Constructing and Rendering Physically Valid Light Fields

Meng Sun

Contents

List of Figures

1 Introduction

Traditional image synthesis rendering has meant simulating the flow of light from a source, reflecting it from a geometric and material description of a model, into a simulated camera and onto a film plane to produce an image.

In recent years there has been increased interest, within the computer graphics community, in image-based rendering systems. These systems are fundamentally different from traditional geometry-based rendering systems. Image-based rendering (IBR) techniques generate new images from other images rather than geometric primitives. We will give a formal definition to IBR in section 1.1.

The study of image-based modelling and rendering techniques is essentially the study of sampled representations of 3D objects. In computer graphics, the progression toward image-based rendering system was initially motivated by the desire to increase the visual realism of the approximate geometric descriptions by mapping images onto their surface (texture mapping). Next, image were used to approximate global illumination effects (environment mapping) [15]. Recently, ground breaking new ideas in this area produced impressive results by combining techniques from computer graphics and computer vision.

In this paper, we restrict our attention to image-based rendering methods which produces physically valid images. Many morphing algorithms apply distortions to the input images for artistic reasons, and the resulting images may not be physically valid, so such morphing techniques are beyond the scope of this paper. Also, we assume that we are dealing with reference images obtained from either video captured data or computer generated scene, and we view the synthesized images by simply looking at the computer screen without employing any special devices.

In this section, we will give a high-level overview of this research area, and specific contributions will be discussed in following sections. Here is an outline of the introduction section. In section 1.1, we give a formal definition for IBR using the "plenoptic function". In section 1.2, we discuss the motivation for IBR research. In section 1.3, we talk about the two main research fields which inspired the recent new ideas in IBR. In section 1.4, we overview the structure of this paper.

1.1 Definition

Adelson and Bergen [1] assigned the name *plenoptic function* to the pencil of rays visible from any point in space, at any time, and over any range of wavelengths. They used this function to develop a taxonomy for evaluating models of low-level vision. The plenoptic function describes all of the radiant energy that can be perceived from the point of view of the observer rather than the point of view of the source.

Adelson and Bergen formalized this functional description by providing a parameter space over which the plenoptic function is valid, as shown in Figure 1. Imagine an idealized eye which we are free to place at any point in space (V_x, V_y, V_z). From there, we can select any of the viewable rays by choosing an azimuth and elevation angle (θ, ϕ) as well as a wavelength, λ, which we wish to consider. In the case of a dynamic scene, we can additionally choose the time, t, at which we wish to evaluate

the function. This results in the following form for the plenoptic function:

$$p = P(\theta, \phi, \lambda, V_x, V_y, V_z, t). \tag{1}$$

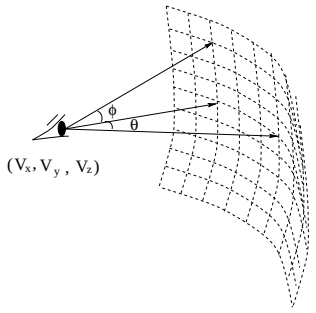

Figure 1: The plenoptic function describes all of the image information visible from a particular viewing position.

In computer graphics terminology, the plenoptic function describes the set of all possible environment maps for a given scene. For the purposes of visualization, one can consider the plenoptic function as a scene representation. In order to generate a view from a given point in a particular direction, we would need to merely plug in appropriate values for (V_x, V_y, V_z) and select from a range of (θ, ϕ) for some constant t.

We define a complete sample of the plenoptic function as a solid spherical map for an image-based rendering. Given a set of discrete samples based on specific values for the parameters from the plenoptic function, the goal of image-based rendering is to generate a continuous representation of that function.

1.2 Motivation

In traditional computer graphics, we normally represent the geometric and illumination information in a scene explicitly. For instance, set up lighting parameters for ambient and directional light, specify the material properties of objects in the scene, then use an algorithm to compute the visual effect. Recently, computer graphics researchers demonstrated that many of image-based modelling and rendering methods would allow us to replace some geometric and illumination information using an image space transformation or a light space representation.

Let's discuss some of the interesting aspect of image-based modelling and rendering by comparing two useful sources of information – *computer generated images* and *real video images*. Note that, we are using *real video images* as an example to motivate the idea of IBR. Information obtained using video cameras and other input devices will be further discussed later.

While computer graphics has made great strides towards increased realism in modelling shape and lighting effects, neither the models nor the hardware is close to the point where it can give convincing real-time images of environments comparable in visual complexity to the real world. For instance, many virtual reality systems

often compromise by displaying low quality images and/or simplified environments in order to meet the real-time display constraint.

Because of the inadequacy of the existing methods, researchers decided to explore new approaches which utilize *video captured images*. Real-world scenes contain enormously rich details. We want to be able to use real-world scenery directly without going through computer modelling and rendering.

Algorithms and techniques developed in recent IBR research allow us to to process and re-organize video captured data to build digital representations of real life objects or scenes.

Furthermore, since such techniques can also be applied to computer generated objects and scenes, representation techniques in image-based rendering have in effect blurred some differences between *computer generated scenes* and *video captured scenes*.

The advantage of IBR methods include the easy acquisition of models from images, certain computational requirements that are independent of scene complexity, and the rendering of some rich details or complicated illumination effect in the scene while avoiding the costs of physical simulation.

1.3 IBR vs. Graphics and Vision

Current IBR research draws its inspiration from research results in computer graphics and computer vision. Here, we will discuss how image-based modelling and rendering might fit into the overall picture of computer graphics and vision research.

1.3.1 IBR in Computer Graphics

Currently, the impact image-based modelling and rendering has in computer graphics is somewhat similar to that of introducing photogrammetry into cartography.

Let's have a brief look at the early history of photogrammetry. The first-known photographs were produced in 1839. In 1849, Aime Laussedat, an officer in the Engineer Corps of the French Army, embarked upon a determined effort to prove that photograph could be used with advantage in the preparation of topographic maps. He experimented with a glass-plate camera in the air, first supported by a string of kites and later by a captive balloon. He succeeded in developing a mathematical analysis for converting overlapping perspective views into orthographic projections on any plane. At the Paris Exposition in 1867, Laussedat publicly exhibited the first-known phototheodolite and also a plan of Paris based up his photographic surveys. This plan, or map, compared favorably with plans made earlier by ground-survey methods. As of today, photogrammetry is established as a basic procedure in all types of mapping. Moreover, the basic photogrammetry principles Col. Laussedat laid down and demonstrated by practical applications are still in use, not only in cartography, but also in other land surveying tasks, as well as computer vision research.

The influence of image-based modelling and rendering research have on computer graphics might have a similar flavor or a potentially more crucial impact in the future. We will comment on the issue concerning image quality in Section 5 after we review

the main image-based modelling and rendering methods.

1.3.2 IBR vs. Computer Vision

The current IBR research and the ongoing computer vision research appear to be somewhat orthogonal.

To some extent, this might be similar to the case of utilizing the digital camera technology. The technology itself has not yet fully matured. However, with the existing digital cameras, users have already been able to accomplish more and more creative tasks. The development and improvement of the tool is one thing, and what people use the existing tool to create is something else.

Part of the computer vision research is like Columbus or Vespucci's *exploration* expedition, searching for the land which is yet unknown to them. Whence one region is "discovered", the explores moved on to search for the next uncharted territory. Whereas, the IBR research might be similar to the rural/urban *development* process carried out by the early European settlement in North and South America. Given the known territory and the existing tools, people took advantage of raw material and natural resources locally. They used their ingenuity to develop and create new useful and/or artistic facilities, structure cities, invent telephones and airplanes for fast communication, etc. (figure 2). In some sense, the exploration expedition (vision) is a horizontal extension, and the rural/urban development process (IBR) is local and vertical growth.

To some extent, this difference in structure is determined by the different tasks, objectives and constraints in vision and IBR research problems. For instance, for a robot to navigate around an office environment, if we use a stereo matching algorithm which works for a large number of scenarios but does not work for several ambiguous cases, the robot might run into furnitures. Whereas, for computer graphics image-based rendering tasks, the same stereo algorithm which works for a large number of scenarios is good enough for generating many impressive results. Even if ambiguous cases arise, since the modelling process does not have to be done on the fly, we can take advantage of small amount of initial user input to guide the stereo matching algorithm to give us correct output.

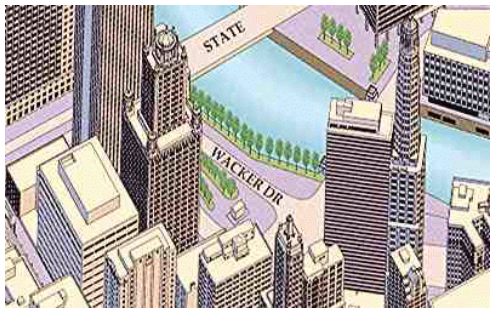

Figure 2: Vertical development: utilize the known resources and tools to create useful and/or artistic facilities.

In the author's opinion, the relationship between computer vision and IBR will

be similar to the relationship between physics and animation/rendering. Research in animation/rendering utilizes many physics principles, and might sometimes assist physics research. But the two areas might remain focusing on different issues, and hence might remain orthogonal to each other.

1.4 Overview

In the first part of section 2, we review several computer vision related research topics. These research topics are the fundamental building blocks in the recent IBR research. By discussing the existing approaches to these problems and the difficulties involved, we may have a better understanding of the similar challenges and limitations which we face in IBR research. In the second part of section 2, we will discuss issues related to input devices, because image acquisition is an important part of IBR. In section 3, we discuss the view interpolation problem in details, because it reveals many of the most basic ideas behind image-based rendering. In section 4, we survey the other image-based rendering applications and major results. In section 5, we summarize the paper, and discuss the possible future research directions and the potential limitations.

2 Background

The process of recovering 3D structure from 2D images has been a central endeavor within computer vision, and the process of rendering such recovered structures is a subject receiving increasing interest in computer graphics. Although no general technique exists to derive models from images, three particular areas of research have provided results that are applicable to the technical problems in image-based rendering. They are: Camera Calibration, Structure from Motion and Stereo Correspondence. We will discuss each of these in term in section 2.1. Also, in image-based modelling and rendering, *images* is the main source of information. Sometimes, we need to capture scenes or objects in the real world. There are a number of input device related issues which are worth noting. In section 2.2, we will discuss some of these issues which are important to image-based modelling and rendering.

2.1 Computer Vision Related Issues

2.1.1 Camera Calibration and Pose Estimation

Camera calibration and pose estimation can be thought of as two parts of a single process: determining a mapping between screen pixels and rays in the world, as shown in Figure 3(a). In this figure, C_{left} and C_{right} are the optical centers of the left camera and the right camera, respectively. The *principal point* of a camera is the foot of the perpendicular from the camera optical center to the image plane. The *baseline*, $C_{left}C_{right}$, is the line connecting the camera optical centers. The parameters associated with the camera calibration process naturally divide into two sets: extrinsic parameters, which define the camera's pose (a rigid rotation and translation), and intrinsic parameters, which define a mapping of 3D camera coordinates onto the screen. This latter mapping not only includes a perspective (pinhole) projection from

the 3D coordinates to undistorted image coordinates, but also a radial distortion transformation and a final translation and scaling into screen coordinates. Camera calibration is a well-studied problem both in photogrammetry and computer vision [10].

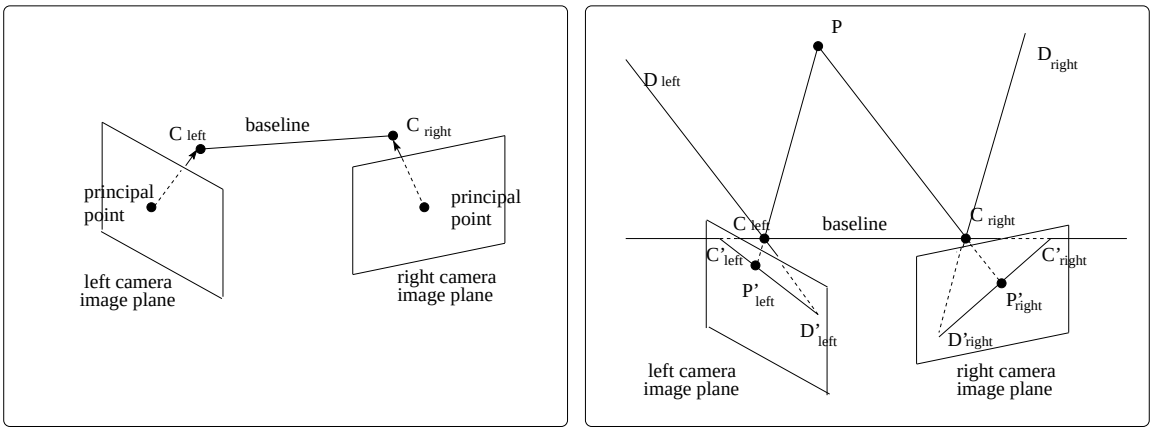

Figure 3: Camera calibration.

Since the traditional methods of camera calibration use known world coordinates, they are not suitable for some image-based rendering application where the 3D coordinates of points in the scene are unknown. To calibrate a camera using only feature coordinates in image plane, one must have more than a single image. It has been shown that we can calibrate the intrinsic parameters of a camera relatively accurately [10] [32]. Given the intrinsic parameters of the camera, the problem of recovering the extrinsic parameters of the camera is exactly the *Structure from Motion* problem which we will discuss in the next section.

2.1.2 Structure from Motion

In this section, we assume that the intrinsic parameters are known, but that the camera's motion is unknown.

Given the 2D projection of a point in the world, its position in 3D space could be anywhere on a ray extending out in a particular direction from that camera's optical center. This is demonstrated in Figure 3(b), where point P in the world is projected to P'_{left} and P'_{right} on the left and right image planes, respectively. C_{left} and C_{right} are the left and right camera optical centers. If we have only one image, say, the left image, then the position of P in space could be anywhere on the ray passing points P'_{left} and C_{left}.

When P and the camera optical centers C_{left} and C_{right} are not co-linear, they form a plane. The intersection of this plane with the left image plane is the line $C'_{left}P'_{left}$, where C'_{left} is the projection of C_{right} on the left image plane. A similar line $C'_{right}P'_{right}$ is obtained in the right image plane where C'_{right} is the projection of C_{left} on the right image plane. These lines, $C'_{left}P'_{left}$ and $C'_{right}P'_{right}$, are the *epipolar lines*. An object imaged on the epipolar line in the left image can only be imaged on

the corresponding epipolar line in the right image, if it is imaged at all. When the projections of a sufficient number of *tokens* in the world are observed in two or three images from different positions, it is theoretically possible to deduce the 3D locations of the points as well as the positions of the original cameras up to an unknown scale factor.

This problem has been studied in the area of photogrammetry for the principal purpose of producing topographic maps. The best understood case is when the *tokens* are points. In 1913, Kruppa proved the fundamental result that given two planar views of five distinct points, one could recover the rotation and translation between the two camera positions as well as the 3D locations of the points (up to a scale factor). The five-points algorithm is highly nonlinear since it requires solving for the roots of a tenth-degree polynomial. In 1981, Loguet-Higgins proposed an eight-point algorithm which reduces the complexity of the resolution method by increasing the number of points.

An alternative formulation of the problem uses lines rather than points as image measurements (i.e., *tokens*), and at least three views are needed. This case is much less well understood. Faugeras's book [10] overviews the state of the art as of 1992.

2.1.3 Stereo Correspondence

The geometrical theory of structure from motion assumes that one is able to solve the correspondence problem, which is to identify that points in two or more images that are projections of the same point in the world.

Years of research [10] have shown that determining stereo correspondences by computer is a difficult problem. In general, current methods are successful only when the images are similar in appearance, as in the case of human vision, which is usually obtained by using cameras that are closely spaced relative to the objects in the scene. When the distance between the cameras (often called the baseline) becomes large, surfaces in the images exhibit different degrees of foreshortening, objects have different patterns of occlusion, and corresponding points have large disparities in their locations in the two images, all of which makes it much more difficult to determine correct stereo correspondences. Unfortunately, the alternative of improving stereo correspondence by using images taken from nearby locations has the disadvantage that computing depth becomes very sensitive to noise in image measurements.

2.2 Input Device Related Issues

2.2.1 Recovering High Dynamic Range from Photographs

When we photograph a scene, either with film or an electronic imaging array, and digitize the photograph to obtain a two-dimensional array of "brightness" value, these values are rarely true measurements of relative radiance in the scene. For example, if one pixel has twice the value of another, it is unlikely that it observed twice the radiance. Instead, there is usually an unknown, nonlinear mapping that determines how radiance in the scene becomes pixel values in the image.

The area of image-based modelling and rendering is working toward recovering more advanced reflection models of the surface in the scene. These methods, which involve observing surface radiance in various directions under various lighting conditions, require absolute radiance values rather than the nonlinearly mapped pixel values found in the conventional images. Just as important, the recovery of high dynamic range images will allow these methods to obtain accurate radiance values from surfaces specularities and from incident light sources. Cameras (whether analog or digital) tend to have a very limited dynamic range. In some cases, we might need to use a number of photographs taken at different exposures to recover the high dynamic range of the scene.

Mann and Picard [25] proposed a means of combining differently exposed pictures to obtain a single picture of extended dynamic range, and improved color fidelity. They proposed a simple algorithm for finding the pointwise nonlinearity, f, of the image capturing and digitization process that maps the light q projected on a point in the image plane to the pointwise value in the picture, $f(q)$, up to a scale factor. The plot of the mapping $f(q)$ gives us the shape of the response curve of the imaging process. They use the response curve to combine an overexposed and underexposed photograph of the same scene to construct a new picture of extended dynamic range. Debevec and Malik [7] extended this idea by proposing a more elaborate algorithm for recovering the response function of the imaging process. Also, their treatment of of image noise is more detailed.

2.2.2 Range Scanners

In image-based modelling and rendering, we may consider range scanners as an alternative type of input device. A *range scanner* is a device which can acquire a two-dimensional grid or image of depth measurements as measured from a single viewpoint or from a projection plane. The two-dimensional grid of depth is called a *range image*. There are two major types of commercially available range scanners – laser range finders and structured light scanners. There are other variation of range scanners which might be a compromise of the types of scanners we describe here.

A laser range scanner uses a single optical path and computes depth via the phase shift or time delay of a reflected laser beam. The intensities recorded are the amplitudes of the portion of the beam reflected back towards the camera. The laser range finders use light wavelengths far outside the visible spectrum. Highly reflective surfaces can cause problems. This is because not enough of the outgoing beam may scatter in the direction back towards the range camera upon impact with the object surface.

A structured light scanner uses two optical paths, one for a charge coupled device (CCD) and one for projected light. A structured light scanner computes depth via triangulation. It captures a form of intensity image in the course of its normal operation. One primary operating concern of a structured light range scanner is to set a threshold which will discriminate between lighted and unlighted areas in the images captured by the CCD. The factors contributing to the thresholding process includes the amount of ambient light, the power of the projected light, the setting of

the aperture on the CCD, the reflective properties of the surfaces being imaged, and the setting of the threshold. Since there are two optical paths, occlusion may occur. Parts of the scene visible to the CCD may not be visible to the light projector. The resulting pixels in the range image, called *shadow pixels*, do not contain valid range measurements.

During image acquisition, both of these types of range scanners must remain motionless and be imaging motionless objects, because the raster of range measurements is not measured simultaneously.

A laser range finder can give us additional information about the shape of objects or the scene. A structured light range scanner can provide both the $2\frac{1}{2}D$ depth and some image color information. Each type of scanners has its advantages and disadvantages. For instance, it is difficult to a structured light range scanner to scan sculptures at historic sites under sunlight, because the strong ambient light might wash out the light-and-dark pattern projected by the light projector. Whereas, a laser range finder might operate just fine under this condition. However, laser range finder might not be suitable for scanning a person's face, whereas a structured light range scanner is better for such task.

2.3 Summary

In this section, we have discussed three particular areas of computer vision research which have provided results that are applicable to the technical problems in image-based rendering. This gives us the basic vocabulary in describing the different image-based rendering techniques and analyzing their advantages and limitations. Also, it helps us to understand the relationships and differences between them. In addition, we discussed a number of input device related issues. We will this background information in the applications in section 4. In the next section, we discuss one kind of image-based rendering, namely view interpolation, in detail. This will demonstrate a typical application of the techniques we discussed above in image-based rendering.

3 View Interpolation

3.1 Definition

View interpolation is a view synthesis technique where new views of a scene can be expressed as combinations of other views of the same scene. Traditional methods which are able to capture a real object and render it from an arbitrary viewpoint usually use a 3D model of the object. However, view interpolation methods render an object from an arbitrary view point without having its 3D model. Given that a set of reference views covering the whole visible surface of the object is captured, reference views can be accessed directly. What is demanded are the intermediate views. If the correspondence of the reference views is available, it is possible to obtain a new view as a composition of a subset of the reference views close to it.

3.2 Basic Problems

Given the definition of view interpolation, we can see that the following problems must be solved:

1. Given the reference views, how to find the correspondences between them.
2. Knowing the positions and the intensities of corresponding points in reference views, how to predict the position and the intensity of a point in the new view.
3. How to determine the visibility of points in the new view.
4. How to find the optimal set of necessary reference views.

In the next few sections, we will discuss the above problems based on solutions proposed in previous view interpolation research results.

3.2.1 Correspondence Acquisition

We need to find the correspondence for the subsets of the reference views from which the new views are to be constructed. In other words, we are looking for all n-tuples $[\mathbf{x}_1, ..., \mathbf{x}_n]$ of pixels coordinates, where each component x_i is from the i^{th} view, so that every n-tuple contains projections of a single point on the object's surface, without considering errors caused by discretization and noise. Stereo correspondence techniques are used here. We have discussed stereo correspondence in section 2.1.3.

In general, a complete correspondence is impossible to obtain automatically. For instance, several tasks in 3D computer vision are complicated by the *aperture problem*. The *aperture problem* arises due to uniformly colored surfaces in the scene. In the absence of strong lighting effects, a uniform surface in the scene appears nearly uniform in projection. Although it is possible to determine which uniform regions correspond in different images, it is impossible to determine correspondences *within* these regions. As a result, additional smoothness assumptions are needed to solve problems such as optical flow and stereo vision.

Seitz and Dyer [28] showed that for a broad class of scenes and views under a monotonic visibility constraint, the image-based view synthesis is in fact well-posed problem and is not affected by the aperture problem. We will explain this idea here in detail.

The monotonicity condition imposes a strong visibility constraint on the scene. Consider the projections of a set of uniform surfaces into images I_1 and I_2, where each surface is *uniformly colored*, but any two surfaces may have different colors. Figure 4 depicts the cross sections S_1, S_2 and S_3 of three surface surfaces projecting to lines l_1 and l_2 in images I_1 and I_2, respectively . Each connected cross section projects on a *uniform interval* of l_1 and l_2. The *monotonicity* constraint induces a correspondence between the endpoints of the intervals in l_1 and l_2, determined by their relative ordering. The points on S_1, S_2 and S_3 projecting to the interval endpoints are referred to as *visible endpoints* of S_1, S_2 and S_3.

Now consider an in-between view with image $I_{1.5}$, and line $l_{1.5}$ corresponding to l_1 and l_2. S_1, S_2 and S_3 project to a set of uniform intervals of $l_{1.5}$, delimited by the projections of the visible endpoints of S_1, S_2 and S_3. Monotonicity is needed to ensure that the endpoints of each uniform interval in $I_{1.5}$ correspond to the visible endpoints of S_1, S_2 and S_3. Notice that $I_{1.5}$ does not depend on the specific shape of surfaces in the scene, only on the position of the visible endpoints of their cross sections. Any number of distinct scenes could have produced I_1 and I_2, but each one would also project to $I_{1.5}$. Because correspondence or shape information within *uniformly colored* regions is not necessary to predict in-between views, the aperture problem is avoided. An example of Seitz and Dyer's result is shown in Figure 5.

Because monotonicity is needed for view interpolation, this condition limits the set of views that can be interpolated. Nevertheless, monotonicity is satisfied at least locally for a wide range of interesting scenes.

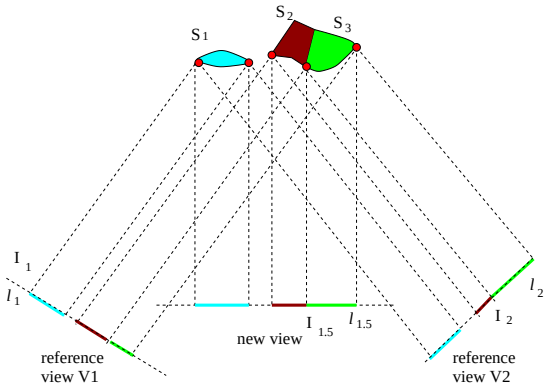

Figure 4: Correspondence Under Monotonicity. Top view of projection of three surface cross-section into corresponding lines of images I_1, I_2 and $I_{1.5}$. Although the projected intervals in I_1 and I_2 do not provide enough information to reconstruct S_1, S_2 and S_3, they are sufficient to predict the appearance of $I_{1.5}$.

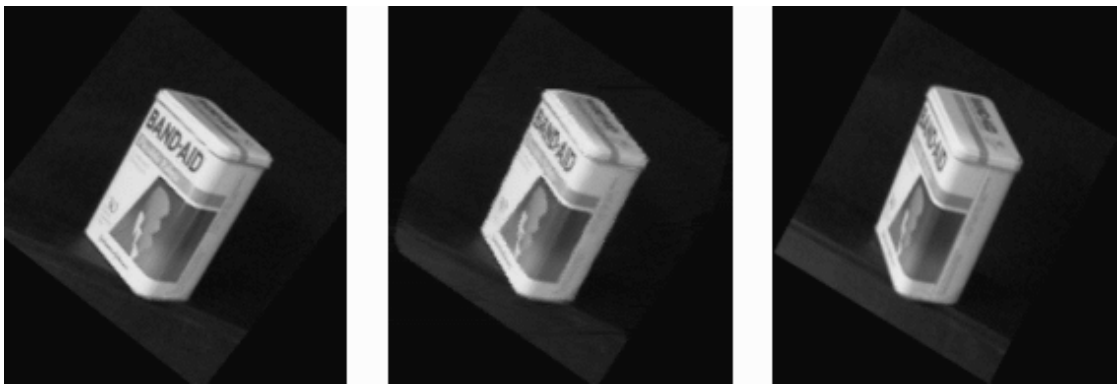

Figure 5: Left: reference view 1; Right: reference view 2; Middle: new view.

3.2.2 Prediction of Positions and Intensity of New Points

Knowing the positions and the intensities of corresponding points in reference views, we want to predict the position and the intensity of a point in the new view. Camera calibration plays an important role here. The new view corresponds to the view of a virtual camera. Knowing the camera parameters of reference views helps us solve this problem.

Ullman and Basri [34] demonstrated that new views can be expressed as linear combinations of other views of the same scene. The modelling of objects using linear combinations of images is based on the following observation. For many continuous transformations of interest in recognition, such as 3D rotation, translation, and scaling, all the possible views of the transforming object can be expressed simply as the linear combination of other views of the same object. The coefficients of these linear combinations often follow in addition to certain functional restrictions. Although the focus of Ullman and Basri's work was recognition, it has clear ramifications for view synthesis, providing a simple mechanism for predicting the positions of features in new views. However, their work does not take into account visibility issues that are crucial to understanding which views can be synthesized.

Seitz and Dyer [28] showed that a transformation can be made to interpolate gaze direction and to generate valid in-between views by first aligning the coordinate axes of the two views. This is accomplished by means of a simple image rectification procedure that aligns epipolar lines in the two images. The result of rectification is that corresponding points in the two rectified images will appear in the same scanline. Then the uniform intervals of corresponding scanlines in the two rectified images are matched. For each scanline, they linearly interpolate positions and intensities of corresponding intervals. Note that this automatically takes care of areas of (almost) constant intensity, as we discussed in the previous section. Although image interpolation is not always physically valid, interpolation of rectified monotonic image always produces valid in-between views of a scene.

Laveau and Faugeras [19] studied *weakly calibrated* views. A stereo rig is said to be *weakly calibrated* if only the *fundamental matrix* is known. To define *fundamental matrix* intuitively, we will again refer to the setup shown in Figure 3(b) where two cameras are looking at the same scene. Recall that the point P is projected to point P'_{left} on the left image plane, and the corresponding *epipolar line* on the right image plane is $C'_{right}P'_{right}$. The *fundamental matrix* is the 3×3 matrix which maps the point P'_{left} to its corresponding epipolar line $C'_{right}P'_{right}$ in image coordinates. Interested readers are referred to Appendix A for a more detailed explanation about the fundamental matrix. Laveau and Faugeras [19] showed that if the views are *weakly calibrated*, and $\mathbf{x_1}$ and $\mathbf{x_2}$ are the projections of a 3D scene point $\mathbf{X}$ in two reference views, then the corresponding point $\mathbf{x}$ in the new view is obtained as an intersection of epipolar lines associated with $\mathbf{x_1}$ and $\mathbf{x_2}$. Here the problem occurs if x lies on the intersection of the projection plane of the new view and the trifocal plane, i.e., the two epipolars are parallel. In fact, if the epipolars are nearly parallel, x can not be determined using this method.

The alternative approach is to use algebraic functions of views **??**, which are al-

gebraic relations among image coordinates of projections of a single scene point in different views (e.g., the epipolar constraint is an algebraic function of two views). For perspective views, the image coordinates (x, y), (x', y') and (x'', y'') of three corresponding points across three perspective views satisfy trilinear equations. Given the corresponding points (x, y) and (x', y') in the first two views in homogeneous image coordinate system, (x'', y'') can be uniquely determined. For orthographics views, the three views are connected via a bilinear function.

Note that when we interpolate between two reference images, we need to find corresponding points in the two images. If the positions of the cameras are known, this is equivalent to finding the depth values of the corresponding points in the camera coordinate system. Automatically finding correspondences between pairs of image is the classic problem of stereo vision, and unfortunately although many algorithms exist, these algorithms are fairly fragile and may not always find the correct correspondences.

Chen and Williams' [3] view interpolation approach requires a depth value for each pixel in the reference image, which is easily provided if the reference images are synthetic images. Given the depth value it is possible to reproject points in the image from different vantage points to map between multiple images. This approach employs incomplete plenoptic samples and image flow fields to reconstruct arbitrary viewpoints with some constraints on gaze angle. The correspondence between images is predetermined automatically using the range data associated with the images. They used pre-rendered synthetic images to determine flow fields from the z-values. To generate an in-between view of a pair of images, the offset vectors are interpolated linearly and the pixels in the source image are moved by the interpolated vector to their destinations. This locally linear approximation is nicely exploited to approximate perspective depth effects, and Chen and Williams show it to be correct for lateral motions relative to the gaze direction.

Establishing flow fields for a view interpolation system can be problematic. In general, accurate flow field information between two samples can only be established for points that are mutually visible to both samples. This points out a shortcoming in the use of partial samples, because reference image seldom have a 100% overlap.

Werner et al. [36] showed an approximation method where almost no calibration of views is needed. Assuming the proximity of the views, the interpolated view can be expressed as a linear combination of the reference views.

3.3 Visibility of Points in the New View

Chen and Williams' [3] approach assumes that the depth of every point in an image is known. The pixel blocks are sorted once by their Z-coordinates and subsequently displayed from back to front to eliminate the overhead of a Z-buffer for visibility determination. Since their method uses the camera transformation and image range data to automatically determine the correspondence between two or more images, the correspondence is in the form of a "forward mapping". A forward mapping process computes destination pixels by accumulating successive source pixel contributions. The key problem with forward mapping is that overlaps and holes may occur in the

interpolated image. Then a inverse mapping is used to handle overlaps and holes. Inverse mapping determines the subset of source pixels that contribute to a given destination pixel. To resolve the overlay problem, Chen and William use a view-independent visible priority method to make sure that pixels are ordered from back to front, then determine the front most pixel for the overlapping pixel area. The holes can be filled using a number of techniques, such as interpolating the color pixels surrounding the hole and using more reference images.

Werner et al. [36] listed the six possible situations which could arise in determining the visibility of interpolated points in the new view from the reference views. They used epipolar geometry to help determining the visibility of the new point.

Laveau and Faugeras [19] uses an ray-tracing like algorithm to disambiguate the visibility of the new point.

3.4 Optimal set of Necessary Reference Views

The question of the choice of the smallest and still sufficient set of the reference views is non-trivial. The more restricted fundamental problem of how to choose a set of so-called *characteristic* views (i.e., the minimum set of views in which all points of a given surface are visible) still remains unsolved for general non-convex objects. As we will see in later sections, other image-based rendering applications, such as light field rendering and panoramic mosaicing, choose to avoid this issue by having a reasonable sampling frequency for all places on the surface. On the one hand, it is possible that the study of optimal set of necessary reference views will lead to better compression for other image-based rendering methods in certain instances. On the other hand, when we are dealing with complex real world scenes, the visibility issues and the complicated unknown surface properties of certain objects might render such effort useless.

4 Image-Based Object and Scene Modelling and Rendering

4.1 Light Field Rendering and the Lumigraph

Image based modelling and rendering has been presented in its purest form in the Light Filed Rendering [20] and the Lumigraph [17] work. In some ways, this has very much the flavor of "What you see is what you get". The flip side is, of course, "What you see is all you get". Chen's [4] object movie making technique is similar to [20] and [17], and its representation and reconstruction are relatively more primitive.

The major idea behind the light field type approaches is a representation of the *light field*, the radiance as a function of position and direction, in regions of space free of occluders (free space). In free space, the light field is a 4D function. An image is a two dimensional slice of the 4D light field. Creating a light field from a set of images corresponds to inserting each 2D slice into the 4D light field representation. Similarly, generating new views corresponds to extracting and resampling a slice.

There are several major challenges to using the light field approach to view 3D scenes on a graphics workstation — representation of the light field, acquisition of data, generation of new views, compression of the information. In the next few sections, we discuss these issues.

4.1.1 Representation

We can use the *plenoptic function* to represent the light field as the radiance at a point in a given direction.

If we consider only a snapshot of the plenoptic function and use only a monochromatic function (in practice three discrete color channels) instead of wavelength, the plenoptic function is reduced to a function of five variables representing position and direction. In free space, at any point in space, one can determine the radiance along any ray in any direction, by tracing backwards along that ray through empty space to the surface of the object. Thus, the plenoptic function due to the object can be reduced to 4 dimensions. In both [20] and [17], the authors use the surface of a cube to hold all the radiance information due to the enclosed object as shown in Figure 6(a). For each cube face, the direction is parameterized using a second plane which is parallel to the first plane. The lines intersecting two planes are used to define the directions of the light rays, as shown in Figure 6(b). The two planes form a *light slab*.

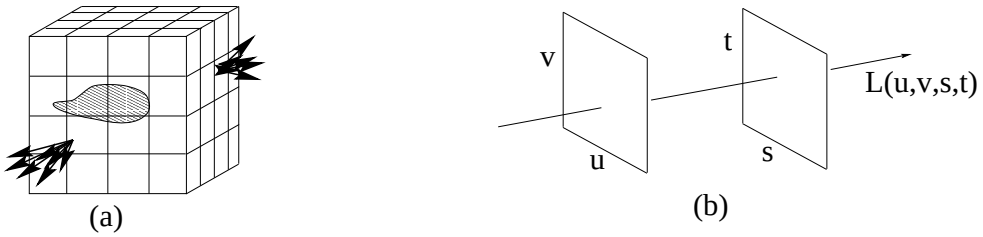

Figure 6: Define the light ray using a line which intersects two planes.

4.1.2 Generation or Acquisition the Light Field

For a virtual environment, a light slab is easily generated simply by rendering a 2D array of images. Each image represents a slice of the 4D light field at a fixed uv value and is formed by placing the center of projection of the virtual camera at the sample location on the uv plane.

To make light field from video captured images, an image capturing step is needed.

In [17], Gortler et al. take an inexpensive approach by moving a hand held camera through the scene, populating the field from the resulting images. As a result, the sample points in the domain can not be pre-specified or controlled. Each pixel in the input video stream coming from the hand-held camera represents a single sample. The number of sample points is reported to be approximately 10^8. Constructing a Lumigraph from these samples is similar to the problem of multidimensional scattered data approximation. Gortler et al. used a hierarchical push-and-pull algorithm to construct the light field from the scattered data. Also, they extracted the geometric

information from the capturing process to do a depth correction on the light field representation.

In [20], Levoy and Hanrahan design the image capturing process differently. They built a computer-controlled camera gantry and digitized images on a regular grid. So the light field capturing process is controlled, and all the camera parameters are known. The number of sample points for the captured object ranges from $50M$ for the buddha example to $1608M$ for the hallway example.

Since Levoy and Hanrahan do not use the same type of demo objects as Gortler et al. in the Lumigraph approach, we can not draw any conclusion about the relative quality of the light fields constructed using these two different capturing process. Hence, it is difficult to judge their choice of sample size. The acquisition method used by Gortler et al. is flexible and inexpensive. It is good for capturing objects or scenes which requires moderate sample size. But when the required sample size is too big, automated capturing process have obvious advantages. The capturing process used by Levoy and Hanrahan is automated. It is relatively costly and time consuming to set up.

4.1.3 Reconstruction of Images

Given a desired camera setting (position, orientation and resolution), the reconstruction step sets each pixel of the output image with the color that this camera would create if it were pointed at the real object. Given the light field representation, one may generate a new image from an arbitrary camera pixel by pixel, ray by ray, using ray tracing. The expense of tracing a ray for each pixel can be avoided by reconstructing images using texture mapping operations. The resampling process is approximated by interpolating the 4D function from the nearby samples. In [17], Gortler et al. also take the depth correction information into account.

4.1.4 Compression

In choosing compression/decompression scheme for the light field representation, a number of factors should be taken into account: data redundancy, random access, compression/decompression asymmetry and computational expenses. In [20], Levoy and Hanrahan use vector quantization followed by entropy coding to do the compression. So the decompression process occurs in two stages. The first stage is gzip decoding, which is the reverse of entropy coding. After this stage, the light field is still compressed by vector quantization, but it is represented in a way that supports random access.

In [17], Gortler et al. suggest applying "a transform code to the 4D array, such as a wavelet transform *or* block DCT". They speculate on a number of possibilities, but they do not specify exactly which compression method they do use.

4.1.5 Uniformly Sampled Light Field

The two plane parameterization of light field which we discussed in section 4.1.1 show noticeable artifacts when the image being rendered uses samples from more than one

light slab. The density of lines induced by the two-plane parameterization is biased towards certain directions and the spatial sampling induced by the representation is different for each direction.

Camahort et al. [2] proposed two techniques for uniformly sampling the light field. These techniques are based on the two-sphere parameterization (2SP) and the sphere-plane parameterization (SSP) as shown in Figure 7. To choose sample directions, they started with a icosahedron as an approximation to an object's bounding sphere, then quad subdivision is used to further refine the approximation. Quad subdivision is as shown in Figure 8. This way, the polyhedra surface is subdivided into k patches. In the 2SP representation, the light field rays are determined by each ordered pair of patches, as shown in Figure 7(a). In the SSP representation, each patch represent a plane Q_k through the bounding sphere's origin with normal defined by the direction vector $\overrightarrow{\omega}_k = \overrightarrow{Q}_c - \overrightarrow{O}$, where Q_c is the patch centroid and O is the sphere origin, as shown in Figure 7(b). Given a directional sample $\overrightarrow{\omega}_k$, they select a set of light field samples in a (u, v) coordinate system local to Q_k. This is done for synthetic images by performing an orthographic projection of the object along direction $\overrightarrow{\omega}_k$ onto the plane Q_k. Both 2SP and SSP use hierarchical representation. The 2SP representation is compressed using vector quantization and Lempel-Ziv coding. The SSP representation is compressed using JPEG for the image data, and it uses Lempel-Ziv and Welch for lossless compression for its depth maps.

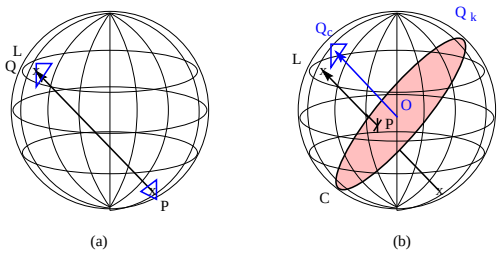

Figure 7: Object bounding sphere and sampled light rays: (a) select two random points P and Q uniformly distributed on the sphere's surface and join them with a line L; (b) select a random great circle C with uniform probability, select a random point P uniformly distributed over C's surface, and choose the normal L to C passing through P.

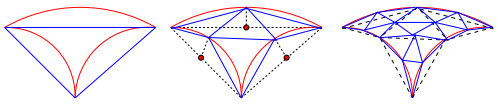

Figure 8: Quad subdivision.

Other than the uniform sampling feature, their hierarchical representations also enables progressive transmission and progressive rendering, smooth transition between different level of details and adaptive frame rate control. The disadvantage is that the rendering of their models takes two or three times longer than a single light slab of the parallel plane representation in section 4.1.1.

4.1.6 Light Field Type Rendering vs. View Interpolation

View interpolation would do exactly what light field rendering does if we were to make the reference camera locations sufficiently dense to cover a sphere surrounding the object to be rendered. To some extent, the light field representation is just a way of re-organizing and re-indexing the light ray information extracted from the reference images which might be used for view interpolation.

We can use the 2D case to demonstrate their connection. For instance, in the situation as shown in Figure 9, the new camera location C_{new} is on the camera line between two neighboring reference camera locations C_1 and C_2. In this case, the 2D view interpolation task is exactly the same as that of 2D light field rendering task. In other words, the light rays collected by reference cameras 1 and 2 for view interpolation are exactly the sample light rays passing through the camera line sample locations $u = C_1$ or $u = C_2$ which are used for light field interpolation.

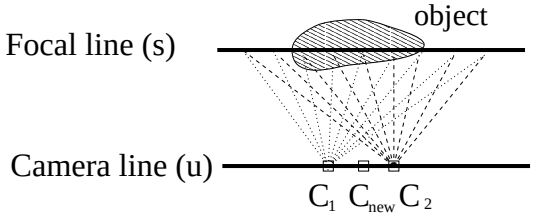

Figure 9: Similarity between light field rendering and view interpolation (example 1).

For the situation as shown in Figure 10 where the new camera location C_{new} is not on the camera line. The light field interpolation task is equivalent to a view interpolation problem which uses a number of reference images of the object captured at several sample camera locations on the camera line as shown in Figure 10.

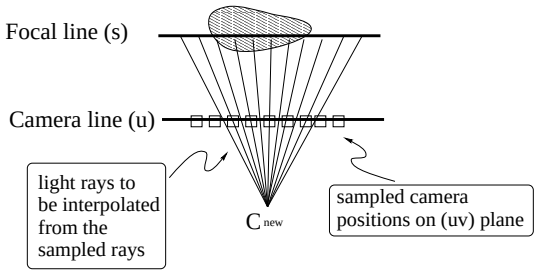

Figure 10: Similarity between light field rendering and view interpolation (example 2).

In view interpolation, the question of the choice of the smallest and still sufficient set of the reference views is non-trivial as we discussed in section 3.4. It remains unsolved for general non-convex objects. Light field rendering "ignores" this question by choosing a large sampling frequency. This is reasonable, especially when we are dealing with objects such as the toy lion in the Lumigraph paper [17] where its surface property is considerably complex. Also, in [20], the authors took advantage of the

large sampling frequency to avoid depth computation, and were still able to obtain reasonable results. The problem of visibility poses constraints on the quality of image produced by light field rendering as well as by view interpolation.

4.2 Light Field Rendering and Lumigraph Related Work

4.2.1 Light Field under Different Lighting Conditions

Wong et al. [38] experimented with extending the light field and lumigraph work by taking into account the change of lighting condition. They tested their idea in virtual environment using direct lighting. They did not try to recover or use any geometrical information (e.g., depth or surface normals) to calculate the illumination. The scene was sampled from different viewpoints and under different illuminations. The sampling process was set up as shown in Figure 11.

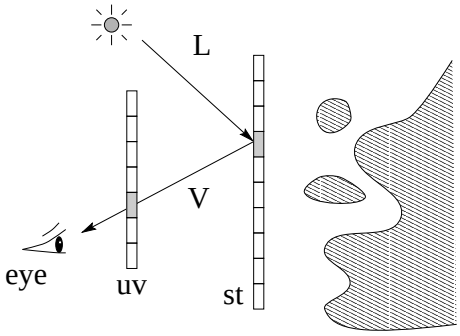

Figure 11: Measuring the BRDF of the pixel under different direct lighting conditions.

The back plane (st) and the front plane (uv) forms a *light slab*. A *view* is an image of the st plane. Each *pixel* on the st plane was treated as a surface element with an *apparent* bidirectional reflectance distribution function (BRDF). Each surface element emits different amounts of radiant energy in different directions under different illuminations. By recording the (apparent) BRDF of each st pixel, the aggregate reflectance of objects visible through that pixel window is captured. The tabular BRDF data of each pixel is transformed to the spherical harmonic domain for efficient storage. Whenever the user changes the illumination setting, the tabular BRDF data for different lighting conditions would be used to reconstruct different views.

This is a preliminary result. In their approach, Wong et al. did not try to recover or use any geometrical information. This will introduce incorrectness in the scene reconstruction when the new view is interpolated from the sampled views. An example of such situation is shown in Figure 12. The sampling process would completely miss the sphere in the scene. The spherical object might be the light source or a shiny surface with bright reflectance highlight under certain illumination conditions. This is one of the main reasons why the light field rendering and lumigraph work chose to deal with free space, i.e., regions of space free of occluders.

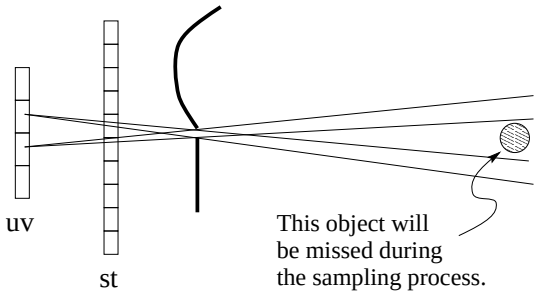

Figure 12: An example where the sampling process misses an object in the scene.

4.2.2 Light Field Application in Global Illumination

Fournier et al. [12], van Liere [22] and Lewis et al. [21] investigated a technique for solving global illumination problem using light field type of method. The proposed technique has a divide-and-conquer flavor. The scene was broken into small regions. Each region is treated as a lumigraph, where the boundaries of neighboring regions are used as virtual dividers and bounding boxes. There are two basic element to the global illumination calculation: (a) In each region, given the incoming light energy, its local illumination is calculated; (b) Then radiance distributions along the boundaries between regions are transferred. Processes (a) and (b) are repeated until the light energy transfer between regions gets below a threshold level.

The main problem with this kind of approach is again visibility. If we treat each small region as a black box without considering the geometry of objects inside it, when we sample the light transfer at the region boundaries, situations such as the one shown in Figure 12 might occur. In order to eliminate such problem, we have consider the geometry of objects inside each region. Moreover, we need to choose the sample points on a virtual boundary depending on the geometry of objects on both sides of this boundary. Therefore, the amount of computation involved is equivalent to doing a global visibility calculation. In terms of computational complexity, such an approach will not be faster than traditional global illumination methods, because global visibility check is the bottleneck in traditional techniques.

4.2.3 Dynamic Scene Created from Multiple Views

Since we can model and render a still scene using images from multiple views, a natural next step is to create visual events using similar techniques. Kanade et al. [18] captured visual events using many cameras that cover the action from all sides. The 3D structure of the event is computed for a few selected directions using a stereo technique. Triangulation and texture mapping enable the placement of a virtual camera to reconstruct the event from any new viewpoint.

There are many problems which need to be resolved with this approach Kanade et al. proposed. For instance, when the sequence of actions is not repeatable, it needs to be captured from different view points simultaneously. An example Kanade et al. give in their paper is a person in the scene playing basketball, as shown in Figure 13. The sequence of actions performed by this person is unique, and it can not be

repeated exactly at a later time. Given the limited number of cameras and camera positions (i.e., viewpoints), the new view constructed using the interpolation of the reference views appears to be distorted. As we can see from the sequence of frames from a new view (Figure 13(b)) constructed using the reference views, the head of the person and the basketball are distorted. This takes us back to the problem we discussed in section 3.4, i.e., that the choice of the smallest and still sufficient set of the reference views is non-trivial.

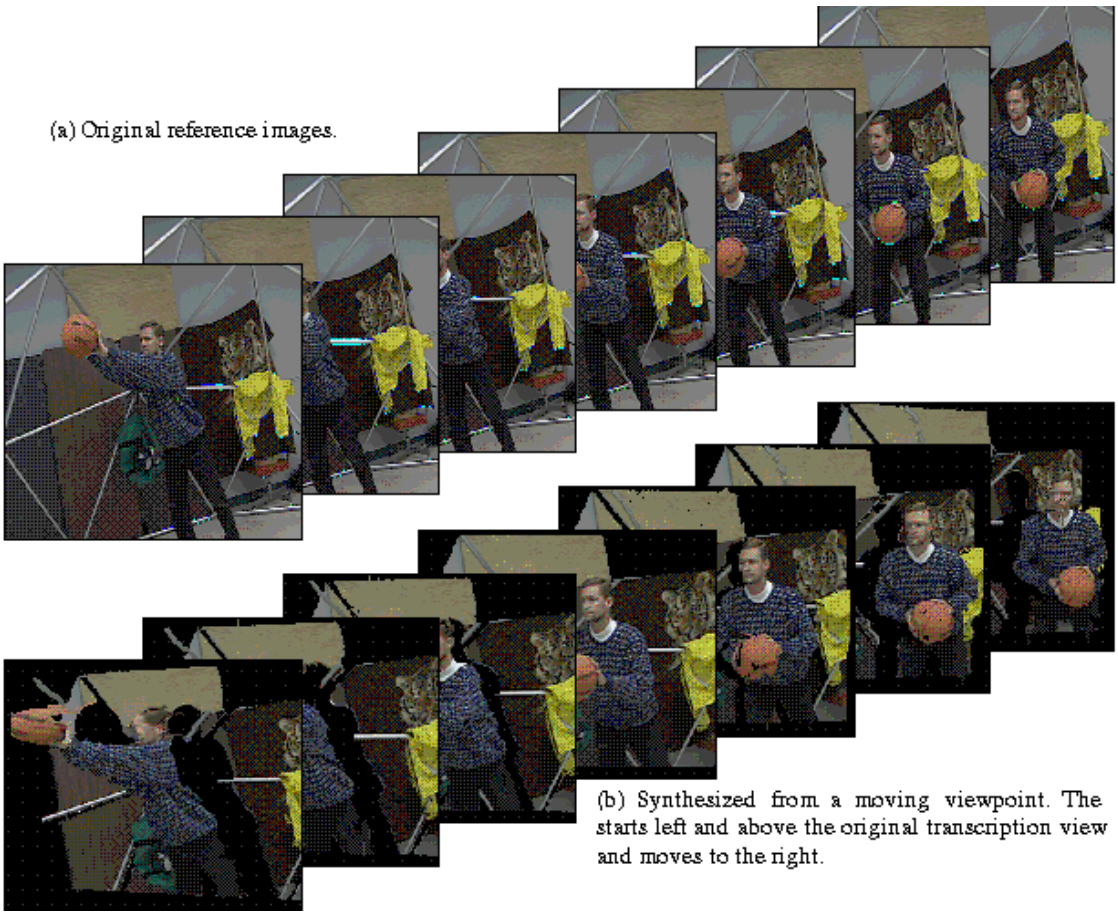

Figure 13: Seven frames of a basketball sequence.

4.3 Panoramic Mosaic Images Construction

A number of techniques have been developed for capturing panoramic images of real-world scenes. One way is to record an image onto a long film strip using a panoramic camera to directly capture a cylindrical panoramic image. Another way is to use a lens with a very large field of view such as a fisheye lens. Mirrored pyramids and parabolic mirrors can also be used to directly capture panoramic images.

A less hardware-intensive method for constructing full view panoramas is to take many regular photographic or video images in order to cover the whole viewing space. These images must then be aligned and composited into complete panoramic images

using an image mosaic or "stitching" algorithm [24] [4] [26] [33]. Most stitching systems require a carefully controlled camera motion (pure pan), and only produce cylindrical images [4] [26]. Uncontrolled 3D camera rotation can be used. The case of general camera rotation has been studied in [24] using an 8-parameter planar perspective motion model. In [33], Szeliski and Shum proposed an algorithm which uses a 3-parameter rotational motion model. Once a mosaic has been constructed, it can be mapped into cylindrical or spherical coordinates, and displayed using a special purpose viewer [4]. In [33], a mosaic is converted to an environment map [15], and they showed how to map a mosaic onto any texture-mapped polyhedron surrounding the origin.

If only the view direction is changing and the viewpoint is stationary, as in the case of pivoting a camera about its nodal point (i.e. the optical center of projection), panoramic mosaic images can be constructed using the techniques described above.

When the view point starts moving, the movement of the viewpoint causes "disparity" between different views of the objects in the scene. The disparity is a result of depth change in the image space when the viewpoint moves. Because of this disparity, we need to obtain panoramic mosaic images at different viewpoints, so that the images from an arbitrary viewpoint can be constructed from the reference images [4] [26]. The main processing steps of such techniques are similar to that of the light field rendering and lumigraph work, i.e., choosing and defining a representation, acquiring and generating the panoramic mosaic images, constructing new views from reference images.

4.4 Rendering of Architecture from Photographs

In the different image-based rendering application we have discussed so far, researchers have mostly avoided the use of traditional modelling steps. However, for certain types of IBR application, recovering some basic geometric information using traditional modelling method will allow us to achieve better quality rendering results with less reference images. Debevec et al. [6] exploited the constraints which are characteristic of architectural scenes. They developed a hybrid geometry- and image-based approach to model and render architecture from photographs.

In their approach, the scene is represented as a constrained hierarchical model of parametric polyhedral primitives, called *blocks*. To ensure the geometric model is consistent with the real life model of the architecture, they use a reconstruction algorithm to improve the geometric model, i.e., minimizing the disparity between the projected edge of the model and its corresponding edge in the reference images. View-dependent texture mapping is used to project the original photographs onto the model. This approach to modelling and rendering architecture requires only a sparse set of photographs.

This is a good example where one can take advantage of the property of the objects we wish to render, and tailor the image-based rendering technique towards the specific application. Debevec et al.'s approach is effective for objects which can be constructed using simple geometric primitives. If the contour and shape of an object are more complex, the basic photogrammetric modelling step might become

much more demanding. In section 5.1, we will further discuss the use of geometric models in image-based rendering.

4.5 Rendering Synthetic Objects into Real Scenes

In [13], Fournier et al. present a technique for approximating the common global illumination for real video images and computer-generated images, assuming some elements of the scene geometry of the real world and common viewing parameters are known. They use the real image as a projection of the exact solution for the global illumination in the real world. The objects in the video captured scene are replaced by few boxes covering them. The image intensity of the real video images is used as the initial surface radiosity of the visible part of the boxes. The surface reflectance of the boxes is approximated by subtracting an estimate of the illumination intensity based on the concept of ambient light. Then, they use radiosity computation to render the surface of the computer-generated object with respect to this new environment and for calculating the amount of image intensity correction needed for surfaces of the real image.

Drettakis et al. [9] extend the approach in [13] by adding a method for performing fast updates of the illumination solution in the case of moving objects.

In [8], Debevec presents a general method that uses measured scene radiance and global illumination in order to add new objects to light-based models with correct lighting. He uses a light probe to measure the incident illumination at the location of the synthetic objects. The probe is a spherical first-surface mirror, such as a polished steel ball. The high dynamic range radiance map of the environment is recovered using photographs of the light probe at different exposure. This radiance map, rather than synthetic light sources, is used to illuminate the new objects. To compute the illumination, the scene is considered as three components: the distant scene, the local scene, and the synthetic objects. The distant scene is assumed to be photometrically unaffected by the objects obviating the need for reflectance model information. The local scene is endowed with estimated reflectance model information, so that it can catch shadows and receive reflected light from the new objects. Renderings are created with a standard global illumination method by simulating the interaction of light amongst the three components.

In [8], the use of a light probe to measure the the incident illumination at the location of the synthetic objects is similar to Greene's idea [15] of capturing the environment map at the location of a synthetic object. This incident illumination capturing step along with its high dynamic range radiance map analysis sets Debevec's approach apart from methods [13] and [9]. Debevec demonstrated his method on synthetic objects with different reflectance properties, and the results are satisfactory.

4.6 Object Shape and Reflectance Modelling from Observation

It is often the case that 3D object models are created manually by users. The input process is normally time-consuming and can be a bottleneck for realistic image syn-

thesis. Therefore, techniques to obtain object model data automatically by observing real objects could have great significance in practical applications. An object model for computer graphics applications should contain two aspects of information: shape and reflectance properties of the object. A number of techniques have been developed for modelling object shapes by observing real objects.

In [27], Sato et al. present a a method for modelling object reflectance properties, as well as object shapes by observing real objects. An object surface shape is reconstructed by merging multiple range images of the object. By using the reconstructed object shape and a sequence of color images of the object, parameters of a reflection model are estimated in a robust manner. The key point of their method is that, first, the diffuse and specular reflection components are separated from the color image sequence, and then, reflectance parameters of each reflection component are estimated separately. This approach enables estimation of reflectance properties of real objects whose surfaces show specularity as well as diffusely reflected lights. The recovered object shape and reflectance properties are then used for synthesizing object images with realistic shading effects under arbitrary illumination conditions.

The separation of diffused and specular components of the object surface has been studied in computer vision for a number of years, because the specular highlight on objects can cause vision algorithms for scene segmentation and shading analysis to produce erroneous results. The strong directional dependence of specular reflection poses serious problems for vision techniques such as binocular stereo and motion detection. In [27], Sato et al. utilized the results in this research area, and combined them with model building methods using range images [5] [37]. The result they presented is impressive. Though they did not provide detailed discussion on the effect different surface properties might have on the robustness of their method.

4.7 Sprites with Depth and Layered Depth Images

For decades, animated cartoons and movie special effects have factored the rendering of a scene into layers that are updated independently and composed in the final display. Each layer produces a 2D image stream as well as a stream of 2D transformations that place the image on the display. Layered rendering naturally integrates 2D elements such as overlaid video, offline rendered sprites, or hand-animated characters into 3D scenes. Sprites are texture maps or images with alphas (transparent pixels) rendered onto planar surfaces. They can be used either for locally caching the results of slower rendering and then generating new view by warping, or they can be used directly as drawing primitives (as in video games).

The descriptive power (realism) of sprites can be greatly enhanced by adding an out-of-plane displacement component at each pixel in the sprite. In the case of sprites representing smoothly varying surfaces, Shade et al. [29] introduce an algorithm for rendering *Sprites with Depth*. The algorithm first forward maps the depth values themselves and then uses this information to add parallax corrections to a standard sprite rendering. For more complex geometries, they introduce the *Layered Depth Images* (LDI) that contains potentially multiple depth pixels at each discrete location in the image. Instead of a 2D array of depth pixels (a pixel with associated depth

information), they store a 2D array of layered depth pixels. A layered depth pixel stores a set of depth pixels along one line of sight. The next pixel in the layered depth pixel samples the next surface seen along that line of sight, etc.. When rendering from an LDI, the requested view can move away from the original LDI view and expose surface that were not visible in the first layer. The previously occluded regions may still be rendered from data stored in some later layer of a layered depth pixel.

There are many advantages to this representation. The size of the representation grows linearly only with the depth complexity of the image. Moreover, because the LDI data are represented in a single image coordinate system, there exists algorithm which can draw pixels in the output image in back to front order allowing proper alpha blending without depth sorting. One disadvantage of the LDI is that its pixel resampling steps might potentially degrade image quality. If some surface is seen at a glancing angle in the LDIs view, the depth complexity for that LDI increases, while the spatial sampling resolution over that surface degrades. A formal analysis of the sampling and aliasing issues involved in the LDI might be helpful. Sprites with Depth and Layered Depth Images provide us with two new image based primitives that can be used in combination with traditional computer graphics primitives such as polygonal models, environment maps and planner sprites. So they are useful tools.

5 Summary and Conclusion

5.1 Light Field, Geometric Model and Rendering Algorithms

Comparing image-based rendering to traditional rendering techniques, we notice the relationship among the three main components – light field, geometric model and rendering algorithms. Since an image is just a 2D slice of the light field, we will treat it as a subset of a light field. In going from a traditional rendering technique to light field rendering, we are shifting from mainly relying on geometric models to trying to find the right balance in combining light field and geometric information (Figure 14).

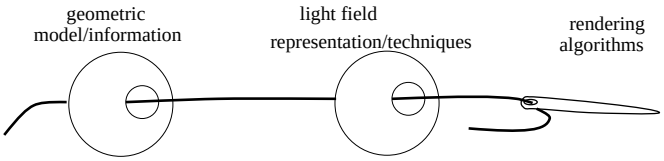

Figure 14: Light field, geometric model/information and rendering algorithms.

Most real world scenes are complex. When we move viewpoints around, we need to obtain certain geometric information (e.g., occlusion, depth of objects, etc.) to establish valid connections between light rays sampled at different viewpoints. The quality of an image constructed using a light field technique depends on the geometry of the scene. Problems such as visibility should be taken into account. Also, a light field representation of a real world scene is relatively large. A geometric model can be viewed as a form of compression. Hence, in light field techniques, geometric

information obtained automatically or with human assistance can be utilized to make the light field representation more efficient.

In traditional computer graphics rendering, complex models of the virtual world are made by users. Image-based rendering shift human effort to different processing steps, as shown in Figure 15. In terms of image appearance, IBR methods such as light field rendering and the lumigraph generate relatively more blurred images in comparison to that of traditional rendering. This blurriness is mainly due to the sampling process during the capturing step, the lack of detailed geometric information, and the interpolation step for constructing new views. The type of methods shown in Figure 15(b) is a compromise between the two extremes, namely all-geometry or almost-no-geometry. As more information is added concerning the scene geometry, such as depth of objects, primitive models of objects, etc., the quality of the image generated by the IBR methods has been improved in comparison to that of using the almost-no-geometry setting.

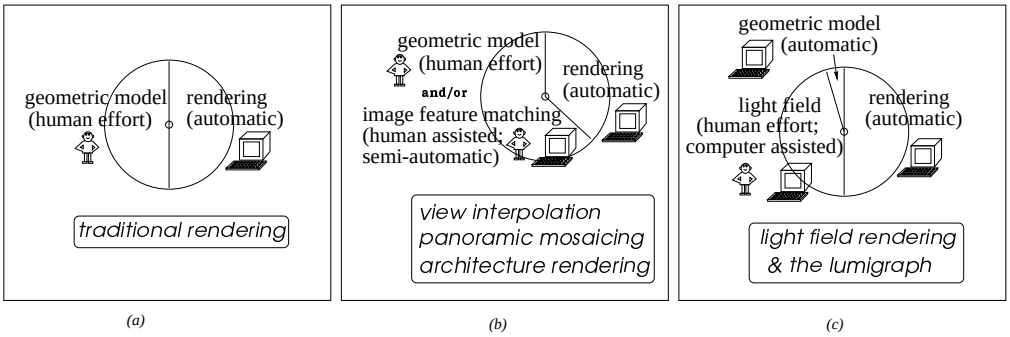

Figure 15: The distribution of human effort and automated computing workload.

5.2 Possible Future Research Directions and Potential Limiting Factors

Since most existing image-based rendering algorithms rely on different research results in computer vision, as we discussed in sections 2 and 3, the improvement and robustness of the IBR approaches might be limited by its dependence on difficult computer vision problems such as image-correspondence and camera calibration.

The vision problem – determining shape from images – is known to be hard. If we are willing to represent shape as images, perhaps as a set of inconsistent range maps rather than as a geometric model, does this simplify the vision problem? Another difficult problem is computing global illumination problem – determining surface reflectance from measured radiance in the presence of interreflections. It has never been solved except on very simple scenes. The question is – can image-based representations help solve these problems.

In terms of future work in image-based rendering, there are many possible research directions.

We might want to combine two sources of information, for example, video captured data (VCD) and computer generated scenes (CGS). Recent research in image-based

rendering have in effect blurred some differences between VCD and CGS. For instance, a light field type representation might provide a relatively more unified way of processing information from both VCD and CGS. To combine VCD and CGS, the main problems can be divided into geometric issues and illumination issues. The geometric issues in turn divide into viewing parameters and visibility problems. The viewing problem is to establish common viewing parameters between the VCD and the CGS. The visibility problem consists in resolving mutual priority while compositing the the video captured scene and the computer generated scene. The illumination problem is to compute both local and global illumination of video captured objects being illuminated by computer generated objects (including light sources) and vice versa. Previous research in similar directions are mainly done in 2D [14], such as 2D image compositing using real video images and computer generated images, estimating illumination characteristics, highlight detection, determining light direction from shading, establishing common global illumination. Some of these problems are still not well understood. Image-based rendering techniques might help us to extend some of the existing research results to 3D and providing some solutions to the open problems.

To extend the current image-based rendering methods and applications, we might want to take *time* into consideration. As we discussed in section 4.2.3, when researchers try to apply the image-based rendering type of idea to modelling an action or event which took place over a time interval, simultaneously capturing images from different viewpoints is possible. However, the sparse viewpoints due to limited hardware resources and the visibility problem due to scene complexity need to be taken care of in order to improve the quality of the sequence of images constructed by such technique.

Apart from various open problems related to the variables in the plenoptic function, more efficient image-based representations is also worth looking into. Researchers are studying various hybrid approaches based on partial factorization into geometric and reflectance structure and representation of the remaining information in the form of unfactored image maps.

Since image-based rendering draws its inspiration from both computer graphics and computer vision research, its dependence on both research areas might make it subject to certain unsolved problems in both areas. However, computer graphics and computer vision tasks differ in their goals and constraints. For instance, some tasks in computer vision and robotics need to be fully automatic, and the best algorithms are still not robust enough. But when similar algorithm is used in image-based rendering, we are not under the same constraint. A reasonable amount of human assistance might be introduced, which may help the algorithm produce more satisfactory results. By taking advantage of the differences and similarity between computer graphics and computer vision, we can further explore and improve the image-based rendering paradigm.

Currently, we are investigating a hybrid geometry- and image-based approach for extending the Lumigraph approach [17]. The Lumigraph is a "what you see is what you get" and "what you see is all you get" type of method, where the image based representation is obtained under a fixed lighting condition. In order to incorporate

different lighting condition, we need to employ certain mechanisms to control the diffuse and specular components of the appearance of the objects. The color of the diffused component has the color of the object, and the color of the specular component is largely affected by the color of the light. Separating the diffused and specular components is one way of providing control over image-based rendering and scene synthesis process. Simple geometric models extracted from the input images are useful for interpreting the color information in the input and re-constructing the scene under different lighting conditions.

Striving to achieve different effects using image-based modelling and rendering techniques is not just a set of isolated interesting intellectual exercises. Computer graphics itself is moving towards a direction of changing from being just a tool for scentific visualization and expensive entertainment production to becoming a *medium*, a *medium* people can use to record ideas and share experiences. Despite of the complicated vision problems, what makes image-based rendering special is its "intuitiveness" – direct visual translation of the real world. It is almost like how photography compares to traditional painting techniques. Most people are not good photographers at all, but almost everybody takes photographs and uses photography as a *medium* of communication. Image-based rendering fits into this movement of making computer graphics becoming more of a ubiquitous true *medium*.

A Fundamental Matrix

Given two reference views/images of the same scene, the *fundamental matrix* is a 3×3 matrix which maps a point in one view to its corresponding epipolar line in the other view. In this appendix, we will explain how the fundamental matrix is derived. This derivation is not essential for understanding the rest of the material in this paper.

Note that during the derivation, when we need to distinguish between a projective quantity and an affine or euclidean one, we will add a $\tilde{}$ on top of the projective quantity.

A camera can be considered a system that performs a linear projective transformation from the projective space $\mathcal{P}^3$ into the projective plane $\mathcal{P}^2$. The projection of a point $\mathbf{Q}$ in 3D space to a point $\mathbf{q}$ in an image plane is given by the equation

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \tilde{\mathbf{P}} \begin{bmatrix} X \\ Y \\ Z \\ T \end{bmatrix},$$

where $\tilde{\mathbf{q}} = [U, V, S]^T$ and $\tilde{\mathbf{Q}} = [X, Y, Z, T]^T$, and $\tilde{\mathbf{P}}$ is a 3×4 transformation matrix which defines a pinhole camera up to a scale factor and $\tilde{\mathbf{P}}$ is of rank 3 [23].

Since this will be used in what follows, let us now see how the optical center $\mathbf{C}$ can be recovered from the matrix $\tilde{\mathbf{P}}$. Let us decompose $\tilde{\mathbf{P}}$ as follows

$$\tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{P} & \mathbf{p} \end{bmatrix},$$

where $\mathbf{P}$ is a 3×3 matrix of rank 3 and $\mathbf{p}$ is a 3×1 vector. Let us assume without loss of

generality that $\mathbf{C}$ is not at infinity and let $\tilde{\mathbf{C}} = \left[\mathbf{C}^T 1\right]^T$ be a projective representation of this point, where $\mathbf{C}$ is its 3×1 Euclidean vector of coordinates and the component equal to 1 accounts for the fact that $\mathbf{C}$ is not at infinity. $\mathbf{C}$ satisfies the equation $\tilde{\mathbf{P}}\tilde{\mathbf{C}} = 0$ from which we conclude

$$\mathbf{C} = -\mathbf{P}^{-1}\mathbf{p}.$$

We now consider the case of two cameras looking at the same scene. The epipolar geometry is as shown in Figure 16. $\mathbf{C}$ and $\mathbf{C}'$ are the optical centers of the two cameras. $\mathbf{M}$ is a point in 3D. Points $\mathbf{m}$ and $\mathbf{m}'$ are the projections of $\mathbf{M}$ in the left and right image planes, respectively. The plane formed by points $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{C}'$ intersects the left and right image planes at the epipolar lines $l_{m'}$ and l'_m, respectively. The line $\mathbf{CC}'$ intersects the two epipolar lines at the epipoles $\mathbf{e}$ and $\mathbf{e}'$.

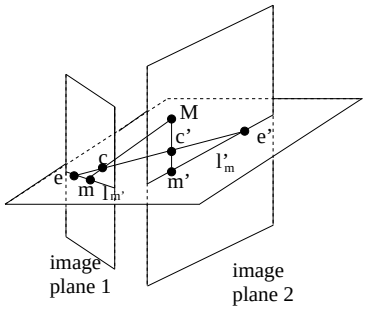

Figure 16: Epipolar geometry.

In the two camera case, let $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{P}}'$ be the perspective projection matrix for the left and right cameras, respectively. Clearly, the epipole $\mathbf{e}'$ in the right image is the projection of the optical center $\mathbf{C}$ of the first camera into the second camera. So we have

$$\mathbf{e}' = \tilde{\mathbf{P}}' \begin{bmatrix} \mathbf{C} \\ 1 \end{bmatrix} = \tilde{\mathbf{P}}' \begin{bmatrix} -\mathbf{P}^{-1}\mathbf{p} \\ 1 \end{bmatrix} = \mathbf{p}' - \mathbf{P}'\mathbf{P}^{-1}\mathbf{p}.$$

The epipolar line in the right image plane is defined by two points: the epipole $\mathbf{e}'$ and the projection of point of infinity of direction $< \mathbf{C}, \mathbf{M} >$ in the right image. Let us assume without loss of generality that $\mathbf{M}$ is not at infinity. To calculate $\mathbf{M} - \mathbf{C}$, we observe that $\mathbf{M}$'s projection in the left image plane is

$$\mathbf{m} = \tilde{\mathbf{P}}\tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{P} & \mathbf{p} \end{bmatrix} \begin{bmatrix} \mathbf{M} \\ 1 \end{bmatrix} = \mathbf{P}\mathbf{M} + \mathbf{p}.$$

Multiply both sides by $\mathbf{P}^{-1}$ from the left and substitute in $\mathbf{C} = -\mathbf{P}^{-1}\mathbf{p}$, we have

$$\mathbf{P}^{-1}\mathbf{m} = \mathbf{M} + \mathbf{P}^{-1}\mathbf{p} = \mathbf{M} - \mathbf{C}.$$

So, the point of infinity of direction $< \mathbf{C}, \mathbf{M} >$ is projected to the right image by

$$\tilde{\mathbf{P}}' \begin{bmatrix} \mathbf{M} - \mathbf{C} \\ 0 \end{bmatrix} = \tilde{\mathbf{P}}' \begin{bmatrix} \mathbf{P}^{-1}\mathbf{m} \\ 0 \end{bmatrix} = \mathbf{P}'\mathbf{P}^{-1}\mathbf{m}.$$

The projective representation of the epipolar line l'_m is obtained by taking the cross-product of $\mathbf{e}'$ and the projection of the infinity of $< \mathbf{C}, \mathbf{M} >$ in the right image,

$$l'_m = \left[\mathbf{p}' - \mathbf{P}'\mathbf{P}^{-1}\mathbf{p}\right] \times \mathbf{P}'\mathbf{P}^{-1}\mathbf{m} = \underbrace{\left[\mathbf{p}' - \mathbf{P}'\mathbf{P}^{-1}\mathbf{p}\right] \times \mathbf{P}'\mathbf{P}^{-1}}_{\mathbf{F}}\mathbf{m}.$$

Hence, the matrix $\mathbf{F}$ is the *fundamental matrix* which maps the point $\mathbf{m}$ in the left image to its corresponding epipolar line l'_m in the right image.

References

[1] Edward H. Adelson and James R. Bergen, "The Plenoptic Function and the Elements of Early Vision", *Computational Models of Visual Processing*, edited by Michael S. Landy and J. Anthony Movshon, The MIT Press, 1991, pp 1-20.

[2] Emilio Camahort, Apostolos Lerios and Donald Fussell, "Uniformly Sampled Light Fields", *Proceedings of the 9th Eurographics Workshop on Rendering*, Vienna, Austria, June 29 – July 1, 1998, pp 117-130.

[3] Shenchang Eric Chen and Lance Williams, "View Interpolation for Image Synthesis", *ACM SIGGRAPH '93*, published as ACM Computer Graphics 1993 Annual Conference Series (August, 1993), pp 279-288.

[4] Shenchang Eric Chen, "QuickTime VR – An Image-Based Approach to Virtual Environment Navigation", *ACM SIGGRAPH '95*, published as ACM Computer Graphics 1995 Annual Conference Series (August, 1995), pp 29-38.

[5] Brian Curless and Marc Levoy, "A volumetric method for building complex models from range images", *ACM SIGGRAPH '96*, published as ACM Computer Graphics 1996 Annual Conference Series (August, 1996), pp 303-312.

[6] Paul E. Debevec, Camillo J. Taylor and Jitendra Malik, "Modelling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach", *ACM SIGGRAPH '96*, published as ACM Computer Graphics 1996 Annual Conference Series (August, 1996), pp 11-20.

[7] Paul E. Debevec and Jitendra Malik, "Recovering High Dynamic Range Radiance Maps from Photographs", *ACM SIGGRAPH '97*, published as ACM Computer Graphics 1997 Annual Conference Series (August, 1997), pp 369-378.

[8] Paul E. Debevec, "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography", *ACM SIGGRAPH '98*, published as ACM Computer Graphics 1998 Annual Conference Series (August, 1998).

[9] G. Drettakis, L. Robert and S. Bougnoux, "Interactive common illumination for computer augmented reality", in *8th Eurographics workshop on Rendering*, St. Etienne, France (May 1997), pp. 45-57.

[10] O. Faugeras, *Three-Dimensional Computer Vision, A Geometric Viewpoint*, MIT Press, Cambridge, MA, 1993.

[11] O. Faugeras and L. Robert, "What Can Two Image Tell Us About a Third one?", *European Conference on Computer Vision*, page 485-492, 1994. Also the technical report No. 2018, 1993, available at http://www.inria.fr/RRRT/RR-2018.html.

[12] Alain Fournier, E. Fiume, M. Ouellette and C. K. Chee, "Fiat Lux: Light driven global illumination", Technical Memo DGP89-1, Dynamic Graphics Project, Department of Computer Science, University of Toronto, 1989.

[13] Alain Fournier, Atjeng S. Gunawan and Chris Romanzin, "Common Illumination between Real and Computer Generated Scenes", *Graphics Interface '93*, 1993, pp 254-262.

[14] Alain Fournier, "Illumination Problems in Computer Augmented Reality", UBC Imgar Lab Technical Reports, 1994, http://www.cs.ubc.ca/nest/imager/tr/fournier.95c.html.

[15] Ned Greene, "Environment Mapping and Other Applications of World Projections," *IEEE Computer Graphics and Applications*, Vol. 6, No. 11, Nov, 1986.

[16] Allen Gersho and Robert M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Norwell, MA, 1992.

[17] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski and Michael F. Cohen, "The Lumigraph", *ACM SIGGRAPH '96*, published as ACM Computer Graphics 1996 Annual Conference Series (August, 1996), pp 43-54.

[18] Takeo Kanade, P. J. Narayanan and Peter W. Rander, "Virtualized Reality: Concepts and Early Results", *IEEE Workshop on the Representation of Visual Scenes*, June 24, 1995.

[19] Stephane Laveau and Olivier Faugeras, "3D Scene Representation as a Collection of Images and Fundamental Matrices", *Technical Report 2205*, INRIA, Sophia-Antipolis, France, February, 1994.

[20] Marc Levoy and Pat Hanrahan, "Light Field Rendering", *ACM SIGGRAPH '96*, published as ACM Computer Graphics 1996 Annual Conference Series (August, 1996), pp 31-42.

[21] Robert R. Lewis and Alain Fournier, "Light-Driven Global Illumination with a Wavelet Representation of Light Transport", *Proceedings of the 7th Eurographics Workshop on Rendering*, Porto, Portugal, 1996.

[22] Robert van Liere, "Divide and Conquer Radiosity", *Eurographics Workshop on Rendering*, Barcelona, Spain, 13-15 May, 1991.

[23] Quan-Tuan Luong and Olivier D. Faugeras, "The Fundamental Matrix: Theory, Algorithms, and Stability Analysis", *International Journal of Computer Vision*, Vol 17, page 43-75, 1996.

[24] Steve Mann and Rosalind W. Picard, "Virtual Bellows: Constructing High Quality Stills From Video", *First IEEE International Conference on Image Processing*, Austin, TX, November, 1994.

[25] Steve Mann and Rosalind W. Picard, "Being 'Undigital' with Digital Cameras: Extending Dynamic Range by Combining Differently Exposed Pictures", in *Proceedings of IS&T 46th annual conference*, (May 1995), pp. 422–428.

[26] Leonard McMillan and Gary Bishop, "Plenoptic Modeling: An Image-Based Rendering System", *ACM SIGGRAPH '95*, published as ACM Computer Graphics 1995 Annual Conference Series (August, 1995).

[27] Yoichi Sato, Mark D. Wheeler and Katsushi Ikeuchi, *ACM SIGGRAPH '97*, published as ACM Computer Graphics 1997 Annual Conference Series (August, 1997).

[28] Steven M. Seitz and Charles R. Dyer, "Physically-Valid View Synthesis by Image Interpolation", *Proceedings IEEE Workshop on Representation of Visual Scenes*, (In conjunction with ICCV'95), Cambridge, MA, June 24, 1995.

[29] Jonathan Shade, Steven Gortler, Li-wei He and Richard Szeliski, "Layered Depth Images", *ACM SIGGRAPH '98*, published as ACM Computer Graphics 1998 Annual Conference Series (August, 1998), pp 231–242.

[30] Jerome M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, December 1993.

[31] Amnon Shashua, "On Geometric and Algebraic Aspects of 3D Affine and Projective Structures from Perspective 2D Views", *MIT Media Laboratory, Technical Report*, No. 236, July, 1993.

[32] G. P. Stein, "Accurate Internal Camera Calibration using Rotation, with Analysis of Sources of Error", *Fifth International Conference on Computer Vision*, (ICCV'95), page 230-236, Cambridge, Massachusetts, June, 1995.

[33] Richard Szeliski and Heung-Yeung Shum, "Creating Full View Panoramic Image Mosaics and Environment Maps", *ACM SIGGRAPH '97*, published as ACM Computer Graphics 1997 Annual Conference Series (August, 1997).

[34] Shimon Ullman and Ronen Basri, "Recognition by Linear Combinations of Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 10, October, 1991, pp. 992-1006.

[35] G. Wallace, "The JPEG Still Picture Compression Standard", *CACM*, Vol. 34, No. 4, April, 1991, pp. 30-44.

[36] Tomas Werner, Roger David Hersch and Vaclav Hlavac, "Rendering Real-World Objects Using View Interpolation", *Fifth International Conference on Computer Vision*, (ICCV'95), Cambridge, MA, June 1995, pp 957-962.

[37] Mark D. Wheeler, Yoichi Sato and Katsushi Ikeuchi, "Consensus Surfaces for Modeling 3D Objects from Multiple Range Images", *DARPA Image Understanding Workshop*, 1997.

[38] Tien-Tsin Wong, Pheng-Ann Heng, Siu-Hang Or and Wai-Ying Ng, "Image-based Rendering with controllable Illumination", *Rendering Techniques '97*, Proceedings of the Eurographics Workshop in St. Etienne, France, June, 1997, pp 13-22.