# Light Field Rendering

Marc Levoy and Pat Hanrahan
Computer Science Department
Stanford University

## Abstract

A number of techniques have been proposed for flying through scenes by redisplaying previously rendered or digitized views. Techniques have also been proposed for interpolating between views by warping input images, using depth information or correspondences between multiple images. In this paper, we describe a simple and robust method for generating new views from arbitrary camera positions without depth information or feature matching, simply by combining and resampling the available images. The key to this technique lies in interpreting the input images as 2D slices of a 4D function - the light field. This function completely characterizes the flow of light through unobstructed space in a static scene with fixed illumination.

We describe a sampled representation for light fields that allows for both efficient creation and display of inward and outward looking views. We have created light fields from large arrays of both rendered and digitized images. The latter are acquired using a video camera mounted on a computer-controlled gantry. Once a light field has been created, new views may be constructed in real time by extracting slices in appropriate directions. Since the success of the method depends on having a high sample rate, we describe a compression system that is able to compress the light fields we have generated by more than a factor of 100:1 with very little loss of fidelity. We also address the issues of antialiasing during creation, and resampling during slice extraction.

**CR Categories**: I.3.2 [Computer Graphics]: Picture/Image Generation — *Digitizing and scanning, Viewing algorithms*; I.4.2 [Computer Graphics]: Compression — *Approximate methods*

**Additional keywords**: image-based rendering, light field, holographic stereogram, vector quantization, epipolar analysis

## 1. Introduction

Traditionally the input to a 3D graphics system is a scene consisting of geometric primitives composed of different materials and a set of lights. Based on this input specification, the rendering system computes and outputs an image. Recently a new approach to rendering has emerged: *image-based rendering*. Image-based rendering systems generate different views of an environment from a set of pre-acquired imagery. There are several advantages to this approach:

- The display algorithms for image-based rendering require modest computational resources and are thus suitable for real-time implementation on workstations and personal computers.
- The cost of interactively viewing the scene is independent of scene complexity.
- The source of the pre-acquired images can be from a real or virtual environment, i.e. from digitized photographs or from rendered models. In fact, the two can be mixed together.

The forerunner to these techniques is the use of environment maps to capture the incoming light in a texture map [Blinn76, Greene86]. An environment map records the incident light arriving from all directions at a point. The original use of environment maps was to efficiently approximate reflections of the environment on a surface. However, environment maps also may be used to quickly display any outward looking view of the environment from a fixed location but at a variable orientation. This is the basis of the Apple QuickTimeVR system [Chen95]. In this system environment maps are created at key locations in the scene. The user is able to navigate discretely from location to location, and while at each location continuously change the viewing direction.

The major limitation of rendering systems based on environment maps is that the viewpoint is fixed. One way to relax this fixed position constraint is to use view interpolation [Chen93, Greene94, Fuchs94, McMillan95a, McMillan95b, Narayanan95]. Most of these methods require a depth value for each pixel in the environment map, which is easily provided if the environment maps are synthetic images. Given the depth value it is possible to reproject points in the environment map from different vantage points to warp between multiple images. The key challenge in this warping approach is to "fill in the gaps" when previously occluded areas become visible.

Another approach to interpolating between acquired images is to find corresponding points in the two [Laveau94, McMillan95b, Seitz95]. If the positions of the cameras are known, this is equivalent to finding the depth values of the corresponding points. Automatically finding correspondences between pairs of images 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.

In this paper we propose a new technique that is robust and allows much more freedom in the range of possible views. The major idea behind the technique 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, not a 5D 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.

---

Address: Gates Computer Science Building 3B
Stanford University
Stanford, CA 94305

levoy@cs.stanford.edu
hanrahan@cs.stanford.edu
http://www-graphics.stanford.edu

Generating a new image from a light field is quite different than previous view interpolation approaches. First, the new image is generally formed from many different pieces of the original input images, and need not look like any of them. Second, no model information, such as depth values or image correspondences, is needed to extract the image values. Third, image generation involves only resampling, a simple linear process.

This representation of the light field is similar to the epipolar volumes used in computer vision [Bolles87] and to horizontal-parallax-only holographic stereograms [Benton83]. An epipolar volume is formed from an array of images created by translating a camera in equal increments in a single direction. Such a representation has recently been used to perform view interpolation [Katayama95]. A holographic stereogram is formed by exposing a piece of film to an array of images captured by a camera moving sideways. Halle has discussed how to set the camera aperture to properly acquire images for holographic stereograms [Halle94], and that theory is applicable to this work. Gavin Miller has also recognized the potential synergy between true 3D display technologies and computer graphics algorithms [Miller95].

There are several major challenges to using the light field approach to view 3D scenes on a graphics workstation. First, there is the choice of parameterization and representation of the light field. Related to this is the choice of sampling pattern for the field. Second, there is the issue of how to generate or acquire the light field. Third, there is the problem of fast generation of different views. This requires that the slice representing rays through a point be easily extracted, and that the slice be properly resampled to avoid artifacts in the final image. Fourth, the obvious disadvantage of this approach is the large amount of data that may be required. Intuitively one suspects that the light field is coherent and that it may be compressed greatly. In the remaining sections we discuss these issues and our proposed solutions.

## 2. Representation

We define the light field as the radiance at a point in a given direction. Note that our definition is equivalent to the *plenoptic function* introduced by Adelson and Bergen [Adelson91]. The phrase light field was coined by A. Gershun in his classic paper describing the radiometric properties of light in a space [Gershun36].[1] McMillan and Bishop [McMillan95b] discuss the representation of 5D light fields as a set of panoramic images at different 3D locations.

However, the 5D representation may be reduced to 4D in free space (regions free of occluders). This is a consequence of the fact that the radiance does not change along a line unless blocked. 4D light fields may be interpreted as functions on the space of oriented lines. The redundancy of the 5D representation is undesirable for two reasons: first, redundancy increases the size of the total dataset, and second, redundancy complicates the reconstruction of the radiance function from its samples. This reduction in dimension has been used to simplify the representation of radiance emitted by luminaires [Levin71, Ashdown93]. For the remainder of this paper we will be only concerned with 4D light fields.

[1] For those familiar with Gershun's paper, he actually uses the term light field to mean the irradiance vector as a function of position. For this reason P. Moon in a later book [Moon81] uses the term photic field to denote what we call the light field.

Although restricting the validity of the representation to free space may seem like a limitation, there are two common situations where this assumption is useful. First, most geometric models are bounded. In this case free space is the region outside the convex hull of the object, and hence all views of an object from outside its convex hull may be generated from a 4D light field. Second, if we are moving through an architectural model or an outdoor scene we are usually moving through a region of free space; therefore, any view from inside this region, of objects outside the region, may be generated.

The major issue in choosing a representation of the 4D light field is how to parameterize the space of oriented lines. There are several issues in choosing the parameterization:

**Efficient calculation.** The computation of the position of a line from its parameters should be fast. More importantly, for the purposes of calculating new views, it should be easy to compute the line parameters given the viewing transformation and a pixel location.

**Control over the set of lines.** The space of all lines is infinite, but only a finite subset of line space is ever needed. For example, in the case of viewing an object we need only lines intersecting the convex hull of the object. Thus, there should be an intuitive connection between the actual lines in 3-space and line parameters.

**Uniform sampling.** Given equally spaced samples in line parameter space, the pattern of lines in 3-space should also be uniform. In this sense, a uniform sampling pattern is one where the *number of lines* in intervals between samples is constant everywhere. Note that the correct measure for number of lines is related to the form factor kernel [Sbert93].

The solution we propose is to parameterize lines by their intersections with two planes in arbitrary position (see figure 1). By convention, the coordinate system on the first plane is $(u, v)$ and on the second plane is $(s, t)$. An oriented line is defined by connecting a point on the uv plane to a point on the st plane. In practice we restrict $u$, $v$, $s$, and $t$ to lie between 0 and 1, and thus points on each plane are restricted to lie within a convex quadrilateral. We call this representation a *light slab*. Intuitively, a light slab represents the beam of light entering one quadrilateral and exiting another quadrilateral.

A nice feature of this representation is that one of the planes may be placed at infinity. This is convenient since then lines may be parameterized by a point and a direction. The latter will prove useful for constructing light fields either from orthographic images or images with a fixed field of view. Furthermore, if all calculations are performed using homogeneous coordinates, the two cases may be handled at no additional cost.

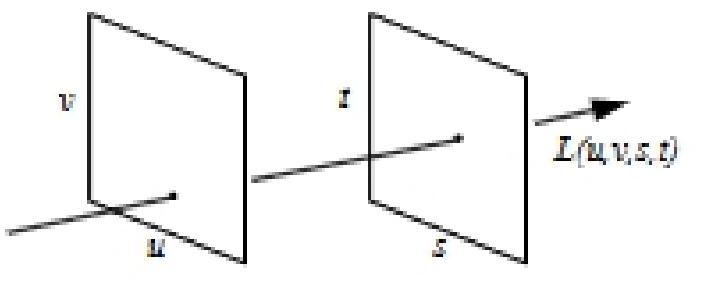


**Figure 1:** The light slab representation.

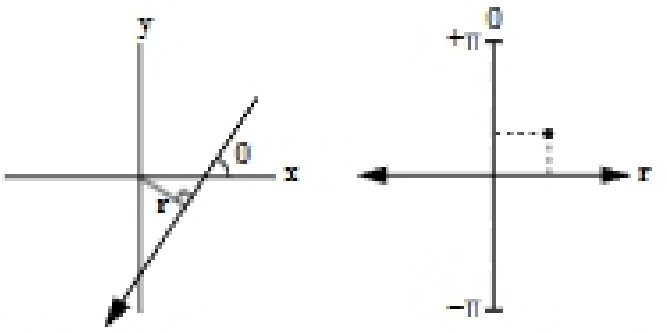


**Figure 2:** Definition of the line space we use to visualize sets of light rays. Each oriented line in Cartesian space (at left) is represented in line space (at right) by a point. To simplify the visualizations, we show only lines in 2D; the extension to 3D is straightforward.

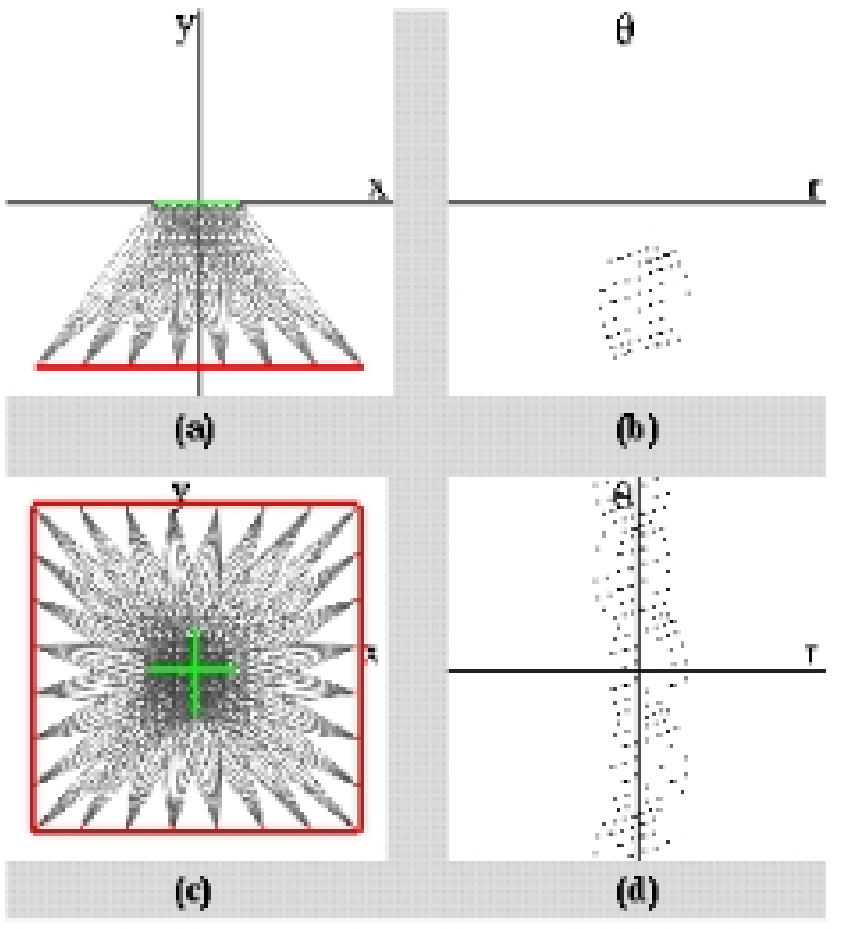


**Figure 3:** Using line space to visualize ray coverage. (a) shows a single light slab. Light rays (drawn in gray) connect points on two defining lines (drawn in red and green). (c) shows an arrangement of four rotated copies of (a). (b) and (d) show the corresponding line space visualizations. For any set of lines in Cartesian space, the envelope formed by the corresponding points in line space indicates our coverage of position and direction; ideally the coverage should be complete in $\theta$ and as wide as possible in $r$. As these figures show, the single slab in (a) does not provide full coverage in $\theta$, but the four-slab arrangement in (c) does. (c) is, however, narrow in $r$. Such an arrangement is suitable for inward-looking views of a small object placed at the origin. It was used to generate the lion light field in figure 14d.

---

A big advantage of this representation is the efficiency of geometric calculations. Mapping from $(u, v)$ to points on the plane is a projective map and involves only linear algebra (multiplying by a 3x3 matrix). More importantly, as will be discussed in section 5, the inverse mapping from an image pixel $(x, y)$ to $(u, v, s, t)$ is also a projective map. Methods using spherical or cylindrical coordinates require substantially more computation.

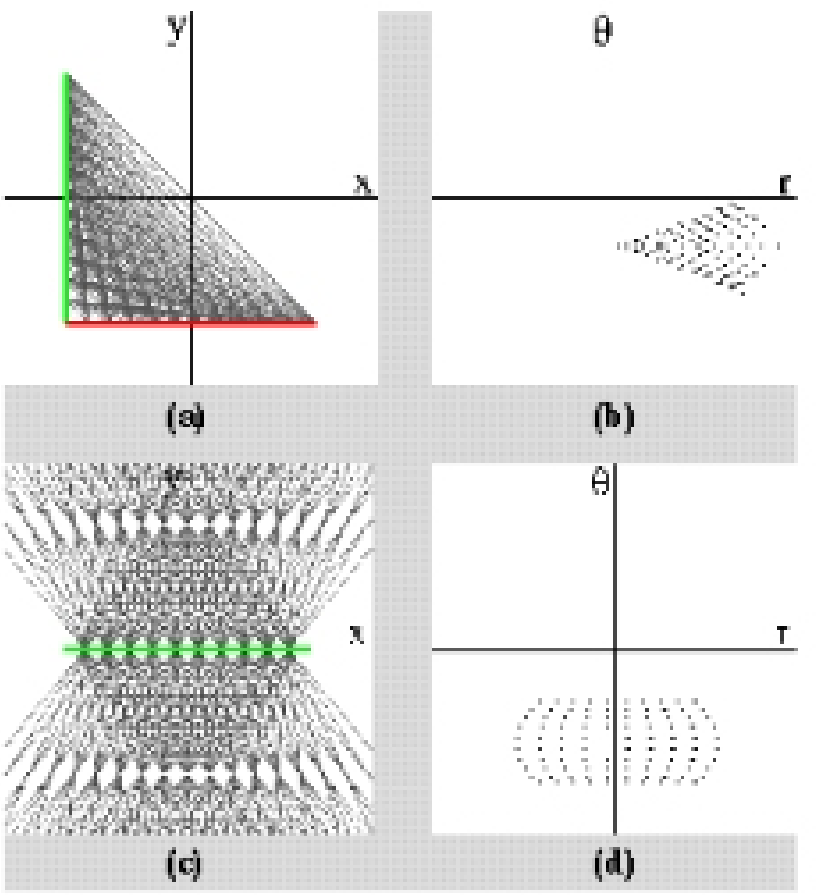


**Figure 4:** Using line space to visualize sampling uniformity. (a) shows a light slab defined by two lines at right angles. (c) shows a light slab where one defining line is at infinity. This arrangement generates rays passing through the other defining line with an angle between -45° and +45°. (b) and (d) show the corresponding line space visualizations. Our use of $(r, \theta)$ to parameterize line space has the property that equal areas in line space correspond to equally dense sampling of position and orientation in Cartesian space; ideally the density of points in line space should be uniform. As these figures show, the singularity at the corner in (a) leads to a highly nonuniform and therefore inefficient sampling pattern, indicated by dark areas in (b) at angles of 0 and $-\pi/2$. (c) generates a more uniform set of lines. Although (c) does not provide full coverage of $\theta$, four rotated copies do. Such an arrangement is suitable for outward-looking views by an observer standing near the origin. It was used to generate the hallway light field in figure 14c.

---

Many properties of light fields are easier to understand in line space (figures 2 through 4). In line space, each oriented line is represented by a point and each set of lines by a region. In particular, the set of lines represented by a light slab and the set of lines intersecting the convex hull of an object are both regions in line space. All views of an object could be generated from one light slab if its set of lines include all lines intersecting the convex hull of the object. Unfortunately, this is not possible. Therefore, it takes multiple light slabs to represent all possible views of an object. We therefore tile line space with a collection of light slabs, as shown in figure 3.

An important issue related to the parameterization is the sampling pattern. Assuming that all views are equally likely to be generated, then any line is equally likely to be needed. Thus all regions of line space should have an equal density of samples. Figure 4 shows the density of samples in line space for different arrangements of slabs. Note that no slab arrangement is perfect: arrangements with a singularity such as two polygons joined at a corner (4a) are bad and should be avoided, whereas slabs formed