Incorporating Camera Models, Ocular Models, and Actual Patient Eye Data for Photo-Realistic and Vision-Realistic Rendering

Brian A. Barsky, Billy P. Chen, Alexander C. Berg,
Maxence Moutet, Daniel D. Garcia, Stanley A. Klein

Abstract. This paper introduces a new concept, that of vision realistic rendering. New rendering techniques are developed for the computer generation of synthetic images that incorporate accurate optics, especially for the human visual system, both as a model and from specific patient data. Two approaches are developed. In the first approach, we convolve an image with a point spread function, using high dynamic range source images to achieve a much more realistic estimate of glare and loss of contrast. The second approach traces rays from a retina or film plane through a system of lenses and into a three-dimensional scene. Both of these approaches can be used for camera models, eye models, and actual corneal topographic data. We are incorporating an accurate distribution of photoreceptors, with the density of sampling decreasing away from the foveal region of the retina. There are two distinct impacts of this research, one from the perspective of computer graphics and the other from the point of view from optometry and ophthalmology.

§1. Introduction

Context

The field of computer graphics is concerned with techniques for the generation of realistic synthetic images using computers. One of the primary goals has been photo-realistic rendering, that is, the computer creation of synthetic images that are visually indistinguishable from photographs of real scenes. This quest for visual realism in computer graphics has been remarkably successful as the field has developed and matured since the mid-1960's.

However, even the most compelling realistic synthetic images, generated by techniques such as ray tracing and radiosity, do not model the optics of the camera nor of the human visual system. The so-called "camera model" in computer graphics is in fact a misnomer, meaning little more than the

Mathematical Methods in CAGD: Oslo 2000

Tom Lyche and Larry L. Schumaker (eds.), pp. 1–10.

ISBN 0-8265-xxxx-x.

specification of the position and orientation of the perspective projection. The complexities of an individual's human visual system are not taken into account. In fact, almost all images in computer graphics are generated on the basis of the over-simplified pinhole camera model. For example, an effect such as the blur of the background of a scene is usually handled in an ad-hoc manner.

Two New Approaches to Rendering

We are investigating camera models and eye models as well as the use of corneal topographic data obtained from videokeratography. To this end, we are exploring two independent approaches. In the first approach, we convolve an image with a point spread function, using high dynamic range source images to achieve a much more realistic estimate of glare and loss of contrast. The second approach traces rays from a retina or film plane through a system of lenses and into a three-dimensional scene. Both these approaches can be used for camera models, eye models, and actual corneal topographic data. When using an eye model or actual corneal data in the second approach, we incorporate an accurate distribution of photoreceptors, with the density of sampling decreasing away from the foveal region of the retina.

§2. Approach I: Convolution of a 2D HDR Image with a Point Spread Function

Overview

Our overall goal is the accurate simulation of human vision. In this approach, we use the Point Spread Function (PSF), a technique used to measure the quality of optical systems. We use the PSF with our corneal models to derive a very good first approximation of what a particular person actually sees. This should be particularly effective in simulating visual phenomena such as nighttime haze and glare, a possible side effect of laser corneal refractive surgeries.

Background of the Technique

The PSF describes the response of an optical system to a single point of light; it is a two-dimensional retinal energy histogram. It provides a lot of information about predicted visual acuity, as it is akin to the impulse response of a circuit. Thus, to simulate the response of the optical system to any visual stimuli, we simply convolve the PSF with a digital image reproduction of that stimuli to get a very good simulation of what the patient actually sees [8, 9, 10].

Computational Method to Create the PSF from the Cornea

To compute the PSF, we shower the cornea with parallel incoming rays and record where they fall on the retina as a 2-D histogram. This retinal distribution was studied by Strehl [1] who developed an interesting metric, called the Strehl Ratio, to characterize the function. The Strehl Ratio describes the ratio of the height of the normalized PSF histogram compared to the ideal diffraction-limited height. We ignore diffraction effects, so we call this computation the Geometric Strehl ($Strehl_G$) ratio. The PSF itself is usually represented as a simple 2-D height-map or pseudo-color map of light ray density.

Note that most corneas have astigmatism which dominates the PSF response. Thus, we must add a correcting cylinder lens, just as patients who are astigmatic must wear correcting spectacles or contact lenses. We search through a three-dimensional space of correcting terms (overall resolving power, astigmatic correction angle and intensity) until we find the particular combination which achieves "best" acuity. We measure best acuity as the highest geometric Strehl ratio.

Once we have the PSF and a digital image of the visual stimuli, we then convolve the PSF with the stimuli to create a blurred image, very similar to what the patient would see. This convolution is very fast, as it involves performing a Fourier transform of both images, multiplying them, and taking the inverse transform. Both the transform of the stimuli and PSF can be pre-computed.

The input image can be anything: a photograph, a computer-generated 2-D image, or even a standard Snellen acuity eye chart. This last stimuli is very revealing, since it shows what the patient would see during an eye examination, and provides an accurate picture of his or her vision.

Simulating Accurate Haze and Glare Using High Dynamic Range Images

One of the shortcomings of using input images with Low Dynamic Range (LDR) (i.e., 8 bits per pixel with 256 levels of luminosity) is that they do not fully capture the nonlinearities of human vision. To properly model visual phenomena such as glare and haze requires High Dynamic Range (HDR) stimuli, which have at least tens of thousands of different levels of luminosity.

§3. Approach II: Optical Ray Tracing in 3D

Introduction

The second approach traces rays from a retina or film plane through a optical system of lenses (ocular or camera-based) and compute radiances of rays permeating through a three-dimensional scene. The illumination from the scene along a ray can be determined either from a synthetic computer-modeled environment or from data captured by a series of images taken in a real environment. In the synthetic case, the light can be computed using traditional ray tracing or an experimental virtual light field technique [21]. For a real scene, the flow of light is represented using a light field or Lumigraph [11, 18] representation of the plenoptic function. We provide a brief discussion on the applicability of light fields for completeness.

Camera Models

Realistic models of cameras are an important part of rendering photo-realistic images. However, with only a handful of exceptions, [3, 4, 7, 16, 17, 20], this has received almost no attention in the computer graphics literature. Combining light fields with realistic camera models provides an opportunity to create photo-realistic renderings of synthetic scenes captured in a light field.

Realistic models of cameras for rendering comprise a system of lenses focusing light onto a film plane. As with physical cameras, an aperture control is included as part of the system. By modeling lenses and apertures, the realistic camera model introduces depth of field blurring, and distortion effects such as coma and pincushion. These effects are not present in the simpler pinhole camera model used for most computer graphics rendering.

Rendering with a realistic camera model is accomplished by tracing rays from the film plane, through the system of lenses and apertures, out into a synthetic environment. For each point on the film plane, a number of rays need to be traced in slightly different directions to simulate the different paths that light can take to arrive at a point on the film plane. These multiple paths produce blur in images taken with a real camera. An (ideal) pinhole camera, where light can reach a point on the film plane in exactly only one direction, creates unrealistic images with no distortion or blur.

This camera model allows the lens system to be directed at a certain point in the scene and then focused at a given distance. For instance, the camera could be pointed at a virtual tree and then focused (by adjusting the distance between elements in the camera's lens system) to produce a crisp image of the tree, but a somewhat blurry image of mountains in the background. In addition the aperture could be adjusted to be very small, bringing distant features such as mountains into better focus, just as with a physical camera. Some of the subtle distortions and variations in illumination that physical cameras produce can be modeled. The typical pinhole model used in computer graphics does not provide this realism.

Tracing a few rays from each point on the film plane is sufficient for camera models with close to perfect lenses. As more imperfections or aberrations are introduced into the lens system, it is necessary to sample more rays per location on the film plane to account for a greater variation in the paths that light can traverse to reach each location.

Building the Camera Model

Camera lenses are clasically composed of spherical/planar glass "elements" as well as apertures, all centered on the optical axis [22]. We build the system of lenses [22] from the right to left, parameterizing each spherical lens by the following properties: 1) radius, 2) z position along the principal axis of the lens system 3) aperture radius and 4) Indices of refraction of adjacent surfaces.

One other important property of the lens system is the exit pupil. Similar to the pupil of the human eye, the exit pupil asserts a maximum range for rays emanating from the lens system. To compute the exit pupil, for each subset of lenses between the image plane and the aperture stop, we compute the thick lens approximation and project the aperture stop through the approximation [2]. The exit pupil is characterized by the minimum angle (with respect to the principal axis) over all subsets of lenses. We use the plane on which the exit pupil lies to compute sample rays from the image plane.

Sampling for the Camera Model

Our lens/cornea research uses sampling to obtain quanta from the continuous space of the "world" outside the camera/eye and to map these points to the image plane. We sample a finite number of rays defined by one endpoint at the point on the image plane and the other at a random point that is within a planar region specified to be some distance away from the image plane (the exit pupil plane). We perform this operation over all points on the image plane, effectively blurring areas of high frequency transitions on the projected image plane.

When sampling, we also need to account for the aliasing that occurs. [19] Our solution is to use a variation of unweighted area sampling (prefiltering). [6, 15] When we create the set of rays from any given point, we perturb the "base point" (starting point on the image plane) up to a specified random amount. This prefiltering effectively anti-aliases regions of high frequency, smoothing out the visually objectionable "jaggies".

Our sampling algorithm is as follows: For each discrete point on the image plane, compute a finite set of rays (which are described by their two endpoints) by randomly picking the second endpoints of each ray on the plane that describes the geometry of the projected exit pupil. If this random point lies within a specified area on the plane, then send this ray out into the ray tracer. Otherwise, pick another random point, create another ray and perform this test again until enough rays have been computed.

After sampling a set of rays for a given point, we combine the ray samples using a cos() function: the weight of any given color is a function of cos(d), where d is the angle between the normal vector of the image plane and the ray being cast. The intuition is that rays that are closer to the normal are the ones with the most influence on the color of that point on the image plane.

The Ocular Model

In the ocular model, we send rays from a small image plane through a corneal surface and compute radiances along refracted rays in the surrounding environment. The corneal surface is obtained from our spline surface algorithm for reconstruction from a videokeratographic reflection pattern using spline surface fitting of normals [12, 13, 14]. This data is represented as a Taylor series.

Sampling for Corneal Topographic Data

We send rays constructed from sampling points on the cornea surface and base points on the image plane. The sampled points can be arbitrarily dense and computing intersections between these rays and the cornea surface becomes trivial. In addition, the normals are also known at these sampled points, so computing the refracted ray by Snell's law is also simple. This method also allows for more control in terms of the sampling region on the surface of the cornea. For example, we may want to sample more densely towards the center of the cornea and less densely on the outer regions.

Image Based Rendering and Light Fields

Given our two refraction models (ocular and camera-based), we also investigate representations of scene geometry other than the "classical" ray tracing approach, which is limited to synthetic, computer-modeled geometry. It would be beneficial to trace rays from "real" geometry. Another drawback is that this approach generates a single image from a fixed vantage point. These shortcomings can be overcome by using a light field (or lumigraph) [11, 18].

Light fields can be generated from digitized images of real-world objects and environments, thereby providing a database of light rays from real objects. In recent years, there has been interest in the approach of image-based modeling and rendering. By using real photographs, the need for certain techniques to create realistic images is obviated. This light field approach employs a discrete sample of the distribution of radiance in a space constructed from a set of 2D images with known camera parameters. In addition, once the light field is generated, it can be quickly queried for the radiance of any given ray; this is advantageous compared to ray tracing which would require non-trivial time to compute this radiance.

A light field is ideally employed in situations where the images are captured by a digital camera from real scenes. We can refer to this as a captured light field (CLF) approach. A light field for virtual scenes can also be constructed [21]. In this case, an existing rendering system may be used to produce the required array of 2D images. In the case of a light field for virtual scenes, if the illumination is local, then there is clearly no problem to produce the required 2D images; for example, OpenGL can be used for this purpose. However, a new approach [21] is to directly produce light fields for globally illuminated virtual scenes without the intermediate need to render a large number of 2D images. This is called the virtual light field (VLF) approach.

Our proposed representation of a light field is similar to the 4D light fields used to represent a region of non-occluded space [18]. We parameterize lines by their intersections with two planes; the first plane is parameterized by (u, v) and the second by (s, t). We select n discrete points from the first plane and m discrete points from the second plane; then, for each point (x, y) on the uv-plane, we compute the radiance along the lines to each point on the st-plane. The resulting set of lines (and their radiances) represents the light field for that region. In other words, the light field is the set of radiances associated with the lines that are formed from the nm pairs of intersections between the first and second planes.

To produce images, a virtual "observer" can be placed anywhere in the scene at any viewing orientation. Consider, first, the simplest case of a single convex thin lens. The lens is represented as a rectangular planar polygon, and all rays through the lens can be trivially and rapidly generated. The intersection corresponding to any ray striking the lens can be looked up. If the intersection carries radiance in the appropriate direction (to inside the lens), then the lens formula may be used to compute the refracted ray, which can then be intersected with the image plane, and the radiance along the ray is accumulated on the image plane pixels. For a single convex lens, all rays in a parallel subfield are refracted through a single point on the focal plane. The image plane can be mapped to a two-dimensional window on a workstation display.

The lens properties (focal length, etc.) can be altered as desired between frames at no cost. The time for this operation of capturing an image onto the image plane through a lens is essentially constant for any eye position anywhere in the scene. It is equivalent to rasterization of one polygon (the lens) as many times as there are rotations of the parallel subfield plus the cost of the rotations.

This simple case of a single convex thin lens can be extended to any set of optical elements that can represent any combination of camera lenses, spectacles, contact lenses, and elements of the human eye.

§4. Example Images

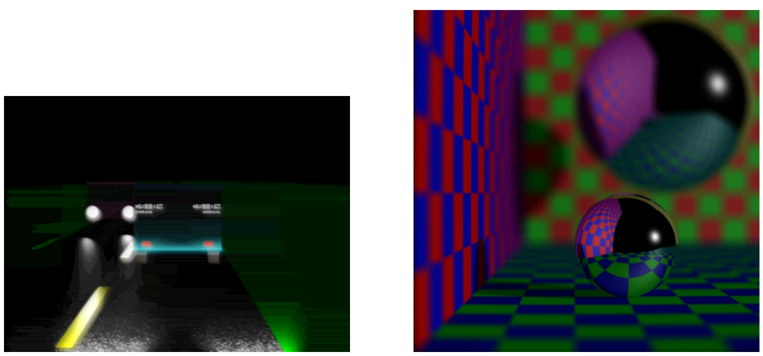

Fig. 1 & 2. A scene rendered using the PSF (left) and a scene with a camera/lens system (right).

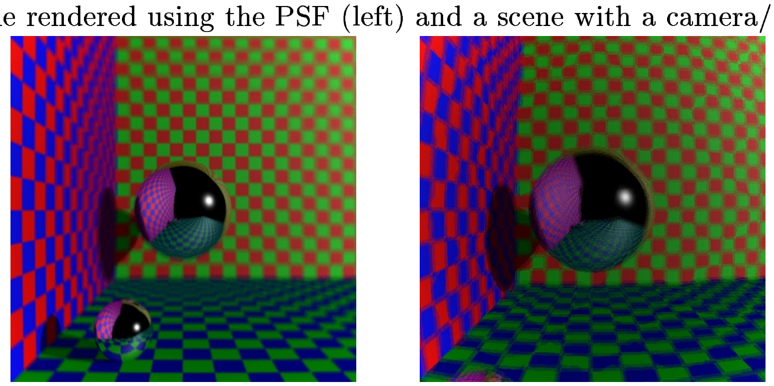

Fig. 3 & 4. Scene rendered through a normal cornea (right), and through cornea with keratoconus (left).

Figures 1-4 provide some images rendered using these techniques. Figure 1 shows the glare from the headlights of an approaching car rendered by convolving a synthetic image with a point spread function that was generated from actual patient corneal topographic data. Figure 2 illustrates rendering through a lens system of a camera, demonstrating blur as well as spherical aberration at the edges of the image. Figure 3 is an image traced through a normal cornea. Figure 4 is an image traced through the keratoconic cornea of the first author.

§5. Conclusion: The Next Step in Adding Rigor to Computer Graphics

Taking a broad view of the development of computer graphics, it can be seen that throughout its history, there has been a tradition of beginning with ad hoc, heuristic models, and then refining them by appealing to more rigorous science and mathematics. One example is in the early "cosine to a power" in the Phong specular reflection model which had no basis in physical reality and which was later improved by models based on physics such as the Cook-Torrance [5] model. Another example is in motion specification in animation which benefited from many physics-based models such as dynamics [23]. These examples are from the early 1980's, and as the computer graphics field has matured and has been so successful in improving the quality of computer-generated imagery, there have been diminishing opportunities for finding such gaps where appealing to more rigorous science and mathematics can have a significant impact. In this context, the new concept of vision realistic rendering incorporating involves the identification of just such a gap.

Acknowledgments. We would like to thank Timothy Tze-Him Lam, Celia Lin, Manda Sujitono, Koichi Tsunoda, Amirali Valiani, Matthew T. Webb, Sam Ka-Shing Wu, and Woojin M. Yu, students in the Computer Science Division at the University of California, Berkeley, for their contributions to the work described in this paper.

This work was supported in part by the National Science Foundation grant number ASC-9720252, "Visualization and Simulation in Scientific Computing for the Cornea", and grant number CDA-9726362, "Virtual Environments for Telesurgery and Surgical, Training: Efficient Computation, Visualization, and Interaction".

References

1. Ajoy K. Ghatak and K. Thyagarajan, *Contemporary Optics*, Plenum, New York, 1978).
2. Born, Max and Emil Wolf, *Principles of Optics*, 7th Edition, Cambridge University Press, Cambridge, UK, 1980.
3. Cook, Robert L., Stochastic Sampling in Computer Graphics, *ACM Transactions on Graphics*, **5** (1), January 1986, 51–72.

4. Cook, Robert L., Thomas Porter, and Loren Carpenter, Distributed Ray Tracing, *Computer Graphics (SIGGRAPH '84)*, **18** (3), 23-27 July 1984, 137–145.

5. Cook, Robert L., and Kenneth Torrance, A Reflectance Model for Computer Graphics, *ACM Transactions on Graphics*, **1** (1), Jan. 1982, 7–24. (Original version in *ACM SIGGRAPH '81 Conference Proceedings*, Aug. 1981, 307–316.)

6. Crow, Franklin C., A Comparison of Antialiasing Techniques, *IEEE Computer Graphics and Applications*, **1** (1), 1981, 40–49.

7. Dippe, Mark A. Z., and Erling H. Wold, Antialiasing Through Stochastic Sampling, *Computer Graphics (SIGGRAPH '85)*, **19** (3), 22–26 July 1985, 69–78.

8. Garcia, Daniel D., CWhatUC : Software Tools for Predicting, Visualizing and Simulating Corneal Visual Acuity, Ph.D. Thesis, May 2000.

9. Garcia, Daniel D., Brian A. Barsky, and Stanley A. Klein, The OPTICAL Project at UC Berkeley: Simulating Visual Acuity, *Medicine Meets Virtual Reality: 6 (Art, Science, Technology: Healthcare (r)Evolution)*, San Diego, January 28-31, 1998.

10. Garcia, Daniel D., Brian A. Barsky, and Stanley A. Klein, CWhatUC : A Visual Acuity Simulator, *Conference on Human Vision and Electronic Imaging, SPIE/IS&T Symposium on Electronic Imaging: Science and Technology*, San Jose, California, January 24-30, 1998.

11. Gortler, Steven J., Radek Grzeszczuk, Richard Szeliski, and Michael Cohen, The Lumigraph, *Computer Graphics (SIGGRAPH '96)*, Annual Conference Series, 4–9 August 1996, 43–52.

12. Halstead, Mark A., Brian A. Barsky, Stanley A. Klein and Robert B. Mandell, Geometric Modeling of the Cornea Using Videokeratography, in: *Daelhen, Morton; Lyche, Tom; and Schumaker*, Larry L., editors, *Mathematical Methods for Curves and Surfaces*, Nashville: Vanderbilt University Press, 1995, 213–223.

13. Halstead, Mark A., Brian A. Barsky, Stanley A. Klein, and Robert B. Mandell, A Spline Surface Algorithm for Reconstruction of Corneal Topography from a Videokeratographic Reflection Pattern, *Optometry and Vision Science*, **72** (11), November 1995, 821–827.

14. Halstead, Mark A., Brian A. Barsky, Stanley A. Klein, and Robert B. Mandell, Reconstructing Curved Surfaces From Specular Reflection Patterns Using Spline Surface Fitting of Normals, *ACM/SIGGRAPH '96*, New Orleans, 4-9 August 1996, 335–342.

15. Kirk, David and James Arvo, Unbiased Sampling Techniques for Image Synthesis, in proceedings of SIGGRAPH '91, *Computer Graphics*, **25** (4), 1991, 153–156.

16. Kolb, Craig, Don Mitchell and Pat Hanrahan, A Realistic Camera Model for Computer Graphics, *Computer Graphics (SIGGRAPH '95)*, Annual Conference Series, 6–11 August 1995, 317–324.

17. Lee, Mark E., Richard A. Redner, and Samuel P. Uselton, Statistically Optimized Sampling for Distributed Ray Tracing, *Computer Graphics (SIGGRAPH '85)*, **19** (3), 22–26 July 1985, 61–67.

18. Levoy, Marc and Pat Hanrahan, Light Field Rendering, *Computer Graphics (SIGGRAPH '96)*, Annual Conference Series, 4–9 August 1996, 31–42.

19. Oppenheim, Alan V. and Ronald W. Schafer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ 1975.

20. Potmesil, Michael and Indranil Chakravarty, Synthetic Image Generation with a Lens and Camera Model for *Computer Graphics, ACM Transactions on Graphics*, **1** (2), April 1982, 85–108.

21. Slater, Mel, A Note on Virtual Light Fields, Department of Computer Science Research Note RN/00/26/, University College London, April 5th 2000.

22. Smith, Warren J.,*Modern Lens Design*, McGraw Hill, New York, 1992.

23. Wilhelms, Jane and Brian A. Barsky, Using Dynamic Analysis for the Animation of Articulated Bodies such as Humans and Robots, 97–104 in *Proceedings of Graphics Interface '85*, Montreal, 27–31 May 1985. Revised version published in *Computer-Generated Images - The State of the Art*, edited by Nadia Magnenat-Thalmann and Daniel Thalmann, Springer-Verlag, 1985, 209–229.

Brian A. Barsky
University of California, Berkeley
EECS Computer Science Division
387 Soda Hall # 1776
Berkeley CA 94720-1776
U.S.A.
barsky@cs.berkeley.edu