A Warping-based Refinement of Lumigraphs

Wolfgang Heidrich, Hartmut Schirmacher, Hendrik Kück, Hans-Peter Seidel

Computer Graphics Group
University of Erlangen
{heidrich,schirmacher,hkkueck,seidel}@immd9.informatik.uni-erlangen.de

ABSTRACT

Light fields and Lumigraphs have recently received a lot of attention in the computer graphics community, since they allow to render scenes from a database of images very efficiently. While warping based approaches are still too slow to achieve truly interactive frame rates, Lumigraph rendering can easily achieve tens of frames per second in full screen on a contemporary graphics workstation. On the down side, high resolution Lumigraphs require large amounts of storage and are expensive to acquire, while low resolutions yield low quality images with lots of blurring.
In this paper, we propose to refine an initial sparse Lumigraph through the use of a specialized warping algorithm. This algorithm increases the resolution of the Lumigraph by inserting new views, which can then be used to generate higher quality images through normal quadri-linear interpolation. This way, storage requirements and acquisition costs remain low, but image quality is increased drastically.

Keywords: image based rendering, image warping, light field, Lumigraph, view morphing

1 Introduction

The light field [LH96] and Lumigraph [GGSC96] approaches to image based rendering are based on a dense sampling of the plenoptic function [AB91], which describes the flow of light in a scene. The advantage of this approach is that images of the scene from new camera positions can simply be computed by interpolating radiance values in 4 dimensions. This is a very inexpensive operation that can be further accelerated through the use of OpenGL-compliant graphics hardware, as has been shown in [GGSC96].

The disadvantage of this method is that the image quality is only good as long as the sampling density is extremely high, which, of course, means large memory requirements, and, for transmission, the need for extremely large bandwidths. A Lumigraph of moderate resolution can easily have a size of multiple Gigabytes.

One approach for solving this problem is to compress the Lumigraph, for example using vector quantization, like in [LH96]. This does reduce the amount of memory required for final storage, but the initial memory consumption before compression is not changed.

Furthermore, the acquisition of large amounts of images is expensive and time consuming. For example, if a light field is to be generated from a globally illuminated synthetic scene, the rendering time per frame can easily exceed several hours, especially if specular effects are to be simulated. The acquisition costs for real world scenes also strongly depend on the number of images. Either a photo camera has to be carefully positioned for each image, as in [LH96], or the images are generated through *rebinning* the information from video streams [GGSC96]. In the latter case longer video streams are required for higher resolutions, and the computational effort for rebinning grows drastically.

An alternative approach to image-based rendering are warping methods that only use a few initial images. Prominent examples for this class of algorithms are view interpolation [CW93], plenoptic modeling [MB95, McM97] and post rendering 3D warping [MMB97]. These methods both solve the acquisition problems and reduce the storage cost. They are, however, generally not fast enough for truly interactive purposes at the moment. Moreover, most of these methods exhibit severe artifacts such as visibility problems and gaps resulting from undersampling. These problems can only be avoided in very specific cases, but not for arbitrary camera movements.

With this paper we bridge the gap between high-performance algorithms based on dense sampling, and low-performance warping algorithms that only use a handful of images. Starting with a sparse Lumigraph, we increase the number of images contained in it by applying a warping algorithm. As we will show, the special geometry of a Lumigraph not only allows for a fast, efficient warping, but also reduces the number of warping artifacts. After enough warped images have been inserted, the new, higher resolution Lumigraph can be rendered using the fast quadri-linear interpolation algorithm. Even graphics hardware can be employed just as before.

Our warping method is in a sense similar to the work on view interpolation [CW93], except that we do not establish explicit correspondences between pixels in the respective source images. Moreover, we warp every pixel separately instead of using a quadtree-based approach which can warp whole blocks of approximately equal depth pixels. This gives the view morphing algorithm a performance advantage, but introduces warping artifacts, which, depending on the prescribed error tolerance, can vary between subpixel size and several pixels. For textured objects, these artifacts are quite visible. Moreover, the quadtree subdivision into blocks only works for scenes with highly coherent depth images.

The remainder of this paper is organized as follows. Section 2 briefly reviews the work on light fields and Lumigraphs. Section 3 discusses the warping algorithm we have developed for the refinement of Lumigraphs, before we present results in Section 4 and conclude in Section 5.

2 Light Fields and Lumigraphs

Our work builds strongly on the light field [LH96], and the Lumigraph [GGSC96]. A light field is a dense sampling of the 5-dimensional plenoptic function [AB91], which describes the radiance at every point in space in every direction. Since radiance does not change along a ray in empty space, the dimensionality can be reduced by one, if an appropriate parameterization is found, that reflects this property.

The so-called two-plane parameterization used by the light field representation fulfills this requirement. It represents a ray via its intersection points with two parallel planes. Since each of these points is characterized by two parameters in the plane, this results in a 4-dimensional function that is sampled through a regular grid on each plane (see Figure 1).

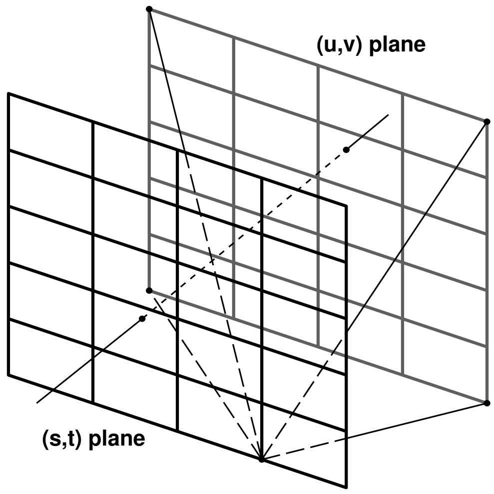

Figure 1: A light field is a 2-dimensional array of images taken from a regular grid of eye points on the (s,t)-plane through a window on the (u,v)-plane. The two planes are parallel, and the window is the same for all eye points.

One useful property of the two-plane parameterization is that all the rays passing through a single point on the (s,t)-plane form a perspective image of the scene, with the (s,t) point being the center of projection. Thus, a light field can be considered a 2-dimensional array of images with eye points regularly spaced on the (s,t)-plane.

Moreover, since we assume that the sampling is dense, the radiance along an arbitrary ray passing through the two planes can be interpolated from the known radiance values in nearby grid points. Each such ray passes through one of the grid cells on the (s,t)-plane and one on the (u,v)-plane. These are bounded by four grid points on the respective plane, and the radiance from any of the (u,v)-points to any of the (s,t)-points is stored in the data structure. This makes for a total of 16 radiance values, from which the radiance along the ray can be interpolated quadri-linearly. As shown in [GGSC96], this can also be done in hardware.

This method works well as long as the resolution of the light field is high. For low resolutions, the interpolation only yields a sharp image for objects in the (u,v)-plane. The further away points are from this plane, the more blurred they appear in the interpolated image.

The Lumigraph extends the concept of a light field by adding some geometric information which helps compensating for this problem. A coarse polygon mesh is stored together with the images. The mesh is used to first find the approximate depth along the ray to be reconstructed, and this depth is used to correct the weights for the interpolation.

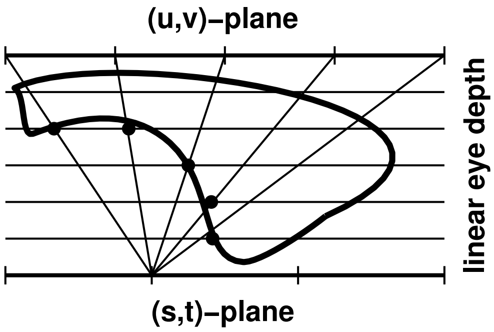

Figure 2: In this paper the geometry is stored as a z value for each sample in the Lumigraph. This is a linear function of the distance from the two planes.

In this paper we also use geometric information, but in a format that is different from the original Lumigraph work. Instead of storing a polygon mesh, we store the depth of each sample in the Lumigraph as the z coordinate in eye space, which is a linear function of the distance from the two light field planes (see Figure 2). Note that the two representations for geometry can easily be transformed into each other: ray-casting yields the depth values from the polygon mesh, and on the other hand, it is easy to generate an initial fine mesh from the depth values, and then apply one of the well-known mesh decimation algorithms (for example [Hop96]) to reduce the complexity.

3 Lumigraph Warping

In this section, we introduce a warping algorithm that is specifically tailored towards refining a Lumigraph by inserting images from new view points. Because this is a very specific geometric setting, the complexity of the warping equations is significantly reduced, and the warping is very fast.

As mentioned above, the Lumigraph consists of a number of images taken from a regular grid of eye points on one plane, the (s,t)-plane, through a rectangular region on a second, parallel plane, the (u,v)- or viewing-plane. This implies that the images are sheared perspective views of the scene, which share a common viewing direction perpendicular to the two planes.

In the following, let this viewing direction be the negative z-axis. Furthermore, let $\mathbf{0}$, the origin, be located on the viewing plane. This means that any point on the (u,v)-plane is given as $\mathbf{p} = \mathbf{0} + u \cdot \mathbf{x} + v \cdot \mathbf{y}$, and any point on the (s,t)-plane as $\mathbf{q} = \mathbf{0} + s \cdot \mathbf{x} + t \cdot \mathbf{y} + d \cdot \mathbf{z}$, where d is the distance of the two planes. This situation is depicted in Figure 3.

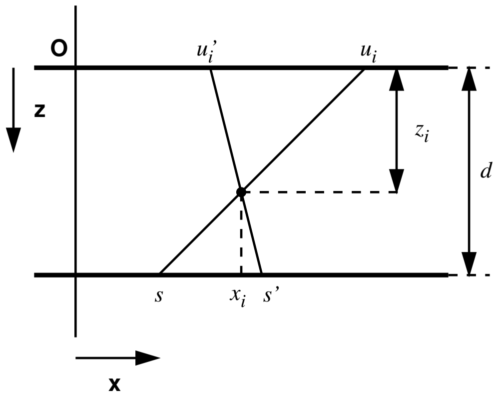

Figure 3: Geometry for warping a Lumigraph image located at (s,t) to a new view point (s',t').

Suppose a point at $(x_i, y_i, z_i)^T$ is to be warped from an image with a view point at $q := (s,t,d)^T$ to a new view point located at $q' := (s',t',d)^T$. The corresponding pixel positions u_i, v_i, u_i', and v_i' are related via the following two equations:

$$\frac{x_i - s}{d - z_i} = \frac{u_i - s}{d} \quad \text{and}$$
$$\frac{x_i - s'}{d - z_i} = \frac{u_i' - s'}{d}.$$

Solving these for the new pixel position u_i' yields

$$\begin{aligned} u_i' &= u_i + \frac{z_i}{d - z_i} s - \frac{z_i}{d - z_i} s' \\ &= u_i - \frac{z_i}{d - z_i} \Delta s, \end{aligned} \tag{1}$$

where $\Delta s := s' - s$ is the translation of the view point in s-direction. Similarly, we get $v_i' = v_i - z_i/(d - z_i) \cdot \Delta t$ with $\Delta t := t' - t$.

This means that for this specific setting the warping equation only depends on the pixel's distance from the (s, t)-plane and on the baseline between old and new camera position. This is much simpler than the warping equation for general camera transformations [MB95, McM97].

In particular, it is possible to precompute and store $z_i/(d - z_i)$ for each pixel. The warping of one pixel then only requires two multiplications and two sums (one each for u_i', and the same for v_i').

3.1 Eye Points at Infinity

One interesting variation of the basic two-plane parameterization are light fields and Lumigraphs where the eye points are located at infinity. This setting allows for a strict separation of spatial and directional resolution, which is not possible with the normal two-plane parameterization. In this case, the images are sheared orthographic views of the scene, and the eye points are given in homogeneous coordinates as $q = [s, t, 1, 0]^T$.

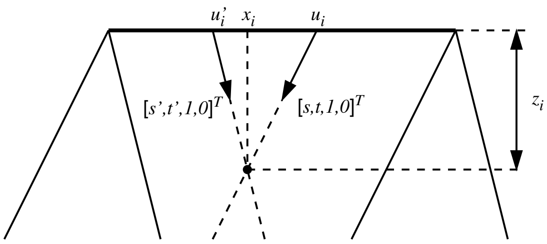

Figure 4: Geometry for warping orthographic images from Lumigraphs where the (s, t)-plane is at infinity.

The warping equation for this case is slightly different, but no more complicated. We get

$$\begin{aligned} x_i &= u_i + s \cdot z_i \quad \text{and} \\ x_i &= u_i' + s' \cdot z_i, \end{aligned}$$

which results in

$$\begin{aligned} u_i' &= u_i + z_i \cdot s - z_i \cdot s' \\ &= u_i - z_i \cdot \Delta s, \end{aligned} \tag{2}$$

3.2 Gap Filling and Warping from Multiple Images

When warping only a single original image to a new viewing position, the parallax will inevitably uncover some areas that have previously been occluded. These regions show up as visibility gaps in the destination image, since no pixels from the source are projected into these areas. These gaps can be seen as problems arising from the extrapolation of view points.

Luckily, the new eye points inserted into the Lumigraph are located within a rectangle of four original eye points. Therefore, the new image can be formed by interpolation similar to [CW93] if the information from all four source images is used. Furthermore, we can assume that the original Lumigraph is a fine enough sampling of the environment, so that each point visible from any position on the (s, t)-plane is also seen by at least one of the original eye points.

The other kind of holes that often occurs in warped images is due to the difference in solid angle under which a surface is seen. If the solid angle in the destination image is larger than in the source image, then the warped pixels will not cover the whole area in the destination image, which results in holes. These holes are caused by the undersampling of the object. The solid angle increases if either the new view point is closer to the object, or if it sees the surface under a more perpendicular angle than the old view point.

In the Lumigraph setting, in which the eye points move perpendicularly to the viewing direction, the change of distance to an object is relatively small, so that the number of resulting gaps should be minimal. The second cause, however, is possible, as depicted in Figure 5.

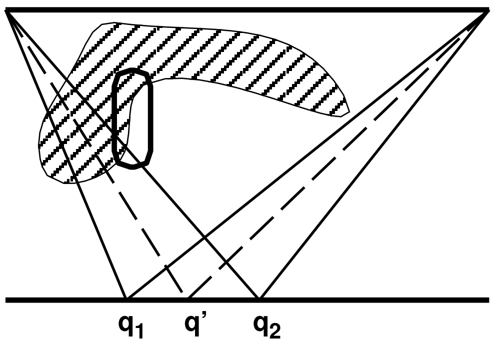

Figure 5: Gaps resulting from undersampling can be caused from the change of the angle under which a surface is seen (circled area). In the Lumigraph setting, however, it is likely that one of the original images sees the surface under a similar or even better angle.

If only view $\mathbf{q}_1$ was to be used as a source image for the new view point $\mathbf{q}'$, then many gaps would occur in the destination image. However, in the setting of a Lumigraph, one of the four neighboring original views, in this case $\mathbf{q}_2$, usually sees the surface under a similar if not better angle than the destination image. Thus, the number of gaps from this cause typically is also smaller than in a more general setting.

To fill any remaining gaps, we employ a hierarchical approach. Instead of generating only one image at the final resolution, we generate a whole pyramid of images, where each level has half the resolution of the previous one. This pyramid resembles a mipmap [Wil83] of the destination image. Warping to such a mipmap is inexpensive: let (u'_i, v'_i) be the warped pixel position according to Equation 1 or 2 for some point. Then the pixel positions for the lower resolution images are $(u'_i/2, v'_i/2)$, $(u'_i/4, v'_i/4)$, and so forth.

This way, each of the four source images can be warped into one such pyramid for the new view point. These four pyramids are then combined into one final image using the following algorithm:

```
mergeImages( src_img[4] )
{
  foreach pixel i
  {
    dest_z[i]= far plane distance
    layer= lowest resolution layer
    foreach image j
    {
      if( |dest_z[i]-src_z[j][i]| < ε )
      {
        dest_img[i]=
          blend( dest_img[i],
                 src_img[j][i] )
        layer= min(layer,layer[j][i])
      }
      else
        if( layer== layer[j][i] )
        {
          if( dest_z[i]< src_z[j][i] )
            dest_img[i]= src_img[j][i]
        }
        else
        {
          if( layer> layer[j][i] )
          {
            dest_img[i]= src_img[j][i]
            layer= layer[j][i]
          }
        }
    }
  }
}
```

This algorithm performs the following operations: if the z values of the pixels from multiple pyramids are almost identical, the color values are blended to accommodate for changes in surface color due to non-diffuse reflection characteristics. If the depth values differ, but the pixel information comes from the same layer in each of the pyramids, then the closest pixel is selected. This corresponds to a normal z-buffer algorithm. Finally, if the layers differ, then the pixel color is taken from the highest resolution layer available. The reason for this is that information from higher resolution layers is more reliable than information from lower resolution layers, which are only used to fill gaps.

Although a z-buffer is required for this combination step, the warping algorithm for generating the pyramids does not require a depth test if the pixels are warped in the depth preserving order presented in [McM97]. In this case, the warping only has to set the depth buffer for each pixel, but a depth test is not necessary.

4 Results

For the purpose of evaluating the quality and performance of our algorithm, we have created Lumigraphs from two synthetic scenes depicted in Figure 6. The first scene consists of a little airplane without textures. The second scene contains an elephant with a procedural wood shader applied. The visibility of both scenes is rather complex, and both are ray-traced RenderMan objects. The base resolution Lumigraph contains 4×4 images of size 256×256.

Figure 6: The two scenes used for comparison purposes. A simple airplane, and a textured elephant, both with a rather complex visibility.

The first criterion for the quality of our warping algorithm is the number of holes that need to be filled by the pyramid warping approach. This is best depicted by showing the depth image of a warped view where only the highest resolution layers have been used (Figure 7). The holes show up as small dots in the indicated regions. It can be seen that there are only very few holes, and since they only consist of isolated pixels, they are easily filled using the second level of the image pyramids. In the example in Figure 7, the number of pixels where this had to be done was seven for the airplane. Due to the increased geometric complexity, the number of holes for the elephant lies between five and twenty. Note that, although parts of the elephant's back are covered by its ears in two of the source images, the other two source images are able to fill in this missing information.

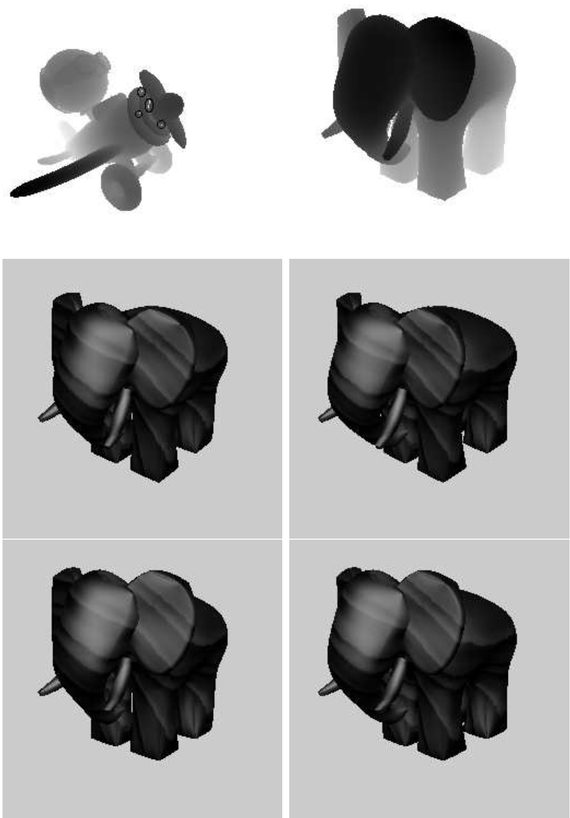

Figure 7: Top: combined depth image of a warped view where only the highest resolution layers of the pyramids have been used. Holes are marked with black circles. Center and bottom: source images used for warping the elephant.

In order to evaluate the quality of the warped images, we compare them to both ray-traced images and images generated using the quadri-linear interpolation (Figure 8). Due to the large parallax between the source images, the interpolated images exhibit strong blurring and ghosting effects, while the warped version is almost indistinguishable from the ray-traced image.

The warping of these 256×256 images proceeds at more than 8 images/sec. on an R10000 SGI O2, while the quadri-linear interpolation using

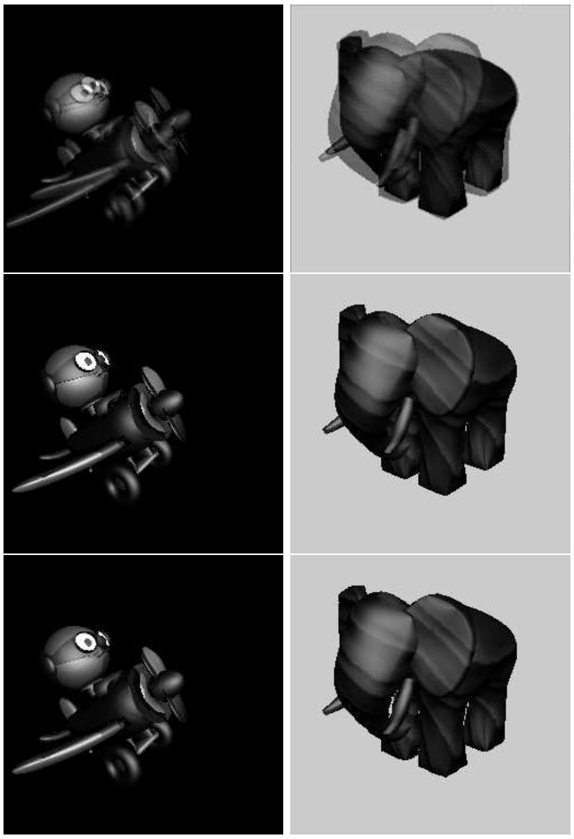

Figure 8: Top: images generated through quadri-linear interpolation from four source images. Center: ray-traced images. Bottom: images generated by our warping algorithm.

the graphics hardware on the same machine is more than 3 times as fast ($\approx$30 frames/sec.). Ray-tracing takes several minutes for each of the two scenes. These numbers show that our warping as a stand-alone method is almost fast enough for interactive viewing, with an image quality that is very much comparable to ray-tracing. On the other hand, the location of the view points is of course limited with our method.

The final criterion for the evaluation of our approach is the improvement of image quality when quadri-linear interpolation is applied to a refined Lumigraph. A comparison is depicted in Figure 9. It shows images reconstructed from the 4×4 Lumigraph of the elephant, and from a refined version with 13×13 images, as well as a sequence of resolutions for the airplane (2×2, 4×4, 7×7, and 13×13 images). As can be seen, the ghosting and blurring artifacts are dramatically reduced with increasing Lumigraph resolution. Independent of the Lumigraph size, these images can be rendered at full screen resolution at roughly 20 frames/second on an SGI O2.

5 Conclusions and Future Work

In this paper we have proposed a warping algorithm that is tailored towards refining the resolution of Lumigraphs. Due to the special geometric structure of this setting, this warper is able to generate new, high quality images at several frames per second. The refinement of the Lumigraph drastically improves the quality when using the efficient quadri-linear interpolation for generating images from arbitrary view points.

Using morphing to refine a Lumigraph has two important advantages over generating a higher resolution Lumigraph to begin with. First, the requirements for storage space and transmission bandwidths are minimized, and in many cases the loading of the data set proceeds faster (warping an image is often faster than loading an uncompressed image of the same resolution over Fast Ethernet). Secondly, the acquisition of lower resolution Lumigraphs is less painful and time consuming than the acquisition of higher resolution ones.

At the moment, the warping is only used as a preprocessing step to increase the resolution of a Lumigraph to a fixed size during the loading from disk. The new eye points are always placed on a uniform grid. In the future, we would like to adaptively refine the Lumigraph in those areas which are particularly important for the current user interaction. A cache of the warped images will make sure that locally even higher resolutions will be possible without requiring excessive amounts of memory. We also plan to investigate compression issues, especially the compression of the depth values.

REFERENCES

[AB91] E. H. Adelson and J. R. Bergen. *Computational Models of Visual Processing*, chapter 1 (The Plenoptic Function and the Elements of Early Vision). MIT Press, Cambridge, MA, 1991.

[CW93] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, pages 279–288, August 1993.

[GGSC96] Steven J. Gortler, Radek Grzeszczuk, Richard Szelinski, and Michael F. Cohen. The Lumigraph. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 43–54, August 1996.

[Hop96] Hugues Hoppe. Progressive meshes. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 99–108, August 1996.

[LH96] Marc Levoy and Pat Hanrahan. Light field rendering. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 31–42, August 1996.

[MB95] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 39–46, August 1995.

[McM97] Leonard McMillan. *An Image-Based Approach to Three-Dimensional Computer Graphics*. PhD thesis, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, 1997.

[MMB97] William R. Mark, Leonard McMillan, and Gary Bishop. Post-rendering 3D warping. In *Proceedings of 1997 Symposium on Interactive 3D Graphics*, pages 7–16, April 1997.

[Wil83] Lance Williams. Pyramidal parametrics. In *Computer Graphics (SIGGRAPH '83 Proceedings)*, pages 1–11, July 1983.

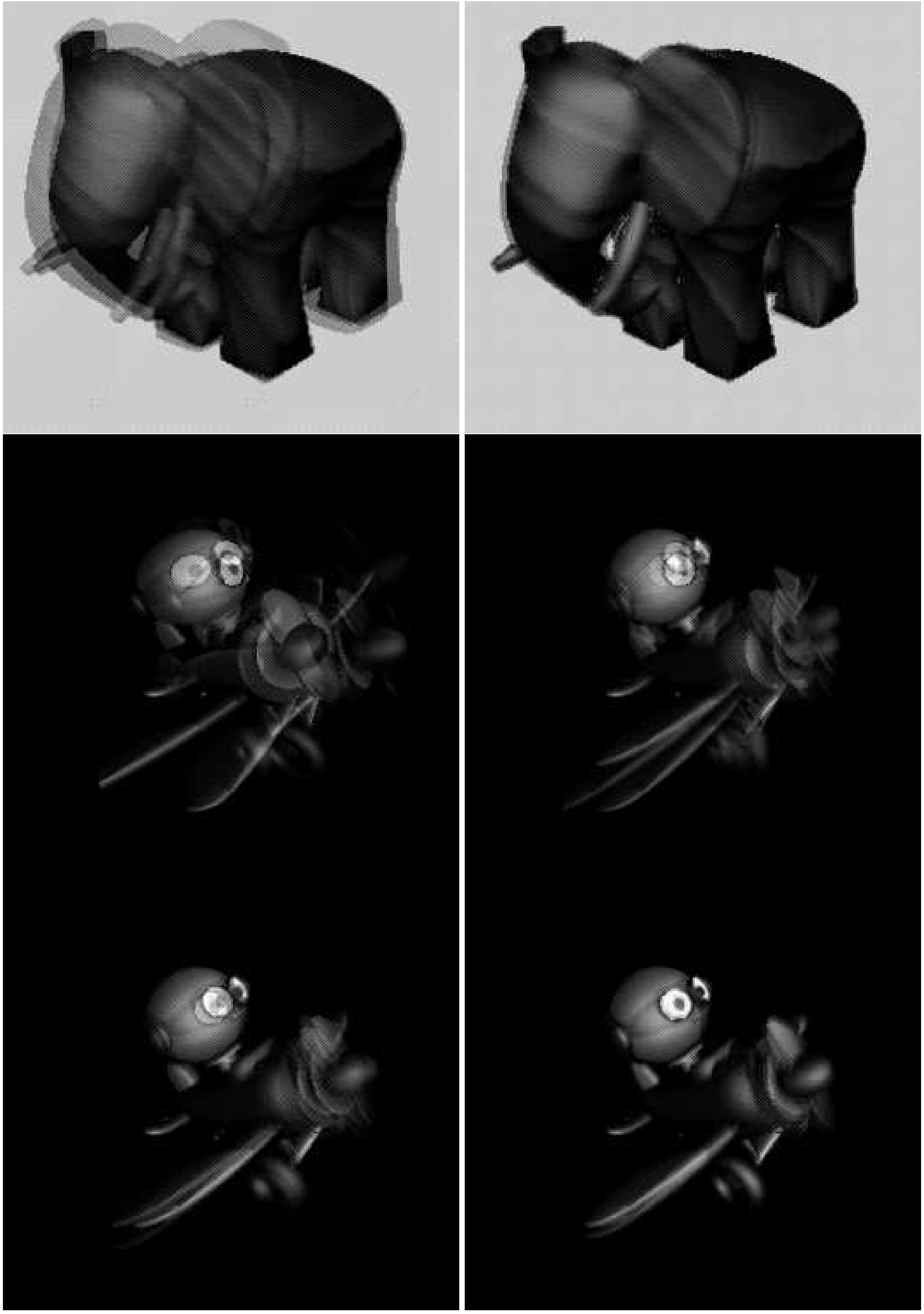

Figure 9: Top: quadri-linear reconstruction from a 4×4 and a 13×13 Lumigraph. Center and bottom: quadri-linear reconstruction from 2×2, 4×4, 7×7, and 13×13 Lumigraphs. The 7×7 and 13×13 Lumigraphs have been generated via our refinement algorithm from their respective 4×4 versions.