Viewing Complex Environments with Hierarchical Light Fields

Category: research

Abstract

We propose a hierarchical light field framework in which it is possible to deal with complex environments that cannot be simply represented by a basic 4D (non-hierarchical) light field. The hierarchical light field framework extends the basic light field structure by defining bounds on the region of freespace over which a light field is valid. It consists of multiple light fields hierarchically placed in 3D space, with each light field only sampling part of the scene. By compositing the images from the subset of light fields that are valid for any viewpoint, the viewer is given much more freedom in the range of possible views compared to the basic light field representation. The sampling resolution is hierarchical and view dependent to guarantee efficient sampling, which avoids overrsampling and aliasing during minification. The hierarchy and sampling parameters are generated systematically so that a given lower bound on image quality can be maintained. Also, the rendering time is independent of the scene complexity. Because of these advantages, the hierarchical light field framework is suitable for visualizing complex environments.

In this paper, the advantages of hierarchical light fields are described and analyzed, along with techniques for reduction of light field memory requirements. A detailed description is given of the steps undertaken during generation of hierarchical light fields and the reconstruction process. The capabilities and advantages of the representation are shown by examples of an immersive walkthrough and a 3D fly-around of an object.

1 Introduction

Image-based rendering, the practice of generating new views from a set of previously rendered or acquired views, has been an active area of research during the past few years. There are several advantages to modeling and rendering a scene using an image-based representation instead of a traditional geometry-based representation. Since the primitives in image-based rendering are pixels, the amount of computation required to process an image is in general proportional to the number of pixels in the image, making the cost of rendering a view independent of the scene's geometric and shading complexity. Furthermore, since operations on pixels are generally inexpensive, repetitive, and incremental, they can usually be performed faster than rendering from geometric models. This makes image-based rendering very attractive for use on general-purpose computers without special-purpose graphics hardware. In terms of model generation, the ability to acquire source images from the real world makes it possible to render more realistic, complex, and visually rich scenes that would otherwise be hard to model using traditional geometric and shading primitives. In terms of level-of-detail control, an image having the nature of a 2D discrete signal allows classical signal processing methods such as sampling, filtering, and frequency-domain decomposition to be directly applied.

Among various image-based rendering approaches, the light field function (or, equivalently, the plenoptic function), which is defined as the radiance of a point in a given direction, has been proposed as a framework for interpreting source images and generating new views [1, 7, 13, 16]. In general, this function is a 5D function. A representation of this 5D function has been proposed by McMillan and Bishop as a set of panoramic images taken at different 3D locations [16]. As pointed out by Levoy and Hanrahan, the 5D light field can be reduced to a 4D function, defined on the space of oriented lines, in regions free of occluders (free space) [13]. Within this formulation, an image is interpreted as a two-dimensional sampled slice of this 4D function, and generating new views corresponds to extracting and interpolating known samples from this light field. This representation has several advantages. First, the use of light fields enables reconstruction of geometrically correct images when the viewpoint is either translated or rotated, without requiring model information such as pixel correspondence or depth values to resolve occlusion. Second, the reconstruction process to generate new images is simple and robust. Third, the light field formulation provides a well-defined framework for flexible level-of-detail control by specification of sampling patterns and parameters.

The light field formulation has two weaknesses that limit its application to viewing complex environments. First, a 4D low pass filter is used during the creation of the source images to correctly anti-alias the light field. This filter integrates all of the rays represented by each sample of the light field. However, this process may also introduce blurriness to the images. In general, the farther an object is from the focal plane, the more blurry the generated images are. The blurriness of the images limits the spatial range of objects that the light field can clearly represent. Second, a particular light field is only valid for viewpoints that are outside the convex hull of the represented objects. For a viewpoint that is inside the convex hull, the free-space assumption is invalid, and a different light field is needed. In walking through a complex environment, the viewer is constantly walking into or out of the convex hull of different sets of objects. The systematic creation and parameterization of multiple light fields is still an open problem.

Another key issue in image-based rendering is the efficiency of a sampled representation. If an object or a portion of an object occupies very few pixels in the source images but needs to occupy more pixels in a new view to reveal more detail, there may not be enough information to infer those details. The resulting image quality is poor due to undersampling. The inverse case, in which the source images contain more pixels than the interpolated images need, is called oversampling. Undersampling of a scene results in blurriness and loss of detail, while oversampling results in inefficient use of storage. The most storage-efficient sampling representation for a scene contains the minimum number of samples of the scene required for a desired resolution, as specified by some minimum resolvable solid angle over the valid viewing volume. The most time-efficient representation for rendering corresponds to access of the minimum required number of samples in the source images for each pixel rendered on screen.

In this paper, we propose the hierarchical light field (HLF) representation, which consists of multiple light fields hierarchically placed in 3D space, with each light field only sampling part of the scene. A alpha(opacity) value is stored for every sample along with its color. With a reconstruction scheme that always selects the subset of light fields that are valid for any given view point, and composites the images from different light fields together according to the opacity of pixels of each layer, this representation enables the viewpoint to move into or out of the convex hull of a set of objects. This gives the viewer much more freedom in the range of possible views compared to the previous (non-hierarchical, free space) light field representation. The sampling resolution is hierarchical and view dependent, which avoids oversampling and is more compact. Moreover, the hierarchy and sampling parameters are generated systematically so that a lower bound on image quality can be maintained. We also present a corresponding reconstruction scheme with moderate depth complexity that resamples these light fields and composites them to generate the final image for a given viewpoint.

The remainder of this paper is organized as follows. Section 2 describes related work. Section 3 describes the hierarchical light field representation and its advantages. Section 4 discusses implementation issues, including the generation and reconstruction of hierarchical light fields and techniques for reduction of light field data size. Section 5 gives results, and Section 6 contains discussion and ideas for future work.

2 Related work

One form of scene representation based on images is a set of environment maps. An environment map records incoming light from all directions at a given point. It can be used to quickly display any outward looking view at any orientation from that point. An example of the use of environment maps is the Apple QuickTime VR system, which contains environment maps created for a set of key locations [4]. The user can continuously change viewing direction at each location, but can only “jump” discretely from location to location because views in between the key locations cannot be generated. Strictly speaking, in order to allow for continuous viewpoint translation, a new environment map is needed for every possible viewpoint, which will result in extremely large storage requirements. In practice, various approximation techniques are proposed. Regan and Pose [17] divide the scene into layers by depth and create an environment map for each layer. Environment maps of far away objects are updated at a slower rate than those of closer objects. Greene and Kass [9] generate z-buffered environment maps of the far geometry for a cubic lattice of reference points and record only those polygons which are visible from those reference points. To avoid holes using this approach, polygons whose screen projection is smaller than a pixel are enlarged to one pixel. An approximation of the far environment from an arbitrary viewpoint is generated by rendering the list of polygons associated with the environment maps at adjacent lattice points from the new viewpoint. The system demonstrated at Siggraph’97 by UNC also stores environment maps for a 3D grid of viewpoints and approximates far geometry for a given viewpoint by resampling the environment maps.

A planar array of environment maps can be viewed as a special case of a light field with one focal plane at infinity. In other words, the hierarchical light field representation proposed in this paper can be viewed as an extension to the above presentation based on an array of environment maps. Similar to Greene’s work and the UNC system, hierarchical light fields are created for a regular lattice of viewpoints. Similar to Regan's priority rendering, for each viewpoint, there are multiple layers of light fields for objects at different depths.

Image warping has been used as a way of reconstructing new views from previously generated views. A number of variations of this approach have been proposed for use with different representations [5, 8, 12, 15, 16, 18]. A more complete review of warping techniques can be found in [15]. A common characteristic of these warping techniques is that they require a depth value, or information equivalent to depth values, such as pixel correspondences for each pixel. The reconstruction process of hierarchical light fields involves resampling each light field associated with a given viewpoint, then compositing them using the painter's algorithm, neither of which requires additional z-buffer information.

Another image-related technique that can be employed in interactive rendering to exploit frame-to-frame coherence is "image sprites", which are the 2D projections of an object or a collection of objects. These image components are usually dynamically generated and re-used for a range of viewpoints, and must be regenerated for viewpoints outside that range. In that sense, Regan's priority rendering is also a form of image sprites. The Talisman system divides a scene into layers of images. Each layer can be warped using affine transformation, then multiple layers can be composited together. In the work by Shade, et al., images of nodes on various levels of a BSP-tree are cached for potential re-use in the subsequent frames [19]. Rather than caching regions of a scene, Schaufler dynamically creates and re-uses images called "impostors" on a per-object basis and shows some advantages. Although image caching and light fields are considerably different in terms of their application and goal, they share some common aspects. The analysis Schaufler did on the validity of impostors has provided helpful insights to us for determination of where to place focal planes in hierarchical light fields.

It is also insightful to look at all of these image-based representations in terms of whether they are "view-centered" or "object-centered". Dally, et al. proposed a data structure called the delta tree as an "object-centered" approach to image-based rendering [6]. Each node stores an image taken from a point on a sphere that encloses an object. Space-efficiency is achieved by discarding pixels in an image that can be reconstructed from reprojecting its ancestor's image to that node's viewpoint. Reconstruction of an arbitrary view involves traversing the tree and compositing images or partial images of one or more leaf nodes. The representation is object-centered in the sense that it contains subdivided viewing of a space/region with respect to an object. In contrast, view-centered approaches subdivide the object space with respect to a view [4, 5, 8, 12, 16]. The hierarchical light field representation, however, can be viewed as a more general framework that encompasses both approaches because it allows subdivision in either the view space, or the object space, or both.

Another relevant area of previous work is the literature on hierarchical representations and approaches in the global illumination community. Hanrahan, et al. have proposed a hierarchical data structure and procedure to rapidly solve the radiosity equations with bounded error [10]. Lewis and Fournier [14] proposed a wavelet-based system to compute radiative transport and surface interaction. While the hierarchical radiosity representation records interaction between real objects like patches, Lewis and Fournier's hierarchy records interaction between 3D regions, which may or may not contain objects. Although these algorithms have different goals, the fact that a light field is defined as radiance on lines between two planes suggests that principles for a hierarchical radiance representation can be borrowed. In fact, as will be explained in detail in the next section, the hierarchical light field data structure also records interaction between regions of 3D space, like in Lewis and Fournier's work. But the space subdivision criterion and hierarchy generation procedure in hierarchical light fields are similar to procedures described in hierarchical radiosity.

3 Hierarchical light fields

In this section we first give an overall description of the hierarchical light field representation. We then proceed to describe and analyze the advantages of this representation.

3.1 Representation

Figure 1 shows an example of a 1D hierarchical light field. In general, a hierarchical light field consists of a *view tree* and an *object tree*. A node of a view tree represents a view cell, which is a cubic region in space containing possible viewpoints. A node of an object tree represents an *object cell*, which is a cubic region in space containing a set of objects. A view cell and an object cell may overlap. The root view cell contains all possible viewpoints, and the object cell contains all objects in the scene. In the most general case, the root view cell and root object cell refers to the same 3D region. A cell associated with a node can be subdivided into 2, 4, or 8 subcells depending on whether a 1D, 2D, or 3D hierarchy is used. The set of subcells of a given cell is represented by the children of the node associated with that cell. The whole view region and object region are subdivided into a hierarchy of cells.

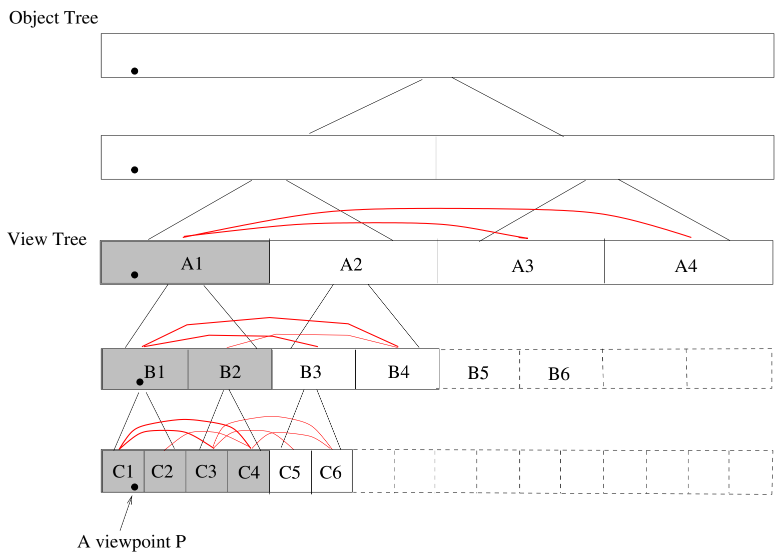

Figure 1: An example of hierarchical light fields with 1D subdivision. The structure consists of a view tree (shown in gray) and an object tree. (The view tree headed by node happens to be exactly overlapping the object subtree headed by A1, but in general the two trees corresponds to 3D regions which may or may not overlap) Each red arc corresponds to a link between a view cell and an object cell (and only the links looking toward right are considered here). Each cell has a fixed number of links to other cells (in this case, either 1 or 2 links per cell). To reconstruct a viewpoint P in C1, cells A1, B1, and C1 are traversed in order and all links associated with these three cells, which are {(A1,A4),(A1,A3),(B1,B4),(B1,B3),(C1,C4),(C1,C3)} are resampled. Images extracted from these links are then composited.

There are links connecting view cells and object cells at all levels of the hierarchy. A link is a 4D light field associated with a view cell and an object cell, parameterized by two planar *light slabs* as described in [13]. It represents all views of the object cell as seen from any position in the view cell. There can be a link between a view cell and an object cell only when: (1) the two cells do not overlap and the solid angle subtended by each cell with respect to the other cell is less than a user-specified threshold; and (2) no link has been established between ancestors of the two cells. Condition (1) means that two cells can only link when the ratio between their size and the distance between them is smaller than a threshold. Condition (2) is a result of the recurseive subdivision scheme presented in Section 4.1. The spatial density of samples on both slabs is constant for links at the same level. Links between parent nodes have a sample density half that of links between children one level down in the hierarchy. Cells connected by links at a higher level of the hierarchy are always separated by a larger distance than cells at a lower level.

To reconstruct the image from any viewpoint, the view tree is traversed from top to bottom and each node containing the viewpoint is visited. All links belonging to the nodes on the path are evaluated, and the appropriate image is extracted from each light field. These images are then composited together. To enable compositing, an alpha (opacity) value needs to be stored along with the color for each sample in the light field.

3.2 Advantages of hierarchical light fields

In this subsection we describe and analyze the advantages of the hierarchical light field representation, which include bounded image blurriness, sampling efficiency, and validity for all view positions.

3.2.1 Bounded image blurriness

While the light field function itself is a continuous function defined on the 4D space of oriented lines, the representation actually contained in memory only consists of discrete samples of this continuous function. As with any sampling process, aliasing can occur when the light field contains high frequency components. To minimize the aliasing effect, a 4D low-pass filter on the space of lines needs to be applied. As described in [13], this ray filter can be considered as the combination of a pixel filter and an aperture filter. It computes an average of radiances on all lines connecting a square region on the st plane bounded by a pixel, and a square region on the uv plane bounded by a (synthetic) aperture whose diameter is the spacing between adjacent camera positions (see Figure 2). Although the prefiltering anti-aliases the light field, it also introduces blurriness to the image.

The effect of aperture filtering is illustrated in Figure 3. A uv slab (the camera plane) and an st slab (the

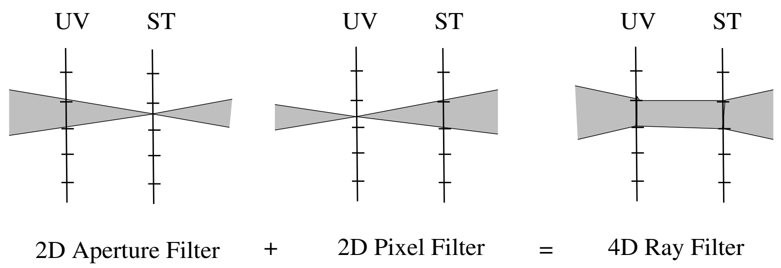

Figure 2: In order to avoid aliasing, a 4D low-pass filter is applied to the radiance funtion before sampling.

focal plane) are separated by a distance Z. Suppose that the cubic cells with which they are associated have sizes U and S, respectively, and that adjacent cameras are placed Δu apart. Consider an object point p at a distance Z_s from the focal plane. This point would be projected to different locations on the focal plane from different camera locations. Let the distance between the two projected points be Δs. Since the result of aperture filtering is the average of the two images, this point appears to occupy a region as wide as Δs on the focal plane after filtering. (Δs is usually referred to as the *circle of confusion*.) The spatial acuity of the human eye is usually measured by the *angle of confusion* $\Delta\theta$, the angle that the circle of confusion subtends at the eye [11], which is $\Delta s/Z$.

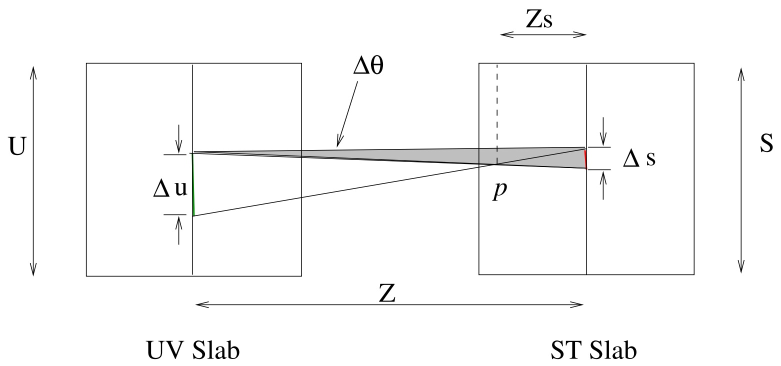

Figure 3: The effect of aperture filtering. A distance of Δu between adjacent cameras will cause a circle of confusion Δs. The blurriness of an image is related to the angle of confusion $\Delta\theta$, which is defined as $\Delta s/Z$.

Employing simple geometry, we can see that

$$\Delta s = \Delta u \frac{Z_s}{Z \pm Z_s}$$

and

$$\Delta s \approx \Delta u \frac{Z_s}{Z}$$

if $Z \gg Z_s$. Thus, the angle of confusion

$$\Delta\theta = \frac{\Delta s}{Z} \approx \frac{\Delta u}{Z} * \frac{Z_s}{Z}$$

It can be easily seen that the larger Z_s is, i.e. the farther away an object is from the focal plane, the larger the angle of confusion is. In a non-hierarchical light field, the angle of confusion $\Delta\theta$ is unbounded since there is no restriction on the value of Z_s. In the hierarchical light field representation, the objects that can be represented by a pair of slabs are restricted to be in the object cell associated with the st slab. This implies that Z_s has a maximum value of $S/2$. Also, $\Delta u = U/n_u$ where n_u is the preset number of sample in the u dimension. Therefore,

$$max(\Delta\theta) = \frac{1}{2n_u} * (\frac{U}{Z}) * (\frac{S}{Z})$$

where U/Z and S/Z are the solid angles subtended by the view cell and the object cell, respectively. Since both U/Z and S/Z must be below a threshold for a link (light field), as described in Section 3.1, the angle of confusion $\Delta\theta$ is bounded.

If we have a scene with objects at drastically different positions and use only one pair of slabs, as would be required in the regular (non-hierarchical) light field representation, then it is likely that no matter where the focal plane is placed, some object will be very far from the focal plane. Such an object will be severely blurred, since there is no restriction as to how far the object can be away from the focal plane. However, in the hierarchical light field representation, a pair of light slabs are only allowed to represent objects in the object cell where the st slab is located. This means that $max(Z_s) = 1/2S$, where S is the size of the cube, and thus that $max(\Delta\theta) = 1/2(S/Z) * (\Delta u/Z)$. As we go down or up levels of the hierarchy, the solid angle subtended by a cell is always kept below a threshold, i.e. (S/Z) is upper-bounded. $(\Delta u/Z)$ is also upper-bounded since the number of samples in the u dimension Δu is kept unchanged. Therefore, the maximum blurriness of source images is bounded when the hierarchical light field representation is used, and therefore a lower bound on image quality can be maintained. A corollary to this is that since the solid angle subtended by two adjacent light field samples in the hierarchy is bounded, the scene is sampled uniformly in terms of angular resolution.

3.2.2 Sampling efficiency

As stated previously, our goal is to be able to look from any point in the root view cell to any object in the object cell, with some lower bound on image quality. If both the root view cell and the root object cell are divided equally into n sections in each dimension, a brute-force solution would create a link from each of the n^3 view cells to each of the n^3 object cells, resulting a total of n^6 links. The hierarchical method should result in fewer links, since links between many cells separated by a large distance are clustered into one link between two bigger cells at a higher level of the hierarchy.

Refer again to the one dimensional case shown in Figure 1. It can be seen that all nodes share a similar linking pattern. Consider the linking pattern for any node in the interior of the tree, say B1, for example. Since the solid angle criterion forbids any neighboring cells to link, B1 can not link to its immediate neighbor B2 . The interactions between immediate neighbors must handled by B1's children. Similarly, B1 only needs to be responsible for interactions between its parent and the immediate neighbors of its parent. Generally speaking, a cell only needs to link to children of its parent's immediate neighbors, subject to the solid angle criterion. This implies that each node is connected to a fixed number of links, since the number of immediate neighbors of its parent is fixed. Therefore, the total number of links is proportional to the number of nodes in the tree, which is $\mathcal{O}(n^3)$.

It should also be noted that tree pruning can be used to further reduce the number of links in a hierarchy. If a node in the object tree contains no object, all links to that node and its children can be deleted. If it is known in advance that the viewer will not enter a specific view cell, all links to that view cell and its children can be deleted. In an environment where most of the objects are concentrated near a plane, the number of non-empty nodes is usually $\mathcal{O}(n^2)$. This means that the total number of links in a hierarchy for such a scene is only $\mathcal{O}(n^2)$.

Since all links of a node containing the viewpoint need to be evaluated during the image reconstruction process, having a fixed number of links per node also implies that the cost of rendering a frame at runtime is fixed and almost independent of the complexity of the scene.

3.2.3 Validity for all view positions

One of the limitations of the basic 4D light field representation is that the use of this representation results in spurious occlusion effects when the viewpoint enters the convex hull of an object. The hierarchical light field representation provides a framework that automatically divides the space into a hierarchy of cells with links between them. As described in Section 3.1, two overlapping cells are not allowed to link. (If a view cell and an object cell do overlap, they must both be subdivided and interaction may only occur between a child of the view cell and a child of the object cell that are not overlapping). Consequently, for any object cell, the view cells to which it is allowed to link are always outside the object cell; in other words, the viewer is always kept outside the convex hull of the set of objects being viewed, thus ensuring that spurious occlusion effects do not occur. The validity of any reconstructed view is always guaranteed within this framework.

4 Implementation issues

The section describes implementation issues in the generation of hierarchical light fields and the reconstruction process. Techniques for reduction of light field data size are also described.

4.1 Generation of hierarchical light fields

Generating a hierarchical light field consists of three steps. First, a space subdivision procedure is employed to divide the root view cell and the root object cell into a hierarchy of sub-cells. Appropriate links are generated for cells satisfying the solid angle criterion. Second, since each link needs to be parameterized using two light slabs, the appropriate slab parameters for that link, such as slab location, orientation and resolution, are calculated. Third, given the slab specification from step two, samples of each light field are then generated using some renderer.

The space subdivision procedure used in step one is very similar to the one used in hierarchical radiosity [10]. Given a view cell and an object cell, the solid angles subtended by each cell with respect to the other cell are computed. If both angles are less than the user-specified threshold, a link is created between the two cells. If either one of the angles is larger than the threshold, the cell which subtends a larger solid angle is divided into subcells, and the subdivision procedure is called recursively with the smaller subcells. The subdivision is terminated when some termination criterion, such as a minimum cell size or a maximum number of levels, is reached. The result of step one is a view tree and an object tree with appropriate links between nodes signifying a allowable interaction.

The next step is to parameterize the links. As mentioned earlier, a link between a view cell and an object cell is a 4D light field, defined on all rays from the view cell to the object cell and sampled at a preset sampling resolution. A link is allowed to represent views from the view cell looking at the objects in the object cell only. Using the notation of the two-plane parameterization in Levoy and Hanrahan's paper [13], we place the uv slab (camera plane) at the center of the view cell, and the st slab (focal plane) at the center of the object cell, axis-aligned and parallel to each other, as shown in Figure 4. The reason we place the focal plane in the center of the object cell is as follows. As we have shown in Figure 3 of Section 3.2.1, the larger the distance Z_s from an object to the focal plane is the more blurry its image becomes. By placing the focal plane in the center of the object cell, we limit the maximum value of Z_s to be one-half the cell size. If we place the focal plane elsewhere, the maximum value of Z_s is larger. Thus placing the focal plane in the center of the object cell minimizes the blurriness of images for that cell. The size of the slabs is such that they can catch all rays between the two cells. The sample density is determined by the level of the link and the initial sample density for the root level. The sample density of a link in each of the (u,v,s,t) dimensions is twice as large as that of links one level higher in the hierarchy. In addition to the slab parameters such as slab position and sample resolution, the bounding volume of the view cell and the bounding volume of the object cell are also recorded for each link.

Once the slab specification for all the links are generated, a rendering program is used to generate all the source samples of the light fields. The renderer generates an image for each camera position on the uv slab with the appropriate aperture and pixel prefiltering, in the same way in which a basic 4D light field would be authored. However, the renderer differs from a basic light field authoring program in two ways. First, since each link in a hierarchy only represents the objects in the object cell for that link, any geometry outside the object cell should not be rendered for that link. Thus 6 clipping planes are used in the hierarchical light field renderer to clip out the geometry outside the object cell. The second difference is that, in addition to red, green, and blue values, an alpha (opacity) value is also calculated and stored for

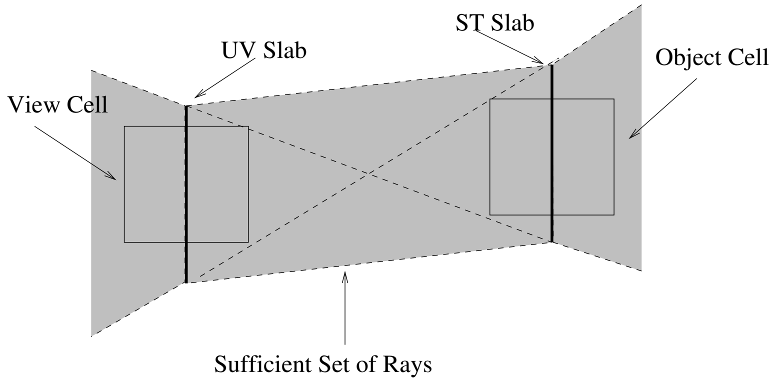

Figure 4: A link from a view cell to an object cell in the hierarchical light field representation. Slabs are placed in the center of each cell and aligned with the cell. The slab size should be big enough to cover any ray connecting a point in the view cell and a point in the object cell.

each sample. An alpha value of 0 indicates that there is no object points in the object cell that intersects the ray. An alpha of 1 indicates there is an intersection with the ray. Partial alpha values would occur at object edges after averaging multiple jittered rays during prefiltering. The r,g, and b values of each sample are pre-multiplied by the alpha of that sample and all filtering, resampling and compositing operates on these *associated color* values[2]. The reason we store an alpha and alpha associated colors rather than pure r, g and b values is to enable proper resampling and compositing in the reconstruction, as will be explained in the next subsection.

4.2 Reconstruction algorithm

A view from an arbitrary viewpoint in the initial root view cell is generated by traversing the view tree from the root node down to a leaf node. At each level, the node that contains the viewpoint is identified. Each link associated with that node is then resampled with respect to that view. The resampled images then need to be composited properly. A characteristic of hierarchical light fields is that links at a higher level are always linked to more distant object cells. Thus, traversing the tree nodes from top to bottom guarantees that objects at different levels are rendered in far to near order. If we pre-sort links of the same view cell in far-to-near order, the whole scene is rendered in far-to-near order.

Since the images are extracted from different light fields in a far-to-near order, they are composited using an algorithm described in detail by Blinn [2, 3].(The explanation of why associated colors should be used can also be found there.) Images with alpha values are overlaid into the frame buffer one by one, from back to front. Each pixel is composited with the pixel that is already in the frame buffer using an "over" operation.

However, the "over" operation is only suited for compositing two pixels that are uncorrelated, such as two points on different objects. As with any space-subdivision algorithm, some surfaces can be splitted into two sub-cells in the hierarchical light field representation. The surface points at the boundary of two sub-cells will be projected to a pixel in two images, each with partial alpha value. Since the two pixels are correlated, they should really be composited using the "add" operator rather than "over". If the "over" operator is used, the composited pixel would appear darker than what it should be and the boundary of two cells would appear to be a dark seam in the final composited image. One way to solve this problem is, in the authoring process, to use one bit of the alpha channel to indicate whether a pixel with partial alpha is a correlated pixel or uncorrelated pixel. Then in the reconstruction process, use the "over" or "add" operation accordingly. This may reduce the precision of alpha values but should solve the seam problem.

4.3 More efficient storage with slab combination

As described in Section 4.1, the general way to parameterize a link from a view cell to an object cell is to use two parallel slabs. The slabs are located in the middle of the cells with a certain sampling resolution. This parameterization scheme guarantees that the samples will cover the space completely. But there may be situations where it is not very efficient in terms of storage. In order for a pair of slabs to catch all of the rays between the two cells, the slab size must be bigger than the size of the cells. This causes neighboring slabs to overlap, as shown in Figure 5(a). Consequently, many rays will be recorded multiple times in the representation, such as the blue ray shown in the figure. This redundancy can be removed by combining the slab and its neighboring slab into one bigger slab, as shown in Figure 5(b). Another source of data inefficiency lies in that samples of an object cell behind another object cell linked to the same view cell will always be composited with samples of the cell in front of it. For example, samples in links (V,O1) and (V,O3) will always get combined in the reconstruction process, no matter where the viewpoint is. Since the purpose of dividing scenes into layers is to allow hierarchical sampling resolution, layers at the same level with comparable sampling resolution are not necessary. We can get rid of this extra layer by combining the slabs of the two cells into one. This is done by placing a slab directly between cells O1 and O3, as shown in Figure 5(d). Since slabs being combined are always at the same level of the hierarchy, the sample density on each slab should not change, but the combined slab size may need to be adjusted so that it will cover rays from the view cell to both object cells. Figure 5 only shows the combining of slabs of neighboring object cells. In fact, slabs of neighboring view cells linked to the same object cells can also be combined or composited in a similar fashion.

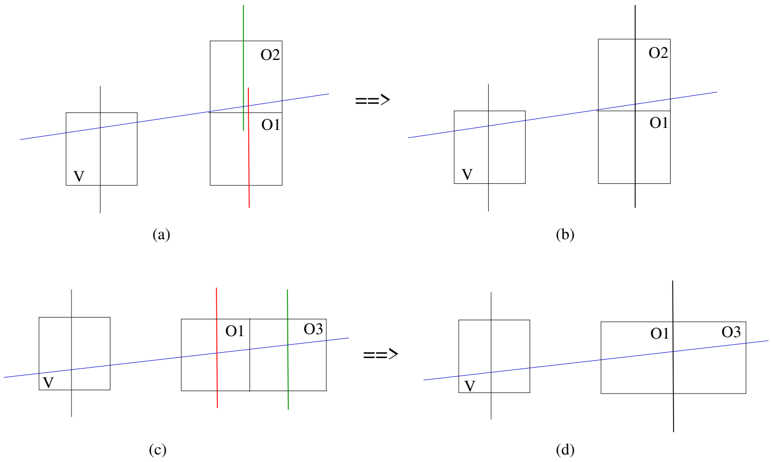

Figure 5: Combining slabs. (a) Links (V,O1) and (V,O2) have overlapping st slabs; (b) The overlapping st slabs in (a) are combined. (c) Links from the same view cell V to two different object cells O1 and O3, where O3 is behind O1. Samples of link (V,O1) will always overwrite (V,O3); (d) The overlapping st slabs in (c) are combined.

Slab combination is extremely helpful in a 2D or 3D hierarchy. In these situations, each cell is usually linked to many other cells, resulting in more redundant samples and longer processing time. By combining neighboring slabs into one big slab, the structure of the hierarchy is greatly simplified, and both the storage and rendering time are reduced. An example of an 2D hierarchy with slab combination is shown in Figure 6.

5 Results

We first demonstrate how hierarchical light fields can be used in a walkthrough of an environment. The scene in Color Plate I contains a texture mapped ground with 5 pairs of trees forming a path. We assume

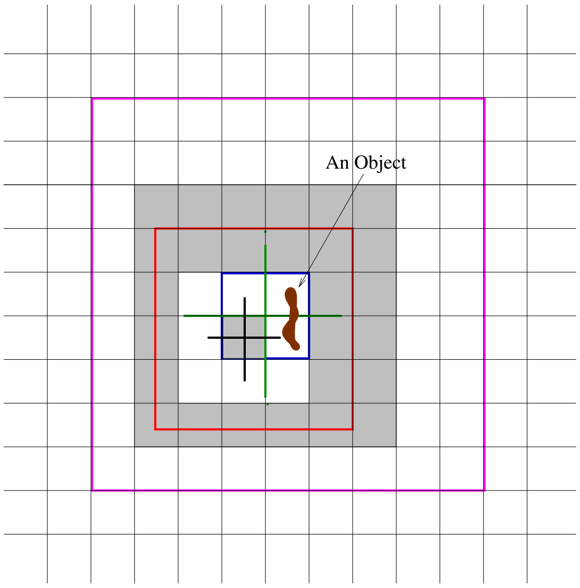

Figure 6: Hierarchical light fields with 2D subdivision with combined slabs. If this is an outward-looking scenario, the uv slabs and st slabs for links to the grey cell in the center are shown shown in black and red, respectively; The uv slabs and st slabs for links to its parent (the blue cell) are shown in green and magenta, respectively. Note each four st slabs forming a square are combined from 24 smaller slabs. Each view cell is surrounded by rings of light fields, and each light field can be viewed as an array of environment maps. In an inward-looking scenario, each object cell is surrounded by rings of light fields where the uv slabs got combined. Notice that in an inward-looking scenario, an object can be broken into several portions and each portion can have its own focal plane (i.e. st slab.)

that the viewer only navigates along the path and looks forward. Therefore, a 1D hierarchical light field is created. The hierarchical structure is similar to the one shown in Figure 1 except that it has one more level, so there are 8 leaf nodes in the view tree. Links to the same view node are combined into one as described in Section 4.3 and shown in Figures 5(c) and (d). There is exactly one link per view node and a total of 15 links. Each link is a light field from a view node to a (combined) object node. The solid angle criterion is set so that cells can only link if there is at least one cell in between them. The sampling resolution is set so that the (n_u, n_v, n_s, n_t) values for a nominal pair of slabs are (24,24,96,96). The total raw data size is 3052 million RGBA samples, which takes 12.2 GB. Employing a vector quantizer using a 2 byte index for a block of 2x2x2x2 RGBA samples, it takes approximately 404MB in memory. The scene contains about 460 thousand polygons. Using a 195 MHz R10000 processor and an Infinite Reality Engine graphics board, it took about 24 hours to pre-generate all the source samples with each sample uvst-filtered from 16 subsamples.

The first advantage of hierarchical light fields shown by this example is that hierarchical light fields extend the original light field concept to be able to handle scenes with occluders. As the viewer walks down the path, he is walking from cell to cell and the light fields satisfying the free space assumption for view points in that cell are automatically chosen and rendered. In this example, the view tree has four levels and each view is composited from four resampled light fields, one at each level from the top to the bottom of the tree. Color plate I shows three views from a walkthrough and the resampled light fields from which they are composited at each level.

Another of the advantages of using a hierarchy of light fields is the sampling efficiency. By subdividing the scene into hierarchical layers, light fields at a higher level (corresponding to more distant objects) are sampled at a coarser resolution and cover a larger region of space. This means that they take less storage and are valid for a larger range of view points. As can be seen in Color plate I, light fields for a higher level view node are re-used for all of its child views. As the viewer moves from cell to cell, only the lower level light fields need to be switched.

An alternative way of doing a walkthrough could be to just create a 4D light field for each of the leaf cells for all objects surrounding that cell (or equivalently, a planar array of environment maps.) As the viewer moves within that cell, the regular 4D light field resampling is performed. When the viewer moves to another cell, the light field being resampled also switches to the light field for the other cell. It is a simple scheme with a depth complexity of 1. However, such a representation is not very efficient in sampling, because far-away objects are duplicated in every cell's light field. For example, in Figure 1 there are 8 leaf nodes and we need to create 8 arrays of environment maps, one for each leaf node. Each image in the environment map includes all objects, near and far. In other words, objects that are contained at the root level in a hierarchical light field would require 8 copies in an independent map implementation.

The advantage of hierarchical light fields with respect to data size is dependent on the depth complexity of the scene. For sparse scenes, most of the object octree's leaf nodes would not contain any geometry. For these empty nodes, a light field does not need to be stored. In contrast, a dense scene with high depth complexity (such as a tropical Brazilian rain forest) has significant geometry in every leaf node. This would lead to dense light fields at every level of the hierarchy, and negate the savings of being able to reuse data at the higher levels. For example, our 1D tree light field has geometry in almost every node of the object octree. As a result of this object density, we save only 26% over an equivalent set of independent light fields. However, the savings can be much greater when the scene contains open spaces.

We expect (although we have not yet verified) that a 3D spatial subdivision will lead to far greater differences between a hierarchical light field and a set of independent light fields (or array of environment maps), since many more of the links should be empty. For example, in our 1D subdivision in Figure 1, we divide space into strips along the path. However, every strip contains a piece of the ground; thus, every cell in the object hierarchy contains geometry, and all 15 light field links must be present, whether a tree occupies the cell or not. In a 3D spatial subdivision, a ground plane would occupy only $1/N$ of the cells, where N is the number of subdivisions along the vertical axis. Other cells will inevitably contain geometry wherever there are objects, but there will also be many empty cells that would not require any light field links. If there are many links that connect to empty cells that are nearby, the use of a hierarchical light field is extremely beneficial in comparison to the use of an array of environment maps. That's because in an array of environment maps, a 4D light field has to be generated for every leaf node with the highest sampling resolution, whereas in a hierarchical light field representation, many of those high-resolution links are pruned and links that connect distant objects are sampled at a lower resolution.

Figure 7 (a) summarizes the number of non-transparent samples in the light fields for each level of the hierarchy for the 1D tree example. Part (b) compares these counts to the number of non-transparent samples that an array of independent environment maps would contain. It can be seen that the total number of non-transparent samples in a hierarchical light field is smaller than the total number of non-transparent samples in an array of environment maps for a scene with moderate depth complexity.

The second example, shown in Color Plate II, is an inward-looking scenario where an object is being observed from all directions and at different distances. The root object cell is a cube that contains the dragon, while the root viewing cell is a much larger cube that encloses the object cell. A hierarchy is generated, similar to the one shown in Figure 6, but in 3D rather than 2D. Thus, each object cell is surrounded by 6 uv slabs forming a cube, and each object cell at a higher level is surrounded by 6 uv slabs

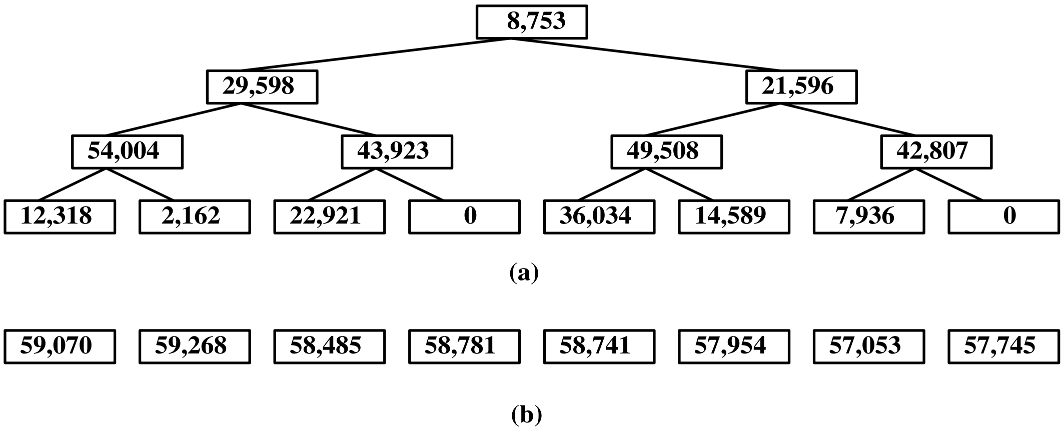

Figure 7: Comparison of (a) the number of non-transparent pixels in a 1D hierarchical light field to (b) an array of independent light fields of the same resolution. The hierarchical light field has a total of 346,158 pixels, while the array has 467,097 pixels.

that are twice as big in each dimension and twice as far from the object cell. This structure is similar to a Delta tree [6]. The difference is that, first, the viewpoints are defined on cubic surfaces rather than on a sphere; second, viewpoints of different levels reside on cubes of different sizes. Since hierarchical light fields divide both view cells and object cells, different levels of viewpoints are centered around different levels of object cells. The dragon is placed at the location of the brown object in Figure 6 and three levels of hierarchy are generated. The sampling frequency is set to a (u,v,s,t) resolution of (6,6,128,128) for a slab pair of size 1x1 at a distance of two times the leaf node size. There are a total of 1833 million RGBA samples and 230MB of memory after a 32:1 vector quantization. The model contains 220 thousand polygons. The original source images were rendered by a ray tracer.

Color Plate II shows an example of an inward looking hierarchical light field. In this case, full 360-degree light fields surrounding the object are created. Links connected to the same object cell are combined. The resulting hierarchy can be thought of as follows. The whole dragon has been subdivided into a hierarchy of sub-object cells, and each cell has a ring of uv slabs surrounding it at different resolutions.

Another advantage of using hierarchical light fields is proper sampling and filtering. In a regular 4D light field where one pair of slabs is used to view an object, as the viewer moves farther and farther back from the viewing plane, the light field undergoes minification. Similar to unmipmapped textures, the lack of proper filtering produces aliasing. In a hierarchical light field, light slabs are created for a particular angular resolution. Thus, they have an appropriate prefiltering kernel, and aliasing is greatly reduced. Color Plate II (a) shows images of the dragon at three resolutions, demonstrating this relatively alias-free minification.

It is also interesting to compare the dragon example using hierarchical light fields with the same example using a basic 4D light field. One notable artifact of the basic light field is that, from any angle, the st plane passes through the middle of the dragon, roughly perpendicular to the viewing direction. The st plane has significantly higher resolution than the uv plane, in order to maximize detail. However, this means that blurriness increases as the geometry gets farther away from the st plane. From a side view, the dragon is fairly flat, so it does not diverge much from the st plane. Therefore, the image is sharp. However, from a head-on view in Color Plate II (b), the head and tail of the dragon are a substantial distance from the st plane. This causes excessive blurriness in these regions. In contrast, the hierarchical light field uses several different st planes to represent the different regions of the object volume. Thus, the geometry is always closer to its local st plane, and more in focus. This makes the head and tail of the dragon as sharp as the middle section.

6 Discussion and future work

We have proposed a new image-based representation, that extends the basic 4D light field representation into a hierarchy of light fields. Regular spatial subdivision is used to subdivide the scene into a tree-structure, and each node in the tree is linked to a fixed number of other nodes based on a minimum solid angle criterion. Each link is a 4D light field representing objects in one node as seen from the other nodes. To reconstruct an arbitrary view, links belonging to all nodes that contain the viewpoint are visited from top to bottom in the tree. The appropriate image as seen from the reconstructed view point is resampled from each of the light fields, and all of the images are composited in a far-to-near order. This representation basically stores multiple samples per ray. With the reconstruction scheme always selecting the valid subset of samples, the representation overcomes the free space limitation in the basic light field representation, and allows the viewer to move into and out of the convex hull of a set of objects. The hierarchy, parameterization scheme, and sampling resolution of the light field are generated systematically so that a lower bound on the image quality can be maintained. The cost of rendering a view is nearly constant and independent of the scene complexity. Thus, it is well-suited to visualizing complex environments.

It is interesting to compare the hierarchical light field representation with some other image-based representations. For example, as described in the last section, a planar array of environment maps is a special case of hierarchical light fields with only one level of hierarchy. Another related representation is priority rendering, which also divides the scene into layers of environment maps. However, the advantage of hierarchical light fields over priority rendering is that hierarchical light fields are a more general framework which offers more freedom in sampling the radiance function. By placing the focal plane at the center of object cells rather than at infinity, a layer is valid for a larger range of views than the layers in priority rendering.

Layered depth images is another representation that allows construction of new views from a set of image samples. It organizes the color and depth information from a number of views as an array of color and depths per pixel as seen from one camera position. Both the hierarchical light field and LDI sort the scene by depth. The difference between the depth sorting approaches used by hierarchical light fields and LDI is that in hierarchical light fields, the sorting is done for spatial regions and the sorting resolution is regular and hierarchical; in LDI the sorting is done by objects and the sorting resolution is irregular and depends on the depth complexity of the scene. However, LDI has two disadvantages. First, resampling images twice can cause potential quality degradation in reconstruction. Second, LDI is unable to model radiance changes as a function of direction, since the entire scene is organized with respect to one viewpoint.

One major limitation of hierarchical light fields is that they require a large number of samples, since each light field requires a high sampling density to avoid image blurriness. The current tree-structured vector quantization is a valuable first step toward compression, but further compression based on removal of redundancy in the light field data has not yet been explored. An important part of the future work is to determine more specialized compression schemes for light fields. For example, coherence between images taken from very close views can be removed by inter-frame coding, such as that used in MPEG. In addition, occlusion culling can be used to remove samples that are occluded by another sample along the same ray and that will never be used in reconstruction. Another interesting avenue of research is to approximate the 4D light field function by lower-dimensional functions (such as texture-mapped planes) in regions with relatively small variation in a particular dimension.

One characteristic of hierarchical light fields is that the solid angle criterion forbids the linking of two cells that are very close to each other. Two close cells must subdivide and the solid angles are evaluated at the next finer level, until some termination criterion is reached. This implies that, in order to view very close objects, the hierarchy needs to have more levels, which results in an even larger number of samples. One alternative is to render the near objects from geometry rather than light fields and to composite the geometry with rendered light fields. One issue of using this approach is the question of how to correctly avoid seams in compositing.

The subdivision of a scene into layers or subcells is not hard for synthetic environments where the geometry of the scene is known. But the subdivision of a real scene and the acquisition of such light fields appears to be a challenge, since the exact geometry of the real scene is hard to acquire. However, since the depth information is used mainly to sort the object into different depth bins (high precision is not required). It may be possible to use vision algorithms to separate a composited image into layers.

Currently, all images in a basic light field or in hierarchical light fields are pre-generated. Another area of future research is to use light fields as dynamically generated 4D image sprites. As the viewer moves through an environment, a hierarchical light field structure as described in this paper is populated with images from previous frames and images from lazy evaluation. Compression can be done during the idle cycles of the processor. Later, viewing of the complex environment can be accelerated a great deal by extracting and compositing images from the hierarchical light fields.

Acknowledgments

References

[1] ADELSON, E. H., AND BERGEN, J. R. *Computational Models of Visual Processing*. MIT Press, Cambridge, MA, 1991, ch. 1 (The Plenoptic Function and the Elements of Early Vision).

[2] BLINN, J. Compositing, part 1: Theory. *IEEE Computer Graphics and Applications* (Septempber 1994), 83–87.

[3] BLINN, J. Compositing, part 2: Practice. *IEEE Computer Graphics and Applications* (Novenber 1994), 78–82.

[4] CHEN, S. E. Quicktime vr - an image-based approach to virtual environment navigation. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '95)* (1995), pp. 29–38.

[5] CHEN, S. E., AND WILLIAMS, L. View interpolation for image synthesis. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '93)* (1993), pp. 279–288.

[6] DALLY, W., MCMILLAN, L., BISHOP, G., AND FUCHS, H. The delta tree: An object-centered approach to image-based rendering. Tech. rep., Massachussetts Institute of Technology, 1996.

[7] GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. F. The lumigraph. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '96)* (1996), pp. 43–54.

[8] GORTLER, S. J., HE, L.-W., AND COHEN, M. F. Rendering layered depth images. Tech. Rep. MSTR-TR-97-09, Microsoft Research Advanced Technology Division, Redmond, WA, March 19 1997.

[9] GREENE, N., AND KASS, M. Approximating visibility with environment maps. Tech. Rep. 41, Apple Computer, Inc., Cupertino, CA, 1993.

[10] HANRAHAN, P., SALZMAN, D., AND AUPPERLE, L. A rapid hierarchical radiosity algorithem. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '91)* (1991), pp. 197–206.

[11] KINGSLAKE, R. *Optics in Photograpy*. SPIE Optical Engineering Press, Bellingham, WA, 1992, ch. 5 (Depth of Field).

[12] LAVEAU, S., AND FAUGERAS, O. 3-d scene representation as a collection of images. Tech. Rep. Technical Report RR-2205, INRIA - The French Institute for Research in Computer Science and Control, February 1994. Available from http:www.inria.fr.

[13] LEVOY, M., AND HANRAHAN, P. Light field rendering. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '96)* (1996), pp. 31–42.

[14] LEWIS, B., AND FOURNIER, A. Light-driven global illumination with a wavelet representation of light transport. In *7th Eurographics Workshop on Rendering* (1996).

[15] MARK, W. R., MCMILLAN, L., AND BISHOP, G. Post-rendering 3d warping. In *Symposium on Interactive 3D Graphics* (Providence,RI, 1997), pp. 7–16.

[16] MCMILLAN, L., AND BISHOP, G. Plenoptic modeling: An image-based rendering system. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '95)* (1995), pp. 39–46.

[17] REGAN, M., AND POSE, R. Priority rendering with a virtual reality address recalculation pipeline. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)* (1994), pp. 155–162.

[18] SEITZ, S. M., AND DYER, C. R. Physically-valid view synthesis by image interpolation. In *IEEE Computer Society Workshop: Representation of Visual Scenes* (Los Alamitos, CA, June 1995), IEEE Computer Society Press, pp. 18–27. Available from ftp.cs.wisc.edu.

[19] SHADE, J., LISCHINSKI, D., SALESIN, D. H., DEROSE, T., AND SNYDER, J. Hierarchical image caching for accelerated walkthroughs of complex environments. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '96)* (1996), pp. 65–74.

Viewing Complex Environments with Hierarchical Light Fields

Xing Chen, Lucas Pereira, Pat Hanrahan

Computer System Laboratory, Stanford University

Category: research

Format: print, video

Contact:	Xing Chen
	381 Gates Building
	Stanford, CA 94305
phone:	(650) 723-0618
fax:	(650) 723-0033
email:	xcchen@graphics.stanford.edu

Estimated # of pages: 8

Keywords: image-based rendering, light field, spatial hierarchy

We propose a hierarchical light field framework in which it is possible to deal with complex environments that cannot be simply represented by a basic 4D (non-hierarchical) light field. The hierarchical light field framework extends the basic light field structure by defining bounds on the region of freespace over which a light field is valid. It consists of multiple light fields hierarchically placed in 3D space, with each light field only sampling part of the scene. By compositing the images from the subset of light fields that are valid for any viewpoint, the viewer is given much more freedom in the range of possible views compared to the basic light field representation. The sampling resolution is hierarchical and view dependent to guarantee efficient sampling, which avoids overrsampling and aliasing during minification. The hierarchy and sampling parameters are generated systematically so that a given lower bound on image quality can be maintained. Also, the rendering time is independent of the scene complexity. Because of these advantages, the hierarchical light field framework is suitable for visualizing complex environments.

In this paper, the advantages of hierarchical light fields are described and analyzed, along with techniques for reduction of light field memory requirements. A detailed description is given of the steps undertaken during generation of hierarchical light fields and the reconstruction process. The capabilities and advantages of the representation are shown by examples of an immersive walkthrough and a 3D fly-around of an object.