An Efficient Error-Bounded General Camera Model

Voicu Popescu, Jordan Dauble, Chunhui Mei, Elisha Sacks
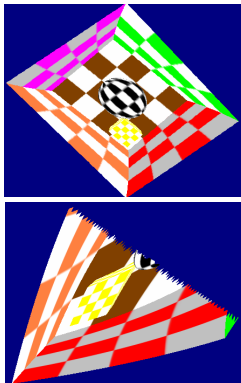Purdue University

Figure 2. Images rendered with bilinear 4-ray (*top*) and 6-ray (*bottom*) cameras.
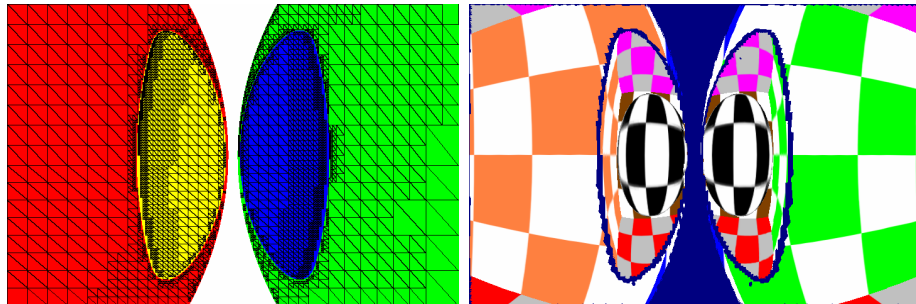

Figure 1. (*Left*) Approximation of catadioptric system with two partial-sphere mirrors and a perspective camera, creating first (red and green) and second (yellow and blue) order reflected rays. The image is triangulated into simple cameras ($\varepsilon = 1\text{pix}$). The right image was rendered with the resulting compound camera model.

Abstract

Camera models are essential infrastructure in computer vision, computer graphics, and visualization. The most frequently used camera models are based on the single-viewpoint constraint. Removing this constraint brings the advantage of improved flexibility in camera design. However, prior camera models that eliminate the single-viewpoint constraint are inefficient.

We describe an approximate model for coherent general cameras, which projects efficiently with user chosen accuracy. The rays of the general camera are partitioned into simple cameras that approximate the camera locally. The simple cameras are modeled with k-ray cameras, a novel class of non-pinhole cameras. The rays of a k-ray camera interpolate between k construction rays. We analyze several variants of k-ray cameras. The resulting compound camera model is efficient because the number of simple cameras is orders of magnitude lower than the number of rays in the original general camera, and because each simple camera provides closed-form projection.

1. Introduction

The invention of synthetic imaging systems and of graphics computers has created two research fields for which camera models are essential infrastructure. By far the most popular camera model in computer vision and computer graphics is the planar pinhole camera. One reason for this is that manufacturing imaging systems is a challenging endeavor and researchers more often than not have to rely on cameras developed for applications where human-vision-like images are needed. Another reason is that many computer graphics applications require a high degree of realism, such that the output image is a believable replica of what the user would see in a similar real world scenario. Finally, the planar pinhole camera model is simple, it is well understood, and it can be used to efficiently analyze or create images.

The simplicity of the planar pinhole camera model also implies important limitations: the scene is sampled from a single viewpoint, the field of view is relatively small, and building a physical camera that precisely respects the theoretical camera model is difficult. Camera models have been developed to remove the field of view limitation by sampling the scene from all directions, and to account for departures from ideal camera models.

However, most camera models maintain the constraint

that all captured rays have to pass through a common point, because such a single viewpoint image can be trivially re-sampled into a familiar planar pinhole camera image. Although this advantage is certainly important, removing the single viewpoint constraint would greatly increase flexibility in the design of imaging systems, and would allow optimizing exclusively for the requirements of the application. One example is designing camera models that improve scene coverage in the presence of occlusions. Another example is modeling the deviation from the single viewpoint theoretical model in the case of an actual catadioptric omnidirectional camera.

Such general camera models also have applicability in computer graphics. Rendering accurate reflections on curved surfaces presently requires sacrificing interactivity. Camera models that capture the rays reflected off the specular surfaces in the scene would enable rendering reflections accurately with the efficient feed-forward graphics pipeline. In image-based rendering the scene is rendered from pre-acquired or pre-rendered reference images. Using single viewpoint reference images produces disocclusion errors, artifacts that occur when a surface is visible in the output image but is not sampled in the reference images. Reference images that sample the scene from more than one viewpoint have the potential to alleviate this problem.

A third domain where general camera models are useful is visualization. Artists have known for a long time that deviating from the precise rules of perspective viewing can enhance the eloquence of an image. This idea is exploited in information visualization to devise rules for producing visualizations that deemphasize realism and focus on maximizing the effectiveness of the information transfer. Such visualizations could be produced automatically by devising appropriate camera models.

We are not the first to point out the usefulness of general cameras, nor are we the first to propose a general camera model. The contribution of this paper is to introduce an *efficient* general camera model, which has the potential to make general cameras practical for many applications. We describe an algorithm that accepts as input a camera given by its rays and an error threshold ε measured in pixels. The algorithm produces an approximate model of the input camera, which projects efficiently with an error bounded by ε.

We assume that the input camera projects a 3D point to a finite number of image locations—cameras with non-zero aperture can be modeled as a collection of such cameras. Another precondition on the input camera is coherence. We informally define a coherent camera as a camera for which the number of pixel discontinuities is $O(\sqrt{N})$, where N is the number of rays. A camera is discontinuous at a pixel if the origin or direction of the ray is substantially different from that of the rays at neighboring pixels.

Coherence is a condition readily met by most physical cameras. We would like to point out however that looking beyond the technological challenges of building a non-coherent camera, such a camera could prove to be useful. For example, a camera that samples randomly and dynamically the five dimensional space of rays in a scene with complex occlusions could have attractive statistical scene-coverage properties.

The algorithm exploits the coherence of the input camera to partition its rays into groups, each handled by a simple camera. A group is sufficiently small such that replacing it with the simple camera respects the projection error threshold ε. An obvious choice for the model of the simple cameras is the pinhole camera model. However, pinholes poorly approximate the non-concurrent input rays, which results in a large number of simple cameras and an increased complexity of the overall compound camera. Instead, we model simple cameras as *k-ray cameras*, a novel class of non-pinhole cameras that interpolate between k given rays. We analyze several instances of k-ray cameras and show that up to $k = 6$, the k-ray cameras have closed-form projection, which makes them well suited for use as simple cameras. k-ray cameras are a generalization of general linear cameras (GLCs) [Yu 2004]; a GLC is equivalent to the 3-ray camera, the first member of the k-ray class.

A 3D point is projected with the simple cameras that contain it. Efficient projection with user controllable accuracy is useful for many basic operations such as projecting a 3D scene marker during calibration or tracking, finding the locus of possible correspondences of an image point in a second image, or finding the output image projection of a triangle vertex that is first reflected in a curved surface.

The paper is organized as follows. The next section reviews prior work. Section 3 gives an overview of the algorithm. Section 4 introduces the k-ray camera class and analyzes several types of such cameras. Section 5 describes an extension to ensure C^0 projection continuity between two neighboring k-ray cameras. The last section discusses our results.

2. Prior work

The first limitation of the planar pinhole camera model addressed was the limited field of view. One approach is the use of fisheye lenses [Wood 1906, Miyamoto 1964]. Another approach is to rotate a planar pinhole camera about its center of projection [McMillan 1995, Chen 1995] and to stitch a wide field of view image from the individually captured frames.

The catadioptric approach employs mirrors to extend the field of view of a regular camera. Conical [Yagi 1990], spherical [Hong 1991], hyperboloidal [Yamazawa

1995], faceted [Nalwa 1996], and paraboloidal [Nayar] mirrors are used in conjunction with perspective or orthographic cameras to capture coherent 2D arrays of pixels that sample the scene in all directions. The early conical and spherical variants do not respect the single viewpoint constraint, whereas the improved hyperboloidal, faceted and paraboloidal mirror systems generate concurrent reflected rays.

Although the lack of a single effective viewpoint was originally viewed as a disadvantage, researchers have begun investigating the geometry and useful properties of camera models that have a continuum of view points. We generically call such cameras non-pinholes. Examples of non-pinhole cameras are the pushbroom [Gupta 1997], which collects rays in parallel planes by sweeping a line, and the two-slit camera [Pajda 2002], which captures all rays passing through two non-coplanar lines. The pushbroom, two-slit, planar pinhole, and orthographic cameras, are subclasses of the general linear camera (GLC) that collect linear combinations of three rays [Yu 2004].

The GLC has proven to be a powerful tool for locally characterizing reflections on arbitrary surfaces [Yu 2005]. A GLC centered at a reflected ray is used to probe the reflective properties of the surface at the point where that ray originates. The focus of the framework developed by Yu and McMillan is analysis of the complex distortions caused by arbitrary mirror surfaces, whereas our goal is replacing the rays of a complex imaging system with a small set of powerful but efficient simple cameras.

In computer graphics camera model extensions have been called upon to address challenging problems. One such problem is the problem of disocclusion errors. The layered depth image [Shade 1998], the multiple-center-of-projection camera [Rademacher 1998], and the occlusion camera [Mei 2005] are examples of camera models which produce reference images less prone to disocclusion errors. None of these camera models are sufficiently flexible to approximate general cameras.

Another problem that would benefit from an efficient general camera model is that of rendering reflections on arbitrary surfaces. The reflectors in a scene and the desired view planar pinhole camera can be seen as a dynamic large-scale catadioptric imaging system, used by the reflection rendering algorithm to synthesize images. The classical approach to high-quality reflections is ray tracing [Glassner 1980], which evaluates each ray of the catadioptric system individually.

The approach is inefficient since only a few of the ray/geometric-primitive pairs yield a valid intersection. A variety of acceleration schemes have been developed to avoid considering invalid ray-primitive pairs, but efficiency is hindered by the high complexity of the rays and the scene. When the reflector surface is smooth, the reflected rays are coherent and our approach can be used to replace the set of reflected rays with a considerably smaller set of simple cameras. The simple cameras can be readily integrated in most ray tracing acceleration schemes. Consequently, the efficient general camera model we describe has the potential to accelerate accurate reflection rendering.

Heidrich et al. propose an image-based lens model [Heidrich 1997] for adding depth-of-field effects to computer rendered imagery. The non-zero aperture is sampled from several points. For each point, the rays captured by the lens system are grouped in simple cameras. The approach is similar to ours, the difference is that the simple cameras are modeled as pinhole cameras, resulting in a much coarser approximation of the imaging system. The higher approximation error is acceptable in their case since the contribution of all the aperture sample points is averaged, but would produce unacceptable results in the case of a sharply focused catadioptric system.

Several general camera models have been proposed. The light field [Gortler 96, Levoy 96] samples a 4D subset of the plenoptic function [Adelson 1991]. Although most imaging systems could be modeled with light fields, for many of them, the uniform sampling of the 4D ray space is unnecessary. Grossberg and Nayar describe a general camera model that stores only the actual rays of the imaging system [Grossberg 2001]. They present a calibration procedure that takes an unknown imaging system and produces a camera model expressed as a collection of N raxels, where N is the number of pixels in the image. A raxel is a 7-dimensional vector that associates a ray $(o_x, o_y, o_z, \theta, \varphi)$ to a pixel (u, v).

3. Algorithm overview

Our algorithm takes as input a general camera and a desired projection accuracy ε. The camera can be given by an analytical

$$M(u, v) = (o_x(u, v), o_y(u, v), o_z(u, v), \theta(u, v), \varphi(u, v)) \quad (1)$$
$$0 < u < w, \ 0 < v < h$$

or discrete, raxel like,

$$M(ui, vi) = (o_{xi}, o_{yi}, o_{zi}, \theta_i, \varphi_i), i = 1 \ to \ N, \quad (2)$$

pixel to ray mapping. The image resolution is w x $h = N$. Not all pixels need to be assigned a ray, which allows modeling, for example, catadioptric systems where the mirrors do not cover the entire image. The output of the algorithm is a set of n simple cameras

$$M^* = \{SC_1, SC_2, ..., SC_N\} \quad (3)$$

with the property that M and $M*$ produce the same number of projections for any 3D point P, and for any projection p produced by M there is a projection p^* produced by M^*, such that $|p\text{-}p^*| < \varepsilon$.

The algorithm subdivides the image in quad tree fashion. The subdivision halts if the current image tile is smaller than a minimum acceptable size, if it is empty, or if it can be approximated well enough with one or two simple cameras. A tile is empty if none of its pixels contains a valid ray. The approximation of a tile with simple cameras is described in Section 4.5. If the approximation is successful, the one or two newly created simple cameras are added to the list M^*. If none of the halting conditions are met, the tile is subdivided in 4 and the algorithm proceeds recursively on each of the 4 quarters.

4. The k-ray camera class

The goal is to devise a camera model that is sufficiently powerful to closely approximate a large set of contiguous, coherent, but non-concurrent rays, yet is simple enough to provide efficient projection. Although in theory numerical non-linear optimization methods could be employed to solve for arbitrarily complex projection equations, we have opted for camera models that offer closed-form projection. In addition to efficiency and robustness, closed-form projection also has the advantage of being suitable for hardware implementation. The projection equations of the cameras developed in this paper can already be implemented on modern GPUs by taking advantage of vertex-level programmability. Future hardware could support them at full speed as part of the "fixed" pipeline.

4.1. The k-ray camera

A k-ray camera is computed from k construction rays $(o_{xi}, o_{yi}, o_{zi}, \theta_i, \varphi_i)$, $i = 1$ to n as follows. The first step is to compute the image plane Π by least squares fitting a plane to the k ray origins (o_{xi}, o_{yi}, o_{zi}). Then a coordinate system local to the camera is computed. The z axis is given by the normal of Π (Figure 3). The axis is oriented such that all k ray tips lie in the positive z half-space. If this is not possible, the construction rays are degenerate and the construction fails. In practice the maximum angle between two rays is small (the rays have similar directions), and the construction typically succeeds. The x and y axes, and the origin of the local coordinate system O are chosen arbitrarily in Π.

The next step is to transform the input rays to the local coordinate system and to compute the (q_i, r_i) and (s_i, t_i) intersections of each ray with the local planes $z = 0$, and $z = 1$, respectively. The ray at image plane coordinates (q, r) is given by two points $(q, r, s(q, r), t(q, r))$, where s and t
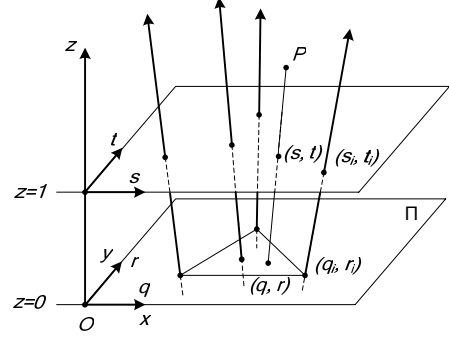

Figure 3 k-ray camera construction ($k = 4$), and projection of point P at local image plane location (q, r).

are expressions in q and r. The coefficients of $s(q, r)$ and $t(q, r)$ are computed by solving the system of $2k$ equations:

$$s_i = s(q_i, r_i), t_i = t(q_i, r_i), i = 1\ to\ k \qquad (4)$$

In order for the construction to succeed, the system of equations has to be fully determined and have $2k$ unknowns, which means that the expressions $s(q, r)$ and $t(q, r)$ must total $2k$ coefficients. Many expressions are possible. This means that each value of k defines a subclass of cameras. We choose polynomials in q and r with k coefficients each:

$$\begin{aligned} s(q, r) &= P_s(a_1, a_2, ..., a_k, q, r) \\ t(q, r) &= P_t(b_1, b_2, ..., b_k, q, r) \end{aligned} \qquad (5)$$

Polynomials have the advantage that the model camera system of equations (4) is linear, and the coefficients can be computed trivially.

In order to project a 3D point P, the point is first transformed to the camera's local coordinate system. Let (x, y, z) be its local coordinates. The projection equation is given by the system of 3 equations and 3 unknowns (q, r, w):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} q \\ r \\ 0 \end{bmatrix} + \begin{bmatrix} s(q,r) - q \\ t(q,r) - r \\ 1 \end{bmatrix} w \qquad (6)$$

The third equation provides w so the projection equation is simplified to a system of 2 equations and 2 unknowns:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} q + (s(q,r) - q)z \\ r + (t(q,r) - r)z \end{bmatrix} \qquad (7)$$

Since z is constant, the order of the projection equations (7) is given by the order of the polynomials $s(q, r)$ and $t(q, r)$.

For most imaging systems the precise modeling of the minimum distance from the image plane for a point to have a valid projection is not important, and the approximation provided by the image plane fitted to the origins of the rays is sufficient. If the tails of the rays have to be taken into account exactly, a hither surface can be fitted to the tails (e.g. a polyhedron with triangular faces, or a polynomial similar to the expressions of s and t). For computer graphics applications a far clipping plane is needed to limit the maximum values that are stored in the z-buffer, and to improve z-buffering precision. In order to simplify the clipping with the lateral faces of the viewing volume of the camera, we ensure that 3 of the k construction rays form a base triangle that contains the other k-3 rays. This way, projections that are outside the projection of the base triangle onto the local image plane (Figure 3) are easily clipped.

Since it is our intention to use k-ray cameras to model a set of rays of a general camera, the k-ray camera has to provide the general camera image location where a 3D point projects. For this the construction rays are augmented with two scalars (u, v) which specify the final image coordinates for the ray. During camera construction, we compute two additional polynomials P_u and P_v for interpolation of the final u and v values. During projection, once the local image plane point (q, r) is computed, the final (u, v) values are computed as $(P_u(q, r), P_v(q, r))$.

4.2. The 3-ray camera

The simplest k-ray camera is the 3-ray camera. There are no construction rays inside the base triangle, since k-3=0. The s and t polynomials are linear in q and r:

$$s(q, r) = a_1q + a_2r + a_3$$
$$t(q, r) = b_1q + b_2r + b_3 \qquad (8)$$

The projection equations are also linear:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} q + (a_1q + a_2r + a_3 - q)z \\ r + (b_1q + b_2r + b_3 - r)z \end{bmatrix} \qquad (9)$$

4.3. The 4-ray camera

A useful variant of the 4-ray camera is the bilinear 4-ray camera. The 4 construction rays form a convex quadrilateral on the image plane. The rays of the camera are obtained by bilinearly interpolating the 4 construction rays:

$$q(\alpha,\beta) = a_{11}\alpha\beta + a_{12}\alpha + a_{13}\beta + a_{14}$$
$$r(\alpha,\beta) = a_{21}\alpha\beta + a_{22}\alpha + a_{23}\beta + a_{24}$$
$$s(\alpha,\beta) = a_{31}\alpha\beta + a_{32}\alpha + a_{33}\beta + a_{34}$$
$$t(\alpha,\beta) = a_{41}\alpha\beta + a_{42}\alpha + a_{43}\beta + a_{44} \qquad (10)$$

In order to project a point P, α and β are computed from the plane through P that is parallel to the local image plane. If (s_i', t_i') are the intersections of the k rays with this plane, α and β are found by inverse bilinear interpolation. The frustum of the bilinear 4-ray camera is delimited by the conditions that α and β be positive fractional numbers. Once α and β are known, the local image plane projection (q, r) is computed from equations (10) to complete the projection. See Figure 2 (*top*) and the accompanying video for images rendered with the bilinear 4-ray camera.
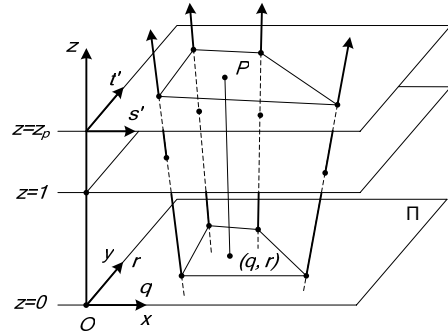

Figure 4 Bilinear 4-ray camera projection. The fractional quantities α and β are computed as the bilinear interpolation coefficients that generate P when applied to the vertices of the quadrilateral in plane (s', t').

4.4. The 6-ray camera

In order to answer the question "what is the most powerful k-ray camera with closed form projection?", recall that a system of two equations with two unknowns can be solved in closed form if the system degree, computed as the product of the total degrees of the individual equations, is less than four.

A complete two-variable quadratic has 6 coefficients, thus the most powerful camera model with closed-form projection is a 6-ray camera. The 6-ray camera has 6-3=3 construction rays inside a base triangle. The s and t polynomials are:

$$s(q, r) = a_1q^2 + a_2r^2 + a_3rq + a_4r + a_5q + a_6$$
$$t(q, r) = b_1q^2 + b_2r^2 + b_3rq + b_4r + b_5q + b_6 \qquad (11)$$

The projection equations are:

5

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} q + (a_1 q^2 + a_2 r^2 + a_3 qr + a_4 r + a_5 q + a_6 - q)z \\ r + (b_1 q^2 + b_2 r^2 + b_3 qr + b_4 r + b_5 q + b_6 - r)z \end{bmatrix} \quad (12)$$

Each of the two equations has degree two and all terms are present. The system is solved using a resultant, which simplifies it to a quartic equation in a single variable. Out of the four solutions, the one inside the base triangle is kept. The relatively simple reflectors used in our tests did not reveal situations where a unique solution could not be separated; as future work we will investigate the conditions when multiple "valid" projections occur. See Figure 2 (*bottom*) and the accompanying video for images rendered with the bilinear 4-ray camera.

4.5. Error-bounded k-ray camera approximation

Recall that the recursive image subdivision stops if the current rectangular image tile can be approximated with 1 or 2 k-ray cameras. Our current implementation models all simple cameras with a single type of k-ray cameras. The 3- and 6-ray cameras have a triangular base so the tile is first split in two using one of its diagonals. The bilinear 4-ray camera has a quadrilateral base and maps directly to a tile.

The *k* construction rays are given by the vertices of the base in the case of the 3-ray and bilinear 4-ray camera. For the 6-ray camera the 3 additional construction rays are chosen at base triangle barycentric coordinates *(a, (1-a)/2, (1-a)/2)* and permutations, for a uniform sampling of the base triangle. A typical value for *a* is 0.7. If the construction succeeds, the projection approximation introduced by the k-ray camera is evaluated at a few original rays inside the k-ray camera frustum.

The projection error is 0 at the construction rays since they are original camera rays and since they are also part of the k-ray camera. The error grows with the distance to the construction rays. Since the rays are coherent, it is generally sufficient to measure the projection error for one or a few rays. A conservative algorithm would test at all rays replaced by the k-ray camera. We choose the test ray at the center of the base. The projection error at a test ray is estimated by projecting the origin and the tip of the ray using the k-ray camera, and computing the distance to the true projection, as indicated by the *(u, v)* values of the ray. A distance smaller than ε indicates that the k-ray camera captures the original well enough and the recursion stops.

5. C⁰-continuous projection

We have shown how the projection error is controlled within a k-ray camera, which also limits the overall projection error. In addition to the small absolute error, another desirable property of the overall projection
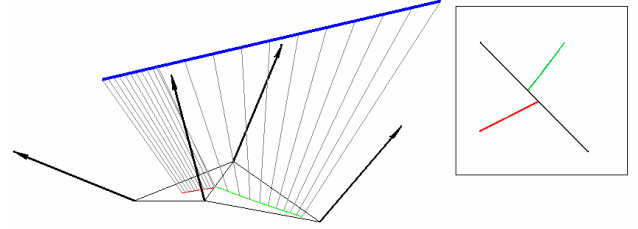

Figure 5 Projection discontinuity between two 3-ray cameras with a shared edge and different image planes. In the 3D view (*left*), the blue 3D line projects in both 3-ray cameras (red and green segments). The 2D view (*right*) magnifies the intersection region to visualize the discontinuity.

function is continuity between adjacent simple cameras. When the image planes of the two adjacent k-ray cameras are the same, the rays generated on the shared edge are the same, and the projection is C⁰-continuous. When the field of view of the overall camera is small, it is possible to use the same two planes for all simple cameras, chosen to be perpendicular to the view direction. However, it is not always possible to use the same planes. For example, in the case of a pinhole aimed at a sphere, the reflected rays cannot be parameterized with the same two planes. When the image planes of the adjacent k-ray cameras are different, the rays at the endpoints of the shared edge have different length, and the projection is discontinuous (Figure 5).

A continuous projection can be obtained if the 3-ray camera is modified to interpolate between rays of unit length (Figure 6). This way the rays at the endpoints of the shared edge are exactly the same. The normalized rays imply that the second parameterization plane *(s, t)* is not parallel to the image plane. The projection equation of the continuous 3-ray camera is more complicated.
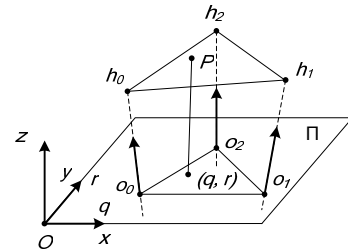

Figure 6 Continuous 3-ray camera.

When projecting a point *P,* the first step is to find the plane that passes through *P* and cuts segments of same length on the 3 construction rays. Let the construction rays have origins o_i and directions d_i in local camera coordinates. The 3 intersections h_i with the construction rays, and then the plane's normal *n* are given by:
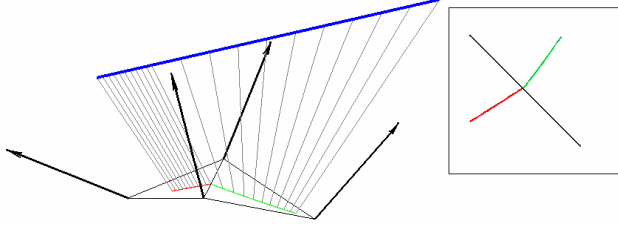
Figure 7 Projection with adjacent continuous 3-ray cameras.

$$h_i = o_i + d_i w, \; i = 1, 2, 3$$
$$n = (h_1 - h_0) \times (h_2 - h_0) \tag{13}$$

where w is the length of the construction ray segments and can be found from equation 14.

$$(P - h_0)n = 0 \tag{14}$$

Equation 14 is cubic in w, with one real solution in front of the image plane and inside the frustum of the 3-ray camera. Once w is known, the points h_i are known, the barycentric coordinates of P are computed in the triangle $h_0 h_1 h_2$, and are used to find the image plane projection *(q, r)*. In conclusion, the projection discontinuity is removed (Figure 7, Figure 8) at the price of a more expensive projection equation. For $k > 3$, the continuous k-ray camera does not have closed-form projection.
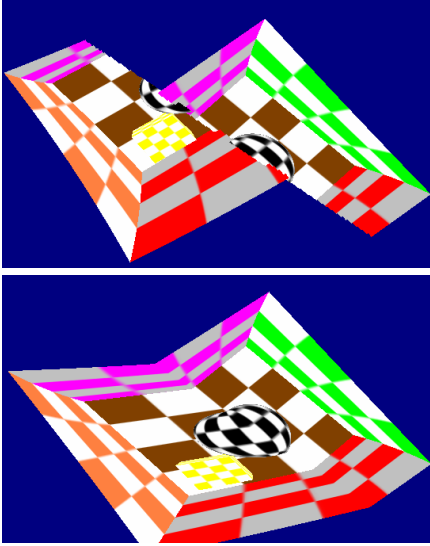

Figure 8. Images rendered with two adjacent 3-ray cameras (*top*), and two adjacent continuous 3-ray cameras (*bottom*).

6. Results and discussion

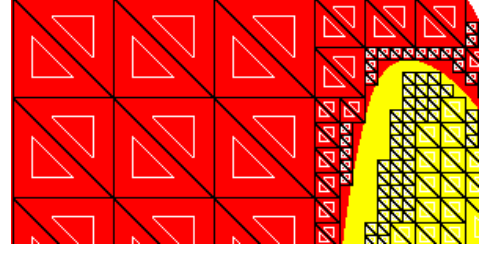We have described how to subdivide coherent general

Figure 9. Visualization of 6-ray cameras created for table entry 1,484 (*fragment*). Rays (1, 2, 3) and (4, 5, 6) are shown with black and white triangles, respectiv.

cameras in disjoint simple cameras. To model simple cameras efficiently we have introduced k-ray cameras, a novel class of non-pinhole cameras. We have developed three novel cameras part of this class: the bilinear 4-, the 6-, and the continuous 3-ray camera. The k-ray cameras have closed-form projection. The projection times for 1 million vertices are 0.42s, 0.55s, 0.72s, 0.85s, and 3.47s, for the planar pinhole, 3-, 4-, continuous 3-, and 6-ray camera, respectively. Timing measurements were performed on a 3.4GHz 2GB Pentium 4 Xeon.

We tested our algorithm on 4 simulated catadioptric systems with non-concurrent reflected rays. Three systems consisted of a spherical mirror and a planar pinhole camera (horizontal field of view 60°, resolution 720x480). The mirror radius and the distance from the camera to the mirror surface were (5m/0.15m, 1m/0.15m, and 0.1m/0.05m). Although spherical, the mirrors are modeled as triangle meshes, with normals computed at a vertex by averaging the normals of triangles that share the vertex.

The fourth system had two spherical mirrors with radius 0.1m, and produced both first and second order reflected rays (see Figure 1). Table 1 reports the number of simple cameras for each of the imaging systems. As expected, the complexity of the compound camera model is lower for lower curvature reflectors, for higher projection error thresholds, and for the more sophisticated simple camera.

	$\varepsilon = 5$pix		$\varepsilon = 1$pix		$\varepsilon = 0.1$pix	
	3-ray	6-ray	3-ray	6-ray	3-ray	6-ray
R = 5m	48	12	48	12	498	48
R = 1m	48	12	204	48	2,442	192
R = 0.1m	204	48	1,020	144	10,536	732
2 x (R = 0.1m)	3,360	956	5,660	1032	17,500	1,484

Table 1 Number of simple cameras of compound camera model.

The original set of over 300,000 rays is simplified to fewer than 20,000 3-ray simple cameras, or to fewer than 2,000 6-ray simple cameras, while the projection accuracy is better than 0.1pixels. The improvement brought by the

6-ray camera over the 3-ray camera is even larger for small ε values. Figure 9 and Figure 10 visualize the simple cameras constructed by the algorithm.

Many of the simple cameras are created to follow the curves that delimit the first order reflected rays from the second reflected rays and from the blind (background) pixels. In our experiments we used a minimum tile size of 2 pixels, which leaves a small gap in the simple camera triangulation. Triangle vertices that are seen by missing simple cameras do not have a valid projection and the triangles that use them are discarded, which causes a gap at the borders, in the rendered image (Figure 1). If larger gaps can be tolerated, the minimum tile size can be increased, which considerably reduces the overall number of simple cameras. Borders between first order ray and no ray regions can be handled well and inexpensively by extending the reflector for the subdivision phase. A coarse triangulation can be safely used since the jagged borders are off the actual surface of the reflector.
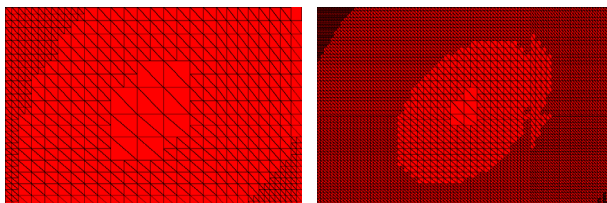

Figure 10. Visualization of 3-ray cameras. $R = 0.1$m, $\varepsilon = 1$pix (*left*). $R = 0.1$m, $\varepsilon = 0.1$pix (*right*).

The k-ray camera class and the approach to modeling complex camera models show that models that deviate from the single view point constraint can be practical. The camera models developed here are only the beginning. Many interpolation functions are possible and most are yet to be explored. A possible enhancement for catadioptric systems is to decouple final image coordinate interpolation from ray interpolation: the k-ray cameras project onto their local image planes, which the planar pinhole camera of the system foreshortens appropriately, increasing accuracy.

The camera models developed here can be readily used in a variety of applications. A possibly more important contribution is the argument made for considering cameras flexible tools that can be adjusted to best address a given problem in computer vision, computer graphics, or visualization.

References

[1] E. H. Adelson and J. R. Bergen. The Plenoptic Function and the Elements of Early Vision. The MIT Press, 1991.
[2] E. Chen. QuickTime VR – An Image Based Approach to virtual Environment Navigation. Computer Graphics: Proc. of SIGGRAPH 95, pages 29-38, August 1995.
[3] A. Glassner. An introduction to ray tracing. Academic Press, 1989.
[4] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. In Computer Graphics, Proc. SIGGRAPH, page 43, 1996.
[5] D. Grossberg and S. Nayar. A General Imaging Model and a Method for Finding its Parameters. In Proceedings of ICCV 2001.
[6] R. Gupta, and R. Hartley. Linear Pushbroom Cameras. IEEE Trans. Pattern Analysis and Machine Intell. vol. 19, no. 9 (1997) 963–975.
[7] W. Heidrich, P. Slusallek, and H.-P. Seidel. An Image-Based Model for Realistic Lens Systems in Interactive Computer Graphics. In Proc. of Graphics Interface 1997.
[8] J. Hong. Image based Homing. In Proc. of IEEE International Conference on Robotics and Automation, May 1991.
[9] A. Krishnan and N. Ahuja. Panoramic Image Acquisition. In Proc. of IEEE CVPR, June 1996.
[10] M. Levoy and P. Hanrahan. Light field rendering. In Computer Graphics, Proc. SIGGRAPH, page 31, 1996.
[11] L. McMillan and G. Bishop. Plenoptic Modeling: An Image-Based Rendering System. Computer Graphics: Proc. of SIGGRAPH 1995.
[12] C. Mei, V. Popescu, E. Sacks. The Occlusion Camera. Computer Graphics Forum, volume 24, issue 3, September 2005.
[13] K. Miyamoto. Fish eye lens. Journal of Optical Society of America, 54(8):1060–1061, August 1964.
[14] V. Nalwa. A True Omnidirectional Viewer. Technical report. Bell Laboratories, Holmdel, NJ, U.S.A. 1996.
[15] S. Nayar. Catadioptric Omnidirectional Camera. In Proceedings of CVPR 1997.
[16] T. Pajdla. Geometry of Two-Slit Camera. Research Report CTU–CMP–2002–02.
[17] P. Rademacher and G. Bishop. Multiple-Center of Projection Images. In Proc. of SIGGRAPH 1998, pp. 199-206.
[18] J. Shade et al. Layered Depth Images. In Proc. of SIGGRAPH 1998, pp231-242.
[19] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision. International Journal of Robotics and Automation, 3(4):323–344, Aug 1987.
[20] R. Wood. Fish-eye views, and vision under water. Philosophical Magazine, 12(Series 6):159-162, 1906.
[21] Y. Yagi and S Kawato. Panoramic Scene Analysis with Conic Projection. In Proc. of International Conference on Robots and Systems IROS, 1990.
[22] K. Yamazawa, Y. Yagi and M. Yachida. Obstacle Avoidance with Omnidirectional Image Sensor HyperOmni Vision. Proc. of IEEE International Conference on Robotics and Automation May 1995.
[23] J. Yu and L. McMillan. General linear cameras. In 8th European Conference on Computer Vision, vol. 2, 2004, pp. 14–27.
[24] J. Yu and L. McMillan. Modelling Reflections via Multiperspective imaging. In Proc. of IEEE CVPR 2005.