INTERACTIVE HAND-HELD LIGHT FIELD CAPTURE

AN HONORS THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
OF STANFORD UNIVERSITY

Myers Abraham Davis
Principal Adviser: Marc Levoy
May 2010

Abstract

Regular images and video often struggle to convey a compelling sense of shape and scale in photographed scenes. Light fields, which are created from large collections of images of a scene, offer a promising alternative by letting users interactively navigate a photographed environment as they would any other virtual space. However, collections of photos that result in high quality light fields are very difficult to capture without specialized hardware. As a result, light field authorship has traditionally been restricted to experts in Image Based Rendering (IBR). This thesis presents the design and implementation of an interactive system for capturing light fields that uses an off the shelf webcamera connected to a laptop computer. The system consists of a process for capture as well as a method for viewing captured light field data that is immediately available for review at any time during the system's use. This system makes it possible for users to create higher quality light fields in a fraction of the time taken by previous methods that do not rely on specialized hardware. This creates a compelling new way for a larger community of users to communicate by capturing and sharing interactive representations of their environment.

Acknowledgements

First and foremost, I would like to thank Marc Levoy. Marc's guidance and mentorship as my adviser for the past two years has had an immeasurable impact on the work that I have done. It was also Marc who first came up with the idea of a "4D Fax Machine," which motivated most of the work described in this thesis. I would also like to thank all of Marc's students - Andrew Adams, Eddy Talvala, Zhengyun Zhang, Jongmin Baek, David Jacobs, Jennifer Dolson, and Sung Hee Park. They were always willing to explain new concepts, listen to ideas, and give advice when asked, and are responsible for teaching me many things that made this work possible.

I would like to thank Noah Snavely who has been a tremendous help over the past year in both writing this thesis and developing the ideas it describes. I would like to thank Fredo Durand for his feedback and advice on this thesis, and I would also like to thank Pat Hanrahan, Mark Horowitz, Kari Pulli, and Natasha Gelfand, whose input on my work in Marc's lab has been invaluable.

Finally, I would like to thank all of my family and friends whose patience and support has kept me sane through the countless presentations and deadlines related to this work.

Contents

Chapter 1

Introduction

1.1 Motivation

Regular images and video often struggle to convey a compelling sense of structure and scale. Image Based Rendering (IBR) offers a promising alternative. The ability to generate synthetic images of a scene makes it possible to interactively navigate that scene as one would any other virtual space. This lets users observe parallax as the result of deliberate camera motion and creates a sense of presence in a virtual environment.

The central task of IBR is novel view synthesis. By controlling a virtual camera a user indicates perspectives from which they wish to view a scene. In general, the set of perspectives from which a user might want to view a scene is much larger than the set of images that have been captured of that scene. As a result, novel views of the scene must be synthesized to meet the requests of the user. The task of photographing a subject and then resampling captured photographs to generate new perspectives of the subject can be interpreted as sampling and reconstruction in a particular domain - the light field.

In order to generate samples of a light field, one captures images of a subject they wish to later view. There are many challenges that make it difficult to capture a good distribution of samples in a light field. These challenges have traditionally served to limit the authorship of light field data to experts. The focus of this thesis is to create a way to author light fields that is accessible to a much larger community of users. The system presented here lets users capture light fields with common off the shelf hardware. Specifically, it assists a user in capturing orbital light fields with an off the shelf webcam connected to a laptop computer. These orbital light fields are light fields that focus on some central subject of interest in a scene. Such light fields let users interactively examine a subject of capture through a virtual camera (see figure 1.1 middle) or create images with a shallow synthetic depth of field which can be refocused on different parts of the scene (see figure 1.1 bottom). By making it easy for non-experts to create this kind of light field, the system presented here will hopefully make this kind of interactive photorealistic media accessible to a much larger community of users.

1.2 Related Work

There is a rich history of research on light field capture and IBR. This section gives an overview of just some of that work with a stronger focus on light field capture, which is most relevant to this thesis.

The conceptual groundwork of sampling and reconstruction in the light field domain comes from the Light Field work of Levoy and Hanrahan 1996 [5] and the Lumigraph work of Gortler et al. 1996 [3]. These works introduced the idea of resampling individual pixels, which represent rays, to reconstruct an image. Both works produced high quality reconstructions, but used methods for light field capture that restricted authorship to lab settings. Levoy and Hanrahan used a robotic arm to capture images, which let them define precisely what perspectives of the scene they would sample. Gortler et al used a handheld camera to capture objects placed on a calibrated stage covered with fiduciary markers. Again, the need for this stage restricted authorship to lab settings.

The Unstructured Lumigraph of Buehler et al. 2001 [1] combined ideas from the Light Field and Lumigraph work with the work of Debevec et al 1998 [2] on view dependent texture mapping (VDTM). This work presented a technique for IBR that could use relatively arbitrary distributions of perspectives in the scene. And, in contrast to the Light Field and Lumigraph work, images did not have to be resampled into a structured grid prior to rendering. In their paper, [1] demonstrated their technique on the images from video captured with a handheld camera. They noted, however, that without available scene geometry their hard-held input resulted in reconstructions that exhibited parallax only along the one-dimensional trajectory defined by a camera's path.

Wilburn et al 2005 [9] observed that by arranging several cameras along the kind of path taken by a robotic arm used by Levoy and Hanrahan, they could capture a high quality light field video. Applying this concept to a planar arrangement of cameras they created the camera array. Ng 2005 [6] applied a similar concept to an individual camera at the sensor level. This work resulted in the creation of the plenoptic camera, which allows refocusing and limited novel view synthesis from a single photograph. The plenoptic camera is particularly interesting in that it is a mobile way to capture light fields. However, the light field captured by a plenoptic camera is restricted to the camera's aperture. This can be effective for simulating different intrinsic camera parameters, but it restricts the users ability to move about the reconstructed scene to the camera's aperture. This much movement is almost imperceptible for anything but very small objects and scenes.

Probably the most accessible example of prior work in IBR comes from the work of Snavely et al 2008 [7] and Snavely et al 2006 [8] on photo tourism (later made into the product Photosynth). The photo tourism work focuses on how to reconstruct scenes from large photo collections like those available on the Internet. The basic approach is to use SfM techniques to register large numbers of photos against one another and then to provide users with an intuitive way to navigate these photos spatially. This approach is extremely effective when a very large collection of photos is available for a scene. In a sense the resulting system can be interpreted as a tool for community authorship of light field data. However, this system was not designed with authorship by an individual user as its main goal. As a result, it can make generating a compelling data set difficult and time consuming for an individual user.

1.3 Goals and Challenges

This section offers an overview of some goals that were set for the system and the challenges that faced meeting these goals.

1.3.1 Goals

Here is a list of goals that were set for the system. Progress made toward each of these goals constitutes a key contribution of the work described in this thesis:

- *Hardware accessibility* - Many approaches to light field authorship use specialized hardware to address some of the challenges associated with capture. In contrast, we would like a system that uses only common off the shelf hardware. This restricts us to the use of a handheld camera for input and prohibits us from depending on a calibrated capture environment (such as the stage used in [3]).
- *Speed* - Speed is a critical part of usability for any approach to light field capture. There are many measures of speed to consider in evaluating a technique for light field capture. These include the amount of time it takes for a user to physically capture images, the amount of time it takes to process images and create an interactive light field visualization, and the total amount of time it takes for a system to let a user capture and view a light field. We would like for each of these measures to be as quick as possible.
- *Coverage Quality* - Most techniques for IBR that use data captured with an individual handheld camera can only consistently produce results that exhibit parallax in one dimension. In contrast, techniques that use specialized hardware for capture are often able to reproduce parallax in multiple dimensions. We would like for a system to capture light fields that consistently reproduce parallax in multiple dimensions.

1.3.2 Challenges

There are many challenges that make light field authorship hard. Here those challenges are divided into those concerned with image registration and those concerned with obtaining good coverage of a light field.

Image Registration

Image registration is what limits our ability to add individual images to a light field. The hardest part of image registration is pose estimation. Without being able to estimate camera pose, we cannot include an image in a light field. That is, if we cannot locate the camera that captured a particular image then we cannot locate the light field samples represented by the pixels of that image. Many previous approaches to light field capture have used some kind of specialized device to obtain camera pose. Examples of this include the robotic arm used by [5] and the calibrated stage used by [3]. Other approaches have used Structure from Motion (SfM) techniques to compute camera pose for images captured with a handheld camera. Examples of this include [1] and [7]. In contrast, we use a handheld camera connected to a laptop computer and run the technique for Parallel Tracking and Mapping (PTAM) presented by [4] to compute pose estimation at interactive rates during capture.

Obtaining Good Coverage

We want the user's movement of the capturing camera to approximately cover the space of views we might later wish to reconstruct. The coverage of a region of the light field corresponds to the density of samples, or captured images, in that region. The task of capturing a good coverage of the light field then corresponds to acquiring a dense distribution of samples in the space of views we may later wish to reconstruct. This presents two major challenges for the user:

- *Navigation:* The space of camera poses is high dimensional and can be very difficult for a user to navigate. It is hard for a user to gauge which parts of the light field they have already visited and which parts remain uncaptured.

- *Resource Limits:* The light field contains an infinite amount of information, but we have limited resources for capture. Even with no limit on memory a user will always have limited patience, and the time and precision asked of the user becomes too great if we request too dense a sampling of the light field. On the other hand, if we request a sampling density that is too sparse then the quality of our reconstructions will suffer.

To address the navigation issue the system presents the user with real time visual feedback in the form of a coverage map. This coverage map indicates the density of captured samples in different parts of the light field. By looking at the coverage map it is easy for a user to tell which parts of the light field are under sampled.

To address our resource limits the system decides every frame whether to record or to reject the image presented to the camera. This decision is made by evaluating whether the current camera pose is in a region of the light field with a density of captured samples above or below a certain threshold. The user can decide to increase or decrease this threshold by evaluating the quality of the data they have already captured. The user evaluates this quality by reviewing the reconstruction that results from the current set of captured images, which they can do at any point during the capture process.

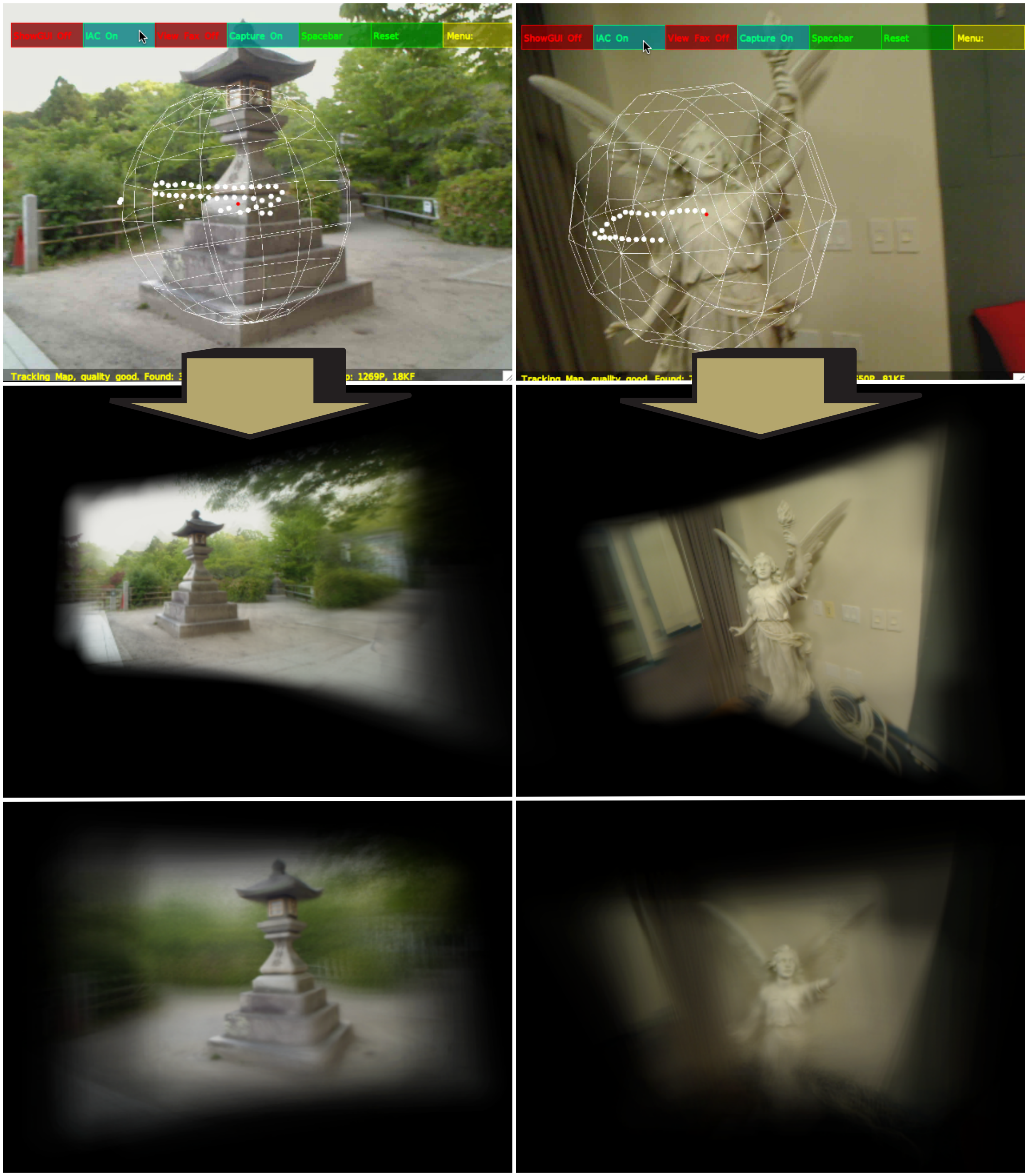


Figure 1.1: *The system includes a capture process (top) and a process for rendering captured data (middle, bottom). The light fields created with this system let users examine captured objects in 3D (middle) and create synthetic depth of field effects by refocusing a virtual camera (bottom).*

Chapter 2

Capturing a Light Field

This chapter describes how the system is used to capture a light field.

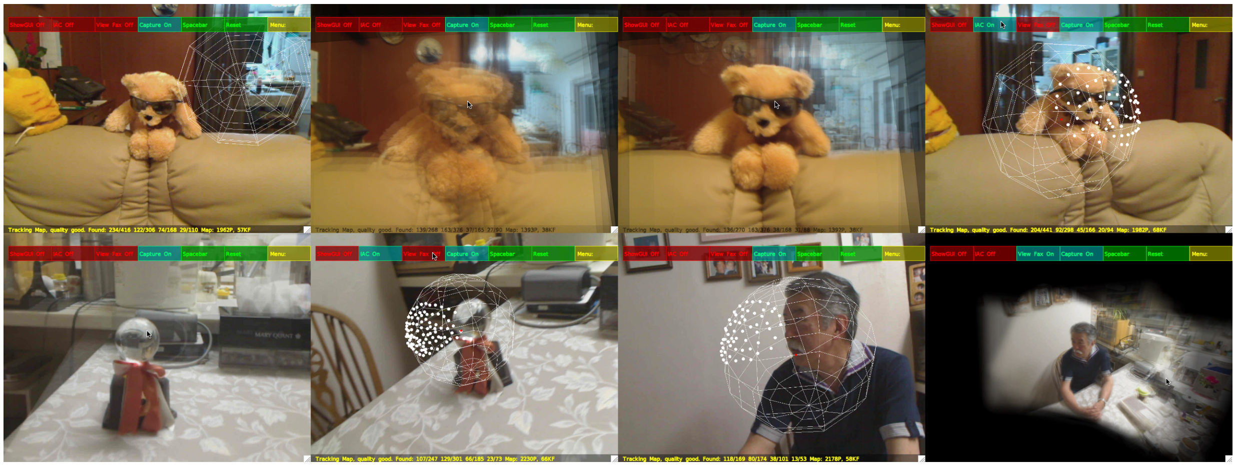
2.1 Setup

The hardware used by the current implementation of the system consists of an off the shelf Logitech Quickcam Pro 9000 usb webcam attached to a MacBook Pro laptop computer. As PTAM requires a calibration profile for the camera, camera calibration software is run before the camera's first use and the resulting profile is stored for all future uses. Before capture is started, the user initializes the pose estimation of PTAM. This process typically takes between 30 seconds and 3 minutes depending on the size of the scene and the number of good image features available. For more details on this step we refer to [4]. Once the user is satisfied with the pose estimation provided by PTAM, they press a button to begin the capture process. As the user points the capturing camera at the scene, a live view of what the camera sees is displayed on the laptop computer.

2.2 Subject Selection

Before the software begins to automatically record images, the user must indicate the part of the scene they wish to capture. The user does this by placing a wireframe sphere in the scene. The system currently provides two ways to do this. The first lets a user click on the subject they wish to capture in the live view of the camera. This click indicates some ray in the space of the scene. The sphere is then placed at some depth along this ray. If the sphere is placed at an incorrect depth in the scene, the user can slide the center of the sphere along the selected ray until it reaches their intended target.

This first approach to subject selection has the disadvantage of making it difficult to tell whether the sphere has been placed at the correct depth. With only the wireframe model as a visual cue, the depth must be verified by moving the camera to check and see that the sphere remains fixed to the image of the intended subject. To address this difficulty the system allows a second approach to subject selection. In this approach the user first switches to what is called the augmented aperture mode. In this mode the live view of the camera is augmented with previously captured frames to create a live shallow synthetic depth of field view of the scene. To create a larger synthetic aperture and decrease the depth of field in this view, the user waves their camera in a small circle and captures just a few images of the scene. The user then simply focuses the shallow depth of field in their augmented aperture view on the subject they wish to capture. Since we know the depth of focus used to reconstruct this wide aperture view, we can infer the depth of the



Tracking Map, quality good. Found: 234/416 122/306 74/168 29/110 Map: 1962P, 57KF

Tracking Map, quality good. Found: 139/268 163/376 37/165 27/90 Map: 1393P, 38KF

Tracking Map, quality good. Found: 136/270 163/376 38/168 31/88 Map: 1392P, 38KF

Tracking Map, quality good. Found: 204/441 92/298 45/166 20/94 Map: 1982P, 68KF

Tracking Map, quality good. Found: 107/247 129/301 66/185 23/73 Map: 2230P, 66KF

Tracking Map, quality good. Found: 118/169 80/174 38/101 13/53 Map: 2178P, 58KF

Figure 2.1: *Top Row: Using the augmented aperture to focus capture. (1st) When capturing a stuffed animal, the subject sphere is initially placed in the feature-rich background. (2nd) After waving the camera in a small circle and taking just a few pictures, the augmented aperture is turned on. (3rd) The user refocuses the augmented aperture on the teddy bear. (4th) The user proceeds with the semi-automated capture.*
Bottom Row: Examples of capturing objects that are transparent, specular, or dynamic. (1st) The user uses the augmented aperture to focus on a crystal ball and specular ribbon. (2nd) The semi-automated capture of the crystal ball. (3rd) The same process is used to capture this human subject. (4th) The user can immediately review the data that they have captured of the human subject at any point in time during capture.

subject from the user's choice of where to focus.

2.3 Capture

After the wireframe sphere has been placed in the scene, the user initiates a semi-automated capture process. Once this process has been initiated, the software will automatically record new images whenever the camera is being presented with a perspective of the subject that is sufficiently different

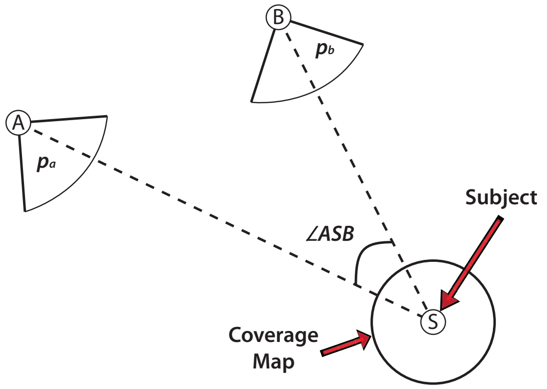


Figure 2.2: *The difference of two perspectives, p_a and p_b, is measured as the angle $\angle ASB$ formed by their centers of projection, A and B, and the center of the coverage map, S.*

from any previously captured image. The difference of two perspectives in this context is measured as the angle that their centers of projection form with the center of the coverage map (see figure 2.2). A new image is captured whenever the distance between the perspective for that image and any previously captured perspective is above a certain threshold.

The sphere that the user has placed in the scene becomes a coverage map to direct the user's movement of the camera. The current location of the camera is projected onto the surface of this coverage map as a red dot, and every previously recorded image of the scene is projected onto the coverage map as a white dot (as shown in 2.3). The user's goal then becomes to control the movement of the red dot and to "paint" the surface of the sphere with white dots. Doing so will capture enough data to generate a high quality reconstruction of the scene.

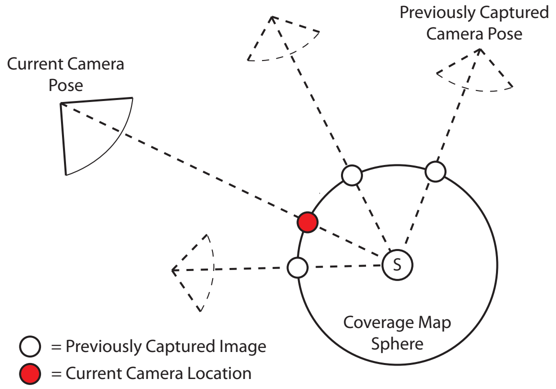


Figure 2.3: *Coverage Map: The projection of a perspective onto the coverage map is the intersection of the line connecting that perspective's center of projection to the center of the coverage sphere with the surface of the coverage sphere.*

2.4 Review

At any point in time during capture the user can switch to a review mode where the current set of captured images is used to generate an interactive visualization of the subject. The switch between capture mode and view mode is immediate. In review mode the user can examine the reconstruction of their subject by controlling the movement of a virtual camera. This can be done using a classic keyboard and mouse interface, or by using the pose of the user's capturing camera in the real scene to control the pose of the virtual camera used in the reconstruction. In this later case, the system simply renders the scene from the current pose of the real camera at every frame. This is similar to what is done in the augmented aperture mode, except the scene is now reconstructed with a much larger synthetic depth of field.

The review mode is used to evaluate the quality of the current set of captured images. If the user is not satisfied with the quality of their reconstruction, they can relax the threshold used to determine when to capture new images and then switch back to capture mode to capture a denser sampling of the light field. If the user finds that there are errors in some of the images that they have captured - say due to failures in pose estimation or some temporary occluder that entered the scene - then they can navigate their virtual camera close to the perspectives with error and press a button to wipe the bad images from the light field so that they may be re-captured.

Chapter 3

Rendering

The data captured with this system can be rendered using any existing technique for IBR from unstructured sets of images. The highest quality reconstructions can be achieved through extensive post processing of the captured data. However, in order to present the user with an immediate visualization of recorded data to assist in the capture process, the system needs some technique for rendering that runs at interactive rates and requires no additional post processing. An approach like the one described by [1] may be a good choice for this if the user's coverage of the light field is very dense and the content of the captured scene has not changed significantly over the course of capture. For light fields with much sparser coverage or light fields with changing scene content an approach like the one described in [7] is often more suitable. To balance some of the advantages of these different approaches, a new technique for rendering was developed for this system.

One consideration for rendering is whether to reconstruct different parts of the reconstructed image using different captured images. This amounts to deciding whether to use a blending field as described in [1] or whether to use only whole images in a reconstruction as in [7]. The approach used for this system is closer to [7] in that it only uses whole images at a time in the reconstruction. This is done for two reasons. First, when testing the system on a variety of scenes it was found that using whole images produced less objectionable artifacts when the scene changed over the course of capture (an observation also made by [7]). This makes using whole images more suitable for objects that move, such as human subjects. Second, using whole images makes it easier for a user to find images that they want to remove from the light field during review. Though the system does not compute an explicit blending field like the one described in [1], the boundaries of the images used in our reconstruction are faded out so that many images can be rendered on top of one another without introducing high frequency artifacts at image borders. This creates an implicit blending field that respects the smooth blending field argument made by [7].

Here a bit of notation is introduced to help describe how light fields are rendered in the system:

- $\mathbf{C}$ - The set of perspectives of the scene that have been captured by the system. The perspective $\mathbf{C}_i \in \mathbf{C}$ is given by an image and its corresponding pose estimate.

- p_{req} - Some requested perspective of the scene. This is given by the current pose of the virtual camera that the user controls while viewing a light field.

- S - A 3D point in the scene being captured that will be a focus point for the reconstruction. S will generally be the same as the center point of our coverage map.

- $penalty(p_a, p_b)$ - A function that evaluates the difference between p_a

and p_b as described in figure 2.2.

To render a requested perspective p_{req}, the system begins by finding the closest k captured perspectives to p_{req}. This can be done by sorting $\mathbf{C}$ in increasing order of $penalty(p_{req}, \mathbf{C}_i)$ and taking the first k elements of this sorted list. The images from these k perspectives are each be projected onto planes in the virtual scene. These planes, each textured by one of the k projected images, are then be viewed from the perspective p_{req}. For the examples provided in this thesis, k is typically between 40 and 60.

Many approaches to IBR follow the same basic technique described above. That is, captured images are projected onto some geometry in the virtual scene and then this textured geometry is viewed from a requested perspective. What differs among these techniques the most is usually the geometry that images are projected onto and the weights with which these images are blended. The geometry that images are projected onto is often referred to as a geometric proxy. The next two sections describe the geometric proxy and the blending weights used to render a lights field in the system.

3.1 Choosing Proxy Planes

Planes make a good geometric proxy in the absence of very accurate geometric information about a captured scene. This is because a plane has no self occlusions unless it is viewed from a point on its surface. As a result, planes will not introduce sharp depth discontinuities where there were none in a captured scene - a property that prevents some of the more objectionable image artifacts that are common in IBR.

There are two common choices of how to position proxy planes for rendering

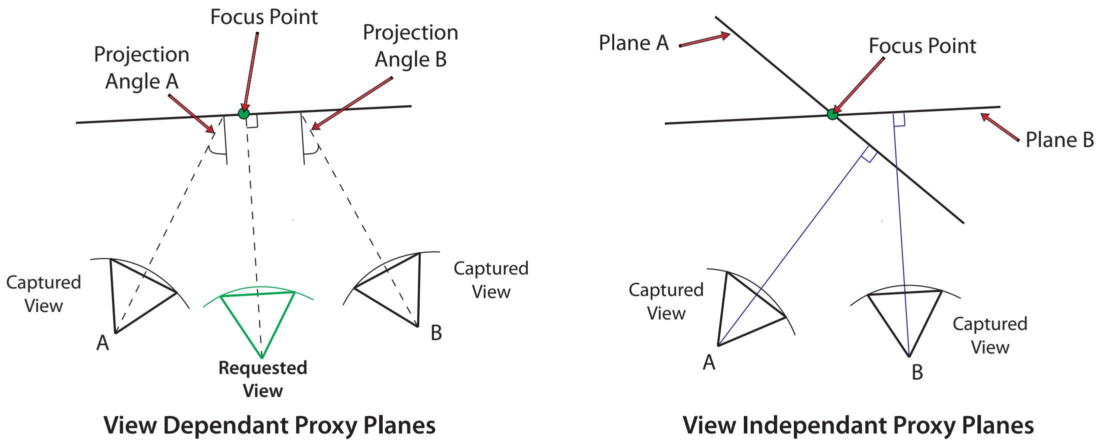


Figure 3.1: *Two different choices of proxy plane orientation.*

a scene (see figure 3.1). In either case, every proxy plane passes through a focus point, represented by S in the system described here. The difference between the two options is in the orientation of planes. The first choice is what we will call the use of a view dependent proxy plane (see figure 3.1 left). In this case captured perspectives are projected onto a plane that passes through the focus point S and is perpendicular to the optical axis of the requested perspective p_{req}. This is a view dependent proxy because the orientation of the plane changes with the requested view. The use of a view dependent planar proxy is common when rendering densely sampled light fields. The second option is what we call the use of view independent proxy planes. View independent proxy planes are different for each image being project into the scene, but do not change with the requested perspective. A good choice of orientation for view independent proxy planes is perpendicular to the optical axis of a projected perspective. That is, each of the k images used in reconstruction is projected onto its own plane that passes through the focus point S and is perpendicular to that perspective's optical axis. View independent proxy planes are used in [7] because they can produce better results when rendering with a sparse set of captured images.

The system described here initially let users switch between using view dependent and view independent proxy planes. After observing the differences between the two techniques for many different light fields, it was concluded that for small projection angles a view dependent proxy plane resulted in less ghosting than view independent planes. When the projection angle was large it was found that the warping that resulted from projecting onto a view dependent proxy plane at an extreme angle became more objectionable than the ghosting produced by view independent planes. To balance the effects of ghosting and warping, the system now sets the orientation of each projection plane to be a spherical linear combination of the view dependent and view independent orientations (expressed as quaternions). When the angle of projection is 0, the weight of the view dependent orientation is 1 and the weight of the view independent proxy is 0. As the angle of projection approaches 90° the weight of the view dependent orientation approaches 0 linearly and the weight of the view independent orientation approaches 1 linearly. This approach to selecting proxy plane orientation effectively balances the advantages of view dependent and view independent proxy planes.

3.2 Choosing Weights

When blending the k perspectives used to render the light field, the spatial distribution of blending weights determines the size of the synthetic aperture used in the reconstruction. That is, it determines the size of the perceived depth of field in the rendered image. The synthetic aperture will approach the size of the neighborhood defined by the k perspectives as those perspectives are weighted more evenly. If perspectives closer to p_{req} are weighted much more heavily then the perceived depth of field becomes smaller.

The system provides two choices for blending the k perspectives used to reconstruct p_req. The first option is wide aperture mode (see figure 3.2 (left)). This takes the k perspectives and first assigns a weight to each $\mathbf{C}_i$ that is equal to $penalty(p_{req}, \mathbf{C}_i)$. Next an exponent is applied to each weight. The weights are then normalized and used as alpha values when rendering each of the k textured proxy planes. The value of the exponent applied to each weight controls how much more perspectives are weighted when they are close to p_{req}. This controls the perceived defocus blur in the reconstruction. If the exponent is high then a smaller number of perspectives close to p_{req} are weighted much more heavily. This creates a small aperture effect resulting in a large perceived synthetic depth of field. If the exponent is close to 0 then all of the k perspectives are weighted more equally. This creates a large aperture effect and results in a shallow perceived synthetic depth of field. When we render in augmented aperture mode we set this exponent to 0.

The second option for blending images is small aperture mode (see figure 3.2 (right)). In the current implementation of small aperture mode the size of k is capped at 40. Before rendering, the OpenGL blending mode is set to (GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA). The k images are then rendered in order of decreasing penalty from p_{req}. The first half of the images are rendered with alpha values that increase linearly from 0 to 0.4. The remainder of the images are then rendered with alpha values of 0.4. Finally, the perspective with the smallest penalty against p_{req} is rendered with an alpha value of 1. This technique results in a very large perceived synthetic depth of field over the part of the rendered image that is covered by the nearest neighbor to p_{req}, but the rendered image blurs more in parts of the scene that were only visible in perspectives that have a higher penalty with the requested view.

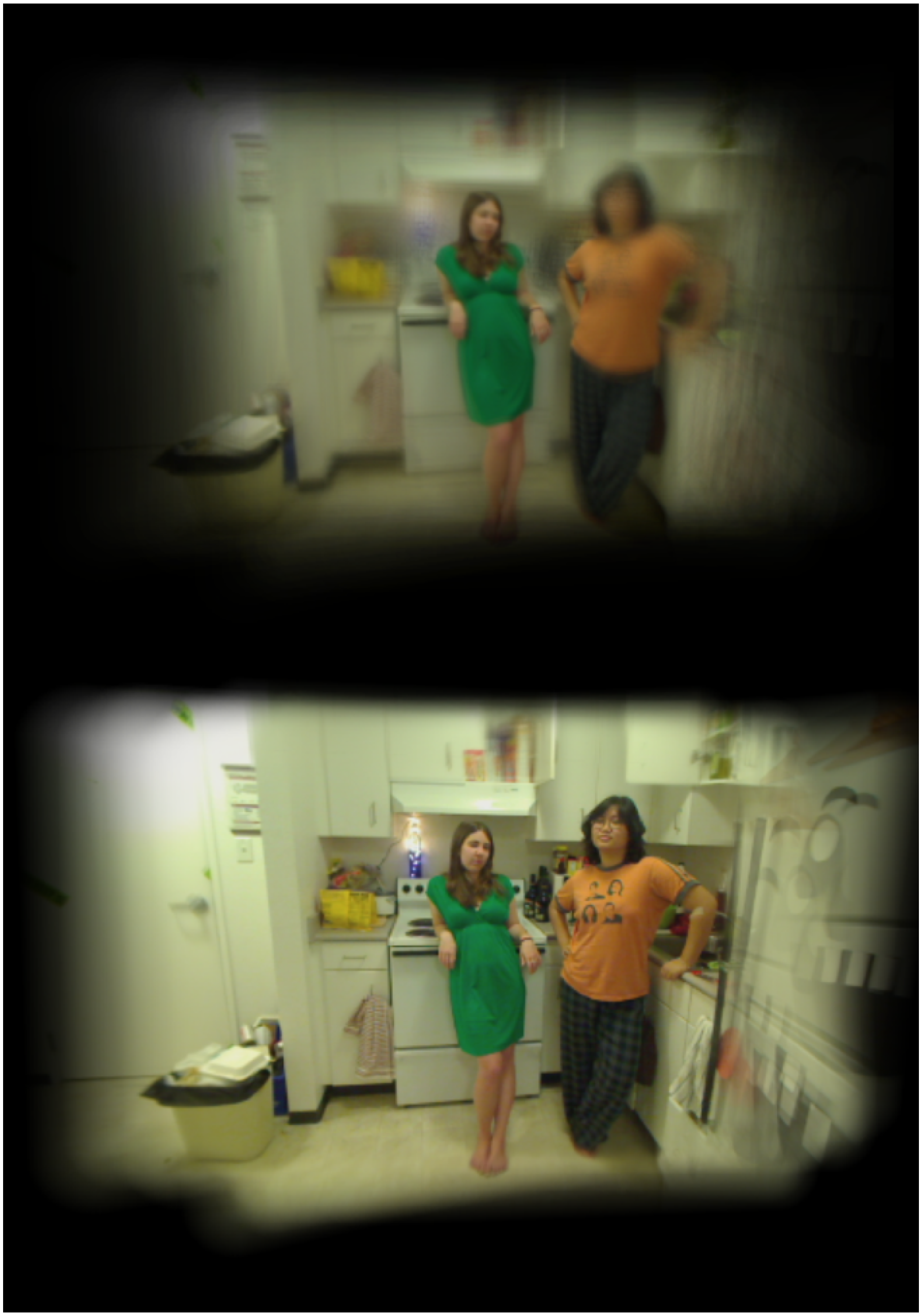


Figure 3.2: *Blending Projected Images: (top) Wide aperture mode with a small exponent applied to image weights. (bottom) Small aperture mode.*

Chapter 4

Discussion

4.1 Results

The system described here makes it very easy to capture light fields of a broad range of subjects. A typical capture takes between 30 seconds and 5 minutes depending on the size of the scene, the number of images to be included in the light field, and the proficiency of the user. In the course of designing and implementing the system it was used to capture well over 100 light fields. Most of these light fields consist of between 100 and 300 images distributed in a variety of different vertical and horizontal resolutions. In general the system is capable of capturing high quality light fields quickly and easily whenever it is not limited by one of two factors. Here those factors are discussed as well as what if anything a user can do to get around them.

Pose Estimation The greatest limiting factor for the system is the need for good image features from which to compute pose estimation. If there are no good image features in a scene then the system cannot be used to capture a light field of that scene. This limitation can often be addressed by moving away from a subject so that the camera can see more of the background in a scene. Features in the background can then be used to compute pose while the user positions the coverage map around a problematic subject of capture. By doing this users can use the system to capture many challenging subjects including humans (figure 4.1 (top)) and subjects that frequently move during capture (figure 4.1 (right)).

User Mobility User mobility can be limited by subject scale or by obstacles in the users environment. When a subject is very large or very far away, the user must move a larger distance in the physical scene to cover a significant range of angles to the subject. This is especially an issue when the system is used to capture large objects such as buildings. The building example shown in figure 4.1 (left) has an average vertical resolution of about 3 images and a horizontal resolution of about 140 images. The difference in vertical and horizontal resolutions is due to the users limited ability to move vertically in the scene. Buildings also generally took the longest to capture of any subject the system was tested on. This was because users had to physically travel a greater distance to obtain good angular coverage of a subject at the scale of a building. Mobility was often an issue for smaller scenes as well though. For instance, the example shown in figure 4.1 (top) was taken in a small room with a large table that limited the user's movement in the scene. This made capturing a light field more difficult. Note that these limitations due to mobility are less limitations of the system and more limitations of the user in a particular kind of environment. As such, they will limit any technique for light field capture that uses a hand-held camera for input.

4.1.1 Thesis Website

As the results of this system are especially difficult to appreciate without seeing the system in use, a video was created to accompany this thesis. That video and a digital copy of this thesis are available at:

`http://graphics.stanford.edu/~abedavis/abedavisthesis.html`

4.2 Contributions

The system presented in this thesis makes three key contributions to light field capture.

4.2.1 Hardware Accessibility

The system uses only common off the shelf hardware to capture light fields. This is significant in contrast to many methods of light field capture that depend on specialized hardware such as a robotic arm or calibrated camera rig.

4.2.2 Speed

The time it takes to capture, process, and view a light field with this system is generally much less than the time spent just on processing a light field in other systems that use a handheld camera for capture.

The advantage in processing speed comes from the use of PTAM, which a technique for Simultaneous Localization and Mapping, rather than any of the SfM techniques used in prior work on light field capture. The advantage in speed offered by SLAM over SfM comes from the assumption that images are coming from a video captured by a single camera. This allows for a small motion prior on the pose of consecutive video frames. While the resulting pose estimation can be less robust than many techniques for SfM, the benefit in speed is enormous.

Perhaps the greatest advantage in speed comes from the fact that this system addresses the capture, processing, and viewing of light fields all in one application. There is no other system that does this without the use of specialized hardware, and as a result a direct comparison of typical use time against any other system with similar hardware accessibility becomes difficult. Even if the capture, processing, and viewing of a light field with this system is compared against just the processing time in other systems, this system is generally 2-10 times faster. When the time spent on capture and the time spent organizing images for processing in other techniques is also considered, the advantage of the system described in this thesis becomes much greater.

4.2.3 Coverage

It is difficult to consider coverage quality and speed in isolation. This is due to the fact that given enough time and resources anyone with a hand-held camera can capture good coverage of a subject. However, previous techniques for light field capture that used a hand-held camera with no additional specialized hardware have only been able to consistently produce light fields that exhibit parallax in a single dimension. In contrast, the system described in this thesis consistently produces light fields that exhibit parallax in multiple dimensions. This is because the of the interactive element of the system. The feedback that a user is given in the form of a coverage map makes it easy to obtain much better coverage of a subject than was possible with previous techniques.

4.3 Conclusion

The system for light field capture described in this thesis makes it very easy for an individual user to generate high quality light fields with common off the shelf hardware. Compared to any previous options for a single user without specialized hardware, our technique is much faster and the light field data that it produces generally provides much higher quality coverage of a subject. This represents significant progress toward making light fields a much more accessible type of media. By making it easy to author this type of media, this system offers a much larger community of users a new way to communicate by capturing and exchanging interactive representations of their environment.

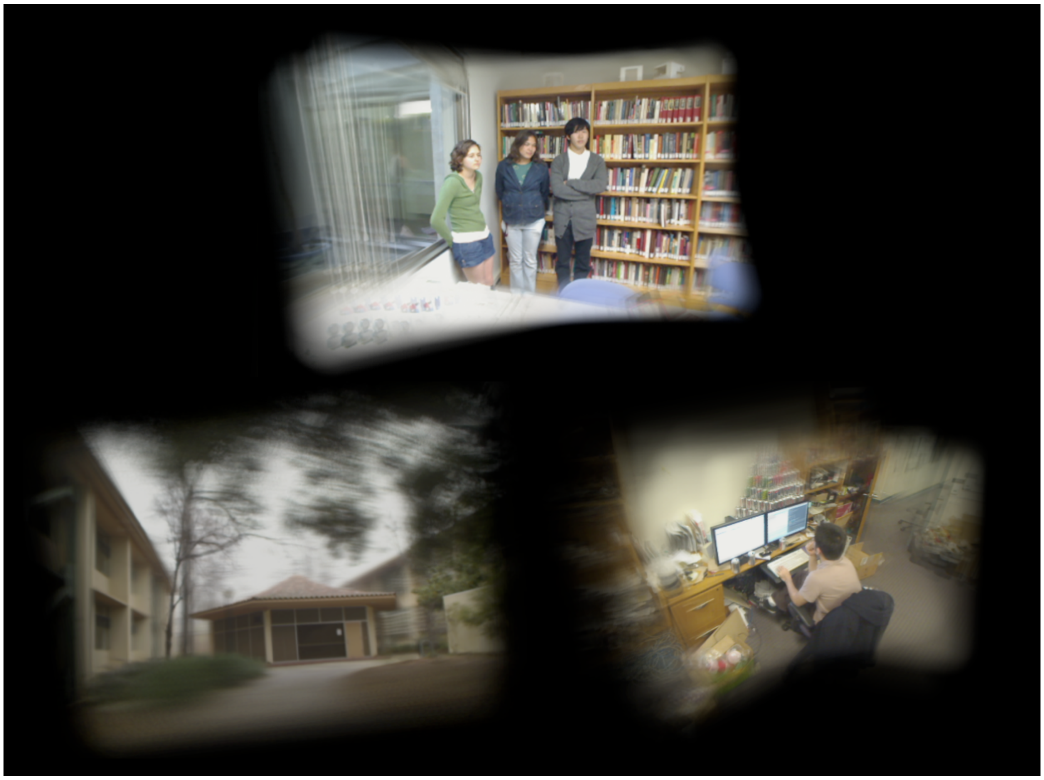


Figure 4.1: *These are examples of light fields that were each captured in less than 5 minutes using the system described in this thesis. The system hangles large objects, human subjects and even moving subjects. (left) A light field of a building that was captured with the system. It took about 5 minutes to capture and consists of 420 images. (top) A light field of three posed human subjects captured with the system. This light field took about 2 minutes to capture and consists of 157 images. (right) A light field of a human subject moving as he works at his desk. This light field took about 4 minutes to capture and consists of 185 images.*

Bibliography

[1] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 425–432, New York, NY, USA, 2001. ACM.

[2] Paul Debevec, Yizhou Yu, and George Boshokov. Efficient view-dependent image-based rendering with projective texture-mapping. Technical report, Berkeley, CA, USA, 1998.

[3] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1996. ACM.

[4] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[5] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, New York, NY, USA, 1996. ACM.

[6] Ren Ng. Fourier slice photography. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 735–744, New York, NY, USA, 2005. ACM.

[7] Noah Snavely, Rahul Garg, Steven M. Seitz, and Richard Szeliski. Finding paths through the world's photos. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–11, New York, NY, USA, 2008. ACM.

[8] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 835–846, New York, NY, USA, 2006. ACM.

[9] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph.*, 24(3):765–776, 2005.