COMPUTER GRAPHICS forum

DOI: 10.1111/j.1467-8659.2009.01587.x

COMPUTER GRAPHICS forum
Volume 29 (2010), number 1 pp. 227–246

# Multi-Perspective Modelling, Rendering and Imaging

J. Yu[1], L. McMillan[2] and P. Sturm[3]

[1]Department of Computer and Information Sciences, University of Delaware, USA
[2]Department of Computer Science, The University of North Carolina at Chapel Hill, USA
[3]INRIA Grenoble – Rhône-Alpes, Montbonnot, France

**Abstract**

*A perspective image represents the spatial relationships of objects in a scene as they appear from a single viewpoint. In contrast, a multi-perspective image combines what is seen from several viewpoints into a single image. Despite their incongruity of view, effective multi-perspective images are able to preserve spatial coherence and can depict, within a single context, details of a scene that are simultaneously inaccessible from a single view, yet easily interpretable by a viewer. In computer vision, multi-perspective images have been used for analysing structure revealed via motion and generating panoramic images with a wide field-of-view using mirrors.*

*In this STAR, we provide a practical guide on topics in multi-perspective modelling and rendering methods and multi-perspective imaging systems. We start with a brief review of multi-perspective image techniques frequently employed by artists such as the visual paradoxes of Escher, the Cubism of Picasso and Braque and multi-perspective panoramas in cel animations. We then characterize existing multi-perspective camera models, with an emphasis on their underlying geometry and image properties. We demonstrate how to use these camera models for creating specific multi-perspective rendering effects. Furthermore, we show that many of these cameras satisfy the multi-perspective stereo constraints and we demonstrate several multi-perspective imaging systems for extracting 3D geometry for computer vision.*

*The participants learn about topics in multi-perspective modelling and rendering for generating compelling pictures for computer graphics and in multi-perspective imaging for extracting 3D geometry for computer vision. We hope to provide enough fundamentals to satisfy the technical specialist without intimidating curious digital artists interested in multi-perspective images.*

*The intended audience includes digital artists, photographers and computer graphics and computer vision researchers using or building multi-perspective cameras. They will learn about multi-perspective modelling and rendering, along with many real world multi-perspective imaging systems.*

## 1. Introduction

Camera models are fundamental to the fields of computer vision, computer graphics and photogrammetry. The classic pinhole and orthographic camera models have long served as the workhorse of 3D imaging applications. However, perspective projection is surprisingly rare in Art: artists, architects and engineers regularly draw using multi-perspective projections. Despite their incongruity of view, effective multi-perspective images are still able to preserve spatial coherence. More importantly, multi-perspective images can depict, within a single context, details of a scene that are simultaneously inaccessible from a single view, yet easily interpretable by a viewer.

Historically, multi-perspective images have been frequently employed by the pre-Renaissance and post-impressionist artists to depict more than can be seen from

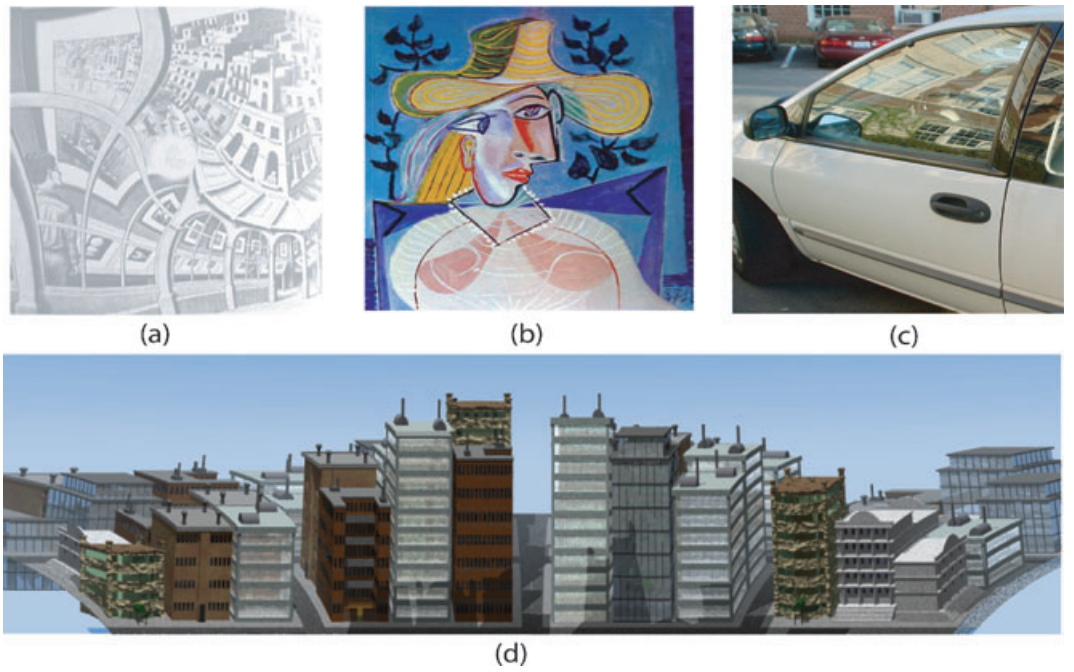


**Figure 1:** *Various types of multi-perspective images. (a) Hall city by M.C. Escher. (b) Portrait of a young girl by Pablo Picasso. (c) A reflection image observed on the car window. (d) A cross-slit panorama rendered using ray tracing.*

any specific point. Classic examples include the visual paradoxes of Escher [Loc81], and the Cubism of Picasso and Matisse [RM84]. Escher uses highly curved projection models to generate 'impossible' perspectives of a scene as shown in Figure 1(a). Picasso and other Cubism pioneers made effective of rearranging different parts of the depicted scene while maintaining their local spatial relationships, which results in an incongruous spatial systems. Despite the large disparities between projections, the overall impression of a three-dimensional space remains intact, as shown in Figure 1(b).

Multi-perspective images have also been used as backdrops in cel animation to effectively depict camera motion in a single panorama [TJ95]. In cel animations, a moving foreground element would translate across the background image with no change in parallax. A local camera is attached with the moving foreground element to generate an illusion of motion. In contrast to the foreground elements, backdrop artists often draw by hand a multi-perspective projection background to create a deeper, more dynamic environment. Computer-generated multi-perspective panoramas, as presented by Wood *et al.* [WFH*97], combined elements of multiple pinhole cameras into a single image using a semi-automatic image registration process. They relied on optimization techniques, as well as optical flow and blending transitions between views.

Finally, multi-perspective images have received attention from the computer vision community for analysing structure revealed via motion [Sei01, PBP01] and generating panoramic images with a wide field-of-view using mirrors [Nay97]. Several researchers have proposed alternative multi-perspective camera models which capture rays from different points in space. These multi-perspective cameras include pushbroom cameras [GH97], which collect rays along parallel planes from points swept along a linear trajectory, and two-slit cameras [Paj02b], which collect all rays passing through two lines. Finally in our everyday lives, we experience multi-perspective images when observing reflections of curved surfaces as shown in Figure 1(c).

Applications of multi-perspective cameras and images are numerous in both computer graphics and computer vision, for example: (i) analysing and rendering reflections and refractions on curved surfaces; (ii) illustrating visual contents of many views in a single image; (iii) capturing much more complete depth information using catadioptric cameras; (iv) synthesizing re-renderable photos via image-based rendering (IBR), etc.

### 1.1. Perspective cameras

The concept of a perspective, or pinhole, camera predates modern history. In particular, the model of a camera obscura has been described throughout history and across many cultures. Precise mathematical models of pinhole cameras are fundamental to the fields of photogrammetry, computer vision and 3D computer graphics.

Geometrically, a pinhole camera collects rays passing through a common 3D point, which is called the centre of projection (COP). The geometry of each pinhole camera, therefore, can be uniquely defined by only three parameters (the position of COP in 3D). The image of a pinhole camera requires specifying an image plane for projecting these rays. The image transformation due to the change of the image plane is referred to as an homography.

The pinhole imaging process, thus, can be decomposed into two parts: projecting the scene geometry into rays and mapping the rays onto the image plane. We refer to the first part as *projection* and the second as *collineation*. It has been shown that the projection and collineation can be uniformly described by the classic $3 \times 4$ pinhole camera matrix [HZ04], which combines six extrinsic and five intrinsic camera parameters into a single operator that maps homogenous 3D points to a 2D image plane. These mappings are unique up to a scale factor, and the same infrastructure can also be adapted to describe orthographic cameras. Because pinhole cameras capture similar images to those we observe from our eyes, that is, in human perspectives, pinhole cameras are also called perspective cameras.

The simple pinhole projection model has many nice properties that are useful to computer graphics and computer vision applications. For instance, under perspective projection, all lines in the scene are projected to lines on the image. Similarly, a triangle in 3D space is projected in a triangle on the pinhole image. Thus, by representing the scene geometry using triangles, one can efficiently render the entire scene by projecting the triangles onto the image plane and then rasterizing the triangles in the image space. Furthermore, in light of two-eye human perception, modern computer vision systems use two pinhole cameras to generate binocular stereo. The perspective projection model also induces the so-called epipolar constraints [BBM87], which significantly reduces the search space for establishing the correspondences that determine a point's depth.

### 1.2. Beyond pinhole cameras

More general camera models do not need to satisfy the same physical constraints that a perspective camera does, that is, not all rays collected by the camera need to pass through a common point. Such cameras are often referred to as *multi-perspective cameras* or non-central cameras, and the corresponding images are called *multi-perspective images*. However, most multi-perspective cameras are still expected to satisfy some properties to capture meaningful (interpretable) images. A list of these properties include:

- 2D Ray Subspace. A multi-perspective camera, like any camera, collects an image. Typical images, which will be the only ones considered in this paper, are a two dimensional subset of the radiant energy along rays.
- Continuity. A multi-perspective camera should collect a 'smoothly' varying set of rays. This guarantees that continuous objects will project to continuous images. If we further assume that all rays can be embedded as points in some *ray space*, then the first and second criteria indicate that a multi-perspective camera captures rays lying on a 2D continuous manifold in this ray space.
- Unique Projection. Except singularities (which are discussed in detail later), a multi-perspective camera should generally image each 3D point $P$ at most once, that is, the camera has at most one ray that passes through $P$. In a perspective camera, only the point lying at the COP lies on multiple rays.

## 2. Multi-Perspective Cameras

In contrast to pinhole and orthographic cameras, which can be uniformly described using the $3 \times 4$ camera matrix, multi-perspective camera models are defined less precisely. In practice, multi-perspective camera models are described by constructions. By this we mean that a system or process is described for generating each specific class but there is not always a closed-form expression for the projection transformation.

### 2.1. Classical multi-perspective cameras

Pushbroom sensors are commonly used in satellite cameras for generating 2D images of the earths surface [GH97]. A pushbroom camera consists of an optical system projecting a scene onto a linear array of sensors. The pushbroom sensor is mounted on a moving rail, and as the platform moves, the view plane sweeps out a volume of space and forms a pushbroom image on the sensor as shown in Figure 6(c). In practice, the pushbroom view plane is often replaced by a perspective camera and the pushbroom image is synthesized by assembling the same column of each perspective image as shown in Figure 2(a). 'True' pushbroom cameras consisting of a linear CCD of thousands of pixels (line scan camera), are routinely used in satellite imagery.

Another popular class of multi-perspective cameras are the cross-slit cameras. A cross-slit camera has two slits $l_1$ and $l_2$ that are oblique (neither parallel nor coplanar) in 3D space. The camera collects rays that simultaneously pass through the two slits and projects them onto an image plane, as shown in Figure 6(d). Zomet *et al.* [ZFPW03] carried out an extensive analysis and modelling of cross-slit cameras. They have shown that a cross-slit camera can be synthesized similar to the pushbroom camera by stitching linearly varying columns across the pinhole viewing cameras, as shown in Figure 2(c).

Pajdla [Paj02a] recently proposed the oblique camera model. The oblique camera is the opposite extremal of the pinhole camera. A pinhole camera collects rays passing


**Figure 2:** *Pushbroom and cross-slit camera. (a) The stationary column sampling routine for synthesizing a pushbroom panorama (b). (c) The linearly varying column sampling routine for synthesizing a cross-slit panorama (d) (courtesy of Steve Seitz [Sei01]).*

**Figure 3:** *A typical catadioptric image.*

through a common point in 3D whereas an oblique camera collects pairwise oblique rays. This means that no two rays in an oblique camera can intersect or be parallel. To give a complete description of the rays collected by the oblique camera, Pajdla used transition closure to find all rays. He further used the quadratic ruling surface to determine its epipolar geometry. One special oblique camera is the bilinear camera [YM04a], where any two rays form a non-degenerate bilinear surface, as shown in Figure 6(g).

### 2.2. Catadioptric cameras

Another class of physically realizable multi-perspective cameras are catadioptric imaging systems that consist of a combination of cameras and mirrors. This is usually done to achieve a large field of view, for example, by pointing a pinhole camera at a curved mirror (see a sample image in Figure 3). A large field of view is obviously beneficial for many applications, such as video surveillance, autonomous navigation, obstacle avoidance and of course panoramic image acquisition. Other motivations for using catadioptric systems are to achieve a multi-perspective imaging device for IBR, to sample the plenoptic function more uniformly than with perspective cameras, or to achieve customized imaging devices with a camera geometry that is optimized for a given application [SNG04].

Most commonly used catadioptric systems are designed to realize a central projection, that is, there exists a single point (optical centre) through which all projection rays pass. This allows to generate perspectively correct images from sections of the acquired image. All possible classes of central catadioptric systems are described in [Nay97, BN99]. The two practical setups consist of a pinhole camera pointed at a hyperboloidal mirror and an orthographic camera, for example, realized using a tele-lens, pointed at a paraboloidal mirror. In both cases, the camera must be carefully placed relative to the mirror [Kan00, SGN01]: the camera's optical centre must coincide with one of the mirror's focii. When this is not the case, then the system becomes a multi-perspective one; however, if the deviation from the earlier requirement is small, a central camera model may still be sufficient in many applications.

To model these multi-perspective cameras, most previous research has been restricted to simple parametric reflectors such as spherical or conical mirrors [CS97, YKT94], and equiangular mirrors [SDM04]. The *envelope* of the reflected rays, often referred to as the caustic, has been used to characterize multi-perspective distortions [SGN01, SGN03]. However, the caustic surface models every ray as originating from a single, but spatially varying, pinhole camera, therefore, it does not provide much insight into the group behaviour of neighbouring rays.

Yu and McMillan [YM05a] provided an analytical framework to locally model reflections as specific multi-perspective cameras. They have shown that local reflections observed by a pinhole or an orthographic camera can be

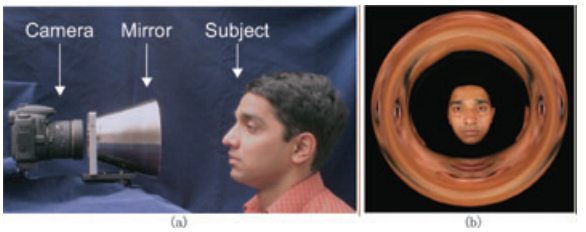


**Figure 4:** *A multi-perspective image (b) captured by the radial imaging system (a) (Courtesy of Shree Nayar [KN06]).*

characterized by only four types of multi-perspective cameras: cross-slit, pushbroom, pinhole or orthographic. By mapping a slit to a linear constraint in the 4D ray space, they have further shown that pushbroom, orthographic and pinhole cameras can all be viewed as special cases of cross-slit cameras: when the two slits intersect, it transforms into a pinhole camera; when one of the slits goes to infinity, the cross-slit transforms into a pushbroom; and when both slits go to infinity, it transforms into an orthographic camera. The imaging properties of these multi-perspective cameras explain the complicated visual distortions seen in a reflected image.

Other catadioptric systems are intentionally designed for multi-perspective image acquisition, for example, systems based on an array of spherical mirrors [LCWT06]. The mirror shape proposed in [HB00] leads to a multi-perspective camera that represents a good compromise between size of field of view and non-perspective image distortions. A number of systems for acquiring stereo pairs or sequences with a single camera exist; they typically use two mirrors, a mirror with a double lobe or a mirror and an additional lens [YA06, JKK05, CSH04, FB05, MSEY05]. It has also been shown how to design systems that acquire optically rectified stereo images [GN02].

Recently, Kuthirummal and Nayar [KN06] proposed a radial imaging system that captures the scene from multiple viewpoints within a single image. Their system consists of a conventional camera looking through a hollow rotationally symmetric mirror polished on the inside, as shown in Figure 4. The field of view of the camera is folded inwards and consequently the scene is captured from multiple viewpoints within a single image. By using a single camera, the radiometric properties are the same across all views. Therefore, no synchronization or calibration is required. The radial imaging system can also be viewed as a multi-perspective imaging system. It has a circular locus of virtual viewpoints and it has the same epipolar geometry as the cyclographs [Sei01]. By capturing two images by translating the object or the imaging system, one can easily reconstruct the 3D structure of the scene. Other applications include acquiring 3D textures, capturing complete objects and sampling and estimating Bidirectional Reflectance Distribution Function (BRDF).

### 2.3. Multi-view imaging

A closely related area to multi-perspective imaging is multi-viewpoint imaging, where the same scene is imaged from different viewpoints. The classical light field cameras [LH96], lumigraphs [GGSC96, BBM*01] and concentric and panoramic mosaics [PBP01, SKS99] move a single camera to capture multiple views towards the scene object. Levoy *et al.* [WJV*05] developed a light field system with multiple synchronized cameras. Conceptually, these methods capture a ray database, or more specifically, a database of radiance measurements along rays, and new views can be synthesized by querying the ray database. In theory, multi-perspective images can also be generated in a similar way [YM04a]. However, since the ray database contains only a finite sampling of the rays, therefore, aliasing can be introduced during initial sampling and final reconstruction.

### 2.4. Multi-perspective photography

Finally, we discuss photographic techniques that have been developed to directly capture multi-perspective images.

The Cirkut camera was introduced by Eastman Kodak in the late 1800s to directly capture panoramas, where the camera is rotated about a vertical axis for scanning the scene in front of the camera. The film moves at the same velocity as the scanning camera and eventually constructs an image containing a view of up to 360° angle. Davidhazy [Dav87] proposed a peripheral camera to record the surface of cylindrical objects. Instead of rotating the camera, peripheral photography captures all sides of an object by imaging a rotating object through a narrow slit placed in front of a moving film. Seitz [Sei01] proposed a similar cyclograph model by stitching different slices from a sequence of images to provide an inward looking panoramic view of the object.

The crossed-slit anamorphoser, credited to Ducos du Hauron, modifies pinhole camera by replacing the pinhole with a pair of narrow, perpendicularly crossed slits spaced apart along the camera axis. It is a physical realization of the crossed-slit camera [ZFPW03]. The pair of slits working together thus constitutes a multi-perspective camera in which the image is stretched or compressed in one direction more than in the other. This type of distortion is called 'anamorphic' or 'anamorphotic' and the degree of anamorphic compression closely matches the estimated distortions using the crossed-slit model.

The emerging field of computational photography has the potential to benefit multi-perspective imaging applications. One important component in computational photography is the generalized optics that treats each optical element as a 4D

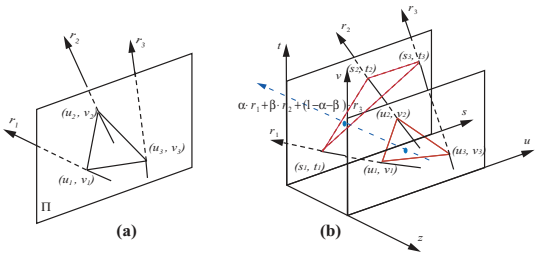


**Figure 5:** *General Linear Camera Model. (a) A GLC is characterized by three rays originated from the image plane. (b) It collects all possible affine combination of three rays.*

ray-bender that modifies the rays in a light field [RTM*06]. The collected ray bundles can then be regrouped into separate measurements of the Plenoptic function [MB05]. Ng *et al.* [NLB*05] developed a hand-held plenoptic camera to record the full 4D light field in a single image. Georgiev *et al.* [GZC*06] modified their design to produce higher spatial resolution by trading-off the light field's angular resolution. Veeraraghavan *et al.* [VRA*07] used a patterned attenuating mask to encode the light field. By inserting the mask at different location in the optical path of the camera, they achieve dynamic frequency modulation of the light field. Unger *et al.* [UWH*03] combined a high-resolution camera and a spherical mirror array to capture the incident light fields with a much wider field-of-view. Since a multi-perspective image can be efficiently synthesized by querying the light field (Section 4), all these cameras can be potentially used as a multi-perspective camera.

### 2.5. General linear cameras

Recently a new camera model has been developed called the general linear camera (GLC) [YM04a]. This single model describes typical pinhole and orthographic cameras, as well as many commonly studied multi-perspective cameras including pushbroom and cross-slit cameras. GLCs also include many lesser known multi-perspective cameras, such as the pencil, twisted orthographic, EPI and bilinear cameras, shown in Figure 6.

A GLC is defined by three generator rays that originate from three points $p_1(u_1, v_1)$, $p_2(u_2, v_2)$ and $p_3(u_3, v_3)$ on an image plane $\Pi_{\text{image}}$, as is shown in Figure 5. A GLC collects radiance measurements along all possible 'affine combinations' of these three rays as defined under a two-plane parametrization (2PP). The 2PP form is commonly used for representing light fields [LH96] and lumigraphs [GGSC96]. Under this parametrization, an affine combination of three rays $r_i(s_i, t_i, u_i, v_i)$, $i = 1, 2, 3$, is defined as

$$r = \alpha(s_1, t_1, u_1, v_1) + \beta(s_2, t_2, u_2, v_2) + (1 - \alpha - \beta)(s_3, t_3, u_3, v_3).$$

This implies that a GLC corresponds to a 2D linear subspace (a 2D hyperplane) in the 4D ray space. In [YM05b],

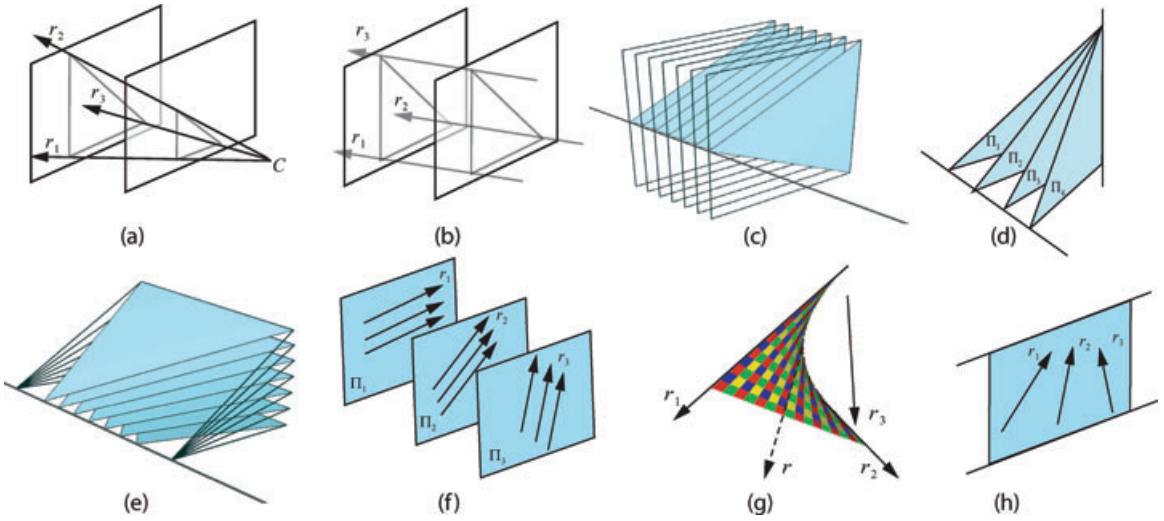


**Figure 6:** *Perspective and Multi-Perspective Cameras. (a) In a pinhole camera, all rays pass through a single point. (b) In an orthographic camera, all rays are parallel. (c) In a pushbroom, all rays lie on a set of parallel planes and pass through a line. (d) In a cross slit camera, all rays pass through two non-coplanar lines. (e) In a pencil camera, all coplanar rays originate from a point on a line and lie on a specific plane through the line. (f) In a twisted orthographic camera, all rays lie on parallel twisted planes and no rays intersect. (g) In a bilinear camera, no two rays are coplanar and no two rays intersect. (h) In an EPI camera, all rays lie on a 2D plane. Sample images for these camera models are shown in Figure 9.*

a closed-form mapping is derived to transform a GLC into a hyperplane representation.

The GLC model satisfies the 2D-manifold, the continuity and the uniqueness criteria for a multi-perspective camera. Specifically, GLCs model all 2-dimensional linear subspaces in the 4D 'ray space' imposed by a two-plane parametrization. Moreover, these 2D subspaces of rays are smoothly varying and form continuous images. Finally, for a specific GLC, any general point in 3D space has a unique mapping to a ray in the GLC. This is because under the $(s, t, u, v)$ parametrization, all rays passing through a 3D point also lie on a 2D hyperplane, and two hyperplanes (one for the point and one for the GLC) generally insect at a unique point in 4D, as shown by Gu *et al.* [GGC97]. Therefore, there is only one ray in each GLC that passes through a given point in a scene.

Yu and McMillan [YM04a] proved that most well-known multi-perspective cameras, such as pinhole, orthographic, pushbroom, cross-slit, linear oblique cameras are GLCs. They further provided a pair of characteristic equations to determine the GLC type. The first characteristic equation computes whether the three rays will simultaneously pass through a slit in 3D space. It is quadratic and has form

$$A \cdot \lambda^2 + B \cdot \lambda + C = 0, \tag{1}$$

where

$$A = \begin{vmatrix} s_1 - u_1 & t_1 - v_1 & 1 \\ s_2 - u_2 & t_2 - v_2 & 1 \\ s_3 - u_3 & t_3 - v_3 & 1 \end{vmatrix} \qquad C = \begin{vmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ u_3 & v_3 & 1 \end{vmatrix} \tag{2}$$

$$B = \begin{vmatrix} s_1 & v_1 & 1 \\ s_2 & v_2 & 1 \\ s_3 & v_3 & 1 \end{vmatrix} - \begin{vmatrix} t_1 & u_1 & 1 \\ t_2 & u_2 & 1 \\ t_3 & u_3 & 1 \end{vmatrix} - 2 \cdot \begin{vmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ u_3 & v_3 & 1 \end{vmatrix}. \tag{3}$$

A second characteristic equation is the edge parallel condition that checks if all three pairs of the corresponding edges of the $u - v$ and $s - t$ triangles formed by the generator rays are parallel.

$$\frac{s_i - s_j}{t_i - t_j} = \frac{u_i - u_j}{v_i - v_j}\, i, \quad j = 1, 2, 3 \quad \text{and} \quad i \neq j \tag{4}$$

The number of solutions to the first characteristic equation and the edge parallel condition can be used to determine the type of the general linear camera for a given set of generator rays. Specific cases are given in Table 1 and illustrated in Figure 6.

The GLC model is capable of describing all perspective (pinhole), orthographic and many multi-perspective (including pushbroom and two-slit) cameras, as well as epipolar plane images. It also includes three new and previously unexplored multi-perspective linear cameras.

**Table 1:** *Characterizing general linear cameras by characteristic equation.*

| Characteristic equation | 2 Solutions | 1 Solution | 0 Solutions | $\infty$ Solutions |
|---|---|---|---|---|
| $A \neq 0$ | XSlit | Pencil/ Pinhole[1] | Bilinear | Ø |
| $A = 0$ | Ø | Pushbroom | Twisted/ Ortho.[1] | EPI |

[1]A GLC satisfying *edge-parallel* condition is pinhole ($A \neq 0$) or orthographic ($A = 0$).

**Twisted orthographic camera.** The characteristic equation of the twisted orthographic camera satisfies $A = 0$, has no solution, and its generators do not satisfy the *edge-parallel* condition. If $r_1$, $r_2$ and $r_3$ are linearly independent, no solution implies $r_3$ will not intersect the bilinear surface $S$. In fact, no two rays intersect in 3D space. In addition, $A = 0$ also implies that all rays are parallel to some plane $\Pi$ in 3D space, therefore the rays on each of these parallel planes must have uniform directions as is shown in Figure 6(f). Therefore, a twisted orthographic camera can be viewed as twisting parallel planes of rays in an orthographic camera along common bilinear sheets.

**Pencil camera.** The characteristic equation of a pencil camera satisfies $A \neq 0$, has one solution and the generators do not satisfy the *edge-parallel* condition. In Figure 6(e), we illustrate a sample pencil camera: rays lie on a pencil of planes that share line $l$. In a pushbroom camera, all rays also pass through a single line. However, pushbroom cameras collect rays along planes transverse to $l$ whereas the planes of a pencil camera contain $l$ (i.e. lie in the pencil of planes through $l$), as is shown in Figures 6(c) and (e).

**Bilinear camera.** By definition, the characteristic equation of a bilinear camera satisfies $A \neq 0$ and the equation has no solution. Therefore, similar to twisted orthographic cameras, no two rays intersect in 3D in a bilinear camera. In addition, since $A \neq 0$, no two rays are parallel either. Therefore, any two rays in a bilinear camera form a non-degenerate bilinear surface, as is shown in Figure 6(g).

### 2.6. Modelling arbitrary multi-perspective cameras

The GLCs can be used to model any multi-perspective camera that describes a continuous set of rays such as the catadioptric mirrors and multi-perspective panoramas. Specifically, let $\Sigma(x, y)$ be a continuous 2D ray manifold implicitly parametrized in $x$ and $y$, that is,

$$\Sigma(x, y) = [s(x, y), t(x, y), u(x, y), v(x, y)]. \tag{5}$$

We can locally approximate the local behaviour of the rays by computing the local tangent plane. The tangent plane can

be computed with two spanning vectors $\vec{d}_1$ and $\vec{d}_2$

$$\vec{d}_1 = [s_x, t_x, u_x, v_x], \vec{d}_2 = [s_y, t_y, u_y, v_y]. \quad (6)$$

Recall that every tangent plane corresponds to a GLC, therefore one can choose three points tangent plane, $\Sigma(x, y)$, $\Sigma(x, y) + \vec{d}_1$ and $\Sigma(x, y) + \vec{d}_2$, and use them to define the GLC. We can then use the GLC characteristic equations to determine the local GLC type. In [YM05a], the local GLC model was used to analyse reflected images seen on arbitrary mirrors. In [YLY07], Yu *et al.* applied the GLC analysis to approximate the local refraction rays for rendering caustics.

### 2.7. Other multi-perspective cameras

Glassner [Gla00] described a camera construction called the 'putty lenses' for collecting rays along the camera path. In his approach, rays are specified by two surfaces with common parametrizations. Recall that the GLC model uses two parametrization planes to specify the rays, therefore, GLCs can be viewed as special putty lenses.

Hall *et al.* introduced a simple but versatile camera model called the Rational Tensor Camera (RTCam) for Non-Photorealistic Rendering [HCS*07]. The RTCam differs from the GLC model in that an RTCam is described by its projection model (a linear tensor) whereas a GLC is constructed by specifying the three generator rays.

Mei *et al.* defined an occlusion camera [MPS05] by specifying a variety of geometric terms (focal length, centre of interest, etc.) on a pair of planes. Their goal is to use a radially distorted multi-perspective camera to capture an omnidirectional view of the target object. The occlusion camera can cover the entire silhouette of an object, and therefore, alleviate disocclusion errors.

## 3. Constructing Multi-Perspective Images

To construct a desirable multi-perspective image, it is a commonplace to combine different multi-perspective cameras in a single camera. Examples include multi-perspective panoramas, Neo-cubist style renderings, and faux-animations from still-life scenes.

### 3.1. Construction by a strip camera

A commonly used technique for creating multi-perspective images is to combine strips from different pinhole cameras. This approach, often called a strip camera, has appeared quite often in Graphics literature. For example, computer generated multi-perspective panoramas, as presented by Wood *et al.* [WFH*97], combined elements of multiple pinhole strips into a single image using a semi-automatic image registration process. They relied on optimization techniques, as well as optical flow and blending transitions between views. The concentric mosaics of [SKS99] and [PBP01] are another type of multi-perspective image that is useful for exploring captured environments.

The multiple centre of projection (MCOP) images of Rademacher [RB98] are another example of unstructured multi-perspective images. They are closely related to images generated by pushbroom cameras, but they are not constrained to follow linear paths. While these images were intended as scene representations, they are also interesting and informative images on their own.

Durand [Dur02] suggests that specifying multi-perspective cameras can also be an interactive process and uses them as an example to distinguish between picture generation and user interaction. Examples of such approaches include the 3D-based interactive rendering systems by Agrawala *et al.* [AZM00] and Hanson and Wernert [HW98].

Roman *et al.* [RL06, RGL04] provide a semi-interactive system that uses a linear camera to combine photographs into panoramas of street scenes. Agrawala *et al.* [AAC*06] proposed to composite large regions of ordinary perspective images. They reduce the degree of user interaction by identifying the dominant plane and then use graph cuts to minimize multi-perspective distortions.

### 3.2. Construction by GLCs

A different multi-perspective image construction method is to use GLCs as primitives. In [YM04b], the problem of multi-perspective rendering is treated as one of specifying and sampling a smooth varying set of rays embedded in a 4D space. They used piecewise planar tessellation of a ray manifold corresponding to a specific collection of GLCs, much like a polygonal model of a curved surface. Specifically, they described an arbitrary multi-perspective image by a triangulation of the image plane along with generator rays attached to each vertex. The continuity of the manifold guarantees that the images produced are coherent. As the tessellation of the image plane increases, this model can approximate arbitrarily smooth 2D manifolds, and hence render arbitrary multi-perspective images.

Since each triangle on the image plane corresponds to a general linear camera, adjacent triangles sharing a common edge represent two GLCs that share two rays. This imposes a constraint on possible pairs of adjacent GLCs. For instance, a pinhole camera cannot share two rays with a different pinhole camera (because rays in two different pinhole cameras pass through two different points). Similarly, a pinhole camera cannot be adjacent to a bilinear camera, because any two rays will intersect in a pinhole while no two rays will intersect in a bilinear camera. In Table 2, we show all possible adjacency relationships between general linear cameras. Triangulations of the image plane into GLCs must

**Table 2:** *Adjacency tables of GLCs.*

| Possible adjacency | P | O | PB | X | PN | T | B |
|---|---|---|---|---|---|---|---|
| Pinhole (P) | N | N | Y | Y | Y | N | N |
| Orthographic (O) | N | N | Y | N | N | N | N |
| Pushbroom (PB) | Y | Y | Y | Y | Y | Y | Y |
| XSlit (X) | Y | N | Y | Y | Y | Y | Y |
| Pencil (PN) | Y | N | Y | Y | Y | Y | Y |
| Twisted Orthographic (T) | N | Y | Y | Y | Y | Y | Y |
| Bilinear (B) | N | N | Y | Y | Y | Y | Y |

satisfy these adjacency constraints to assure $C_0$ continuous images. Furthermore, because any continuous 2D manifold can be locally approximated by tangent planes (i.e. GLCs), the adjacency table shows which types of continuous manifolds, and therefore, multi-perspective images, are possible and which are not. For instance, in the table, no two different pinhole cameras can be adjacent to each other. Thus, there does not exist a multi-perspective image which looks locally like a pinhole camera everywhere. However, there do exist multi-perspective images which look locally like pushbroom or cross-slit images everywhere. In fact, multi-perspective panoramas for cel animations are good examples of these type of multi-perspective images.

While any triangulation of the image plane with generator rays at each vertex describes a multi-perspective rendering, it is not a very intuitive specification. In practice, [YM04b] proposed a design system similar to the automatic layout method described by Wood [WFH*97], but with user guidance. A predefined triangular mesh is placed over the image plane. The user then guides any typical GLC image over the mesh to establish rough spatial relationships. The images can overlap as desired. The mesh then acquires generator rays by extracting rays from the reference images. If more than one image overlaps a vertex various blends of the rays from the reference images can be used, as long as the blend preserves affine combinations. The end result is that corresponding rays are interpolated such that the transition is smooth, as shown in Figure 7.

### 3.3. Applications

Rendering perspectives from multiple viewpoints can be combined in ways other than panoramas. By making subtle changes in viewing direction across the imaging plane, one can depict more of the scene than could be seen from a single point of view. Such images differ from panoramas in that they are intended to be viewed as a whole. Neo-cubism is an example.

Many of the works of Picasso are examples of such multi-perspective images. Figure 7 compares one of Picasso's paintings with an image synthesized using the GLC framework [YM04b]. Starting from a simple layout, it achieves similar multi-perspective effects. In Figures 8(a)–(c), we show the multi-perspective view of a teapot by overlaying image pieces from significantly different perspectives. Figures 8(c) shows a close to 360° view of the teapot, reminiscent of an MCOP image [RB98].


**Figure 7:** *(a) Nusch eluard by Pablo Picasso and (b) A multi-perspective image synthesized using the GLC framework [YM04b].*

It is also possible to use multi-perspective rendering to create fake or faux-animations from still-life scenes. This is particularly useful for animating image based models. In Figures 8(d)–(f), we show three frames from a synthesized animation, each of which corresponds to a multi-perspective image rendered from a 3D light field. Zomet [ZFPW03] used a similar approach by using a single cross-slit camera to achieve rotation effects.

## 4. Rendering Multi-Perspective Images

Once a multi-perspective image is constructed, it can be rendered directly by ray tracing a synthetic scene, stitching the video frames or cutting through pre-captured light fields.

Agrawala *et al.* [AAC*06] combine a series of photographs taken with a hand-held camera from multiple viewpoints along the scene. Their system uses Markov Random Field optimization to construct a composite from arbitrarily shaped regions of the source images according to various properties that the panorama is expected to exhibit.

Alternatively, a multi-perspective image can be rendered using ray tracing. In the GLC framework, each pixel is associated with a unique ray that can be directly traced in a scene. In [DY08], a GLC Ray-Tracer (GLC-RT) was developed based on the legacy Pov-Ray [POV-Ray] framework. The GLC-RT supports rendering both single GLC models and arbitrary multi-perspective images constructed from piecewise GLCs. Figure 1(d) shows a sample cross-slit image rendered using GLC-RT.

Despite recent advances in real-time ray tracing [WPS*03], applying ray tracing for interactive

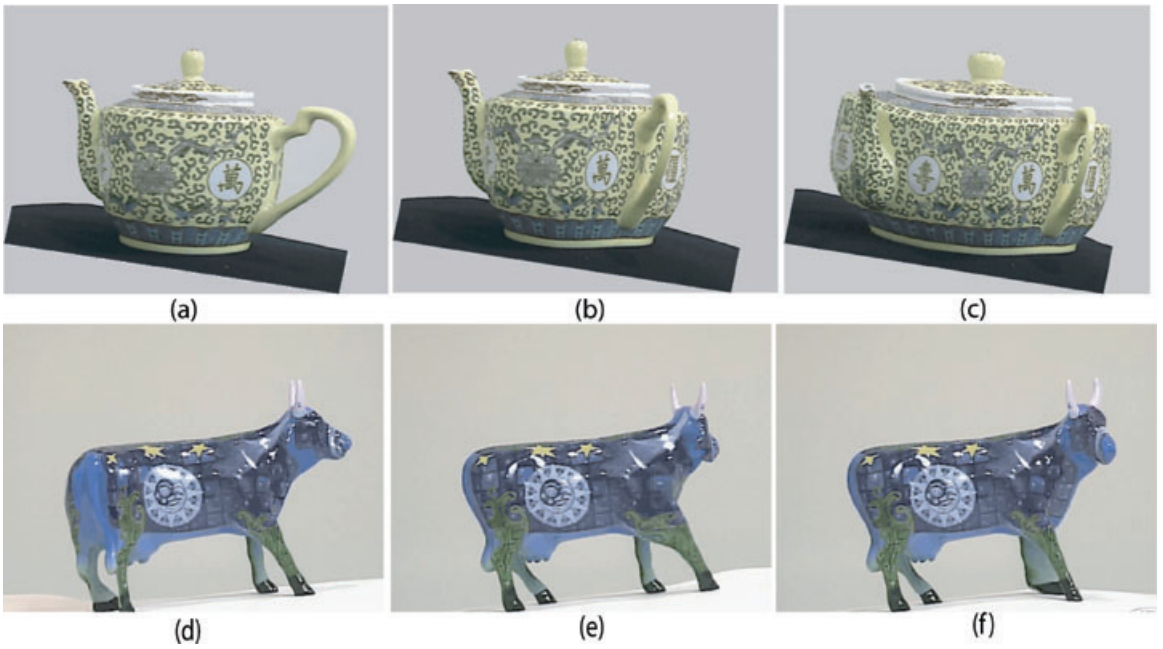


**Figure 8:** *(a) A perspective view of a teapot. (b) A synthesized multi-perspective image that fuses different perspective views of the handle and beak. (c) An omni-perspective image that shows a 360° view of the teapot. (d)–(e) extracted images from a faux-animation generated by [YM04b]. The source images were acquired by rotating a ceramic figure on a turntable. Multi-perspective renderings were used to turn the head and hind quarters of the figure in a fake image-based animation.*

multi-perspective rendering is still a challenging task, especially for complex scenes. Agrawala *et al.* [AZM00] proposed to rasterize the scene from a 'master camera' and discussed the issue of how to depth-order and composite scene components viewed from different perspectives. In the GLC-based panoramas [YM04b], each GLC at the corresponding triangle on the image plane is rendered by cutting through the pre-captured 3D/4D light fields to achieve real-time performance. However, since collecting all rays present in a scene is impractical or impossible for most light fields, aliasing artefacts due to interpolation may appear in the rendered GLC image pieces where the light field is undersampled.

Recently, real-time multi-perspective rendering techniques have been developed based on rasterizing graphics hardware. These include techniques for supporting multiple centres of projection in VR applications [KKYK01, SSP04], rendering general curved reflections or refractions [Wym05], and curved polygon rasterization [HWSG06]. The work by Hou *et al.* [HWSG06] decomposes an arbitrary multi-perspective image into piecewise-linear multi-perspective primitives similar to the GLC multi-perspective rendering approach. They then render each primitive camera by implementing a non-linear beam-tracing using a pair of vertex and fragment programs on programmable graphics hardware.

### 4.1. Distortion, projection and collineation

Although multi-perspective rendering provides potentially advantageous images for understanding the structure of observed scenes, they also exhibit multi-perspective distortions. To analyse these distortions, it is crucial to first derive the closed-form projections and collineations in multi-perspective cameras. Unlike perspective cameras whose projection and collineation can be described using the classic $3 \times 4$ camera matrix [HZ04], multi-perspective cameras follow more complicated projection and collineation models.

Gupta and Hartley [GH97] investigated theoretical insights such as the projection and fundamental matrices as well as the epipolar geometry of linear pushbroom cameras. They showed, although the epipolar constraints are totally different from that of a perspective camera, that a matrix analogous to the fundamental matrix of perspective cameras exists for pushbroom cameras. Zomet *et al.* [ZFPW03] have shown that the projection and collineation of a cross-slit camera is no longer a $3 \times 4$ projection matrix but a $3 \times 4 \times 4$ quadratic tensor.

Yu and McMillan [YM05b] used a plane-sweeping algorithm to derive a closed-form solution to projecting 3D points in a scene to rays in a GLC. They concluded that for projection, singularities can only happen in cross-slits, pushbroom, pencil, pinhole and EPI cameras. When the points lie

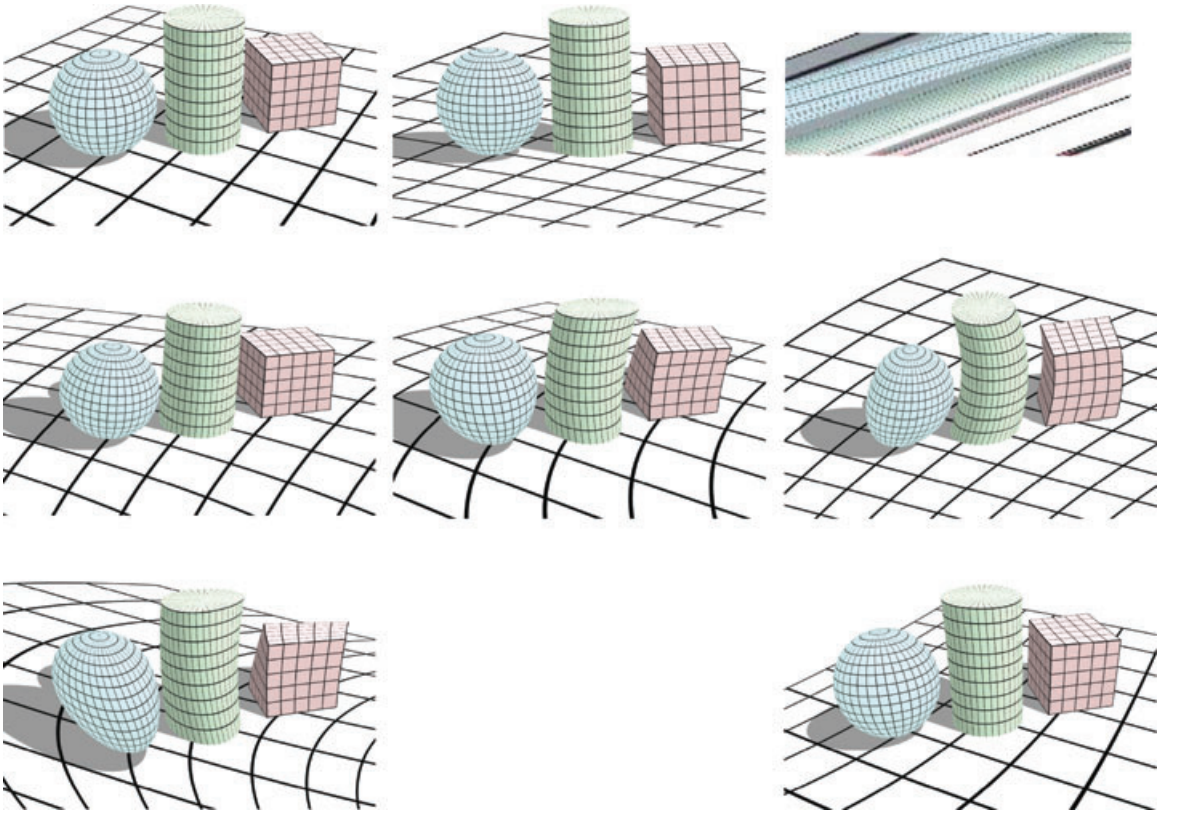

**Figure 9:** *Multi-perspective images rendered using Pov-Ray. From left to right, top row: a pinhole, an orthographic and an EPI; middle row: a pushbroom, a pencil and a twisted orthographic; bottom row: a bilinear and a cross-slit.*

precisely on the slits, duplicated images will occur, because multiple rays in the GLC will pass through these points. They have also derived a closed-form solution to projecting an arbitrary 3D line into a GLC. They proved that if a line is parallel to the 2PP, its image will still be a line. If not, its image will be a conic, as shown in Figure 9.

Besides singularities and curving of lines, additional image distortions such as apparent stretching and shrinking, and duplicated projections of a single point [SGN01, YM05a] can appear in a multi-perspective image. Zomet *et al.* [ZFPW03] have shown that, in the case of cross-slit cameras, these distortions can be analysed in terms of the spatial relationship between the slits and the image plane. Specifically, if one slit is much closer to the image plane than the other, the orientation of the image will be dominated by the corresponding slit [ZFPW03]. Yu and McMillan have shown that the distance between the two slits determines the aspect ratio distortions [YM05a]. For example, when the two slits intersect, the cross-slit transforms into a pinhole camera with small aspect ratio distortion. When one of the slits goes to infinity, the cross-slit transforms into a pushbroom camera with large aspect ratio distortions.

Related to projection, a multi-perspective collineation describes the transformation between the images due to changes in sampling and image plane selection. For GLCs, the collineations can be characterized by a quartic (fourth-order) rational function [YM05b]. The same camera may capture dramatically different images under different collineations, as shown in Figure 10. Yu and McMillan [YM04b] referred to the distortions caused by collineations as 'perspective distortion', whereas distortions introduced by smooth changes in the COP as 'projective distortion'. Projective distortions are inherent to the geometry structure of the camera and are usually desirable to render specific multi-perspective effects such as the Neo-Cubism style by Picasso.

### 4.2. Reducing multi-perspective distortions

A major component in multi-perspective rendering is to reduce perspective distortions to smooth the transitions of projective distortion. In computer vision, image-warping has been commonly used to reduce perspective distortions. Image-warping computes an explicit pixel-to-pixel mapping to warp the original image onto a nearly perspective

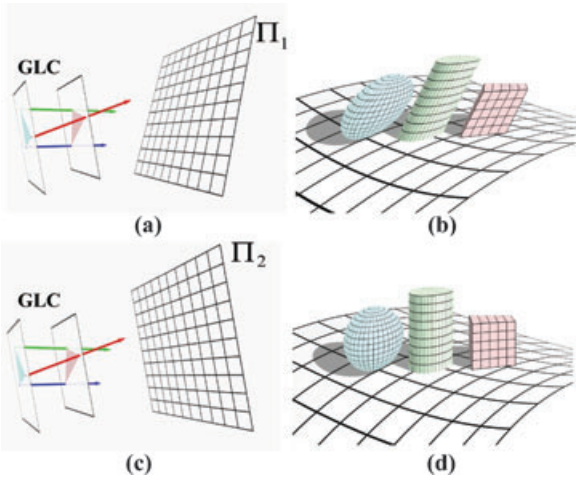


**Figure 10:** *The image (d) of a cross-slit GLC under collineation (c) appears much less distorted than the image (b) of the same camera under collineation (a).*

image. For cameras that roughly maintain a single viewpoint [Nay97], simple parametric functions are sufficient to eliminate perspective, radial and tangential distortions [Che95, DK00]. However, for complex imaging systems, especially those exhibiting severe caustic distortions [SGN01], the warping function is difficult to model and may not have a closed-form solution.

IBR algorithms have also been proposed to reduce perspective distortions [SKS99, GGSC96]. There, the focus has been to estimate the scene structure from a single or multiple images. Zorin and Barr [ZB95] studied the use of multi-perspective and other geometric distortions to improve perceptual qualities of images. Swaminathan and Nayar [SGN03] have shown that simple geometry proxies, such as the plane, sphere and cylinder, are often sufficient to reduce caustic distortions in catadioptric images, provided that a prior on scene structure is known. The geometry proxies have also been used to construct close-to-perspective panoramas [AAC*06, RL06], where the scene geometry is approximated to align the image plane.

Ding and Yu [DY07b] recently proposed a third approach based on multi-perspective collineations. They have developed an interactive system that allows users to select feature rays from the multi-perspective image and position them at the desired pixels. They then compute the optimal collineation to match the projections of the feature rays with the corresponding pixels. Their method can robustly correct highly complex distortions without acquiring the scene geometry, as shown in Figure 11.

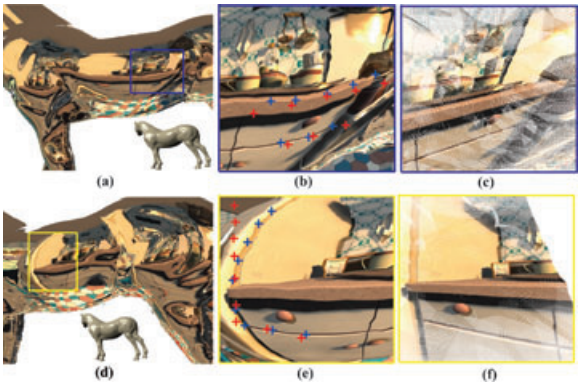


**Figure 11:** *Correcting complex distortions on a horse model. We render a reflective horse model under two different poses (a) and (d) and then select regions (b) and (e). (c) and (f) are the resulting images by matching the selected features (blue) and target pixels (red) in (b) and (e) using collineations [DY07a].*

## 5. Multi-Perspective Geometry

In this section, we focus on analysing the camera geometry of multi-perspective cameras, with an emphasis on how to use these cameras in structure-from-motion. Key concepts of structure-from-motion are camera calibration, epipolar geometry, motion and pose estimation and 3D reconstruction. These are well understood and formulated for perspective and other central cameras. However classical results such as the fundamental matrix representing stereo or epipolar geometry, are not directly applicable for general multi-perspective cameras, not even for general central ones. We first describe an abstract unified camera model that allows to handle any multi-perspective system and explain how calibration and other structure-from-motion problems can be tackled using that model.

We then consider the special cases where multi-perspective camera pairs have a standard stereo geometry, i.e. corresponding points lie on the same scan line in both images; this simplifies image matching and 3D reconstruction. Finally, we present recent work on multi-perspective image pairs that are close to standard stereo geometry, which thus allows to handle larger classes of cameras.

### 5.1. Camera model

Classical camera models provide an analytical 'projection function', allowing to compute the image point where a 3D point is projected to. Usual models depend on up to ten or twenty intrinsic camera parameters, such as the focal length for perspective cameras. Most of these models are applicable to a restricted class of camera technologies each, for example perspective cameras, possibly with radial

distortion, catadioptric cameras, fish-eyes, pushbroom sensors, etc. For most models there does not seem to exist an analytical epipolar geometry, i.e. an equivalent object to the perspective fundamental matrix, that constrains point matches between two images via a bilinear constraint. Slight exceptions are a special case of catadioptric cameras (central catadioptric cameras with a parabolic mirror), for which a fundamental matrix of size $6 \times 6$ exists [Stu02], and linear pushbroom cameras which have a $4 \times 4$ fundamental matrix [GH97].

As will be shown below, these difficulties vanish if instead of reasoning in terms of matching image points, we consider the matching of projection rays. To this end, let us consider a generic camera model, as follows. A camera is modelled by the set of its projection rays, i.e. the lines in 3D such that light incoming along any of them, ends up on the image sensors. We make the assumption that to each point in the image, a single projection ray is associated. This is obviously not the case with real cameras (due to, e.g. blur), but is an assumption made by most geometrical camera models. Most importantly, we do not impose the existence of an analytical relation between the coordinates of a projection ray and the coordinates of the associated image point. Further, for multi-perspective cameras, the projection rays do not have a single common intersection point, as opposed to central cameras.

Calibration of this camera model amounts to determining, for each image point (rather, a finite sample of image points, e.g. all pixel centres), the associated projection ray. In the following, we briefly describe a calibration approach and then how to perform structure-from-motion for cameras calibrated under that model.

### 5.2. Calibration

We briefly review a generic calibration approach developed in [SR04], an extension of [CLSC92, GTK88, GN05], to calibrate different camera systems, especially multi-perspective ones. As mentioned, calibration consists in determining, for every pixel, the projection ray associated with it. In [GN05], this is done as follows: two images of a calibration object with known structure are taken. We suppose that for every pixel, we can determine the point on the calibration object, that is seen by that pixel. For each pixel in the image, we thus obtain two 3D points. Their coordinates are usually only known in a coordinate frame attached to the calibration object; however, if one knows the motion between the two object positions, one can align the coordinate frames. Then, every pixel's projection ray can be computed by simply joining the two observed 3D points.

In [SR04] a more general approach is proposed, that does not require knowledge of the calibration object's displacement. In that case, three images need to be taken at least. The fact that all 3D points observed by the same pixel in different views, are on a line in 3D (the pixel's projection ray), gives a constraint that allows to recover both the motion and the camera's calibration. The constraint is formulated via a set of trifocal tensors, that can be estimated linearly, and from which motion, and then calibration, can be extracted. In [SR04], this approach is first formulated for the use of 3D calibration objects, and for the general imaging model, i.e. for multi-perspective cameras. We also propose variants of the approach, that may be important in practice: first, due to the usefulness of planar calibration patterns, we specialized the approach appropriately. Second, we propose a variant that works specifically for central cameras (pinhole, central catadioptric or any other central camera). More details are given in [SR03].

An important part of this calibration procedure is the matching between images and calibration grids, especially since matches for all pixels are required. A practical way for doing this is to use a structured-light type approach, as in [TS05]: the calibration grid is replaced by a flat computer screen. For each camera position, a series of black-and-white patterns is displayed such that each screen pixel has a unique sequence of blacks and whites. Matching then basically amounts to determining, for each pixel in the image, its sequence of blacks and whites, giving directly the corresponding point in the 'calibration screen'.

### 5.3. Structure-from-motion

Once cameras are calibrated, structure-from-motion can be performed for any type of multi-perspective settings using the same algorithms. We briefly describe how three of the basic structure-from-motion problems – pose and motion estimation as well as 3D reconstruction – can be solved.

#### 5.3.1. *Pose estimation*

A first example is pose estimation, i.e. the problem of computing the relative position and orientation between an object of *known* structure, and a calibrated camera. A literature review on algorithms for perspective cameras is given in [HLON94]. Here, we briefly show how the minimal case can be solved for general cameras [CC04, Nis04a, RLS06]. For perspective cameras, pose can be estimated, up to a finite number of solutions, from three point correspondences (3D-2D) already. The same holds for general cameras. Consider three image points and the associated projection rays, computed using the calibration information. We parametrize generic points on the rays as follows: $\mathbf{A}_i + \lambda_i \mathbf{B}_i$. If we are able to estimate the position of the three object points, i.e. the $\lambda_i$, then the pose problem is solved: the position of three points in general position define the complete position and orientation of an object.

We know the structure of the observed object, meaning that we know the mutual distances $d_{ij}$ between the 3D points. We

can thus write equations on the unknowns $\lambda_i$

$$\|\mathbf{A}_i + \lambda_i \mathbf{B}_i - \mathbf{A}_j - \lambda_j \mathbf{B}_j\|^2 = d_{ij}^2.$$

This gives a total of three quadratic equations in three unknowns. Many methods exist for solving this problem, for example, symbolic computation packages such as MAPLE allow to compute a resultant polynomial of degree 8 in a single unknown, that can be numerically solved using any root finding method.

Like for perspective cameras, there are up to eight theoretical solutions. For perspective cameras, at least four of them can be eliminated because they would correspond to points lying behind the camera [HLON94]. As for general cameras, determining the maximum number of feasible solutions requires further investigation. In any case, a unique solution can be obtained using one or two additional points [HLON94].

#### 5.3.2. *Motion estimation and epipolar geometry*

We describe how to estimate ego-motion, or, more generally, relative position and orientation of two calibrated general cameras. This is done via a generalization of the classical motion estimation problem for perspective cameras and its associated centrepiece, the essential matrix [Lon81]. We briefly summarize how the classical problem is usually solved [HZ04]. Let $\mathsf{R}$ be the rotation matrix and $\mathbf{t}$ the translation vector describing the motion. The essential matrix is defined as $\mathsf{E} = -[\mathbf{t}]_\times \mathsf{R}$. It can be estimated using point correspondences $(\mathbf{x}_1, \mathbf{x}_2)$ across two views, using the epipolar constraint $\mathbf{x}_2^\mathsf{T} \mathsf{E} \mathbf{x}_1 = 0$. This can be done linearly using 8 correspondences or more. In the minimal case of 5 correspondences, an efficient non-linear algorithm, which gives exactly the theoretical maximum of 10 feasible solutions, was only recently introduced [Nis04b]. Once the essential matrix is estimated, the motion parameters $\mathsf{R}$ and $\mathbf{t}$ can be extracted relatively straightforwardly [Nis04b].

In the case of our general imaging model, motion estimation is performed similarly, using pixel correspondences $(\mathbf{x}_1, \mathbf{x}_2)$. Using the calibration information, the associated projection rays can be computed. Let them be represented by their Plücker coordinates, i.e. 6-vectors $\mathbf{L}_1$ and $\mathbf{L}_2$. The epipolar constraint extends naturally to rays, and manifests itself by a $6 \times 6$ essential matrix, introduced by Pless [Ple03]

$$E = \begin{bmatrix} -[\mathbf{t}]_\times \mathsf{R} & \mathsf{R} \\ \mathsf{R} & 0 \end{bmatrix}. \tag{7}$$

The epipolar constraint then writes: $\mathbf{L}_2^\mathsf{T} \mathsf{E} \mathbf{L}_1 = 0$ [Ple03]. Once $\mathsf{E}$ is estimated, motion can again be extracted straightforwardly (e.g. $\mathsf{R}$ can simply be read off $\mathsf{E}$). Linear estimation of $\mathsf{E}$ requires 17 correspondences.

There is an important difference between motion estimation for central and multi-perspective cameras: with central cameras, the translation component can only be recovered up to scale. Multi-perspective ones however, allow to determine even the translation's scale (although this is likely to be inaccurate in practice). This is because a single calibrated multi-perspective camera already carries scale information, via the distance between mutually oblique projection rays. One consequence is that the theoretical minimum number of required correspondences is six instead of five.

#### 5.3.3. *3D point triangulation*

We now describe an algorithm for 3D reconstruction from two or more calibrated images with known relative position. Let $\mathbf{C} = (X, Y, Z)^\mathsf{T}$ be a 3D point that is to be reconstructed, based on its projections in $n$ images. Using calibration information, we can compute the $n$ associated projection rays. Here, we represent the $i$th ray using a starting point $\mathbf{A}_i$ and the direction, represented by a unit vector $\mathbf{B}_i$. We apply the midpoint method [HS97, Ple03], i.e. determine $\mathbf{C}$ that is closest in average to the $n$ rays. Let us represent generic points on rays using position parameters $\lambda_i$. Then, $\mathbf{C}$ is determined by minimizing the following expression over $X, Y, Z$ and the $\lambda_i$: $\sum_{i=1}^{n} \|\mathbf{A}_i + \lambda_i \mathbf{B}_i - \mathbf{C}\|^2$. This is a linear least squares problem, which can be solved, for example, via the Pseudo-Inverse. The solution can actually be obtained in closed-form [RLS06].

#### 5.3.4. *Multi-view geometry*

One concern of multi-view geometry is to study constraints on the positions of matching points in two or more images. With the generic camera model we consider in this section, this translates into matching constraints on projection rays in 3D. In Section 5.3.2, we already described the associated epipolar constraint, i.e. the matching constraint for a pair of images, and its algebraic representation, the essential matrix. Like for perspective cameras, this can be extended to the case of three or four images. Matching constraints are then represented by so-called matching tensors, which constrain corresponding projection rays via multi-linear equations. This issue is too technical for this review; the interested reader is referred to [Stu05].

### 5.4. Multi-perspective stereo

In the previous section, we considered structure-from-motion concepts in terms of projection rays of cameras, making abstraction of the actual images. Let us go back now to images and pixels and re-consider the epipolar geometry between two images. The central question of epipolar geometry is: given a point in one image, what is the locus of the corresponding point in the other image, and how can one compute it? A general answer goes as follows. Consider the projection

ray associated with the given point in the first image and determine all projection rays of the second camera that intersect it. Then, the image points associated with these rays, form the locus of the corresponding point in the second image. For perspective cameras, the locus is known to be a line (actually, a line segment), but for more general camera geometries, especially multi-perspective ones, the locus is usually a more complicated curve and may even be simply a set of isolated points. For example, with central catadioptric cameras with parabolic mirrors, the locus is a conic.

For efficient automatic image matching, it is favourable to have an epipolar geometry where these loci are lines, as with perspective images. This is one of the motivations of the works on multi-perspective stereo theory by Seitz and Pajdla [Sei01, Paj02b]. They provide a complete classification of all possible multi-perspective image pairs in standard stereo configuration, i.e. where corresponding points lie on the same scan line in both images. Their work suggests that only three varieties of epipolar geometry exist: planes, hyperboloids and hyperbolic-paraboloids, all corresponding to double ruled surfaces. The notion of epipolar geometry is thus generalized to apply to multi-perspective images and a unified representation is used to model all classes of stereo views, based on the concept of a quadric view. The multi-perspective stereo theory can be applied to derive new types of stereo image representations with unusual and useful properties.

### 5.5. Epsilon-stereo pairs

Finally, Ding and Yu [DY07a] recently proposed a method for fusing multi-perspective camera pairs that do not have a standard stereo geometry in the sense explained in the previous paragraph. Such pairs may consist of two different cross-slit cameras, a cross-slit and a pushbroom, or two arbitrary multi-perspective cameras. They introduced a notion of *epsilon-stereo pairs*. An epsilon stereo pair consists of two images with a slight vertical parallax. They have shown that many multi-perspective camera pairs which do not satisfy the stereo constraint can still form epsilon stereo pairs. They have also introduced a new ray-space warping algorithm to minimize stereo inconsistencies in an epsilon pair using multi-perspective collineations. This makes epsilon stereo model a promising tool for synthesizing close-to-stereo fusions from many non-stereo pairs.

## 6. Future Direction

Existing results in multi-perspective modelling, rendering and imaging suggest that we might extend computer graphics with new kinds of interactions. In particular, the capability of capturing and rendering what is seen from several viewpoints in a single image may enable many new approaches to solving challenging vision and graphics problems.

### 6.1. Multi-perspective rendering hardware

Most multi-perspective rendering results presented in this report are either ray-traced or synthesized from precaptured light fields or videos. Ray tracing is usually used as an off-line tool for generating high quality multi-perspective images. Light fields can be directly used to render multi-perspective images at an interactive rate. However, the rendering quality relies heavily on the sampling density and the image resolution of the light field.

Ideally, graphics hardware can be used for interactive multi-perspective rendering. The key component in the polygonal graphics pipeline is projecting and rasterizing triangles in the camera. For a multi-perspective camera (e.g. a GLC), the projection of any 3D point (triangle vertices) to the camera may have a closed-form solution and can be easily mapped onto the vertex shader [HWSG06]. However, rasterizing the triangle from the projected vertices in a multi-perspective camera is a challenging problem. For example, a line segment in general projects to a conic in a GLC. Therefore, the rasterizer needs to non-linearly interpolate between the two projected vertices, which cannot be easily achieved on classical polygon-based graphics hardware.

One possible solution is to subdivide scene geometry into smaller triangles so that their images on the multi-perspective camera can also be approximated as triangles [AL07]. However, it is unclear how to control the subdivision level and the computational overhead scales with the number of vertices and triangles in the model. Hou *et al.* [HWSG06] combined the multi-perspective projection and non-linear beam-tracing on the GPUs to approximate a multi-perspective image. However, their method needs to compute the bounding region of rays to reduce the overhead in beam-tracing.

In the future, special graphics hardware may be developed to directly render multi-perspective images. Specifically, it is desirable to make the rasterization unit also programmable to support multi-perspective cameras. Notice that any multi-perspective camera can be locally approximated by the GLCs. Therefore, if the rasterization unit can support the GLC projection and collineation (i.e. fourth-order rational functions), it may be used to render arbitrary multi-perspective effects.

A possible extension to the method by Hou *et al.* [HWSG06] is to develop a multi-perspective culling algorithm similar to the ones developed for a perspective camera. The difficulty lies in that the viewing frustum of a multi-perspective does not form a convex polygon (e.g. it is a bilinear volume for the bilinear GLC). Efficient algorithms may be developed to approximate the multi-perspective frustums as convex frustums.

An interactive multi-perspective renderer will benefit many applications such as interactive design of multi-perspective panoramas and image-based animations. It can

also be used to render accurate reflections and refractions. For instance, one can approximate local reflections and refractions in terms of piecewise GLCs and then render individual GLCs using the multi-perspective renderer.

### 6.2. Multi-perspective acquisition

Multi-perspective cameras could also be used to design new acquisition devices for many IBR and computational photography applications. For example, it will be interesting to design specially curved mirrors to efficiently capture the light fields. The classical pinhole camera arrays is one way to sample the ray space: each pinhole camera corresponds to a 2D hyperplane in the 4D ray space. Alternatively, one can use special-shaped mirrors to more efficiently sample the ray space via a different set of 2D subspaces (e.g. using the GLCs). In addition, customized multi-perspective IBR techniques can be developed to trade-off the image resolution for the spatial resolution. This will effectively reduce the aliasing artefacts in light field rendering due to spatial undersampling.

The multi-perspective acquisition system can also be used to capture appearance data. The spherical mirror array system proposed by Unger *et al.* [UWH*03] has relatively large multi-perspective distortion due to the geometry of the mirrors. Based on the multi-perspective distortion analysis, we can design special-shaped mirror arrays that produce less distortion while maintaining a wide field-of-view. It is also possible to decompose the mirror surfaces into piecewise GLCs and use the multi-perspective calibration techniques to calibrate the entire mirror array.

Furthermore, we can generate a multi-perspective light source by replacing the viewing camera in a catadioptric camera with a point light source. Many image-based relighting approaches are restricted by the geometric constraints of the light source, and by designing a different type of lighting condition, we can improve the way for effectively measuring and sampling the radiance off the surface and, therefore, benefit applications such as measuring the surface BRDF.

A multi-perspective light source will cast special-shaped shadows. In particular, the shadow of a 3D line segment can be a curve under a multi-perspective light source. This may lead to the development of new shape-from-shadow algorithms which determine the depth of the object by analysing the shape of the shadow at the silhouettes.

### 6.3. Computational and differential ray geometry

Finally, we can develop a new theoretical framework based on multi-perspective geometry to characterize and catalogue the structures of ray space. For example, it can be highly useful to model the algebraic ray subspaces (e.g. ray simplices) and analyse how ray geometries are related to specific multi-perspective cameras and affect visual phenomena such as shading and distortion.

In addition, the high-dimensional ray space such as the light fields are typically too large for in-core processing. Therefore, it will be beneficial to develop a ray-space triangulation algorithm using Constrained Delaunay Triangulation and geometric transformation operators to effectively approximate, interpolate and simplify the ray data. Notice that since a multi-perspective camera correspond to a 2D manifold in the 4D ray space, each geometric operator has a unique effect to transform one multi-perspective camera to another.

It will also be useful to relate multi-perspective reflection and refraction distortions to the geometric attributes of the reflector/refractor surface. One possible approach is to model the local reflections or refractions in terms of specific multi-perspective cameras and then derive the surface differential geometry from the camera intrinsics. If we further use the multi-perspective distortions to determine the multi-perspective camera type, we can develop new shape-from-distortion methods for recovering the geometry of specular surfaces, a challenging problem in computer vision for decades [IKM*08].

### References

[AAC*06] Agarwala A., Agrawala M., Cohen M., Salesin D., Szeliski R.: Photographing long scenes with multi-viewpoint panoramas. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)*, pp. 853–861.

[AL07] Adams A., Levoy M.: General linear cameras with finite aperture. In *Proc. Eurographics Symposium on Rendering* (2007).

[AZM00] Agrawala M., Zorin D., Munzner T.: Artistic multiprojection rendering. In *Proc. Eurographics Rendering Workshop* (2000), pp. 125–136.

[BBM87] Bolles R. C., Baker H. H., Marimont D. H.: Epipolar-plane image analysis: an approach to determining structure from motion. *International Journal of Computer Vision*, *1*, 1 (1987), 7–56.

[BBM*01] Buehler C., Bosse M., McMillan L., Gortler S., Cohen M.: Unstructured Lumigraph rendering. In *Proc. of ACM SIGGRAPH* (2001), pp. 425–432.

[BN99] Baker S., Nayar S. K.: A Theory of single-viewpoint catadioptric image formation. *International Journal on Computer Vision*, *35*, 2 (1999), 1–22.

[CC04] Chen C.-S., Chang W.-Y.: On pose recovery for generalized visual sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 7 (2004), 848–861.

[Che95] CHEN S. E.: QuickTime VR—an image-based approach to virtual environment navigation. *Computer Graphics*, *29* (1995), 29–38.

[CLSC92] CHAMPLEBOUX G., LAVALLÉE S., SAUTOT P., CINQUIN P.: Accurate calibration of cameras and range imaging sensors: the NPBS method. In *Proc. International Conference on Robotics and Automation* (1992), pp. 1552–1558.

[CS97] CHAHL J., SRINIVASAN M.: Reflective surfaces for panaromic imaging. *Applied Optics*, *36*, 31 (1997), 8275–8285.

[CSH04] CABRAL E., SOUZA J., HUNOID C.: Omnidirectional stereo vision with a hyperbolic double lobed mirror. In *Proc. International Conference on Pattern Recognition* (2004), pp. 1–4.

[Dav87] DAVIDHAZY A.: Peripheral photography: phooting full circle. *Industrial Photography*, *36* (1987), 28–31.

[DK00] DERRIEN S., KONOLIGE K.: Approximating a single viewpoint in panoramic imaging devices. In *Proc. International Conference on Robotics and Automation* (2000), pp. 3932–3939.

[DY07a] DING Y., YU J.: Epsilon stereo pairs. In *Proc. British Machine Vision Conference I* (2007), pp. 162–171.

[DY07b] DING Y., YU J.: Multiperspective distortion correction using collineations. In *Proc. Asian Conference on Computer Vision 1* (2007), pp. 95–105.

[DY08] DING Y., YU J.: GLC-RT: A Multiperspective ray tracer based on general linear cameras. Technical Report, University of Delaware, 2008.

[Dur02] DURAND F.: An invitation to discuss computer depiction. In *Proc. Symposium on Non-Photorealistic Animation and Rendering (NPAR)* (2002), pp. 111–124.

[Dvo16] Nick Dvoracek's kind hosting of an article from Scientific American, The Slit Camera, February 15, 1916. Report on the slit camera and how images are changed by using a slit instead of a pinhole. http://idea.uwosh.edu/nick/SciAm.pdf.

[FB05] FIALA M., BASU A.: Panoramic stereo reconstruction using non-SVP optics. *Computer Vision and Image Understanding*, *98*, 3 (2005), 363–397.

[GGC97] GU X., GORTLER S. J., COHEN M. F.: Polyhedral geometry and the two-plane parameterization. In *Proc. Eurographics Rendering Workshop* (1997), pp. 1–12.

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

[GH97] GUPTA R., HARTLEY R. I.: Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 9 (1997), 963–975.

[Gla00] GLASSNER A.: Cubism and cameras: free-form optics for computer graphics. Technical Report MSR-TR-2000-05, Microsoft Research, 2000.

[GN02] GLUCKMAN J. M., NAYAR S. K.: Rectified catadioptric stereo sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 2 (2002), 224–236.

[GN05] GROSSBERG M. D., NAYAR S. K.: The raxel imaging model and ray-based calibration. *International Journal of Computer Vision*, *61*, 2 (2005), 119–137.

[GTK88] GREMBAN K. D., THORPE C. E., KANADE T.: Geometric camera calibration using systems of linear equations. In *Proc. International Conference on Robotics and Automation* (1988), pp. 562–567.

[GZC*06] GEORGIEV T., ZHENG K. C., CURLESS B., SALESIN D., NAYAR S. K., INTWALA C.: Spatio-angular resolution tradeoffs in integral photography. In *Proc. Eurographics Symposium on Rendering* (2006), pp. 263–272.

[HB00] HICKS R. A., BAJCSY R.: catadioptric sensors that approximate wide-angle perspective projections. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2001), pp. 1545–1551.

[HCS*07] HALL P. M., COLLOMOSSE J. P., SONG Y.-Z., SHEN P., LI C.: RTcams: a new perspective on nonphotorealistic rendering from photographs. *IEEE Transactions on Visualization and Computer Graphics*, *13*, 5 (2007), 966–979.

[HLON94] HARALICK R. M., LEE C. N., OTTENBERG K., NOLLE M.: Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, *13*, 3 (1994), 331–356.

[HS97] HARTLEY R. I., STURM P.: Triangulation. *Computer Vision and Image Understanding*, *68*, 2 (1997), 146–157.

[HW98] HANSON A. J., WERNERT E. A.: Image-based rendering with occlusions via cubist images. In *Proc. IEEE Visualization'98* (1998), pp. 327–334.

[HWSG06] HOU X., WEI L.-Y., SHUM H.-Y., GUO B.: Real-time multi-perspective rendering on graphics hardware. In *Proc. EUROGRAPHICS Symposium on Rendering* (2006), pp. 93–102.

[HZ04] HARTLEY R. I., ZISSERMAN A.: *Multiple View Geometry in Computer Vision* (2nd edition). Cambridge University Press, 2004.

[IKM*08] IHRKE I., KUTULAKOS K., MAGNOR M., HEIDRICH W., LENSCH H.: State-of-art Report (STAR): Transparent and Reflective Scene Reconstruction. In *Proc. Eurographics* (2008).

[JKK05] JANG G., KIM S., KWEON I.: Single camera catadioptric stereo system. In *Proc. Workshop on Omnidirectional Vision, Camera Networks and Nonclassical cameras* (2005).

[Kan00] KANG S. B.: Catadioptric Self-calibration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition I* (2000), pp. 201–207.

[KKYK01] KITAMURA Y., KONISHI T., YAMAMOTO S., KISHINO F.: Interactive stereoscopic display for three or more users. In *Proc. SIGGRAPH* (2001), pp. 231–240.

[KN06] KUTHIRUMMAL S., NAYAR, S. K.: Multiview radial catadioptric imaging for scene capture. *ACM Transactions on Graphics* (Proceedings of SIGGRAPH 2006), pp. 916–923.

[LCWT06] LANMAN D., CRISPELL D., WACHS M., TAUBIN G.: Spherical catadioptric array: construction, geometry, and calibration. In *Proc. International Symposium on 3D Data Processing, Visualization and Transmission* (2006).

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

[Loc81] LOCHER J. L. (eds): *M.C. Escher: His Life and Complete Graphic Work*. Amsterdam, 1981.

[Lon81] LONGUET-HIGGINS H. C.: A computer program for reconstructing a scene from two projections. *Nature*, *293* (1981), 133–135.

[MB05] MCMILLAN L., BISHOP G.: Plenoptic modeling: an image-based rendering system. *Computer Graphics*, *29* (1995), 39–46.

[MPS05] MEI C., POPESCU V., SACKS E.: The occlusion camera. *Computer Graphics Forum*, *24*, 3 (2005), 335–342.

[MSEY05] MOUADDIB E., SAGAWA R., ECHIGO T., YAGI Y.: Stereo vision with a single camera and multiple mirrors. In *Proc. IEEE International Conference on Robotics and Automation* (2006), pp. 812–817.

[Nay97] NAYAR S. K.: Catadioptric omnidirectional cameras. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (1997), pp. 482–488.

[New64] NEWHALL B.: *The History of Photography, from 1839 to the Present Day*. The Museum of Modern Art (1964), pp. 162.

[Nis04a] NISTÉR D.: A minimal solution to the generalized 3-point pose problem. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition 1* (2004), pp. 560–567.

[Nis04b] NISTÉR D.: An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 6 (2004), 756–777.

[NLB*05] NG R., LEVOY M., BREDIF M., DUVAL G., HOROWITZ M., HANRAHAN P.: Light field photography with a hand-held plenoptic camera. Technical Report, Stanford University, 2005.

[Paj02a] PAJDLA T.: Stereo with Oblique Cameras. *International Journal of Computer Vision*, *47*, 1/2/3 (2002), 161–170.

[Paj02b] PAJDLA T.: Geometry of Two-Slit Camera. Research Report CTU-CMP-2002-02, Czech Technical University, Prague, 2002.

[PBP01] PELEG S., BEN-EZRA M., PRITCH Y.: Omnistereo: panoramic stereo imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*, 3 (2001), 279–290.

[Ple03] PLESS R.: Using many cameras as one. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition II* (2003), pp. 587–593.

[POV-Ray] POV-Ray: The Persistence of Vision Raytracer. http://www.povray.org/.

[RB98] RADEMACHER P., BISHOP G.: Multiple-center-of-projection images. *Computer Graphics*, *32* (1998), 199–206.

[RGL04] ROMAN A., GARG G., LEVOY M.: Interactive design of multi-perspective images for visualizing urban landscapes. In *Proc. IEEE Visualization* (2004).

[RL06] ROMAN A., LENSCH H. P. A.: Automatic multiperspective images. In *Proc. Eurographics Symposium on Rendering* (2006).

[RLS06] RAMALINGAM S., LODHA S., STURM P.: A generic structure-from-motion framework. *Computer Vision and Image Understanding*, *103*, 3 (2006), 218–228.

[RM84] RUCKER R., MIFFLIN H.: *The Fourth Dimension: Toward a Geometry of Higher Reality*. Houghton Mifflin, 1984.

[RTM*06] RASKAR R., TUMBLIN J., MOHAN A., AGRAWAL A., LI Y.: State-of-art report (STAR): computational photography. In *Proc. Eurographics* (2006).

[SDM04] STÜRZL W., DAHMEN H. J., MALLOT H.: The quality of catadioptric imaging—application to omindirectional stereo. In *Proc. European Conference on Computer Vision* (2004), pp. 614–627.

[Sei01] SEITZ S. M.: The Space of All Stereo Images. In *Proc. International Conference on Computer Vision I* (2001), pp. 26–33.

[SGN01] SWAMINATHAN R., GROSSBERG M. D., NAYAR S. K.: Caustics of catadioptric cameras. In *Proc. International Conference on Computer Vision II* (2001), pp. 2–9.

[SGN03] SWAMINATHAN R., GROSSBERG M. D., NAYAR S. K.: A perspective on distortions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2003).

[SKS99] SHUM, H. Y., KALAI A., SEITZ S. M.: Omnivergent stereo. In *Proc. International Conference on Computer Vision* (1999) pp. 22–29.

[SNG04] SWAMINATHAN R., NAYAR S. K., GROSSBERG M. D.: Designing mirrors for catadioptric systems that minimize image errors. In *Proc. Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras* (2004).

[SR03] STURM P., RAMALINGAM S.: A generic calibration concept—theory and algorithms. Research Report 5058, INRIA, 2003.

[SR04] STURM P., RAMALINGAM S.: A generic concept for camera calibration. In *Proc. European Conference on Computer Vision* (2004), pp. 1–13.

[SSP04] SIMON A., SMITH R. C., PAWLICKI R. R.: Omnistereo for panoramic virtual environment display systems. In *Proc. VR* (2004), pp. 67–74.

[Stu02] STURM P.: Mixing catadioptric and perspective cameras. In *Proc. Workshop on Omidirectional Vision* (2002).

[Stu05] STURM P.: Multi-view geometry for general camera models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2005), pp. 206–212.

[TJ95] THOMAS F., JOHNSTON O.: *Disney Animation: The Illusion Of Life* (rev. 1995), Disney Editions, Los Angeles, 1981.

[TS05] TARDIF J.-P., STURM P.: Calibration of cameras with radially symmetric distortion. In *Proc. Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras* (2005), pp. 44–51.

[UWH*03] UNGER J., WENGER A., HAWKINS T., GARDNER A., DEBEVEC P.: Capturing and rendering with incident light fields. In *Proc. EGSR* (2003), pp. 141–149.

[VRA*07] VEERARAGHAVAN A., RASKAR R., AGRAWAL A., MOHAN A., TUMBLIN J.: Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. In *Proc. ACM SIGGRAPH* (2007).

[WPS*03] WALD I., PURCELL T., SCHMITTLER J., BENTHIN C., SLUSALLEK P.: Realtime ray tracing and its use for interactive global illumination. In *Eurographics* State of the Art Reports (2003).

[WFH*97] WOOD D., FINKELSTEIN A., HUGHES J., THAYER C., SALESIN D.: Multiperspective panoramas for cel animation. In *Proc. ACM SIGGRAPH* (1997), pp. 243–250.

[WJV*05] WILBURN B., JOSHI N., VAISH V., TALVALA E., ANTUNEZ E., BARTH A., ADAMS A., HOROWITZ M., LEVOY M.: High performance imaging using large camera arrays. *ACM Transactions on Graphics* (Proc. SIGGRAPH, 2005).

[Wym05] WYMAN C.: Interactive image-space refraction of nearby geometry. In *Proc. GRAPHITE* (2005).

[YA06] YI S., AHUJA N.: An omnidirectional stereo system using a single camera. In *Proc. International Conference on Pattern Recognition* (2006).

[YKT94] YAGI Y., KAWATO S., TSUJI S.: Real-time omnidirectional image sensor (copis) for vision-guided navigation. *Robotics and Automation*, *10*, 1 (1994), 11–22.

[YLY07] YU X., LI F., YU J.: Image-space Caustics and Curvatures. In *Proc. Pacific Graphics* (2007), pp. 181–188.

[YM04a] YU J., MCMILLAN L.: General linear cameras. In *Proc. European Conference on Computer Vision II* (2004), pp. 14–27.

[YM04b] YU J., MCMILLAN L.: A framework for multiperspective rendering. In *Proc. Rendering Techniques, Eurographics Symposium on Rendering* (2004).

[YM05a] YU J., MCMILLAN L.: Modelling reflections via multiperspective imaging. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2005), pp. 117–124.

[YM05b] YU J., MCMILLAN L.: Multiperspective projection and collineation. In *Proc. International Conference on Computer Vision* (2005), pp. 580–587.

[ZB95] ZORIN D., BARR A. H.: Correction of geometric perceptual distortions in pictures. *Computer Graphics 29* (1995), 257–264.

[ZFPW03] ZOMET A., FELDMAN D., PELEG S., WEINSHALL D.: Mosaicing new views: the crossed-slits projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*, 6 (2003), 741–754.

[ZN06] ZOMET A., NAYAR S. K.: Lensless imaging with a controllable aperture. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2006).