Point Light Field for Point Rendering Systems

Kartic Sankar Subr [*] M. Gopi [†] Renato Pajarola [‡] Miguel Sainz [§]

Computer Graphics Lab
Department of Computer Science, University of California Irvine
Technical Report UCI-ICS-03-28

Abstract

In this paper, we introduce the concept of point light fields for point based rendering systems, analogous to surface light fields for polygonal rendering systems. We evaluate two representations, namely, singular value decomposition and spherical harmonics for their representation accuracy, storage efficiency, and real-time reconstruction of point light fields. We have applied our algorithms to both real-world and synthetic point based models. We show the results of our algorithm using an advanced point based rendering system.

Keywords: Surface Light Fields, Point Based Rendering, Singular Value Decomposition, Spherical Harmonics

1 Introduction

The advent of image based rendering eliminated the need for modeling of geometry and physically correct light interaction with the object. The combination of geometry and image based rendering gave rise to surface light fields that represent the view-dependent visual characteristics of the model using images. However, surface light fields require polygonal representation of the object to capture, represent, compress, store, and render the model.

In a parallel line of research, points as rendering primitives have been proved to be a powerful alternative to polygons. Point based representation is typically used when the density of the point set is very high. The strengths of a point based rendering system include its ability to efficiently render highly noisy point data sets and point clouds with no connectivity information.

Sophisticated lighting effects of point samples from real world objects can be handled by point based rendering systems only if view-dependent visual characteristics of the object can be represented. The use of surface light fields for this purpose has many disadvantages. Since the surface light field requires surface representation, a surface reconstruction algorithm is required to convert the point data set to a polygonal model. While high density point data sets work best for high quality point based rendering, surface reconstruction algorithms work best with relatively sparse data points to avoid error prone geoemtric operations (like normal computation) involving very close points. While one of the strengths of the point based rendering algorithm is its ability to handle noisy point data sets, such data sets introduce serious robustness issues in surface reconstruction algorithms. Further, the need for a connected polygonal model undermines the strength of point based rendering systems that do not require connectivity information. Hence there is an imminent need for a representation and reconstruction method to handle view-dependent point colors, and that does not compromise on the strengths of the point rendering systems.

In this paper, we introduce point light fields for point based rendering systems, analogous to surface light fields for polygonal rendering systems. Point light fields also eliminates all the above mentioned drawbacks of surface light fields.

1.1 Main Contributions

The main contributions of this paper are:

- We introduce point light field to represent view-dependent visual characteristics of a point.
- We evaluate two lossy representation schemes, namely, singular value decomposition and spherical harmonics for representation accuracy, storage efficiency, and real-time reconstruction of point light fields.
- We show the results of our algorithm on various point based real-world and synthetic models.

We use a sophisticated point based rendering system [14] to show our results. However, our algorithm is not restricted to a single point rendering system and can be used by any such system.

In the next section, we discuss related work in point based rendering, and light field representations. In Section 3, we formalize our ideas on point light fields and develop algorithms for its representation and reconstruction. Section 4 presents our results and finally we conclude this paper in Section 5.

2 Previous Work

Most image-based rendering (IBR) approaches sample the plenoptic function [5] which is later used for rendering. The plenoptic function describes the distribution of light in 3D space of a (static) scene and can be represented by a 5-dimensional radiance function parameterized by location (three dimensions) and viewing angle (two dimensions). The most closely related IBR techniques to the work proposed in this paper are light-fields and point-based rendering methods.

The traditional light-field methods [10] and [8] index occluder-free light rays in 4D parameter space, while panoramic approaches [12, 6] sample all incoming radiance for a particular point in 3D space. The unstructured lumigraph [4] approach uses an approximate scene representation and a collection of unstructured images and rays to synthesize new views by view-dependent texturing and blending. Plenoptic stitching approaches [1, 2] capture omnidirectional images along paths and reparameterize the image data for efficient reconstruction of new viewpoints. On the other hand, surface light-fields [13, 19, 7] sample the outgoing radiance at points on the surface and thus require accurate knowledge of the object's surface.

[*] kartic@ics.uci.edu

[†] gopi@ics.uci.edu

[‡] pajarola@acm.org

[§] msainz@ics.uci.edu

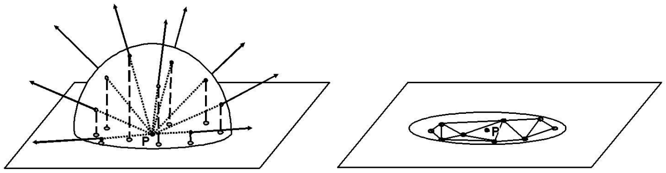


Figure 1: Sampling of point light fields: (a) The input viewing directions are projected onto to the tangent plane at point P. (b) These projected samples are triangulated using Delaunay triangulation.

Compared to the most related work on efficient surface light-field compression and rendering [13, 19, 7] we present novel point light-field representation and reconstruction alternatives based on spherical harmonics or singular value decomposition of either RGB or luminance/chrominance color models. Furthermore, in contrast to previous approaches our point light-field representation does not depend on any continuous parameterization or polygonalization of the object surface and can directly be employed in point based rendering systems.

Points as rendering primitives have been discussed in [11] and revived as a practical approach to render complex geometric objects in [9]. High-performance and space efficient multiresolution point rendering systems have been proposed in [17, 3]. High-quality point-blending and splatting systems have been proposed in [15, 20, 16] and [14]. For color texturing, the proposed point based rendering systems, in general, use a point-sampling of the diffuse surface color, and if supported, interactive illumination is achieved using point-normals and material reflection properties.

Our point light-field representation directly extends existing point based rendering systems such as [20, 16], [3] or [14] to incorporate view-dependent texturing. In particular, the developed techniques support real-time point light-field reconstruction and rendering.

3 Point Light Fields

Point light field $\mathcal{P}(\theta, \phi)$ is defined as the radiance function of a 3D point represented over a hemisphere around the point. This two-parameter function is a special case of surface light field which has two additional parameters to describe the surface. Representation of functions on a hemisphere is frequently seen in many areas in computer graphics. A few examples include BRDF functions for a given input direction, surface radiance functions, and radiosity computation over a hemisphere. Representation of these function values is traditionally done using various methods including closed form expressions, splines, spherical harmonics, principal component analysis, Zernike polynomials, and spherical wavelets.

In this paper, we analyze the use of two fundamentally different function representation techniques – singular value decomposition (SVD) and spherical harmonics (SH). The SVD decomposes the input function along the principal directions of the distribution of the functional values. The SH method expresses the input function in terms of pre-defined harmonic basis functions. We also try two different representations for color, namely, RGB and YIQ. We explore these techniques to represent and reconstruct point light fields, with three distinct objectives: representation accuracy, storage efficiency, and real-time reconstruction.

In this paper, we present our results on representation and rendering of point light fields on various real and synthetic models. The input to our algorithms are calibrated real or virtual camera images of a known geometry. We use the results of [18] to capture real world geometry and corresponding images. There are three steps in processing the captured point light field in order to achieve real-time high-quality rendering performance: resampling, representation, and reconstruction. In the following sections, we describe these three steps for SVD and SH representations of point light fields.

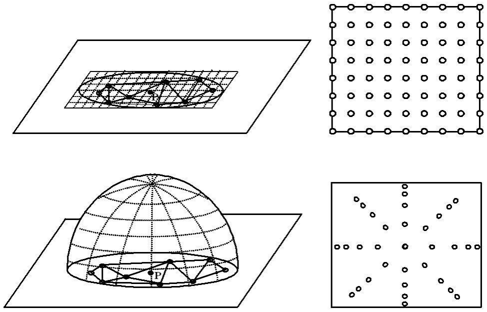


Figure 2: Resampling point light fields: Top row: (Left) Resampling for singular value decomposition involves generating a uniform grid on the tangent plane. (Right) Grid points for SVD. Bottom row: (Left) Resampling for spherical harmonics involves uniform sampling on the sphere. (This figure is just an abstraction and not upto scale.) (Right) Grid points for SH. The values at the grid points are obtained by interpolating the values at the projected input samples [7].

3.1 Singular Value Decomposition (SVD)

Singular value decomposition of an $m \times n$ matrix A yields, $A = U \Sigma V^T$, where U and V are orthogonal matrices and Σ is a diagonal matrix containing the eigen values in decreasing order. Progressive approximation of A can be achieved by choosing only the first few leading eigen values and the corresponding number of columns and rows of U and V^T respectively. Thus, $A \approx \Sigma_{j=1}^{k} U_{col:j} \sigma_j V^T_{row:j}$, where k is the approximation constant. When $k = n$, we get the original matrix A. In the representation of point light fields using SVD, the single parameter k determines the representation accuracy and storage efficiency.

Input Sampling: The given sample points of the hemispherical light field at a point P is orthographically projected onto to the tangent plane of P. Delaunay triangulation is used to triangulate these projected sample points on the plane. Figures 1 shows this process.

Resampling: As shown in Figure 2, the tangent plane is uniformly sampled in both X and Y directions of the local coordinate system into $m \times n$ samples. The color values at these samples are computed using barycentric interpolation of Delaunay triangulation of projected input samples. If the resampled point is outside the convex hull, then the color is a linear combination of the colors of the closest projected input sample points on the covex hull. We compute the mean of these resampled values, and subtract it from each resampled value. The resulting matrix of $m \times n$ values is stored as a matrix A along with the mean value. Our resampling method is adapted from [7].

Representation: We perform the SVD of the above matrix A. Given an approximation constant k, as part of lossy data representation, we store k columns of U and k rows of ΣV^T. Hence, both the representation accuracy and storage efficiency depend on k.

The compactness of representation by SVD does not only depend on k but also the size of the resampling matrix. The SVD represents the resampled function using dominant directions of the function values (eigen vectors). The dimensionality of the eigen vectors, and hence the size of representation, depends on the size of the matrix

A, and thus on the resampling. In case of representations like SH that use standard basis functions, the resampling size will affect only the accuracy of the representation and not its compactness.

Reconstruction: Given the required view point E, the unit vector from the point P to E is transformed to the local coordinate system of P, followed by an orthographic projection on to the tangent plane at P. The projected point is snapped to the closest resampled grid point. The X and Y coordinates of this grid point (after appropriate translation of the origin), would give the column number of ΣV^T and the row number of U respectively. These two k-vectors (k is the approximation constant) are retrieved from the representation data structure. The sum of the dot product of these two k-vectors and the mean, which we subtracted during resampling, would give the color of the point as seen from the given direction E. Unlike in surface light fields, we do not require special texture mapping hardware to reconstruct point light fields. The computations involved in reconstructing every point light field are a simple vector normalizing operation and a dot product of vectors. As shown in Section 4, we can do this operation for several thousand points per-second.

3.1.1 Representing RGB and YIQ

The above representation can be used independently over the three channels of the colors – R, G and B. Alternatively, we can convert RGB to YIQ representation and then represent using the same scheme for Y, I and Q independently. There is an advantage of using YIQ representation over RGB representation. The term Y represents the luminance and, I and Q represent the chrominance. Perceptual studies have shown that luminance is more important than chrominance for human perception. Hence, in our implementation, we represent Y with approximation constant $k > 0$ along with its mean, and I and Q with $k = 0$ in addition to the mean value of I and Q. Thus, the storage requirement for YIQ representation is nearly one-third of the requirement for RGB. But as shown in Section 4, the mean values of I and Q cannot handle the specular chrominance correctly, as the specular chrominance is determined by the chrominance of the light source, and not of the object. Hence, as shown in Figure 6, the specular chrominance appears as part of the diffuse chrominance. The red light that was incident on the blue bunny affects the color of the entire bunny, and makes it appear slightly purple. However, the specular highlights are accurately represented by Y.

3.2 Spherical Harmonics (SH)

Spherical harmonics $Y_l^m(\theta, \phi)$ are single valued, continuous, bounded, complex function on a sphere. They are equivalent to Fourier transforms over a sphere. The integer variable l is called the order, and m is an integer varying from $-l$ to $+l$.

Input Sampling: The given sample points of the hemispherical light field at a point P is orthographically projected on to the tangent plane of P and triangulated as in the case of SVD (Figure 1).

Resampling: Since the SH functions need highly uniform sampling on a sphere, we cannot use the uniform sampling on the plane, as we did in the case of SVD. We use a recursive subdivision of a dodecahedron to get uniform samples on a sphere.

Spherical interpolation of input sample points has to be done to determine the color values for resampled directions. But for the sake of simplicity, as shown in Figure 2, we project uniformly generated spherical samples onto the tangent plane of P, and use the same interpolation technique adopted in case of SVD.

Since the function has to be represented on a sphere to use SH and our point light field is on a hemisphere, we have to ”reflect” the values at the resampled points on the upper hemisphere about the tangent plane to get a smooth function over the entire sphere. This is automatically achieved as the projection of points of reflection about the tangent plane gets mapped on to the same point on the tangent plane and gets the same interpolated value. In this context, the choice of projection plane is very important. The upper hemisphere around the normal vector has “valid” (visible) colors for point light fields, and the lower hemisphere has “invalid” (occluded) colors. The projection plane should separate these “valid” and “invalid” directions. Any other projection plane other than the tangent plane would evaluate incorrect interpolated values for the resampled points.

Representation: Representation of smooth functions uniformly sampled over a sphere can be done by projecting the function values onto the SH basis functions. The coefficient C_l^m for a simple harmonic basis function Y_l^m is computed as $C_l^m = \Sigma_n W(\theta, \phi) f(\theta, \phi) Y_l^m(\theta, \phi)$, where n is the number of resampled points. The weight W is computed based on non-uniformity of sampling. If it is strictly a uniform sampling over a sphere, then $W = 4\pi/n$, a constant. The number of SH basis functions Y is $(l+1)^2$, where l is the order. Hence the representation consists of $(l+1)^2$ coefficient values (C_l^m), one corresponding to each SH basis function.

Reconstruction: Reconstruction of the point light field from SH function is simply the dot product of the coefficient vector and the basis function vector, evaluated for a specific direction (θ, ϕ). That is $\hat{f}(\theta, \phi) = C_l^m \cdot Y_l^m(\theta, \phi)$, where $\hat{f}$ is the approximation of the original input function f.

4 Implementation and Results

We present in this section, results from our study of the two lossy representation techniques, the SVD and the SH. We varied the resampling density and the extent of approximation, to analyze the reconstruction quality and size of the point light field representation. The datasets we used vary from 4000 to 14,000 data points, and the number of input viewing directions varied from 144 to 1,472 views. The point light field resampling, reconstruction, and rendering were performed on a 2.5GHz Pentium 4, with 1GB RAM and an ATI Radeon-9700 video card.

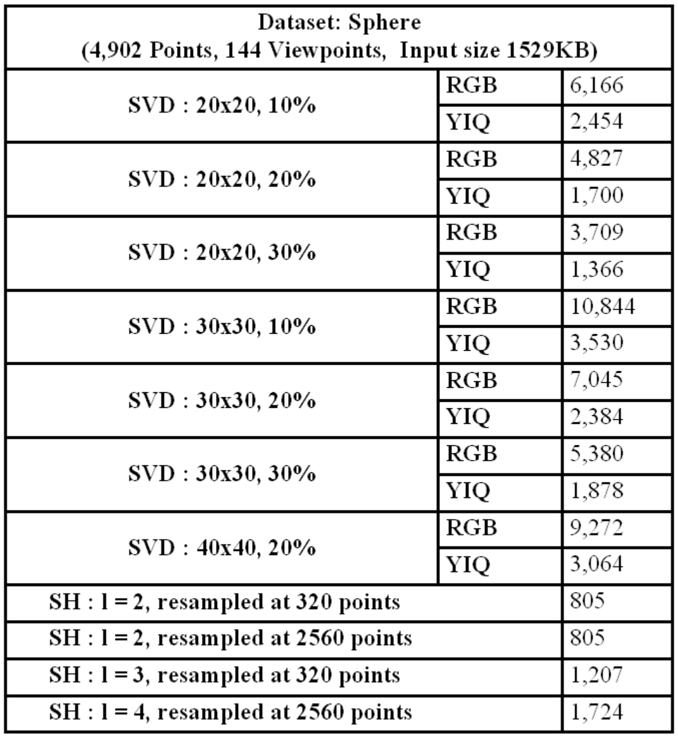
Datasets: The spheres shown in Figure 5 have three spotlights, Red, Green and Blue forming intersecting areas. The cup shown in Figure 5 was generated by visual hull and voxel carving techniques [18] using 14 camera positions.

4.1 Singular Value Decomposition (SVD)

Resampling: For SVD, the resampling density is expressed in terms of the grid size, and was varied from 20x20 to 60x60. As expected, the results show “smoother” images for higher sampling during interactive rendering. Further, as explained in Section 3.1, the resampling grid size also affects the storage requirements as seen from the Figure 3.

Representation: The approximation constant k (Section 3.1) was expressed and implemented in the form of a threshold percentage with respect to the largest eigen value. Thus with a 10% threshold, eigen values below 10% of the maximum eigen value, and the corresponding rows and columns of U and V^T were ignored. We used varying levels of threshold from 10% to 30%. As expected, the point light field representation was more compact for higher values of thresholds as more rows and columns were ignored. This result is shown in the Figure 3. We used the GNU Scientific Library (GSL 1.3) to perform the SVD.

Reconstruction: The reconstruction time when SVD was used was found to take, on an average, 6μ seconds per point. We do not use any functionality of graphics hardware to reconstruct the point light field. We were able to achieve extremely smooth interactive frame



Dataset: Sphere (4,902 Points, 144 Viewpoints, Input size 1529KB)		
SVD : 20x20, 10%	RGB	6,166
	YIQ	2,454
SVD : 20x20, 20%	RGB	4,827
	YIQ	1,700
SVD : 20x20, 30%	RGB	3,709
	YIQ	1,366
SVD : 30x30, 10%	RGB	10,844
	YIQ	3,530
SVD : 30x30, 20%	RGB	7,045
	YIQ	2,384
SVD : 30x30, 30%	RGB	5,380
	YIQ	1,878
SVD : 40x40, 20%	RGB	9,272
	YIQ	3,064
SH : l = 2, resampled at 320 points		805
SH : l = 2, resampled at 2560 points		805
SH : l = 3, resampled at 320 points		1,207
SH : l = 4, resampled at 2560 points		1,724

Figure 3: The table shows the file size of the point light fields representation expressed in KB. Note: no compression techniques were used. The resampling grid size and the threshold parameter used for SVD are given. The order of SH is shown as l.

rates with all our models even when used with sophisticated and compute intensive point based rendering system.
RGB and YIQ: We stored only the mean values of I and Q of a point light field with $k = 0$. As expected, this yielded a more compact point light field than RGB representation. Using YIQ representation, in the case of the Cup dataset(17,312KB) generated from 576 views, SVD for 1600 (40x40) resamped views and at a threshold of 20% resulted in a point light field representation of 13,535 KB. This is encouraging, especially since no special compression technique like vector quantization was employed.

4.2 Spherical Harmonics (SH)

Resampling: For SH, we express resampling in terms of number of samples generated on the sphere, and the the level of approximation, by the order of the spherical harmonic basis functions used. Unlike SVD, resampling size does not affect the compactness of representation of SH. This is also shown in Figure 3.
Representation: We achieved good approximation with low orders. With higher orders, we noticed ”ringing” effect in a few cases. The size of the point light field constructed using SH increases as the square of the order. Figure 7 compares the results of the cup data set for orders $l = 2$ and $l = 4$, rest of the parameters being the same.
Reconstruction: Although SH produced much smoother images and more compact point light fields specular highlights are less pronounced than with the SVD scheme as shown in Figures 5. The reconstruction time, per vertex was around four times slower than for SVD due to complex trignometric computations for the evaluation of SH basis functions during run-time. This can be avoided with clear quantization of sine and cosine values over the spherical angles and using a look-up table for the evaluation of Y_l^ms.

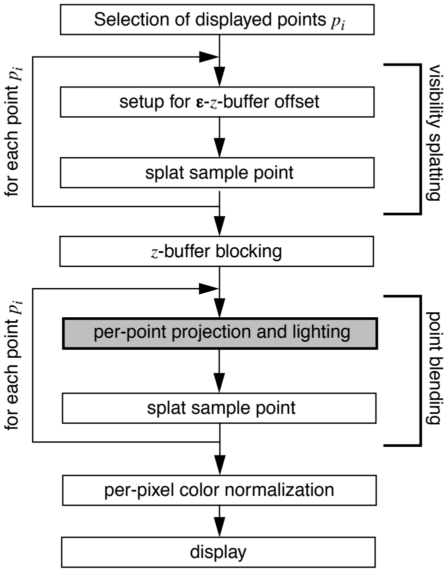


Figure 4: The rendering system: The point light field reconstruction substitutes the rendering system’s own illumination and color computation stage of the pipeline.

4.3 Rendering

To demonstrate our point light-field representation and conduct experiments, we used the point based rendering system introduced in [14]. While it supports level-of-detail (LOD) based rendering of large point sets, no LODs were used here.

The basic rendering process is illustrated in Figure 4 and includes the following steps: (1) selection of the rendered point splats, (2) visibility splatting of points to generate a *z*-buffer at an ε offset in the *z* direction, (3) blending of point splats overlapping within an ε distance range and (4) color normalization of the blended splats. For further details on this point rendering pipeline we refer the reader to [14].

In this or similar point rendering frameworks, the proposed point light field representation comes into play when the illumination is to be computed- highlighted box in Figure 4. At this point the color of a point, given the current viewpoint, is reconstructed from our point light field rather than performing interactive local illumination.

5 Summary

In this paper, we have introduced the concept of a point light field that can be used by point-based rendering systems to render view dependant visual characteristics of points. We have also analyzed two representation schemes for the constructed point light fields with respect to representation accuracy, storage efficiency and real-time reconstruction.

Using singular value decomposition (SVD) to represent the light field, preserved specularities better, while using spherical harmonics resulted in a more compact representation (in terms of storage required). Using SVD on the YIQ color model yielded a more compact point light fields than on RGB, but using RGB produces better results for colored specularity. It is important to note that we have not used any compression technique to compress our point light field data. Using methods like vector quantization, we can achieve

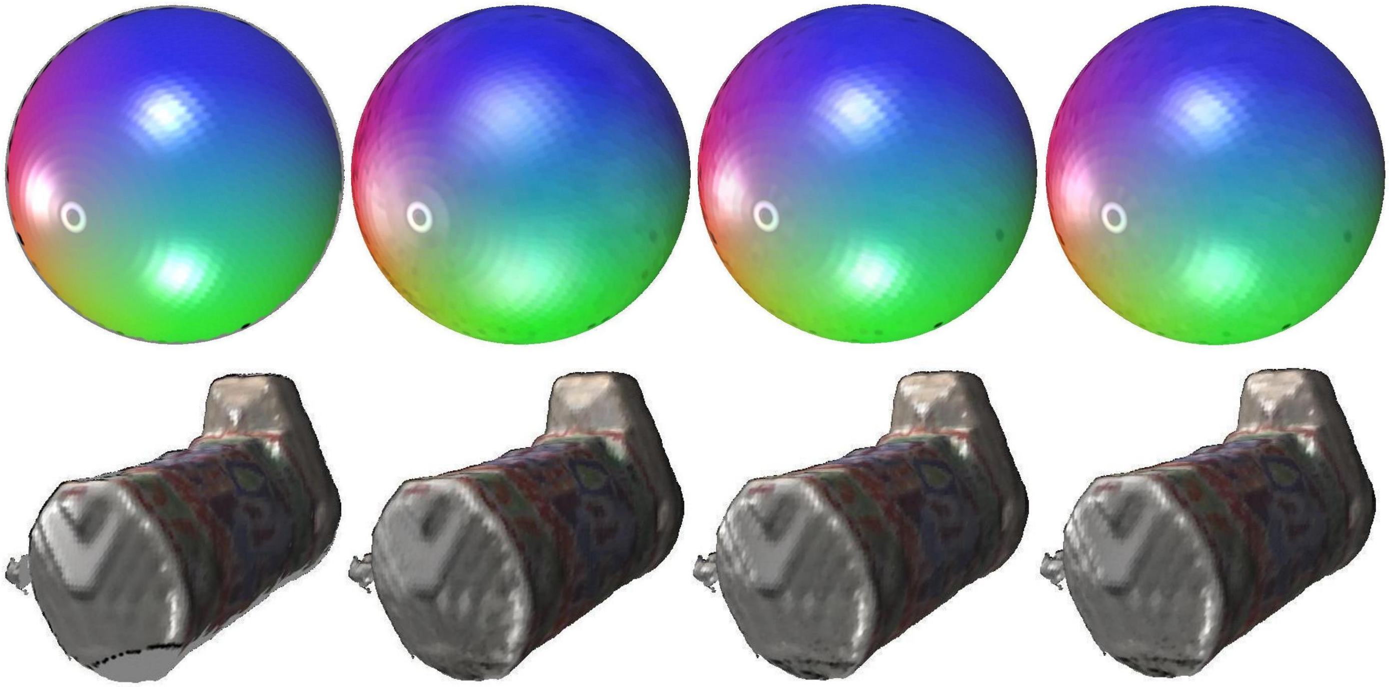


Figure 5: Top row: Sphere with 4902 points. Bottom row: Cup with 13202 points. From left to right: original model, SH with order $l = 4$, SVD on RGB values with 20% approximation tolerance with 40x40 grid, SVD on YIQ with 20% approximation tolerance and 40x40 grid. Notice that the specularity is less pronounced in case of SH.

dramatic compressions for both SVD and SH data sets proportionately.

The concept of point light field, its representation and reconstruction are general enough that it can be used with any point based rendering system. We are still working on run-time efficiency of our implementation to handle data sets in the order of millions of points.

Acknowledgements

We are deeply indebted to Jim Arvo for his enthusiastic willingness to share his spherical harmonics code and sample code to use it. We thank Peter Eisert for sharing the cup data set.

References

[1] Daniel Aliaga and Ingrid Carlbom. Plenoptic stitching: A scalable method for reconstructing 3D interactive walkthroughs. In *Proceedings SIGGRAPH 2001*, pages 443–450. ACM SIGGRAPH, 2001.

[2] Daniel Aliaga, Thomas Funkhouser, Dimah Yanovsky, and Ingrid Carlbom. Sea of images. In *Proceedings IEEE Visualization 2002*, pages 331–338. Computer Society Press, 2002.

[3] Mario Botsch, Andreas Wiratanaya, and Leif Kobbelt. Efficient high quality rendering of point sampled geometry. In *Proceedings Eurographics Workshop on Rendering*, pages –, 2002.

[4] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings SIGGRAPH 2001*, pages 425–432. ACM SIGGRAPH, 2001.

[5] Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. Plenoptic sampling. In *Proceedings SIGGRAPH 2000*, pages 307–318. ACM SIGGRAPH, 2000.

[6] Shenchang Eric Chen. QuickTime VR: An image-based approach to virtual environment navigation. In *Proceedings SIGGRAPH 95*, pages 29–38. ACM SIGGRAPH, 1995.

[7] Wei-Chao Chen, Jean-Yves Bouguet, Michael H. Chu, and Radek Grzeszczuk. Light field mapping: Efficient representation and hardware rendering of surface light fields. *ACM Transactions on Graphics*, 21(3):447–456, 2002.

[8] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings SIGGRAPH 96*, pages 43–54. ACM SIGGRAPH, 1996.

[9] J.P. Grossman and William J. Dally. Point sample rendering. In *Proceedings Eurographics Rendering Workshop 98*, pages 181–192. Eurographics, 1998.

[10] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings SIGGRAPH 96*, pages 31–42. ACM SIGGRAPH, 1996.

[11] Marc Levoy and Turner Whitted. The use of points as display primitives. Technical Report TR 85-022, Department of Computer Science, University of North Carolina at Chapel Hill, 1985.

[12] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Proceedings SIGGRAPH 95*, pages 39–46. ACM SIGGRAPH, 1995.

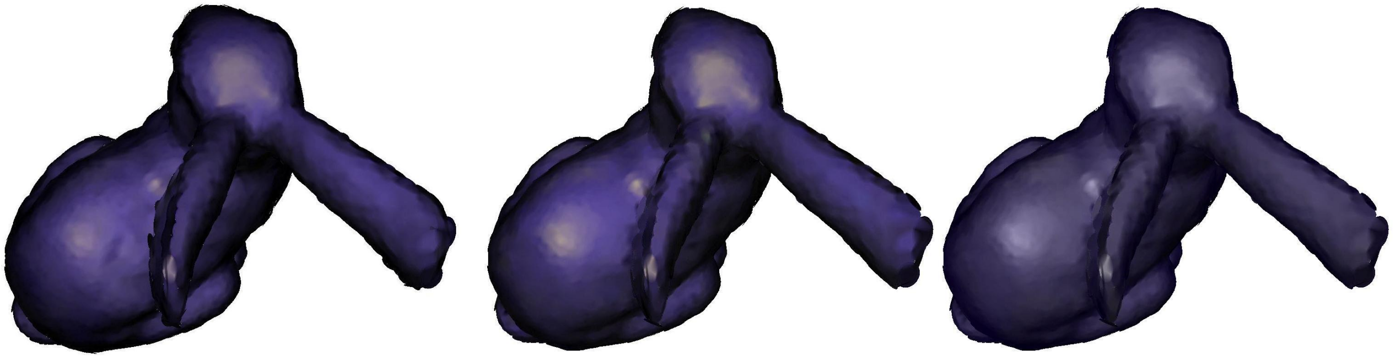


Figure 6: From left to right: SVD on RGB values with 60x60 grid and 10% approximation threshold, SH with order $l = 4$, and SVD on YIQ with 80x80 grid and 10% approximation threshold. This bunny was lit with a red light. Notice the red component spread throughout the model in the case of YIQ representation.

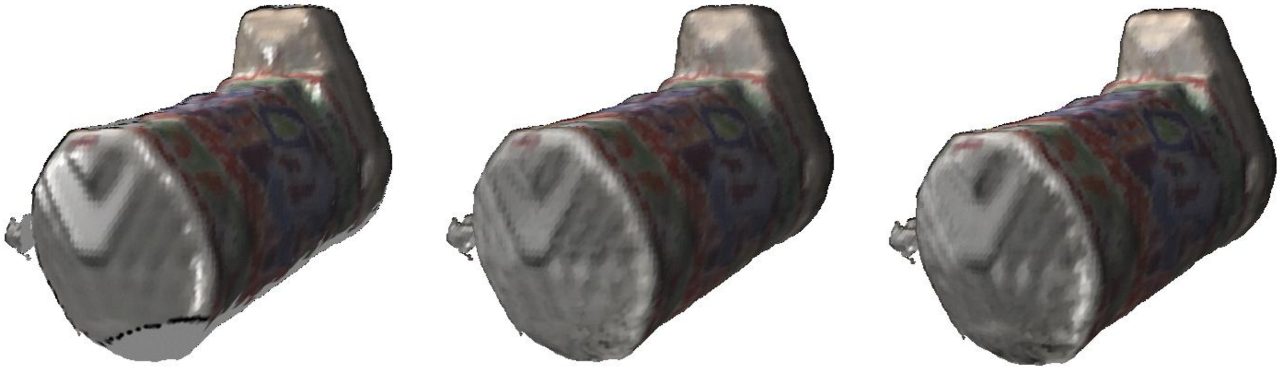


Figure 7: From left to right: Original cup, Reconstructed from SH with order $l = 2$, and order $l = 4$. Sampling of 2560 points in both cases.

[13] Gavin Miller, Steven Rubin, and Dulce Ponceleon. Lazy decompression of surface light fields for precomputed global illumination. In *Proceedings Eurographics Rendering Workshop 98*, pages 281–292. Eurographics, 1998.

[14] Renato Pajarola, Miguel Sainz, and Patrick Guidotti. Object-space blending and splatting of points. Technical Report UCI-ICS-03-01, The School of Information and Computer Science, University of California Irvine, 2003. submitted for publication.

[15] Hanspeter Pfister, Matthias Zwicker, Jeroen van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Proceedings SIGGRAPH 2000*, pages 335–342. ACM SIGGRAPH, 2000.

[16] Liu Ren, Hanspeter Pfister, and Matthias Zwicker. Object space EWA surface splatting: A hardware accelerated approach to high quality point rendering. In *Proceedings EUROGRAPHICS 2002*, pages –, 2002. also in Computer Graphics Forum 21(3).

[17] Szymon Rusinkiewicz and Marc Levoy. Qsplat: A multiresolution point rendering system for large meshes. In *Proceedings SIGGRAPH 2000*, pages 343–352. ACM SIGGRAPH, 2000.

[18] Miguel Sainz, Nader Bagherzadeh, and Antonio Susin. Hardware accelerated voxel carving. In *Proceedings 1st Ibero-American Symposium in Computer Graphics (SIACG 2002)*, pages 289–297, 2002.

[19] Daniel Wood, Daniel Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David Salesin, and Werner Stuetzle. Surface light fields for 3D photography. In *Proceedings SIGGRAPH 2000*, pages 287–296. ACM SIGGRAPH, 2000.

[20] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. Surface splatting. In *Proceedings SIGGRAPH 2001*, pages 371–378. ACM SIGGRAPH, 2001.