An Uncalibrated Lightfield Acquisition System

Puneet Sharma   Angshuman Parashar   Subhashis Banerjee   Prem Kalra
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
{suban,pkalra}@cse.iitd.ernet.in

Abstract

Among the various techniques of Image Based Rendering, Lightfield Rendering has been of special interest. The primary reason behind this is that no depth information or feature matching is required to generate novel views. The process of rendering just involves combining and resampling the given images. Acquistion of image data however remains a challenging and time consuming task. In most methods expensive, complex and bulky setups are being used.

In this paper we describe a simple method of acquiring the image data set. The method requires a normal handheld video camera, which is taken around the object to be rendered. The simple computations involved make the method suitable for online lightfield acquisition.

1. Introduction

Traditionally 3D graphics systems use geometric modeling where the scene is represented as a set of geometric primitives and lights. Such scenes are rendered using the standard geometric rendering pipeline which involves modeling and viewing transforms, projection, clipping, perspective division and scan conversion. A relatively newer approach to rendering is Image Based Rendering (IBR) [9, 1]. IBR offers many advantages:

- The results of IBR are far more photo-realistic and modeling of scenes is easier since images are used to model the scene.
- IBR techniques are less computationally intensive and hence suitable for real-time rendering.
- The rendering speed is independent of the scene compelxity.

Lightfield Rendering as proposed by Levoy and Hanrahan[8] is an Image Based Rendering technique that relies on a convenient representation of the radiance information of a scene as a function of position and direction [10]. The lightfield is similar to the Plenoptic Function [2] except that in a space free of occluders it becomes a 4D function unlike the Plenoptic Function which is a 5D function. Levoy and Hanrahan [8] have suggested a convenient representation of the lightfield using two parallel, parameterized planes or *lightslabs* as shown in Figure 1. The intensity of any ray intersecting the two planes is a function of four parameters, u, v, s and t, where (u, v) is the intersection of the ray with the outer plane and (s, t) is the intersection with the inner plane. As has been shown in their paper an image taken from a viewpoint on one of the planes becomes a 2D slice of the 4D lightfield.

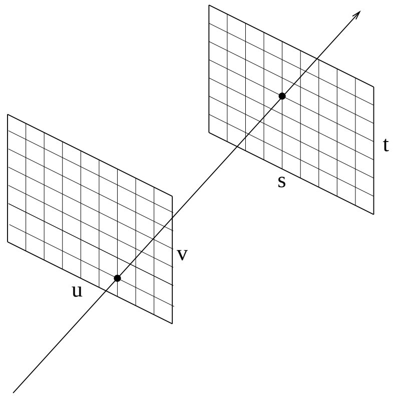

Figure 1: Lightfield representation.

The process of lightfield rendering involves two steps – acquisition of the lightfield, and rendering using the acquired data. To acquire the lightfield a series of images is taken from different viewpoints on one of the two planes used in the lightfield representation. An image taken in this manner is just a set of rays between the viewpoint of the camera and pixels on the image. Continuing in this way, the complete lightfield can be generated. Rendering from the acquired lightfield is a simple task – to generate an image from a new viewpoint, rays are cast from the centre of projection of desired viewpoint to each pixel of the desired image. Intersections of these rays are computed with the planes of the lightfield and nearest ray(s) from the lightfield are looked up for intensity values.

Convenient acquisition of lightfields for real world scenes has been one of the longstanding problems[1] associated with lightfield rendering. Expensive, complex and bulky setups have been used for the purpose and this has been prohibitive to experimenting with Lightfield rendering. In this paper we present a method of lightfield acquisition which is convenient, fast and does not require large calibration objects to be visible.

2. Prior Work

The lightfield acquisition system designed by Levoy and Hanrahan [8] consists of a single camera on a robot arm that translates in a two-dimensional plane as it captures images of an object. The camera has a narrow field of view lens, so the system allows the camera to be rotated towards the object. The images captured thus cannot be used as direct entries to the lightfield database but have to be transformed so as to align them along a plane parallel to the translation plane.

The camera is calibrated once at the onset of the data acquisition process, and all positions are estimated based on the translation velocity of the robot arm. The primary drawback of this system and others similar to this is the infrastructure costs involved in building a high-precision robotized device as described. In addition, such systems suffer in the time required to build a dataset. Reportedly, it took around four hours to capture a typical dataset.

Gortler et al [4] proposed the **Lumigraph** system employing a handheld digital camera which the operator uses to capture images of an object at different angles. Since the camera now has full freedom of movement, it has to be re-calibrated each time an image is captured. The Lumigraph system uses Tsai's camera calibration algorithm [11] for the purpose. The lightfield planes in such a setup are virtual – they are placed in suitable imaginary locations in 3D space. With knowledge of the camera parameters, rays can be shot out of the camera model which transfer intensity values from image pixels to the corresponding intersection points with the two lightfield planes. After some resampling, the lightfield can be updated with the intensity values of the rays with the resampled intersection points.

The system's chief drawback lies in the calibration step – Tsai calibration is an exceedingly complex operation and not very accurate. A second shortcoming is on account of the lack of structured camera movement which result in uncaptured lightfield regions or *holes*. The Lumigraph system uses a *rebinning* step to solve this problem but this results in an inaccurate lightfield.

Isaken, McMillan and Gortler [6] have also recently built a lightfield capture system. Their device uses an X-Y platform to translate a camera, similar to that of Levoy and Hanrahan, but it uses a wide-angle lens so as to avoid rotating the camera. This system only allows for generation of a single lightslab and so allows limited freedom of movement during rendering.

[1]inasmuch as a problem associated with a concept introduced in 1996 can be called longstanding

Again, the drawback to this device is the infrastructure costs. The authors have suggested that using a handheld camera and employing self-calibration techniques could possibly be an interesting alternative approach, no details about the process are available though.

In another approach [7], the image sequence is acquired by simply waving the camera around the scene objects, creating a zigzag scan path over the viewing sphere. They extend the sequential camera tracking of an existing structure-from-motion approach to the calibration of a mesh of viewpoints. Novel views are generated by piecewise mapping and interpolating e new image from the nearest viewpoints according to the viewpoint mesh.

Our approach is in principle similar to that of the Lumigraph in that a handheld camera is used and this has to be "calibrated" for every frame captured. However, we show that it is not really necessary to completely determine the camera parameters in order to establish the mapping of image points from the camera image plane onto the lightfield planes. This results in a system which is faster, easier to use and far more accurate than a fully calibrated system.

3. Our Approach

3.1. Motivation

Let us first examine what exactly has to be done in order to incorporate the information captured by a particular image frame into the lightfield. Essentially, the lightfield is nothing but a set of rays – all rays that pass through the bounded parallel planes of a lightslab are elements of the lightfield.

In order to *update* the lightfield with information gathered from a particular frame, we need to determine the set of rays which the frame has captured and which intersect both of the lightfield planes, and insert these rays (actually, the intensity corresponding to these rays at the position indexed by the rays) into the lightfield. And, in order to compute the parameters of any ray captured by the frame in world coordinates, we need the camera parameters, which is why calibration seems a natural step.

However, on closer inspection in the light of the fact that the frame rays have to be intersected with *planes*, we see that the rays themselves need not be computed at all. All that is required to be computed is the projection, or *homography* [5], from the viewing image plane to both the lightfield planes. Corresponding to every frame image point, the homography onto a lightfield plane would then give the ray intersection point with that plane.

3.2. Setup

Our experimental setup involves taking a handheld video camera around the object for which the lightfield is to be created. The object is placed between two planes, the frontal plane is transparent and both the planes have four identifiable points each, marked on them. The setup is shown in Figure 2.

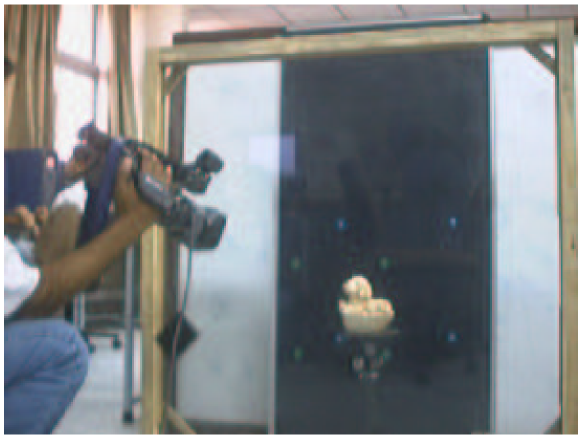

Figure 2: The setup for capturing lighfields.

In the experiments we have restricted ourselves to just one lightslab. This restricts our freedom of movement while rendering. Ideally, all six lightslabs should be constructed for a complete fly-around of the scene. Our single-lightslab restriction is just to keep things simple, there should be no conceptual obstacles in generating all the lightslabs.

3.3. Homography Computation

Computation of the homography matrix requires atleast a four point correspondence, this can be done by placing four recognizable points on each of the lightfield planes. We have used color based region growing for identifying these points. Whenever all four points corresponding to a plane are visible in any camera image, the homography from the camera image to the lightfield plane can be established from the point correspondences using the standard homography equation.

$$\underbrace{\begin{bmatrix} x_h \\ y_h \\ h \end{bmatrix}}_{X'} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{X}$$

The process is illustrated for one lightfield plane in Figure 3.

3.4. Homography Application

Once the homography from the camera plane to the lightfield plane has been established, the intersection point of a ray corresponding to a camera image pixel can be computed by applying the homography matrix to the image pixel. Figure 4 illustrates this.

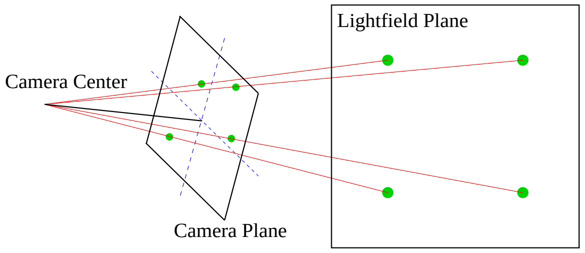

Figure 3: Computation of homography between camera and lightfield plane.

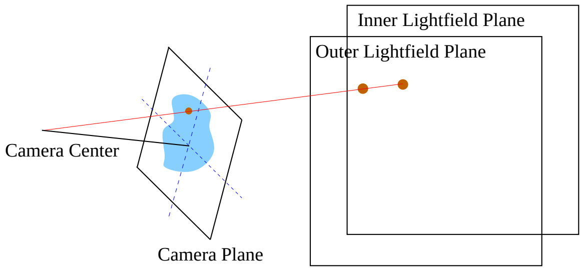

Figure 4: Application of homography to image pixel.

Ray intersections are computed for both the lightfield planes. For modifying the lightfield the point of intersection has to be resampled. We use quadrilinear interpolation for this. There are four nearest neighbors to the intersection point on the *st* plane and four on the uv plane as shown in Figure 5. Hence, there are 16 neighboring rays, each ray passing through one neighbor on the uv plane and one neighbor on the st plane.

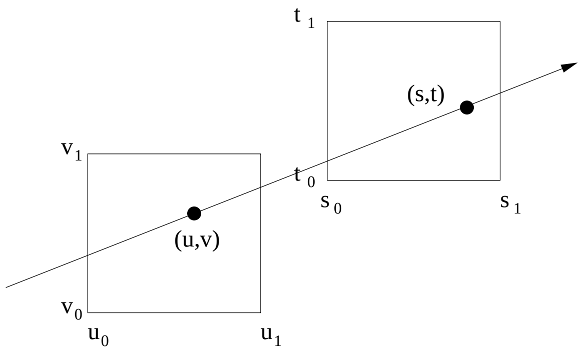

Figure 5: Quadrilinear interpolation.

The interpolation factor, corresponding to each of the neighbor rays is computed using the following equation.

$$F(u_i, v_j, s_k, t_l) = \frac{|(u - u_i)|}{(u_1 - u_0)} \frac{|(v - v_j)|}{(v_1 - v_0)} \frac{|(s - s_k)|}{(s_1 - s_0)} \frac{|(t - t_l)|}{(t_1 - t_0)}$$

3.5. Lightfield Update

For each ray, the lightfield is updated for all the 16 neighbors of the point of intersection of the ray, this is helpful in filling the holes in the lightfield quickly. The lightfield stores a weight corresponding to each (u, v, s, t) along with the lightfield value, changes are made to the lightfield corresponding to a (u, v, s, t) by doing a weighted averaging between the ray intensity and the intensity value stored in the lightfield. If $I(u, v, s, t)$ is the intensity of the ray and $F(u_i, v_j, s_k, t_l)$, $L(u_i, v_j, s_k, t_l)$, $W(u_i, v_j, s_k, t_l)$ are the interpolation factor, lightfield value and weight, respectively, for (u_i, v_j, s_k, t_l), which is one of the neighbors of (u, v, s, t), then the new intensity is given by the following equation.

$$\begin{aligned} L_{new}(u_i, v_j, s_k, t_l) &= \frac{L(u_i, v_j, s_k, t_l).W(u_i, v_j, s_k, t_l)}{W(u_i, v_j, s_k, t_l) + F(u_i, v_j, s_k, t_l)} \\ &+ \frac{I(u, v, s, t).F(u_i, v_j, s_k, t_l)}{W(u_i, v_j, s_k, t_l) + F(u_i, v_j, s_k, t_l)} \end{aligned}$$

The new weight is given by the following equation.

$$\begin{aligned} W_{new}(u_i, v_j, s_k, t_l) &= max(c, W(u_i, v_j, s_k, t_l) \\ &+F(u_i, v_j, s_k, t_l)) \end{aligned}$$

where c is an upper limit on the weight; we have set this to 255.

3.6. Rendering

Rendering from a pre-generated lightfield boils down to shooting rays out of the target camera (in a manner akin to ray-tracing), computing the intersection of these rays with the two lightfield planes, and looking up the lightfield with the intersection points as indices. Resampling methods such as nearest neighbor approximation and quadrilinear interpolation have been incorporated.

Figure 6 shows the effect of nearest neighbor versus quadrilinear interpolation in *uvst* during rendering. As can be seen, aliasing artifacts are reduced when using quadrilinear interpolation during rendering, however quadrilinear interpolation greatly reduces the speed of rendering.

Profiling runs on a renderer based on the above process showed that constructing rays from image pixel locations and computing intersection of these rays with the lightfield planes were demanding the major time slices. Inspired by our technique used for generation, we decided to use homographies to speed up the rendering process.

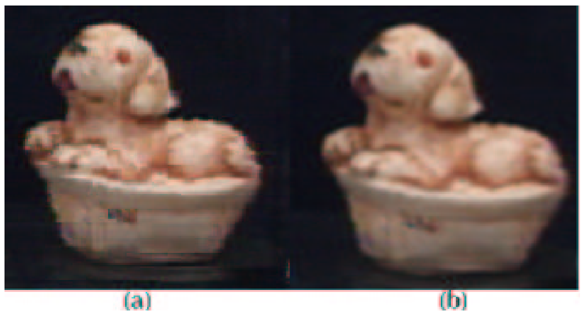

Figure 6: Rendering using (a) Nearest neighbor approximation, (b) Quadrilinear interpolation.

In our method, only four rays undergo the complete process of ray-construction and plane-intersection computation. The four intersection points help determine the homography between the target camera plane and the lightfield planes.

Once the homography is computed, it needs to be applied to each pixel of the target camera get the index required to query the lightfield. The process of homography application is computationally less expensive than ray construction and computation of ray-plane intersections. Further, application of the same homography matrix to successive pixels can be simplified by carrying out the process incrementally.

3.7. Compression

Levoy and Hanrahan have used Vector Quantization for compressing lightfields. Vector Quantization allows for random-access real time decompression – a feature which makes it possible to directly render compressed lightfields online. Also, very high compression ratios can be attained without significant loss in quality.

The downside lies in the compression operation which is very time consuming. First a **code book** has to be generated from a **training set** of samples (which is usually a subset of the data). Then, the data is compressed using the code book. The training stage is especially computation intensive.

We had used vector quantization on our initial lightfield simulations for synthetic scenes. However, our experiments showed clearly that the compression process was too slow to be incorporated into an on-the-fly lightfield capture system.

What instead could be created is an offline compression system which compresses a lightfield after it has been captured. For the sake of simplicity, we have not implemented such a compression module; we work directly on uncompressed lightfields.

4. Implementation & Results

The schematic of the system that we have implemented is shown in Figure 7.

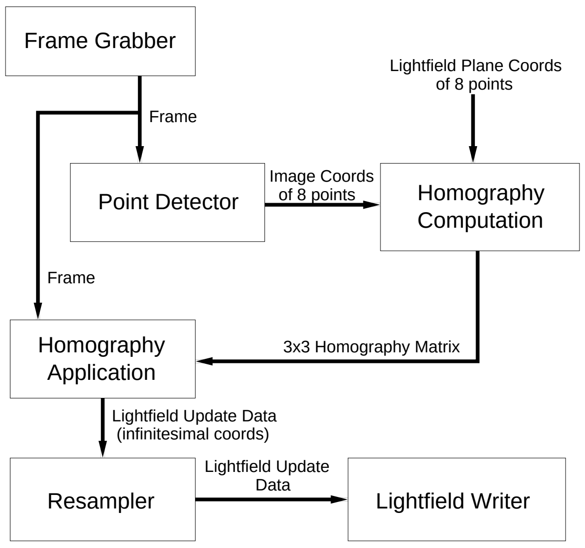

Figure 7: Schematic

We have used a Sony digital video camera with a bttv848 frame grabber card to grab frames. All processing is done on a 1.4GHz Pentium 4 system with 1GB memory. The frames are fed to the point detector module which uses color identification and region growing to identify the eight points (four on each plane). We are not tracking these points, however we expect this module to become faster if tracking is done. Once the points are identified in the image, the homography computation module computes the homography between the image plane and the two lightfield planes. The homography application module projects the image from camera image plane to the two lightfield planes as explained in Section 3.4. The infinitesimal (continuous-space) coordinates are resampled to the (u, v, s, t) coordinates by the resampler module and the lightfield is augmented as explained in Section 3.5.

During lightfield capture, we are able to process frames of resolution 512 x 384 pixels at a rate of around two frames per second. A more efficient implementation with tracking of points would increase this rate. The capture process takes around 15 minutes to generate a fairly dense lightslab. Once the lightslab has been generated, new views can be generated as described in Section 3.6. Frames of resolution 256 x 256 are generated at a rate of over three frames per second using quadrilinear interpolation; rendering using nearest neighboring interpolation is faster.

Generating the entire lightfield in one shot requires delicate movements of the camera and meticulous attention to ensure that all angles have been covered. To overcome this problem we have provided the ability to build on an existing lightfield, thus if a lightfield is found to be of unsatisfactory quality we can identify the approximate location of the holes and build on it by capturing the required frames. Alternatively, a *monitor* window could be used to indicate to the user which frames are required.

The results of rendering two real world models are shown in Figures 8 and 9.

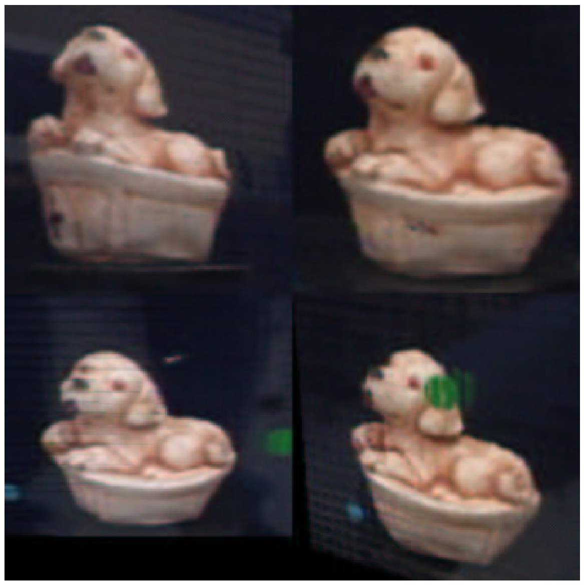

Figure 8: Clay dog model

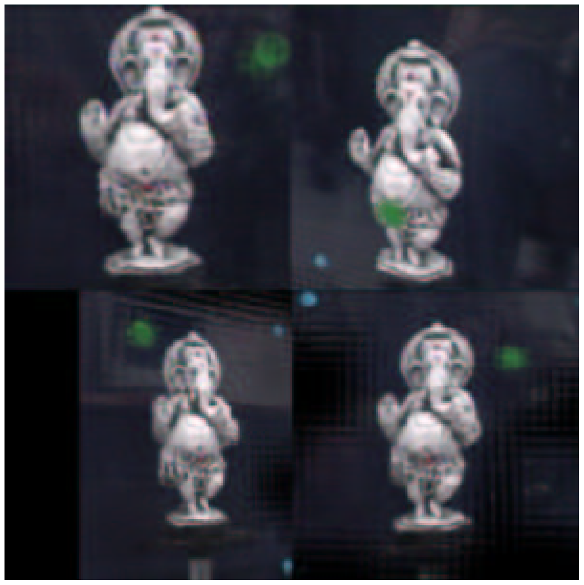

Figure 9: Lord Ganesha

For more results and videos see `http://www.cse.iitd.ernet.in/vglab/demo/lightfield`.

5. Conclusion

Lightfield representation is a convenient method of representing the radiance in a space free of occluders. The method requires no information about the surface properties or geometry of the scene and hence is suitable for real world scenes. However, a few problems are associated with lightfield rendering, lightfield acquisition is one of them.

We have presented a lightfield capturing system which is easily reproducible, cheap and convenient to use. Our method facilitates lightfield capture using a handheld camera without complete camera calibration.

Possibilities for future work include integrating an online compression system which would compress the lightfield data as it is being captured and update it into a compressed lightfield database. This would enable acquisition of high-resolution lightfields on desktop PCs. Currently, the method used to locate the four marker points for each lightfield plane is not very robust, nor is it very efficient. A better tracking mechanism would result in far more valid frames from which lightfield data can be obtained.

References

[1] Image-Based Modeling, Rendering and Lighting. In *SIGGRAPH Course 39*, SIGGRAPH 1999.

[2] E. Adelson and J. Bergen. The Plenoptic Function and the Elements of Early Vision. In *Computation Models of Visual Processing*, chapter 1. MIT Press, 1991.

[3] E. Chen, L. Williams. View Interpolation for Image Synthesis. In *SIGGRAPH '93*, pages 279 – 288, 1993.

[4] R. S. S.J. Gortler, R. Grzeszczuk and M. Cohen. The Lumigraph. In *SIGGRAPH '96*, pages 43 – 51, 1996.

[5] R. Hartley, A. Zisserman. Multiple View Geometry in Computer Vision. *Cambridge University Press*, ISBN: 0521623049, September 2000.

[6] S. G. A. Isaksen, L. McMillan. Dynamically Reparameterized Lightfields. In *SIGGRAPH 2000*, pages 297 – 306, 2000.

[7] R. Koch, M. Pollefeys, B. Heigl, L. Van Gool and H. Niemann. Calibration of Hand-held Camera Sequences for Plenoptic Modeling. In *ICCV '99*.

[8] M. Levoy and P. Hanrahan. Light Field Rendering. In *SIGGRAPH '96*, pages 31 – 41, 1996.

[9] P. J. Narayanan. Image Based Rendering: A Critical Look. In *Indian Conference on Computer Vision, Graphics and Image Processing*, New Delhi, 1998.

[10] M. Slater. Tutorial on Lightfield Rendering. *VRST 2000*, Seoul, Korea, 2000.

[11] R. Tsai. A versatile Camera Calibration technique for high-accuracy 3D Machine Vision Metrology using off-the-shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323 – 344, August 1987.