Ibero-American Symposium on Computer Graphics - SIACG (2006), pp. 1–4
P. Brunet, N. Correia, and G. Baranoski (Editors)

Introducing Extended and Augmented Light Fields for Autostereoscopic Displays

Alejandro Domingo, Miguel Escrivá, Francisco Abad, Roberto Vivó, Emilio Camahort

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia, Spain
{adomingo, mescriva, fjabad, rvivo, camahort}@dsic.upv.es

Abstract

Autostereoscopic displays have recently received a lot of attention because they allow multiple users to view true 3D images of the same object. These devices usually display either 3D volumetric data or 4D light-field data. In this paper we address the issue of representing, building and rendering 4D light-field models like those used in autostereoscopic displays. We present a representation and algorithms for 4D light-field models based on the direction-and-point parameterization, a parameterization with uniformity properties that produce better renderings. Then, we improve on the representation by adding depth information and using it to reduce the amount of storage required for the light-field. Finally, we show how depth information can be used to compose multiple light-fields models and integrate geometric information within the light field.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation–Display algorithms I.3.3 [Computer Graphics]: Picture/Image Generation–Viewing algorithms I.3.6 [Computer Graphics]: Methodology and Techniques–Graphics data structures and data types

1. Introduction

3D spatial and autostereoscopic displays are becoming the subject of recent research efforts in Computer Graphics hardware [PPK00, BFB*05]. The goal is to provide one or more viewers with a 3D image of the object of interest. For these displays this usually requires rendering, storing and displaying 4D light-field data [IMG00, YCH*05].

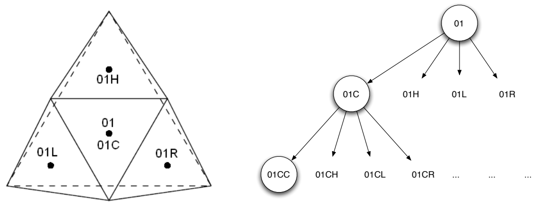
Formally, a 4D light field represents the radiance flowing through all the lines that pass through a scene [LH96, GGSC96]. For a given wavelength, we can represent a static light field as a scalar function $L(s,t,u,v)$ that gives radiance along a line intersecting two parallel planes with coordinates (s,t) and (u,v). This two-plane parameterization is known to introduce biases in the representation of the light field.

Alternative representations use isotropic parameterizations that uniformly sample the light field [CLF98]. We have implemented a light-field modeling and rendering system based on one of them, the direction-and-point parameterization (DPP) [EDA*06]. The system only supports modeling and rendering of single light-field objects with no geometric information.

Van der Linden [vdL03] proposes a rendering method for light-field models with no geometric information. Using a 3-pass algorithm he can also embed multiple light-field instances in a geometric scene. We propose extending our DPP-based representation with depth information to support multiple objects and geometric information.

This paper describes these extensions and reports work in progress in multiple light-field rendering and light-field models augmented with geometry. Our goal is to efficiently build and render these models in order to display them on autostereoscopic displays.

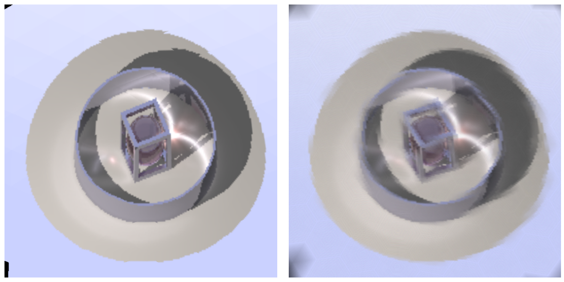
We begin describing our light-field representation and its associated modeling and rendering techniques. Then we present light fields extended with depth information and a new rendering algorithm. In Section 4 we introduce multiple light-field rendering and light fields augmented with geometric information. The paper finishes with some conclusions.



(a) Encoding of the subtriangles of the original icosahedron triangle labeled *01*.

(b) The hierarchical structure of the subdivision: *01*, *01C* and *01CC* represent (roughly) the same direction, so they share the same image.

Figure 1: *Hierarchical structure, encoding and storage of the light field's image data.*



(a) Without interpolation.

(b) With interpolation.

Figure 2: *Two renderings of a light-field model with 20480 directional samples. The model was obtained by ray tracing a geometric model with multiple light sources, reflection, refraction, and caustics. The spatial resolution of each light-field image is 256×256 pixels. The rendering resolution is 512×512 pixels.*

2. DPP-Based Light Fields

Our DPP-based light-field implementation is characterized by its representation, its storage scheme, and its construction and rendering algorithms. The representation relies on a (nearly) uniform discretization of the set of all directions in 3D Cartesian space. We obtain such a discretization by subdividing the surface of the sphere into (nearly) equilateral, (nearly) identical spherical triangles. The center of each triangle corresponds to a directional sample of the representation. This sample approximates the directions that pass through the triangle.

The subdivision starts with the icosahedron, with 20 triangles. Then each triangle is recursively subdivided into four new triangles by adding three new vertices (see Figure 1(a)). Once generated, an encoding is defined for the triangles and the images in the hierarchy. Level-0 triangles are labeled 01 to 20; subtriangles are labeled by adding one of four letters: *C*, *H*, *L* and *R* to their parent's label (see Figure 1).

Given the discretization of directional space, our DPP light field store one image per direction. The images are obtained by rendering parallel projections of the object of interest taken along the directions of the discretization. Images associated to central subtriangles are projections along the same direction as their parents. So they are only represented and stored once, for the parent triangle.

Generation and Storage. To build a DPP light field model we center the target object at the origin and scale it to fit inside of the unit sphere. Then we render each parallel projection onto an orthogonal projection plane centered at the origin. The resulting images are stored in a linear structure sorted according to the hierarchy of directional samples.

The representation requires a large amount of storage. But we can use lossless compression to obtain a 10:1 rate. Or we can use lossy (JPEG) compression to achieve 40:1 rates depending on the amount of background pixels.

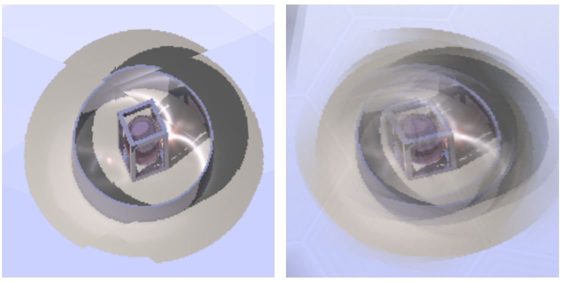
Rendering. The DPP rendering algorithm is an adapted version of the Lumigraph algorithm [GGSC96]. Given the viewing parameters, it starts by placing an imaginary sphere centered at the eye position. The sphere is tessellated into triangles exactly like the sphere representing the set of directional samples of the light-field model. The rendering algorithm determines which pencils of directions intersect the viewing frustum. For each of those pencils, it renders its associated light-field image on the portion of the frustum intersected by the pencil. To do that the triangles of the sphere are texture mapped using the light-field images.

We leave the reprojection and display steps of the algorithm to the rendering hardware. We render each visible triangle with a texture map containing a portion of its light-field image. The texture-map coordinates are obtained by projecting out the triangle's vertices onto the image's supporting plane. This rendering algorithm can be extended to perform interpolation between images associated to neighboring triangles. Figure 2 shows a DPP-based light field generated and rendered with our system.

This representation has a major drawback. To obtain high-quality renderings it uses many images, more than 20000. Uncompressed, these images require 3.7 GBytes of storage. Using less images produces artifacts due to discretization errors (see Figure 3). Seams are visible across triangle boundaries when rendering without interpolation. If we use interpolation the seams turn into ghosting. Lower image resolutions may also be used, but they produce pixelated images.

3. Extended Light Fields

An alternative to higher-resolution light fields uses depth information and lower directional resolutions. Gortler et al. implemented depth-corrected light-field rendering using a representation extended with a low-resolution geometric proxy of the target object [GGSC96]. Shade et al. introduced



(a) Seams. (b) Ghosting.

Figure 3: *Seams and ghosting in a DPP light-field representation of 1280 directional samples.*

layered depth images, images extended with a depth value associated to each pixel [SGHS98]. We use a combined approach.

We extend our representation with depth maps. For each directional sample we store the image and a depth map. Depth maps associate a depth value to each pixel of the light-field images. Given a pixel the depth map stores the orthogonal distance from the geometric object's surface to the image plane. The plane is tangent to the object's bounding ball and orthogonal to its associated directional sample.

Modified Rendering Algorithm. Using depth information we implement an improved rendering algorithm that computes accurate texture coordinates for the visible triangles of our representation. The algorithm is illustrated in Figure 4. For each vertex of a triangle, we construct a ray that starts at the eye and passes through the vertex. The ray is then cast into a volumetric grid, where the depth values represent a surface approximating the object's geometry.

Intersection points are used to compute more accurate texture coordinates for the triangles of the light-field representation. Often we need to subdivide the triangles because a ray misses the object or because the difference between the depths along two rays is too large. This algorithm virtually eliminates the incidence of seams (see Figure 5). Using depth maps can reduce the amount of storage required by a light field's radiance data to $1/4$ or $1/16$.

There are problems with this rendering algorithm. First, it does not support rendering multiple light-field models. Multiple light-field models can be easily implemented by storing more than one image per directional sample. This approach produces a 5D light-field representation. Second, the rendering algorithm does not support rendering of both geometry and light fields.

4. Augmented Light Fields

We want to be able to compose light-field models and add relevant geometric information such as labels, tags and regular objects. To achieve these goals we propose using depth maps, multiple images per directional sample, and improved rendering algorithms.

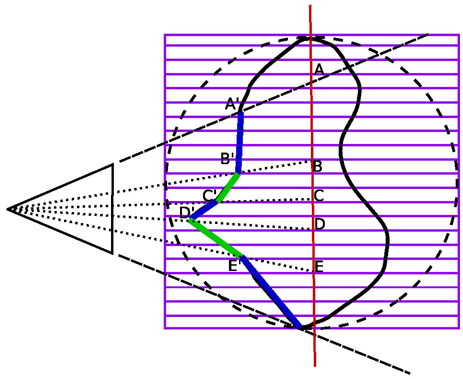
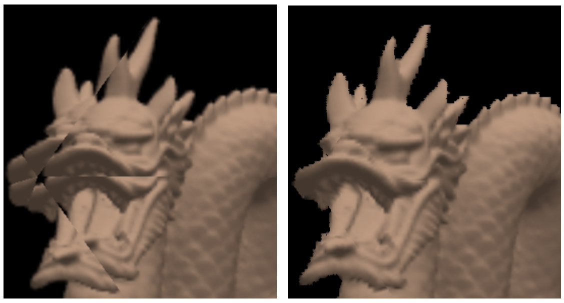


Figure 4: *2D analogy of the depth-map approximation to an object's surface. At rendering time rays are cast starting from the eye and passing through the triangles' vertices. Instead of intersecting the rays with the support plane of our original representation (points A, B, C, ...), we use a depth map to get a better approximation to the object's surface (points A', B', C', ...).*



(a) Without depth maps. (b) With depth maps.

Figure 5: *Renderings of our light-field model with 1280 directional samples.*

Light-Field Composition and Multiple Light-Field Rendering. To handle multiple light fields we store multiple images per directional sample. Images are sorted according to their position in 3D space along each direction. For each visible triangle we now reproject multiple images. And these images must be drawn in proper back-to-front order.

Planar geometric proxies, as in our first algorithm, offer a poor approximation to the objects' geometries (see Figure 6). To solve this problem we need to use depth information. At rendering time we compute for each light-field object the depth associated to each visible triangle vertex. We do this using the algorithm described in the previous section. Then, we use the depths to move the triangle vertex coordinates away from the eye. If the object is closer the triangle will be drawn closer to the viewer, thus properly handling visibility. The algorithm now draws one, possibly subdivided triangle for each directional sample and each light-field model.

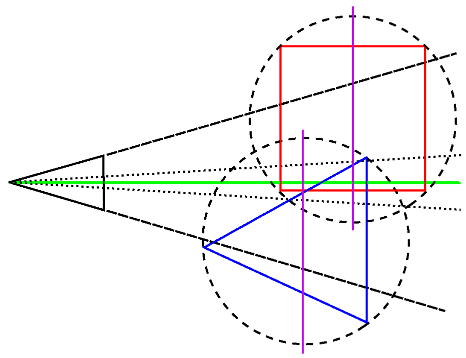


Figure 6: *The red square is in front of the blue object along the green direction. However, its proxy (in pink) is not, and the blue object appears in front of the red square in a multiple light-field rendering without depth information.*

Integrated Geometry and Light-Field Rendering. Geometry and light-field rendering can also be accomplished using depth information associated to the light-field data. Again, texture-mapped triangles are generated for each directional sample and each light-field model. They are drawn in 3D space at roughly the same location as the surface of the light fields' original geometric objects. Then the geometric objects are drawn.

Since triangles approximate the true geometry of both light-field and geometric objects, the result of the algorithm is correct. Furthermore, the objects can be drawn in any order regardless of type. The result is the same. We are currently implementing this new algorithm based on our depth-map based renderer for single light fields. The new implementation will handle multiple light fields (5D light fields) as well as hybrid geometry and light-field models.

5. Conclusions and Future Work

We have introduced extended and augmented light-field models. Extended models store depth information associated to the radiance samples of the light field. We render them using a depth-correction algorithm that provides higher image quality and requires substantially less radiance storage.

Augmented models support multiple light-field objects, as well as models with integrated geometry and light-field representations. We propose a rendering algorithm that extends the algorithm for depth-mapped light fields. The algorithm is almost implemented and we expect to have results within one month. Our ultimate goal is to be able to display our light-field models using autostereoscopic displays.

We also expect to improve our representation and rendering algorithm to handle multiresolution. Our directional discretization algorithm already supports multiresolution by hierarchically subdividing the sphere. Spatial multiresolution can be implemented using texture mipmaps. Multiresolution can be used to build non-uniform models and to adaptively control a model's rendering frame-rate.

Acknowledgements

This work was partially supported by grant TIN2005-08863-C03-01 of the Spanish Ministry of Education and Science and STREP project IST-004363 of the 6th Framework Program of the European Union.

References

[BFB*05] BALOGH T., FORGÁCS T., BALET O., BOUVIER E., BETTIO F., GOBBETTI E., ZANETTI G.: A scalable holographic display for interactive graphics applications. *IEEE VR 2005 Workshop on Emerging Display Technologies* (March 2005).

[CLF98] CAMAHORT E., LERIOS A., FUSSELL D.: Uniformly sampled light fields. In *Proc. Eurographics Rendering Workshop '98* (1998), pp. 117–130.

[EDA*06] ESCRIVÁ M., DOMINGO A., ABAD F., VIVÓ R., CAMAHORT E.: Modeling and Rendering of DPP-Based Light Fields. In *Proceedings of GMAI'2006* (London, UK, July 2006), IEEE Computer Society.

[GGSC96] GORTLER S. J., GRZESZCZUK R., SZELISKI R., COHEN M. F.: The lumigraph. In *Proc. SIGGRAPH '96* (1996), pp. 43–54.

[IMG00] ISAKSEN A., MCMILLAN L., GORTLER S. J.: Dynamically reparameterized light fields. In *Proc. SIGGRAPH '00* (2000), pp. 297–306.

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proc. SIGGRAPH '96* (1996), pp. 31–42.

[PPK00] PERLIN K., PAXIA S., KOLLIN J. S.: An autostereoscopic display. In *Proc. SIGGRAPH '00* (2000), pp. 319–326.

[SGHS98] SHADE J. W., GORTLER S. J., HE L.-W., SZELISKI R.: Layered depth images. In *Proceedings of ACM SIGGRAPH'98* (July 1998), pp. 231–242.

[vdL03] VAN DER LINDEN J.: Multiple light field rendering. In *Proc. 1st international conference on Computer graphics and interactive techniques* (Melbourne, Australia, 2003), pp. 197–202.

[YCH*05] YANG R., CHEN S., HUANG X., LI S., WANG L., JAYNES C.: Towards the light field display. *IEEE VR 2005 Workshop on Emerging Display Technologies* (2005).