetic aperture image focused on the students. (d) The synthetic aperture image in which rays incident on the foliage were excluded. This was done by computing per-camera mattes for the foliage using blue screening and using per-pixel shaped apertures as shown in (b). Images (c) and (d) were computed using a frontoparallel focal plane at the depth of the students.

the desired aperture for every pixel.

5.1 Differences with Prior Work

To synthesize the clearest possible synthetic aperture view of an occluded object, we need to

1. Reconstruct the 3D surface of the object from the camera images.
2. Distinguish between the pixels corresponding to the foreground occluders and the occluded object.

This requires a 3D reconstruction and a foreground-background segmentation. These are two challenging inverse problems in computer vision for which accurate solutions are difficult to obtain, especially for scenes with dense occlusion.

Fortunately, to improve the clarity of synthetic aperture images we do not always require a perfect reconstruction and segmentation. For example, textureless surfaces are an ambiguous case for 3D reconstruction: there can be different surfaces which yield the same images. However, for this very reason the different surface hypotheses yield the same synthetic aperture images: we do not have to do any disambiguation. Thus, textureless surfaces are less of problem for synthetic aperture imaging than for a complete 3D reconstruction. Likewise, a complete foreground-background segmentation is not necessary as we do not need to identify *all* the rays incident on the object we wish to focus on, a non-empty subset suffices.

In this chapter, we compare two approaches for reconstructing 3D surfaces: shape from synthetic aperture focus and shape from stereo. We show that shape from focus requires a stronger surface texture for accurate reconstruction than shape from stereo; but for sufficiently textured surfaces focus performs better than stereo. We also describe two variants of stereo which increase its robustness to occlusions based on color medians and histogram entropy.

A crucial difference between our work and prior approaches is that we desire to reconstruct surfaces that are occluded in many of the camera images. This is a harder problem than classical multi-view stereo in which occlusions are ignored [31]. One approach to handling occlusions is to use shiftable windows and view selection developed by Kang et al [21]. However, this method reconstructs the depths of only those points which are visible in a reference view. Our method of reconstruction using color medians generalizes this so visibility in a reference view is not required.

Several algorithms for 3D reconstruction work by partitioning the scene into layers at different depths [44, 3, 55]. Translucent layers were considered in [37]. These approaches begin by first estimating the foreground layer and using residuals to estimate the rest. For complex foreground occluders whose reconstruction may be error-prone, errors in foreground estimation can severely degrade the reconstruction of the background. Since we seek to develop cost functions robust to occluders (which are considered outliers), we do not require an initial reconstruction of the foreground layer. In our experiments, we restrict the search for the occluded objects to depths behind the foreground.

Algorithms based on voxel coloring [34] attempt a complete scene reconstruction using a front-to-back sweep. Since a threshold is typically used to commit to foreground occluders, erroneous assignments result in background pixels marked as foreground or foreground pixels not deleted in background reconstruction. Both cases degrade the reconstruction of the background. Thresholding effects can be ameliorated to an extent using iterative probabilistic variants [6], but these are not guaranteed to converge. In Section 5.4, we compare our approach with voxel coloring.

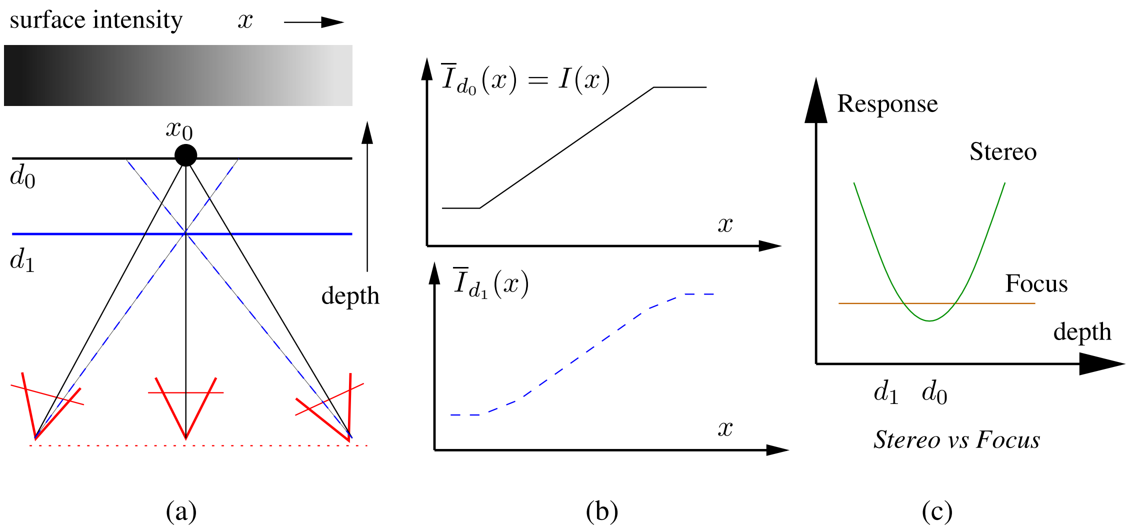


Figure 5.3: Stereo vs. focus for a surface at depth d_0 with a constant intensity gradient. (a) Camera layout, and rays considered for depth hypotheses $d = d_0$ and $d = d_1$. (b) The intensity profiles of the mean of camera images, projected on to depth planes $d = d_0$ (top) and $d = d_1$ (bottom). (c) Comparing the response of stereo (variance) and focus. The variance has a parabolic profile, with the minimum at the correct depth d_0. The mean image does not vary with depth, so shape from focus is not able to reconstruct depth accurately.

5.2 Stereo versus Focus

We begin by analytically comparing shape from synthetic aperture focus and shape from stereo. A common framework to describe the two is the space-sweep approach of Collins [4]. This involves sweeping a plane over a range of depths in the scene. At each depth (more precisely, disparity) d we compute a cost value for every pixel (x, y) on the plane, creating a *disparity space image* (DSI) $D(x, y, d)$ [37]. The main difference between stereo and focus methods is that they use different cost functions in constructing the DSI. A depth map $d(x, y)$ is computed by finding the minimum cost surface from the DSI:

$$d(x, y) = \arg\min_d D(x, y, d)$$

Let us now define the cost functions used by stereo and focus. For simplicity, we work in flatland: the cameras are rectified line cameras and the depth of a pixel determines its horizontal disparity with respect to a reference view [31]. Let $I_{i,d}(x)$ denote the image from camera i projected on the plane corresponding to disparity d. The mean and variance of the warped images from the N cameras are given by

$$\overline{I}_d(x) = \frac{1}{N}\sum_i I_{i,d}(x) \tag{5.1}$$

$$v_d(x) = \frac{1}{N}\sum_i (I_{i,d}(x) - \overline{I}_d(x))^2 \tag{5.2}$$

Shape from stereo uses the variance of rays $v_d(x, y)$ through the 3D point (x, y, d) as the cost function. Shape from focus uses the sharpness of the synthetically focused image $\overline{I}_d(x)$ as the cost function. The measure of sharpness we use is the gradient of the focused image:

$$f_d(x) = -\left[\frac{\partial \overline{I}_d(x)}{\partial x}\right]^2 \tag{5.3}$$

Which method is better? One straightforward observation we can make is that focus is less sensitive to sensor noise (due to averaging) and varying bias across cameras (since it computes a spatial derivative) than stereo. While both would fail for textureless surfaces, we can show that there are textured surfaces which can be reconstructed accurately by stereo but not by focus (Fig 5.3).

Theorem. *Shape from stereo can reconstruct the depth of diffuse surfaces whose 2nd and higher order spatial derivatives of radiance vanish, whereas shape from focus requires the radiance to have non-zero 3rd-order spatial derivatives.*

Proof. Consider a frontoparallel surface at disparity d_0 with intensity given by $I(x)$. We compute the response of stereo and focus at a point x_0 for disparity d_1. Let $\delta = d_1 - d_0$ and x_i the displacement of camera i from the central view. Given rectified images, we can compute $I_{i,d_1}(x)$ using a Taylor series expansion:

$$\begin{aligned} I_{i,d_1}(x_0) &= I(x_0 + x_i\delta) \\ &= I(x_0) + (x_i\delta)I_x(x_0) + \frac{(x_i\delta)^2}{2} I_{xx}(x_0) + \ldots \end{aligned}$$

Using Eqs. 5.1,5.2,5.3 and $\sum_i x_i = 0$, we can compute the mean image, variance and focus to be

$$\begin{aligned} \overline{I}_{d_1}(x_0) &= I(x_0) + I_{xx}(x_0)\delta^2 \sum_i x_i^2/N + O(\delta^3), \\ v_d(x_0) &= \delta^2 I_x^2(x_0) \sum_i x_i^2/N + O(\delta^4), \text{ and} \\ f_d(x_0) &= -[I_x(x_0) + \delta^2 I_{xxx}(x_0) \sum_i x_i^2/N + O(\delta^4)]^2 \end{aligned}$$

We see that for a non-zero gradient $I_x(x_0)$, stereo has a minimum at $\delta = 0$, enabling accurate reconstruction. However, if the 3rd order derivative $I_{xxx}(x_0)$ is zero, the focus response will be approximately constant as δ varies. This completes the proof.

The behavior of stereo and focus for a the special case of a constant gradient texture $I(x)$ is shown in Fig. (5.3). A weaker result for single lens apertures was proved in [7], where they showed that shape from focus requires textures with nonzero *2nd-order* gradients.

In the analysis above, we have ignored occlusions. How would the response of stereo and focus change if the surface we are trying to reconstruct is occluded in some views? In general, the response would depend on the nature of the occluder. For shape from focus, the mean image will have a blurred image of the occluder superimposed on it at the correct depth hypothesis. If the aperture is wide, the blurred image should be approximately constant and should not contribute to the spatial derivatives. However, the focus response will be attenuated by the cameras that do not observe the surface being reconstructed. For shape from stereo, the variance will not approach zero at the correct depth and may not attain a minimum there unless the occluder is of a fairly uniform color. In Section 5.4 we will show experimental comparisons of stereo and focus for reconstructing occluded surfaces.

5.3 Median and Entropy

One problem with stereo and focus is that both methods assign equal weightage to all rays through the 3D point (x, y, d) in constructing the DSI; this corresponds to using the full aperture for every pixel. When reconstructing occluded surfaces, many of these rays will actually hit occluders and should therefore be rejected as outliers (which amounts to shaping the aperture). In this section, we introduce two variants of stereo that try to mitigate the effects of outliers due to occlusions on depth and color reconstruction.

5.3.1 Shape from median

This approach is inspired by the observation that amongst measures of central tendency, the median is more robust to outliers than the mean. Consider the case of grayscale images. Suppose that a surface point (x, y, d) corresponds to pixels $S = \{I_{i,d}(x, y) \ : \ 1 \leq i \leq N\}$ in the N input images. If the point is occluded in some images, the median $I_M = \text{median } S$ will be a better estimator for the surface intensity than the mean. This is precisely why median colors have been used in matte extraction [45]. However, in addition to estimating the surface intensity we need a cost function to quantify the accuracy of the depth estimate. The cost function we use for constructing the DSI is the median distance of all the rays from I_M, i.e.,

$$D(x, y, d) = \text{median}\{|I_{i,d}(x, y) - I_M| \ : \ 1 \leq i \leq N\}$$

When a surface point is visible in more than half the cameras, the median color at the correct depth should correspond to the color of the occluded surface. As the number of occluded cameras increases over 50%, the estimate of the median begins to break down. An interesting question is how to extend shape from median to color images. There are many ways to generalize the notion of median to higher dimensions [1]. We have experimented with the component-wise median and the L1 median, and observed they yield similar results. We prefer to use the component-wise median, since it can be computed in $O(N)$ time.

5.3.2 Shape from entropy

Let us consider the distribution of intensities of the rays through the point (x, y, d), which may be visualized by plotting a histogram. If the depth hypothesis d is incorrect, these rays will hit different points on the foreground occluders and background surface. Given textured surfaces, we would expect the histogram to be spread over a wide range of intensities. If the depth hypothesis d is correct, rays hitting the background surface will be clustered over a small range of intensities, while those hitting the occluder will still be spread out resulting in a more peaked histogram. This suggests that we can use the Shannon entropy of the intensity histogram as a cost function for constructing the DSI. The entropy is maximized for a uniform distribution and decreases as the intensities are clustered together.

For 8-bit grayscale images, we divide the intensity range into $K = 16$ bins. If the number of rays in bin i is b_i, the entropy estimate is given by

$$H = -\sum_{i=1}^{K} \frac{b_i}{N} \log \frac{b_i}{N}.$$

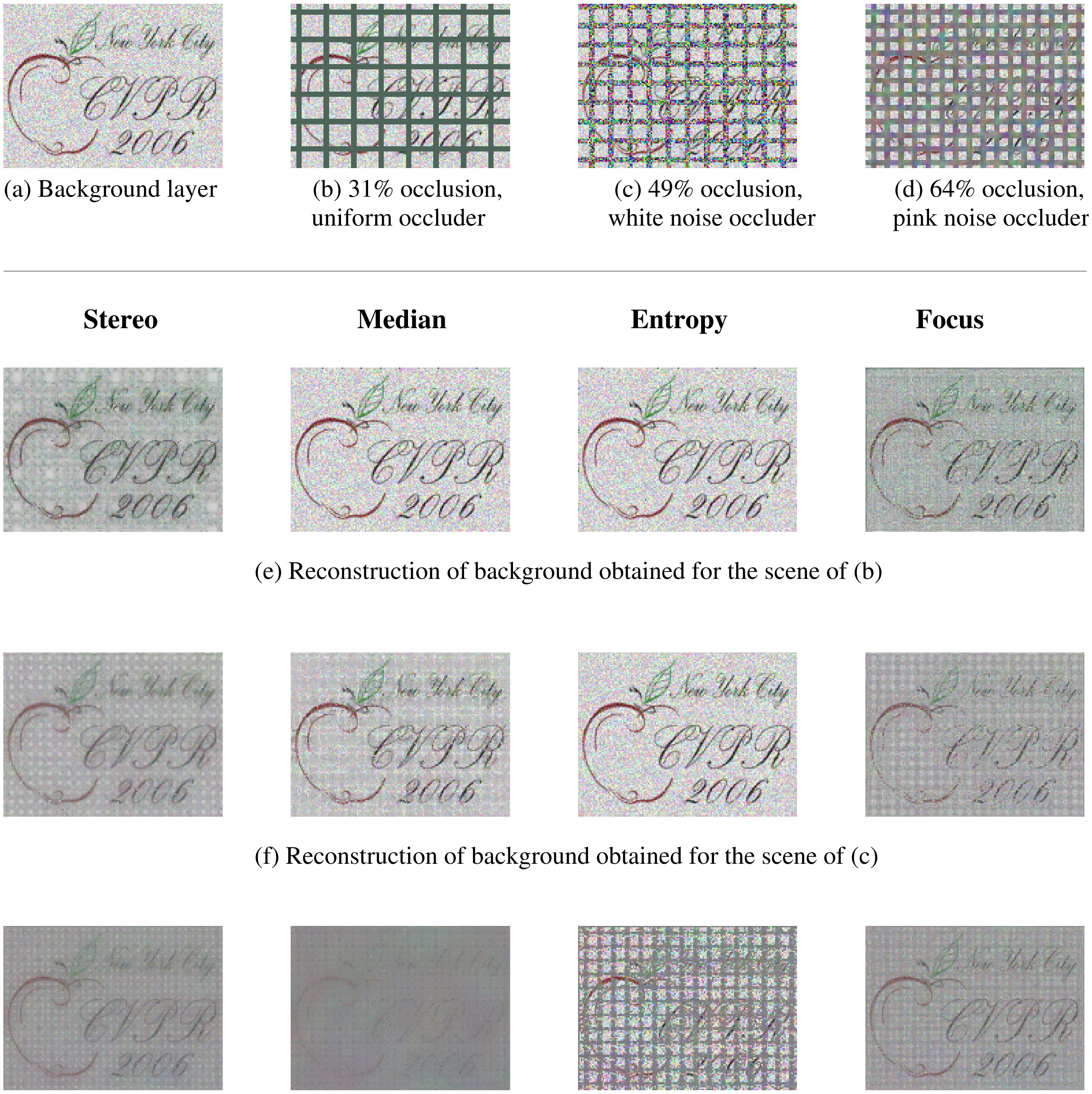
(For color images, we use entropy of a color histogram where bins correspond to cubes in RGB space). The entropy cost function penalizes rays that fall into different bins. Unlike stereo, the penalty *does not increase with the distance between different ray colors.* This helps mitigate the influence of outliers on depth reconstruction.

There are two interesting properties of shape from entropy we would like to mention. First, given N rays, the entropy can vary in increments of $\frac{1}{N}$ from 0 to $\log N$. Thus, the precision of entropy is $O(\frac{1}{N})$, making it more precise as the number of views increases. Secondly, since we are binning the rays, entropy does not use the lower bits of the intensity values (for $K = 16$ bins, we use only 4 bits per channel to compute the bin index for a pixel). This could be exploited for saving storage or bandwidth, and also makes entropy less sensitive to color miscalibration.

Both entropy and shape from median retain the desirable property of shape from stereo of being sensitive to first-order texture derivatives. Both of these start to fail when the occluder is of a fairly uniform color and the amount of occlusion exceeds 50% — in this case, the reconstructed color is the color of the occluder rather than the background surface.

5.4 Experimental Results

We have compared the performance of the four cost functions—stereo (variance), focus, median, and entropy on reconstructing occluded objects in two acquired light fields of complicated scenes and one synthetic scene. As we are avoiding explicit reconstruction of the occluders, we limit our search to a range of depths *behind* the occluders. The accuracy of the depth maps computed by each method is compared with ground truth. To visualize the reconstructed appearance of the occluded surfaces, we construct synthetic aperture images focused on the reconstructed surfaces. In shape from focus and stereo, the mean color of all the rays through a point on the estimated surface is used as its color. This corresponds to using the entire synthetic aperture. In shape from median, we use the median color rather than the mean. For entropy, we use the modal color from the color histogram. This corresponds to a per-pixel aperture consisting of the cameras whose rays fall in the tallest bin of the color histogram for that pixel.



(a) Background layer

(b) 31% occlusion, uniform occluder

(c) 49% occlusion, white noise occluder

(d) 64% occlusion, pink noise occluder

(e) Reconstruction of background obtained for the scene of (b)

(f) Reconstruction of background obtained for the scene of (c)

(g) Reconstruction of background obtained for the scene of (d)

Figure 5.4: Experiment 1: Reconstructions obtained with synthetic scene. (a) A textured plane which we seek to reconstruct. (b)-(d) Some of the occluders used. The amount of occlusion was varied from 19% to 75% by changing the spacing between the bars. (e)-(g) The reconstructions of the background plane obtained for the scenes in (b)-(d) using stereo, median, entropy and focus.

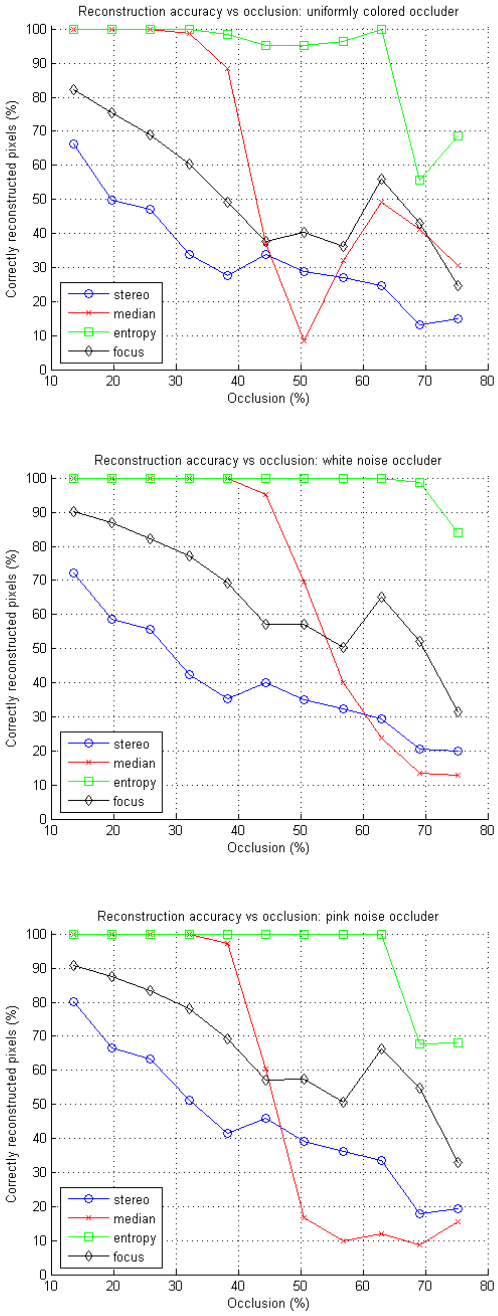


Figure 5.5: Experiment 1: Reconstruction accuracy for the synthetic scene of Fig. (5.4). The plots show the percentage of pixels on the background reconstructed within one disparity level of ground truth versus the amount of occlusion for uniformly colored occluders (top), white noise occluders (middle) and pink noise occluders (bottom).

5.4.1 Experiment 1: Synthetic scene

To measure the performance of our cost functions (stereo, focus, median, entropy) under varying amounts of occlusion, we experimented with a synthetic scene with precisely controlled amounts of occlusion. The scene consists of two planes. The background plane (Fig. 5.4a) has the CVPR logo composited with white noise to ensure there is sufficient texture at each point. The foreground occluder is composed of horizontal and vertical bars. We control the amount of occlusion (between 19% and 75%) by varying the spacing between the bars. We have experimented with three different textures on the foreground bars: white noise (strong texture), pink noise (weak texture, obtained by convolving white noise with a 5×5 box filter) and uniformly colored bars (Fig. 5.4 b-d).

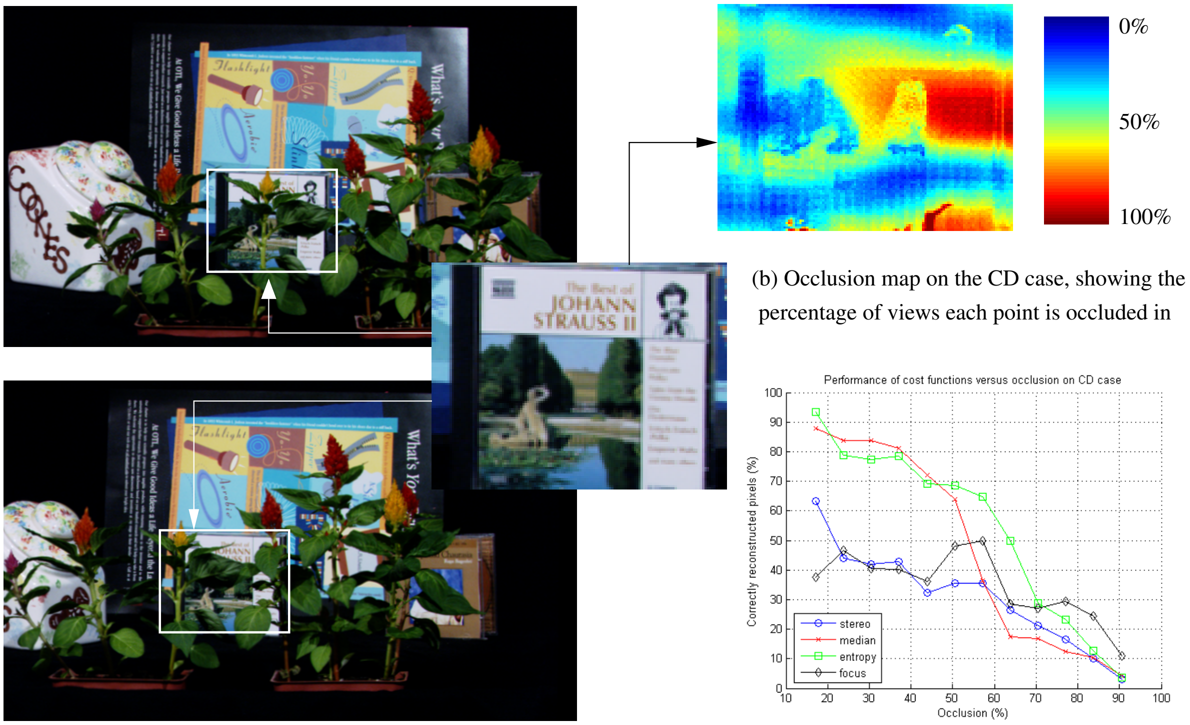
For reconstruction, we used 81 views of this scene. The camera positions were on a 9×9 planar grid perturbed by random offsets. This is to avoid producing a strong correlation between visibility and camera positions, which rarely arises in practice and would introduce addition errors in reconstruction, contaminating our simulation. The two planes were separated by a disparity of 40 pixels across the synthetic aperture. We measured how accurately each of the four cost functions reconstructed the background plane.

Over the range of occlusion densities and foreground textures in our experiments, entropy does best, with near-perfect reconstructions up to 65% occlusion. Shape from median starts to fail as the occlusion density crosses 50%. For all three foreground textures, the fraction of background points correctly reconstructed by focus is about 15% higher than that for regular stereo. This suggests that given an adequately textured surface, focus does better than variance in the presence of occlusions. Note that real scenes could have surface textures that stereo can reconstruct but not focus; in which case stereo would obviously do better. Plots of reconstruction accuracy versus amount of occlusion for the three occluder textures are shown in Fig. (5.5)

5.4.2 Experiment 2: CD case behind plants

This light field was acquired using a single camera on a computer controlled gantry. The camera positions span a 21 × 5 grid (synthetic aperture size 60cm by 10cm). Our goal was to reconstruct a picture of the Strauss CD case behind the plants (Fig. 5.6 a). For ground truth, the depth of the CD case was estimated manually. To estimate the amount of occlusion for the CD case, we captured an identical light field of the same scene without the occluding plants and used image differencing to determine which pixels were occluded. We can thus compute an occlusion map image, where each pixel in the occlusion map stores the number of views in which the corresponding pixel on the CD is occluded (Fig. 5.6 b). Given ground truth and an occlusion value, we determine for each of the four cost functions the percentage of pixels reconstructed correctly (within one disparity level) for different amounts of occlusion (Fig. 5.6 c). The histogram indicates that stereo, median, and entropy start to perform poorly as the amount of occlusion increases, with entropy performing best. The median-based cost function starts to fail once the amount of occlusion exceeds 50%. Depth from focus does not do well on the whole, but it overtakes the stereo-based approaches as occlusion increases beyond 60%.

The synthetic aperture images from each cost function are shown in the top row of Fig. (5.7). The text on the CD case is most legible for entropy and median. The bottom row shows the reconstruction of the CD obtained by voxel coloring after reconstructing and deleting the occluding plants. Voxel coloring uses a threshold to commit to foreground occluders so that pixels corresponding to the occluder are deleted prior to reconstructing the background. Note that there is no single threshold for which the CD is reconstructed properly. At low thresholds, all foreground pixels are not deleted; at higher thresholds, many background pixels are reconstructed as foreground and deleted.



(b) Occlusion map on the CD case, showing the percentage of views each point is occluded in

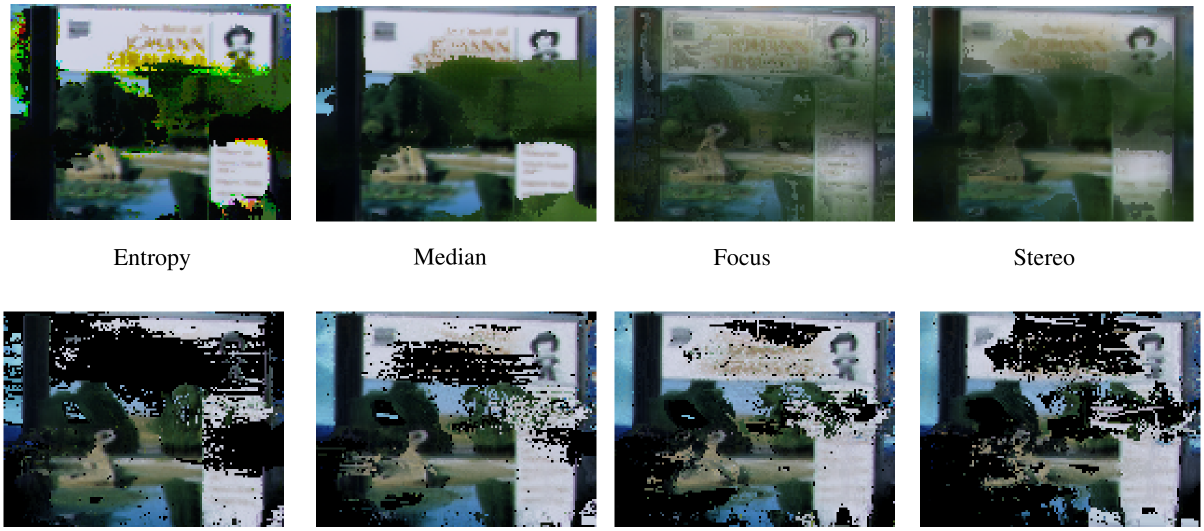
(a) Two images from the light field. We wish to reconstruct the Strauss CD case behind the plants (inset)

(c) Performance of the four cost functions on the the Strauss CD case.

Figure 5.6: Experiment 2: Comparison of cost functions for reconstruction of the CD case behind the plants using a 105 image light field.

5.4.3 Experiment 3: Dense Foliage

This light field (Fig. 5.8a) was captured using an array of 88 cameras. The scene consists of a person and a statue behind a dense ivy wall. We computed a per-camera matte for the occluder using standard blue screen techniques. While we do not have ground truth for this scene, we can estimate visibility near the true depths using the occluder mattes. The objects behind the ivy are occluded in about 70% of the cameras. Entropy, median and voxel coloring all failed to reconstruct any part of the occluded objects accurately. This is due to the high degree of occlusion, the relatively uniform color of the occluder and differences amongst that cameras' colorimetric response which persisted after color calibration. Focus does somewhat better than stereo — it is able to reconstruct the statue's torch and the person's



Entropy Median Focus Stereo

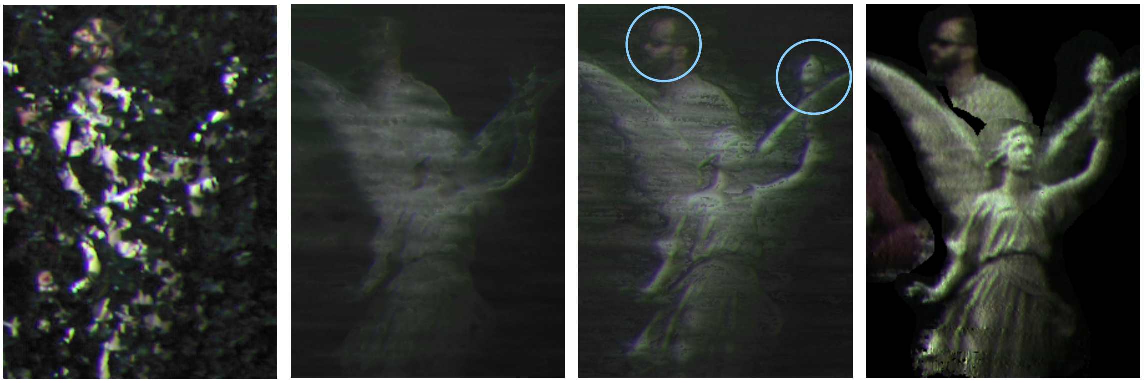
Voxel coloring (threshold increases left to right)

Figure 5.7: Experiment 2: Reconstructing the Strauss CD case behind plants (Fig. 5.6). The top row shows the reconstructed surface and color using entropy, median, focus and stereo. The bottom row shows the reconstruction using voxel coloring: these images are obtained by deleting all voxels up to the depth of the foreground occluders, projecting the rest into the central camera and computing the mean color of the projected voxels. For low thresholds, some points on the CD surface are not reconstructed at all. At high thresholds, the computed depth for points on the CD case can be much closer to the cameras than the true depth, causing them to be deleted with the occluders. These two effects cause the holes visible in the reconstruction.

face where stereo failed (Fig. 5.8 b,c).

5.5 Discussion

We have studied four cost functions for reconstructing occluded surfaces using synthetic apertures. Most existing algorithms use cost functions developed for reconstructing the (unoccluded) foreground. We have studied an alternative approach: designing cost functions that leverage a large synthetic aperture and large number of views to be robust to occlusions. This is useful when the foreground is difficult to



(a) Image from one camera (b) Stereo reconstruction (c) Focus reconstruction (d) Unoccluded objects

Figure 5.8: Experiment 3: Reconstruction of a person and statue behind a dense wall of artificial ivy. The high occlusion (about 70%) and uniform color of the occluder makes this a failure case for median and entropy. Focus does somewhat better than stereo on the right part of the statue and the person's face (shown in blue circles). Image (d) was created by manually compositing layers after deleting the occluded pixels using blue screening.

reconstruct accurately (see, for example, Fig. (5.8) and in applications like surveillance of crowded areas where reconstructing the appearance of occluded objects may be of primary interest.

While the cost functions we have developed show encouraging results, we believe we have merely scratched the surface of this design space. We would like to explore ways to combine information from different cost functions to gain better results. Our approach could also be combined with existing multi-layer estimation algorithms to improve reconstruction of *all* layers. One limitation of our work is that we have not enforced any spatial regularization. We would like to find ways of extending algorithms based on graph cuts or belief propagation to handle outliers due to dense occlusion. While our focus in this work has been on diffuse surfaces, we believe robust cost functions like median and entropy may also be useful for the reconstruction of non-Lambertian surfaces since specular highlights, like occlusions, may be regarded as outliers.

Chapter 6

Conclusion

In this thesis, we have studied the geometry underlying synthetic aperture image formation. We have demonstrated the use of our geometric analysis for calibration of camera arrays, real-time synthetic aperture focusing and 3D reconstruction of partially occluded surfaces. This chapter reviews these contributions, discusses some limitations and potential strategies for overcoming them, and some directions for future work.

6.1 Calibration

We have characterized the minimum calibration required to compute a synthetic aperture image focused on an arbitrary plane:

1. Homographies for projecting camera images on to a reference plane.
2. The location of epipoles (in a reference camera).
3. The μ-vector.

This requires observation of at least of four points on a reference plane, and at least two points not on the reference plane in all cameras [12]. When the camera centers are known to lie in a plane (which is often the case), it suffices to observe a *single point* not on the reference plane. This configuration is useful since we can solve for the calibration parameters (specifically, the epipoles) using linear optimization, which does not require any initial guess.

A key assumption we have made is that the cameras are stationary and synchronized. One can think of situations where this may not be true: for example, the cameras may be mounted on different vehicles moving independently or even on human soldiers. In such cases, calibration becomes significantly more challenging. Some simplification may be achieved by having GPS or inertial sensors attached to the cameras to obtain their relative poses. However it is not possible to achieve pixel accurate alignment in this way which means one would have to resort to general structure-from-motion methods. A promising approach for this would be the system developed by Snavely et al [36] which performs incremental structure from motion using robustly matched SIFT features.

6.2 Real-time Focusing

We have derived a classification of camera arrangements and focal planes corresponding to different families of image warps required for focusing. The key idea to project the camera images on to a fixed reference plane: this computationally demanding step needs to be done only once. We have characterized the configurations where we can move the focus from the reference plane to other focal planes via affine or simpler warps (Table 4.2). Of particular interest are the configurations where changing the focal plane requires simply shifting the images (Fig. 3.4c,d). Since shifts are relatively easy to implement in per-camera hardware, this facilitates a real-time focusing system for frontoparallel focal planes and focal planes in the Scheimpflug configuration.

While our camera array hardware may not be able support focusing on arbitrary planes and surfaces in real-time, one can conceive of other camera array architectures that can. One approach would be to leverage the rendering capabilities of modern GPUs, which have speeds of several gigaflops and can hold entire light fields in their texture memory. Computing synthetic aperture images on GPUs is straightforward: one simply needs to texture map the camera frames on to the desired focal surface and accumulate the rendered images in the A-buffer. Since synthetic aperture imaging benefits from large numbers of cameras, it is important to maintain scalability and minimize the cost per camera. The key challenges in using a GPU-based architecture would include maximizing the bandwidth between the cameras and GPU texture memory and supporting as many cameras per GPU as possible.

6.3 3D Reconstruction

We have shown how to improve the 3D reconstruction of partially occluded surfaces given a large, densely sampled synthetic aperture. Instead of using classical shape from stereo or shape from synthetic aperture focus, we have show we can achieve better reconstruction using color medians and modal colors. Unlike focus and stereo, which are based on the mean and variance respectively, the median and mode are more tolerant of outliers, such as those caused due to occlusion. Experiments indicate these robust approaches can achieve accurate 3D reconstruction of surfaces occluded in over 50% of the cameras. This is facilitated by a synthetic aperture wider than the occluders in the scene which enables seeing around the occluders, and a large number of cameras to sample enough rays that are incident on the partially occluded surface to be reconstructed.

The performance of these robust methods suggests a different approach to matching rays from different cameras for 3D reconstruction. Most existing methods seek to either match rays across *all* views - assuming no occlusions or specularities [31], or explicitly reconstructing occluders and deleting them [34, 24]. The former are restricted to a limited class of scenes, the latter make hard decisions about reconstruction of occluders, and are sensitive to errors in the initial reconstruction. Our work suggests it that given rays from a large number of cameras (typically ≥ 100), it suffices to match a subset of them while treating the remaining rays as outliers. This yields accurate results, avoids making hard decisions (like deleting occluders) and does not require explicit modeling of view-dependent effects like occlusions and non-Lambertian surfaces.

One limitation of our approach is the use of a single depth map to represent the scene we wish to reconstruct. While a single depth map may be an adequate to represent a focal surface for synthetic aperture focusing, there are many surfaces it cannot represent completely. An alternative approach would be to compute a depth map for every view (or a subset of views) in the manner of [20] while continuing to use robust methods to tolerate occlusions. These depth maps could be merged volumetrically to yield a complete 3D model [10]. In contrast to our approach which leverages *all* available views to overcome occlusions, they construct a depth map for every view using normalized cross-correlation across a small number (typically 5) views. Confidence weighted volumetric merging of these depth maps yield accurate and near-complete 3D models. These are far superior to our single-depth-map as a representation of 3D shape, but are time-consuming to compute (computation time varies from hours to days) and incur some loss of accuracy near depth edges due to window-based correlation. An appealing approach would be to design a hybrid of their approach and ours, using the robust cost functions to minimize errors due to windowing and reduce computation time by pruning the search space.

One could also try to improve the quality of our reconstructions by applying spatial regularization. In our implementation, we have reconstructed the depth of each pixel independently of its neighbors. Barring surface discontinuities, the depths nearby pixels are similar. One would also expect a strong correlation between the cameras deemed outliers for two neighboring pixels: if surface point is occluded in one camera it is likely that nearby points are also occluded in that camera. Exploiting this spatial coherence in shape and outliers - perhaps using Markov random fields - should improve the accuracy of reconstruction, especially in regions of low texture.

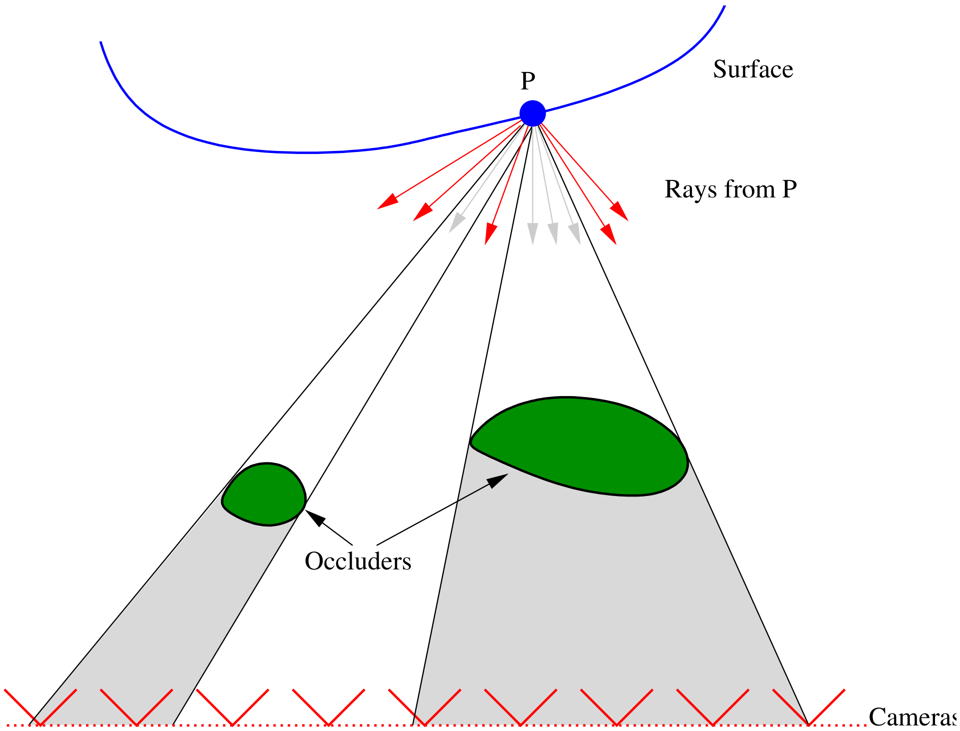


Figure 6.1: Visibility analysis for a surface point P behind occluders being observed by a camera array. Larger occluders shadow out larger chunks of the synthetic aperture. The visibility of P is the fraction of rays from P which avoid the occluders, indicated in red. When the occluders are large, there is a strong correlation between the visibility of adjacent rays.

6.4 Future Work

We would like to conclude by describing two important directions for future research: analyzing the relation between camera layout, scene occluders and visibility, and augmenting camera arrays with arrays of projectors.

6.4.1 Visibility Analysis and Camera Arrangements

Our ability to reconstruct an image of (or reconstruct the shape of) a partially occluded surface patch using synthetic aperture imaging depends on the fraction of cameras in which it is visible. A large aperture enables seeing around large occluders. A large number of cameras enables sampling a larger number of rays which avoid the occluders and are incident on the surface we wish to see or reconstruct. For the designer of a synthetic aperture imaging system who wishes to maximize the visibility surfaces under a wide variety of occluders, understanding the the relationship between surface visibility, aperture size, number of cameras and their layout and distribution of occluder sizes is imperative.

An obvious constraint on the aperture size is that is must be larger than any of the occluders in the scene. The size of the aperture also determines the minimum distance between an occluder of a given size and the surface points which we seek to observe. Fig. (6.1) shows the shadow cast by an occluder on the synthetic aperture. If the distance between P and the occluders is reduced, the fraction of the cameras in shadow will increase. Thus, the aperture size places an upper bound on the size of occluders in the scene and a lower bound on the separation between the surface to be seen and the occluders.

It would be useful to have bounds on the number to cameras required to observe a surface point with some minimum probability. This turns out to be non-trivial for most occluders. Consider the set of rays from a surface point P as shown in Fig. (6.1). Suppose we know (from the average occlusion density) that the fraction of rays starting from P which pass through the occluders is α. Can we compute the minimum number of cameras N required so that the probability P will be visible in at least one view is greater than a given threshold p ? It is tempting to assume the visibility of a ray starting from p is independent of the other rays, in which case the ray visibilities have a binomial distribution and we can obtain a bound on the minimum number of samples (i.e. cameras) required as a function of p and α. However, this assumption of independence is incorrect for most occluders; as shown in Fig. (6.1), there is a strong correlation between nearby rays owing to the size of the occluder. Analyzing the relation between number of cameras and visibility for different classes of occluders as a function of occluder density and occluder size distribution remains an open question for the design of synthetic aperture imaging systems.

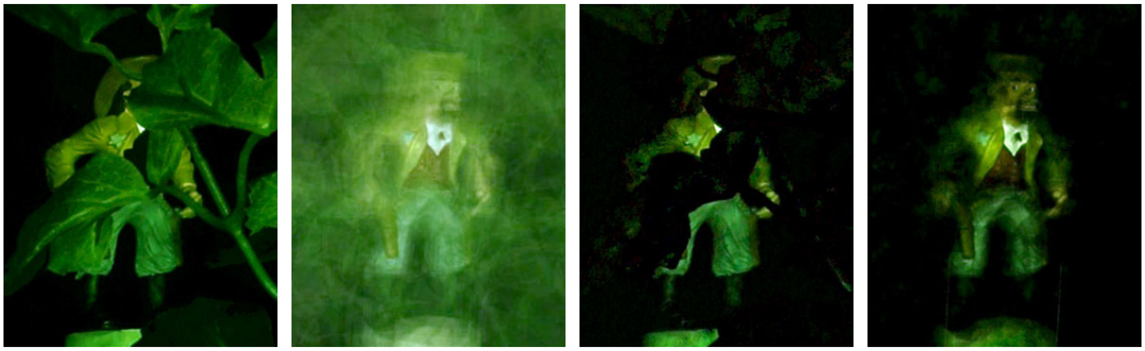

(a) Single view (b) Synthetic aperture image (c) Confocal image (d) Synthetic aperture confocal image

Figure 6.2: Synthetic aperture confocal imaging using an array of 16 (virtual) cameras cameras and projectors [25]. (a) A toy soldier behind foliage. (b) A synthetic aperture image focused on the toy soldier in which the foliage is blurred out. (c) A confocal image in which illumination is synthetically focused on the toy soldier. There is almost no light on the the foliage, leaving it dark. (d) A synthetic aperture confocal image obtained by combining (b) and (c). The occluding foliage is both dark (due to confocal illumination) and blurry (due to synthetic aperture focusing), resulting in better contrast and clarity.

6.4.2 Synthetic Aperture Illumination

We can augment an array of cameras with an array of projectors. A projector is a dual of a camera; just as we can use an array of cameras for synthetic aperture imaging we can use an array of projectors for *synthetic aperture illumination*. A suitably pre-warped image is displayed from each projector so that all the projected images come into alignment on a chosen focal plane (analogous to the focal plane in synthetic aperture imaging). The projectors effectively behave like a single projector with a large synthetic aperture and consequently shallow depth of field.

Using this synthetic aperture projector, we can project any patterned illumination - such as high frequency texture - into the scene such that the illumination is in focus at the desired depth and blurred (low frequency) everywhere else. Just as a large synthetic aperture of a camera array enables us to see objects behind dense occlusion, a large synthetic aperture spanned by a projector array enables us to *project patterned illumination* on to objects behind dense occlusions. Levoy et al have shown that by switching off projector pixels which correspond to rays that are incident on the occluder (i.e. shaping the aperture per pixel), we can illuminate partially occluded objects without lighting up the occluder [25]. This can greatly increase the contrast of synthetic aperture images; we show one of their results in Fig. (6.2). They also showed how synthetic aperture illumination may be used for 3D surface reconstruction where the large synthetic aperture makes the reconstruction more tolerant to occlusion than most structured light systems.

To compute shaped apertures, Levoy et al adapted techniques from confocal microscopy [49]. The resulting method is rather slow, in that it requires a large number of coded patterns to be projected on the scene and imaged. In particular, the number of coded patterns required to get provably correct results is of the order of the number of pixels in the projector. Since these codes were developed in confocal microscopy for single lens aperture, they do not exploit the fact that we multiple cameras imaging the scene. Designing coded patterns which can leverage images from multiple views to help resolve decoding ambiguities should lead to greater efficiency and shorter acquisition times.

A second, more basic challenge is to build projector arrays that match the total resolution and apertures achievable with camera arrays. Levoy et al used a single camera and projector aimed at planar mirrors in their experiments, trading off spatial resolution (of the images) for angular resolution (number of cameras and projectors). Projectors, like cameras, are decreasing in cost and size, while improving in their energy efficiency, making it feasible to build large arrays of digital projectors. Designing and building large projector arrays to augment existing camera arrays will provide a platform for exciting research in synthetic aperture imaging and synthetic aperture illumination.

Appendix A

Error Analysis for Non-planar Cameras

For many of our results, such as parallax-based calibration and real-time focusing we have assumed the camera centers lie on a plane. The cameras are mounted on a rigid, planar framework so that the camera centers do not deviate from a plane by more than 5cm for the telephoto lenses used in outdoor imaging and more than 1cm for the lenses used indoors in our real-time system. We will now quantify the error incurred in synthetic aperture focusing assuming planar camera centers and verify under what conditions this is a reasonable assumption.

Consider the scenario depicted in Fig. (A.1). This shows the reference camera C_0, a "camera plane" on which all camera centers are assumed to lie. C_1 is a camera which is actually some distance ΔZ from the camera plane. Suppose we wish to focus at the depth of some point P. To do so, we will need to shift the image from camera C_1 by an amount equal to the parallax of P, which is $\Delta p_1 + \Delta p$. If we assume camera C_1 is located on the camera plane, we will compute the parallax to be Δp_1. This error in parallax will manifest itself as a misalignment of Δp when focusing on P. We will now compute p in terms of the camera layout and focal distances and show when it is small enough to justify the assumption of planar cameras.

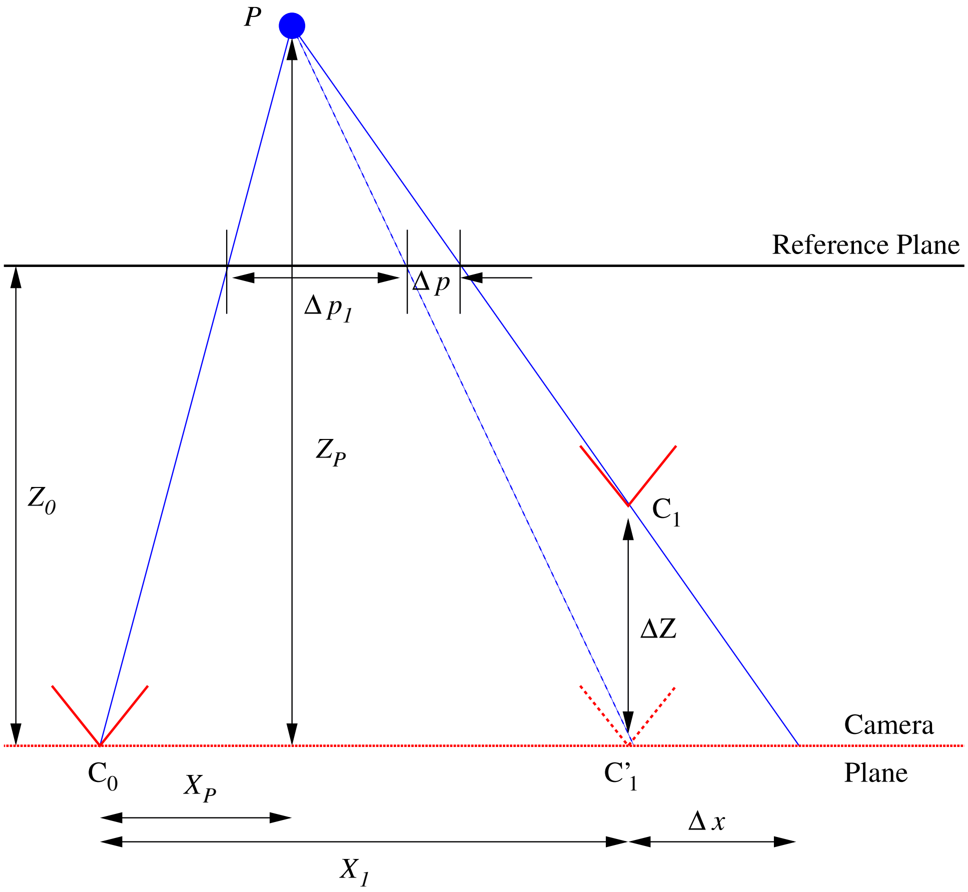


Figure A.1: Quantifying the error incurred by assuming the camera centers lie on a plane. C_0 is the reference camera, C_1 is another camera which is at a distance ΔZ from the assumed plane of cameras shown in red. If we assume camera C_1 is actually located at position C_1', the parallax of P will be Δp_1. The error in parallax will cause a misalignment of Δp when focusing on P.

From similar triangles, we have

$$\frac{\Delta x}{\Delta Z} = \frac{X_1 + \Delta x - X_P}{Z_P}$$

and

$$\frac{\Delta p}{\Delta x} = \frac{Z_P - Z_0}{Z_P}$$

Eliminating Δx and solving for Δp we obtain

$$\Delta p = \frac{\Delta z(X_1 - X_P)}{Z_P - \Delta Z}(1 - \frac{Z_0}{Z_P}) \tag{A.1}$$

In general, the misalignment Δp can be arbitrarily large, such as when $Z_P \approx 0$ or $Z_P \approx \Delta Z$. In most experimental conditions, including ours, we can show that the misalignment is smaller than a single pixel. We make two observations:

1. We focus on surfaces that are within the field of view of most cameras. In Fig. (A.1), we require P to be within the field of view of camera C_1. If the angular field of view is 2θ, this yields:

$$|X_P - X_1| \leq (Z_P - \Delta z) \tan\theta \tag{A.2}$$

2. We focus at depths of the order of Z_0 - at depths distant from Z_0, there is little or no overlap amongst the fields of view of different cameras. In our experiments, we have

$$\frac{2Z_0}{3} \leq Z_P \leq 2Z_0$$

which yields

$$|1 - \frac{Z_0}{Z_P}| \leq \frac{1}{2} \tag{A.3}$$

We are now ready to obtain a bound on Δp.

$$\begin{aligned} |\Delta p| &= |\frac{\Delta z(X_1 - X_P)}{Z_P - \Delta Z}(1 - \frac{Z_0}{Z_P})| && \text{from Eq. (A.1)} \\ &\leq |\frac{\Delta z}{Z_P - \Delta Z}|.(Z_P - \Delta z)\tan\theta.\frac{1}{2} && \text{from Eq. (A.2) and (A.3)} \\ &= \frac{\Delta z \tan\theta}{2} && \text{(A.4)} \end{aligned}$$

To convert this bound from metric units to pixels, suppose that our cameras have images R pixels wide. On the reference plane (i.e. at the depth Δp is defined), these R pixels are distributed over a length of $2Z_0 \tan\theta$ so we have $\frac{R}{2Z_0 \tan\theta}$ pixels per unit length. Thus, we have

$$|\Delta p| \leq |\frac{R\Delta z}{4Z_0}| \qquad \text{(A.5)}$$

For our outdoor imaging setup (Fig. 4.2), we have $R = 640$ pixels, $Z_0 = 28$m, $\Delta z \leq 5$cm which yields

$$\Delta p \leq 0.29 \text{ pixels}$$

For our real-time system used in the laboratory, we have $R = 320$ pixels, $Z_0 = 2$m, $\Delta z \leq 1$cm which yields

$$\Delta p \leq 0.4 \text{ pixels}$$

This amount of misalignment still permits us to obtain fairly sharp synthetic aperture images, as we have seen in our experiments.

Note that Eq. (A.5) is only an upper bound on the misalignment. The misalignment is greatest near the periphery of the field of view (where the bound from Eq. (A.2) is tight) and becomes smaller towards the center. In Fig. (A.1), if P were directly above C_1 so that $X_P = X_1$, the misalignment Δp would be zero. On the other hand, the misalignment increases with the sensor resolution and with the distance of the focal plane from the reference plane.

Bibliography

[1] G. Aloupis. On computing geometric estimators of location. Master's thesis, School of Computer Science, 2001. 56

[2] H. Baker, D. Tanguay, and C. Papadas. Multi-viewpoint uncompressed capture and mosaicking with a high-bandwidth pc camera array. In *Proceedings of OMNIVIS (6th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras)*, 2005. 10

[3] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 434–441, 1998. 51

[4] R. Collins. A space sweep approach to true multi-image matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 358–363, 1996. 26, 52

[5] H. S. M. Coxeter. *Projective Geometry.* Springer Verlag, 2003. 27

[6] J. de Bonet and P. Viola. Roxels: Responsibility weighted 3d volume reconstruction. In *International Conference on Computer Vision*, pages 418–425, 1999. 51

[7] P. Favaro and S. Soatto. Observing shape from defocused images. *International Journal of Computer Vision*, 52(1):25–43, 2003. 11, 54

[8] P. Favaro and S. Soatto. Seeing beyond occlusions (and other marvels of a finite lens aperture). In *IEEE Conference on Computer Vision and Pattern Recognition*, pages (2), 579–586, 2003. 11

[9] G. Garg, E. Talvala, M. Levoy, and H. Lensch. Symmetric photography: Exploiting data-sparseness in reflectance fields. In *Eurographics Symposium on Rendering*, pages 251–262, 2006. 2

[10] M. Goesele, S. Seitz, and B. Curless. Multi-view stereo revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2402 – 2409(2), 2006. 68

[11] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proceedings of ACM SIGGRAPH*, pages 43–54, 1996. 10

[12] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 12, 16, 20, 26, 32, 65

[13] S. Hasinoff and K. Kutulakos. Confocal stereo. In *European Conference on Computer Vision*, pages 620–634, 2006. 11

[14] Integrated systems development, s. a. http://www.isd.gr. 10

[15] M. Irani, P. Anandan, and M. Cohen. Direct recovery of planar parallax from multiple frames. In *Vision Algorithms: Theory and Practice*, pages 85–99. Springer Verlag, LNCS vol. 915, 1999. 12

[16] M. Irani, P. Anandan, and D. Weinshall. From reference frames to reference planes: Multi-view parallax geometry and applications. In *European Conference on Computer Vision*, pages 829–845, 1996. 12

[17] A. Isaksen, L. McMillan, and S. Gortler. Dynamically reparametrized light fields. In *Proceedings of ACM SIGGRAPH*, pages 297–306, 2000. 12

[18] T. Kanade. Carnegie mellon goes to the super bowl. http://www.ri.cmu.edu/events/sb35/tksuperbowl.html, 2001. 1, 10

[19] T. Kanade, H. Saito, and S. Vedula. The 3d room: Digitizing time-varying 3d events by synchronized multiple video streams. Technical Report CMU-RI-TR-98-34, Robotics Institute, Carnegie Mellon University, 1998. 1

[20] S. B. Kang and R. Szeliski. Extracting view-dependent depth maps from a collection of images. *International Journal of Computer Vision*, 58(2):139–163, 2004. 68

[21] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 103–110, 2001. 51

[22] R. Kingslake. *Optics in Photography*. SPIE, 1992. 4, 7

[23] E. Krotkov. Focusing. *International Journal of Computer Vision*, 1(3):223–237, 1987. 11, 14

[24] K. Kutulakos and S. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2002. 67

[25] M. Levoy, B. Chen, V. Vaish, M. Horowitz, I. McDowall, and M. Bolas. Synthetic aperture confocal imaging. In *Proceedings of ACM SIGGRAPH*, pages 825–834, 2004. 2, 8, 71, 72

[26] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of ACM SIGGRAPH*, pages 31–42, 1996. 10, 12, 26

[27] R. Magid. Technobabel. *American Cinematographer*, pages 46–55, April 1999. 1, 10

[28] K. Mileshosky. What is synthetic aperture radar ?, 2005. http://www.sandia.gov/RADAR/whatis.html. 12

[29] Y. Ng. Fourier slice photography. In *Proceedings of ACM SIGGRAPH*, pages 735–744, 2005. 8, 27

[30] Y. Ng, M. Levoy, M. Bredif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. Technical Report CSTR 2005-02, Computer Science Department, Stanford University, 2005. 2

[31] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(4):353–363, 1993. 14, 51, 53, 67

[32] Y. Schechner and N. Kiryati. Depth from defocus vs. stereo: How different really are they ? *International Journal of Computer Vision*, 39(2):141–162, 1993. 3, 12

[33] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comaprison and evalutaion of multi-view stereo reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 519–526, 2006. 13

[34] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1067–1073, 1997. 51, 67

[35] P. Sen, B. Chen, G. Garg, S. Marschner, M. Horowitz, M. Levoy, and H. Lensch. Dual photography. In *Proceedings of ACM SIGGRAPH*, pages 745 – 755, 2005. 2

[36] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *Proceedings of ACM SIGGRAPH*, pages 835–846, 2006. 66

[37] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32(1):45–61, 1999. 26, 51, 52

[38] B. Triggs. Plane + parallax, tensors and factorization. In *European Conference on Computer Vision*, pages 522–538, 2000. 20

[39] V. Vaish. Light field camera calibration. http://graphics.stanford.edu/projects/array/calibration/, 2003. 37

[40] V. Vaish. The stanford calibration grid detector. http://graphics.stanford.edu/software/findgrid/, 2006. 32

[41] V. Vaish, G. Garg, E. Talvala, E. Antunez, B. Wilburn, M. Horowitz, and M. Levoy. Synthetic aperture focusing using a shear-warp factorization of the viewing transform. In *Proc. of Workshop on Advanced 3D Imaging for Safety and Security, in conjunction with IEEE Conference on Computer Vision and Patern Recognition*, 2005. 1, 8, 32

[42] V. Vaish, R. Szeliski, C. L. Zitnick, S. B. Kang, and M. Levoy. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust methods. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages (3),2331–2338, 2006. 8

[43] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy. Using plane + parallax to calibrate dense camera arrays. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages (1) 2–9, 2004. 1, 7, 31

[44] J. Wang and E. Adelson. Layered representation for motion analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–366, 1993. 51

[45] Y. Wexler, A. Fitzgibbon, and A. Zisserman. Bayesian estimation of layers from multiple images. In *European Conference on Computer Vision*, pages (3), 487–501, 2002. 55

[46] R. Wheeler. Notes on view camera geometry. http://www.bobwheeler.com/photo/ViewCam.pdf, 2003. 25

[47] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. High speed video using a dense camera array. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages (2) 294–301, 2004. 10

[48] B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Levoy, and M. Horowitz. High performance imaging using large camera

arrays. In *Proceedings of ACM SIGGRAPH*, pages 765–776, 2005. 1, 2, 8, 10, 11, 26, 42

[49] T. Wilson, R. Juskaitis, M. Neil, and M. Kozubek. Confocal microscopy by aperture correlation. *Optics Letters*, 21(3):1879–81, 1996. 72

[50] J. C. Yang, M. Everett, C. Buehler, and L. McMillan. A real-time distributed light field camera. In *Eurographics Workshop on Rendering*, pages 77–86, 2002. 1

[51] L. Zelnik-Manor and M. Irani. Multi-frame alignment of planes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages (1), 149–156, 1999. 13

[52] L. Zelnik-Manor and M. Irani. Multi-view subspace constraints on homographies. In *International Conference on Computer Vision*, pages (2), 710–715, 1999. 13

[53] C. Zhang and T. Chen. A self-reconfigurable camera array. In *Eurographics Symposium on Rendering*, 2004. 1

[54] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(11):1330–1334, 2000. 37, 40

[55] C. L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *Proceedings of ACM SIGGRAPH*, pages 600–608, 2004. 1, 10, 51