Vis Comput
DOI 10.1007/s00371-013-0786-4

ORIGINAL ARTICLE

Ray geometry in non-pinhole cameras: a survey

Jinwei Ye · Jingyi Yu

Abstract A pinhole camera collects rays passing through a common 3D point and its image resembles what would be seen by human eyes. In contrast, a non-pinhole (multi-perspective) camera combines rays collected by different viewpoints. Despite their incongruity of view, their 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 this paper, we thoroughly discuss the design, modeling, and implementation of a broad class of non-pinhole cameras and their applications in computer graphics and vision. These include mathematical (conceptual) camera models such as the General Linear Cameras and real non-pinhole cameras such as catadioptric cameras and projectors. A unique component of this paper is a ray geometry analysis that uniformly models these non-pinhole cameras as manifolds of rays and ray constraints. We also model the thin lens as a ray transform and study how ray geometry is changed by the thin lens for studying distortions and defocusing. We hope to provide mathematical fundamentals to satisfy computer vision researchers as well as tools and algorithms to aid computer graphics and optical engineering researchers.

Keywords Camera models · Ray geometry · Thin lens · Catadioptric imaging · Computer vision · Computer graphics · Computational photography

1 Introduction

A pinhole camera collects rays passing through a common 3D point, which is called the *Center-of-Project* (*CoP*). Conceptually, it can be effectively viewed as a light-proof box with a small hole in one side, through which light from a scene passes and projects an inverted image on the opposite side of the box as shown in Fig. 1. The history of pinhole cameras can be traced back to Mo Jing, a Mohist philosopher in the fifth century BC in China who described a similar design using a closed room and a hole the wall. In the 10th century, Persian scientist Ibn al-Haytham (Alhazen) wrote about naturally occurring rudimentary pinhole cameras. In 1822, Niepce managed to take the first photograph using the pinhole camera obscura via lithography. Today, the pinhole camera is serving as the most common workhorse for general imaging applications.

The imaging quality of a pinhole camera relies heavily on choosing the proper sized pinhole: a small pinhole produces a sharp image but the image will be dimmer due to insufficient light whereas a large pinhole generates brighter but blurrier images. To address this issue, lenses have been used for converging lights. The goal is to replace the pure pinhole model with a pinhole-like optical model that can admit more light while maintaining image sharpness. For example, a thin, convex lens can be placed at the pinhole position with a focal length equal to the distance to the film plane in order to take pictures of distant objects. This emulates opening up the pinhole significantly. We refer to this thin lens-based pinhole approximation as pinhole optics.

In computer vision and graphics, pinhole cameras are dominating imaging model for two main reasons. First, pinhole geometry is rather simple. Each pinhole camera can be uniquely defined by only three parameters (the position of CoP in 3D). The pinhole imaging process can be decomposed into two parts: projecting the scene geometry into rays

J. Ye (✉) · J. Yu
University of Delaware, Newark, USA
e-mail: jye@cis.udel.edu

Published online: 06 March 2013

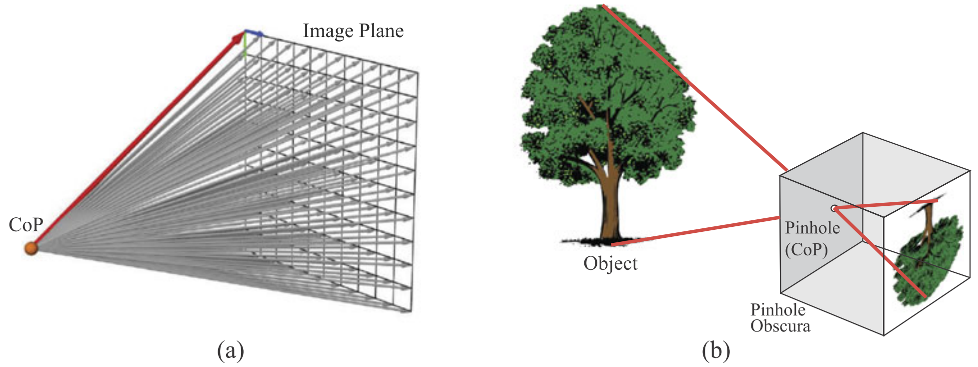


Fig. 1 (**a**) A pinhole camera collects rays passing through a common 3D point (the CoP). (**b**) An illustration of the pinhole obscura

and mapping the rays onto the image plane and they can be uniformly described by the classic 3 × 4 pinhole camera matrix [17]. Under homogeneous coordinates, the imaging process is linear. Second, in bright light, the human eyes act as a virtual pinhole camera where the observed images exhibit all characteristics as a pinhole image, e.g., points map to points, lines map to lines, parallel lines converge at a vanishing point, etc. Pinhole cameras are therefore also referred to as perspective cameras in the graphics and vision literature.

The pinhole imaging model, however, is rare in insect eyes. Compound eyes, which may consist of thousands of individual photoreceptor units or ommatidia are much more common. The image perceived is a combination of inputs from the numerous ommatidia (individual "eye units"), which are located on a convex surface, thus pointing in slightly different directions. Compound eyes hence possess a very large view angle and greatly help detect fast movement. Notice that rays collected by a compound eye will no longer follow pinhole geometry. Rather, they follow multi-viewpoint or multi-perspective imaging geometry.

The idea of non-pinhole imaging model has been widely adopted in art: artists, architects, and engineers regularly draw using non-pinhole projections. Despite their incongruity of views, effective non-pinhole images are still able to preserve spatial coherence. Pre-Renaissance and post-impressionist artists frequently use non-pinhole models to depict more than can be seen from any specific view point. For example, the cubism of Picasso and Matisse [40] can depict, within a single context, details of a scene that are simultaneously inaccessible from a single view, yet easily interpretable by a viewer. The goal of this survey is to carry out a comprehensive review on non-pinhole imaging models and their applications in computer graphics and vision.

Scope On the theory front, this survey presents a unique approach to systematically study non-pinhole imaging models in the ray space. Specifically, we parameterize rays to a 4D ray space using the Two-Plane Parametrization (2PP) [23, 43] and then study geometric ray structures of non-pinhole cameras in the ray space. We show that common non-perspective phenomenon such as reflections, refractions and defocus blurs can all be viewed as ray geometry transformations. Further, commonly used non-pinhole cameras can be effectively modeled as special (planar) 2D manifold in the ray space. The ray manifold model also provides feasible solutions for the forward projection problem, i.e., how to find the projection from a 3D point to its corresponding pixel in a non-pinhole imaging system.

On the application sides, we showcase a broad range of non-pinhole imaging systems. In computer vision, we discuss state-of-the-art solutions that apply non-pinhole cameras for stereo matching, multi-view reconstruction, shape-from-distortion, etc. In computational photography, we discuss emerging solutions that use the non-pinhole camera modelings for designing catadioptric cameras and projectors to acquire/project with a much wider Field-of-View (FoV) as well as various light field camera designs to directly acquire the 4D ray space in a single image. In computer graphics, we demonstrate using non-pinhole camera models for generating panoramas, creating cubism styles, rendering caustics, faux-animations from still-life scenes, rendering beyond occlusions, etc.

This survey is closely related to recent surveys on multi-perspective modeling and rendering [59] and computational photography [39]. Yu et al. [59] provides a general overview of multi-perspective cameras whereas we provide a comprehensive ray-space mathematical model for a broader class of non-pinhole cameras. Raskar et al. [39] focuses mostly on computational photography whereas we discuss the use of conceptual and real non-pinhole cameras for applications in computer vision and computer graphics. Further, our unified ray geometry analysis may fundamentally change people's view on cameras and projectors.

2 Pinhole optics

Pinhole cameras predate modern history. Geometrically, a pinhole camera collects rays passing through the CoP. Each pinhole camera, therefore, can be uniquely defined by only three parameters (the position of CoP). The pinhole imaging process 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 × 4 pinhole camera matrix [17], which combines six extrinsic and five intrinsic camera parameters into a single operator that maps homogeneous 3D points to a 2D image plane. These mappings are unique up to a scale factor, and similar models can be applied to describe orthographic cameras. In this section, we revisit the pinhole imaging process via a ray-space analysis.

2.1 Pinhole in ray space

2.1.1 Ray space

We use the *Two-Plane Parametrization* (*2PP*) that is widely used in light field [23] and lumigraph [6, 15] for representing rays, as shown in Fig. 2(a). Under 2PP, a ray in free space is defined by its intersections with two parallel planes (Π_{uv} and Π_{st}). Usually, Π_{uv} is chosen as the aperture plane ($z = 0$) whose origin is the origin of the coordinate system. Π_{st} is placed at $z = 1$ and chosen to be the default image plane. All rays that are not parallel to Π_{uv} and Π_{st} will intersect the two planes at $[u, v, 0]$ and $[s, t, 1]$, respectively, and we use $[u, v, s, t]$ for parameterizing each ray.

2.1.2 Pinhole ray geometry

Let us consider the pinhole model in ray space. By definition, all rays in a pinhole camera pass through a common 3D point, i.e., the CoP $\dot{C} = [C_x, C_y, C_z]$. For each ray $r = [u, v, s, t]$, there exist some λ that satisfies

$$\lambda[s, t, 1] + (1 - \lambda)[u, v, 0] = [C_x, C_y, C_z] \tag{1}$$

We have $\lambda = C_z$ and

$$\begin{cases} C_z s + (1 - C_z)u = C_x \\ C_z t + (1 - C_z)v = C_y \end{cases} \tag{2}$$

This indicates that rays in a pinhole camera obey two linear constraints, one in s and u and the other in t and v. We call them the *pinhole ray constraints*.

If we divide both sides of Eq. (2) by C_z and let $\dot{C}$ go to infinity, we have

$$\begin{cases} s - u = d_x \\ t - v = d_y \end{cases} \tag{3}$$

i.e., all rays have identical direction $[d_x, d_y, 1]$ and the camera degenerates to an orthographic camera. We call Eq. (3) the *orthographic ray constraints*; they are also linear in $s - u$ and $t - v$.

The pinhole and orthographic models have many nice properties that are useful to computer graphics and computer vision applications. For instance, all lines in the scene are projected to lines on the image. Similarly, a triangle in 3D space is projected as a triangle on the pinhole or orthographic 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.

2.1.3 Slit ray geometry

More general ray configurations, however, do not pass through a common 3D point. To relax the pinhole constraints, we study rays that pass through a line or slit l. We consider the following two cases:

(1) If l is parallel to Π_{uv} and Π_{st}, we can represent it with a point $\dot{P} = [P_x, P_y, P_z]$ on l and its direction $[d_x, d_y, 0]$. If a ray $r = [u, v, s, t]$ intersects l, there exist some λ_1 and λ_2 that satisfy

$$\lambda_1[s, t, 1] + (1 - \lambda_1)[u, v, 0] = [P_x, P_y, P_z] + \lambda_2[d_x, d_y, 0] \tag{4}$$

It is easy to see that $\lambda_1 = P_z$ and we can obtain a linear constraint in $[u, v, s, t]$ as

$$\frac{1 - P_z}{d_x}u - \frac{1 - P_z}{d_y}v + \frac{P_z}{d_x}s - \frac{P_z}{d_y}t + \frac{P_y}{d_y} - \frac{P_x}{d_x} = 0 \tag{5}$$

Yu and McMillan [57] show that it is equivalent to a general linear constraint $Au + Bv + Cs + Dt + E = 0$ where $A/B = C/D$. We call this *parallel slit constraint*.

(2) If l is not parallel to Π_{uv} and Π_{st}, it then can be directly parameterized by a ray under 2PP as $[u_0, v_0, s_0, t_0]$. All rays $r = [u, v, s, t]$ that intersect with l should satisfy

$$\begin{aligned} &\lambda_1[s, t, 1] + (1 - \lambda_1)[u, v, 0] \\ &\quad = \lambda_2[s_0, t_0, 1] + (1 - \lambda_2)[u_0, v_0, 0] \end{aligned} \tag{6}$$

We have $\lambda_1 = \lambda_2$ and

$$\frac{s - s_0}{t - t_0} = \frac{u - u_0}{v - v_0} \tag{7}$$

This is a bilinear constraint that we call *non-parallel slit constraint*. In the following sections, we will use the parallel and non-parallel slit constraints to model a broad class of non-pinhole imaging systems.

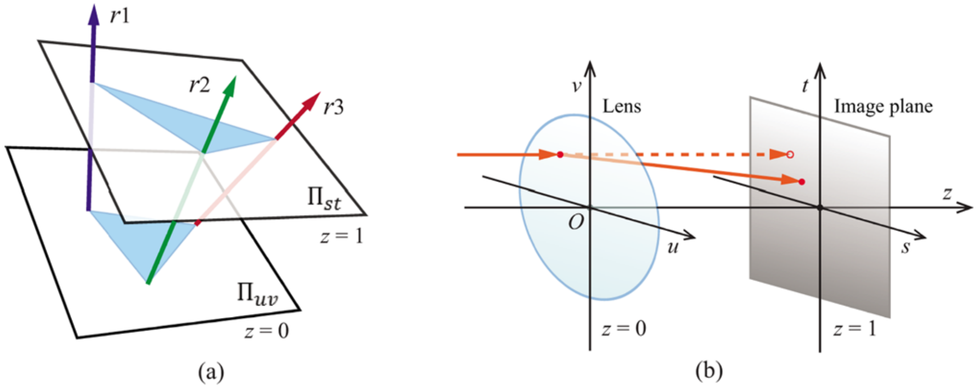


Fig. 2 (**a**) Two-Plane Parametrization: a ray is parameterized by its intersections with two parallel planes Π_{uv} $(z = 0)$ and $\Pi_{st}\,(z = 1)$. (**b**) Ray transformation after passing through a thin lens: the thin lens works as a shearing operator

2.2 The thin lens operator

Recall that practical pinhole cameras are constructed by using a thin lens in order to collect more lights. Although real lenses are typically a complex assembly of multiple lenses, they can still be effectively modeled using the *Thin Lens Equation*:

$$\frac{1}{a} + \frac{1}{b} = \frac{1}{f} \tag{8}$$

where a is the object distance; b is the image distance and f is the thin lens focal length.

The thin lens can be viewed as a workhorse that maps each incident ray $r = [u, v, s, t]$ approaching the lens to the exit ray $r' = [u', v', s', t']$ towards the sensor. Ng [31] and Ding et al. [13] separately derived the *Thin Lens Operator* (*TLO*) to show how rays are transformed after passing through a thin lens. By choosing the aperture plane as Π_{uv} at $z = 0$ and the image sensor plane as Π_{st} at $z = 1$, we have $u' = u$, $v' = v$. Using Eq. (8), it can be shown that the thin lens operator L transforms the ray coordinates as

$$\bigl[u', v', s', t'\bigr] = L\bigl([u, v, s, t]\bigr) = \left[u, v, s - \frac{1}{f}u, t - \frac{1}{f}v\right] \tag{9}$$

This reveals the thin lens L behaves as a linear, or more precisely, shear operator on rays, as shown in Fig. 2(b).

For a toy case study, let us investigate how a thin lens transforms a set of incident rays that follow pinhole geometry. Assume the incident rays originate from the CoP $\dot{C} = [C_x, C_y, C_z]$. By applying TLO (Eq. (9)) to the pinhole constraints (Eq. (2)), we obtain a new pair of constraints for the exiting rays $[u', v', s', t']$ as

$$\begin{cases} C_z s' + (1 - C_z + C_z \cdot \frac{1}{f})u' = C_x \\ C_z t' + (1 - C_z + C_z \cdot \frac{1}{f})v' = C_y \end{cases} \tag{10}$$

If $\dot{C}$ does not lie on the focal plane Π_{L-} of the lens at the world side $(C_z \neq -f)$, then Eq. (10) can be rewritten as

$$\begin{cases} \frac{fC_z}{f+C_z} s' + (1 - \frac{fC_z}{f+C_z})u' = \frac{fC_x}{f+C_z} \\ \frac{fC_z}{f+C_z} t' + (1 - \frac{fC_z}{f+C_z})v' = \frac{fC_y}{f+C_z} \end{cases} \tag{11}$$

Therefore, the exiting rays follow a new set of pinhole constraints with the new CoP at $\frac{f}{f+C_z}[C_x, C_y, C_z]$.

If $\dot{C}$ lies on Π_{L-} $(C_z = -f)$, then Eq. (10) degenerate to the orthographic constraints:

$$\begin{cases} s - u = -\frac{C_x}{f} \\ t - v = -\frac{C_y}{f} \end{cases} \tag{12}$$

In this case, all exiting rays correspond to an orthographic camera with direction $[-\frac{C_x}{f}, -\frac{C_y}{f}, 1]$.

The results derived above are well-known as they can be directly viewed as the image of a 3D point through the thin lens. Nevertheless, for more complex cases when the incident rays do *not* follow pinhole geometry, the TLO analysis is crucial for modeling the exit ray geometry [13]. This case study also reveals that all rays emitting from a 3D scene point $\dot{C}$ will generally converge at a different 3D point $\dot{C}'$ through the thin lens. The cone of rays passing through $\dot{C}'$ will therefore spread onto a disk of pixels on the sensor. This process is commonly described using the Point Spread Function (PSF), i.e., the mapping from a 3D point to a disk of pixels. As shown in Fig. 3, assuming that the sensor moves Δz away from $z = C'_z$ and the lens has circular aperture with diameter D, then the PSF is a disk of size $D_p = D\frac{|\Delta z|}{C'_z}$.

3 Non-pinhole imaging models

More general camera models do not follow pinhole camera geometry, i.e., not all rays collected by the camera need to pass through a common point. Such cameras are often referred to as non-pinhole cameras. In contrast to pinhole and orthographic cameras, which can be uniformly described using the 3×4 camera matrix, non-pinhole camera models are defined less precisely. In practice, many non-pinhole camera models are defined 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. In this section, we apply pinhole constraints and slit constraints (parallel and non-parallel) to study the ray properties in various non-pinhole camera models.

3.1 Classical non-pinhole cameras

Pushbroom cameras, consisting of a linear sensor, are routinely used in satellite imagery [16]. 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. Rays collected by a pushbroom camera should satisfy two constraints: (1) the slit constraint, where the slit is the motion path of the pushbroom sensor; (2) all the sweeping rays are parallel to some plane that is perpendicular to the slit. Assume the common slit is parallel to Π_{uv} and Π_{st} and we parameterize it with a point $[x_0, y_0, z_0]$ on the slit and the slit's direction $[d_x, d_y, 0]$, the two constraints for all rays $[u, v, s, t]$ captured by a pushbroom camera can be formulated as

$$\begin{cases} \frac{1-z_0}{d_x}u - \frac{1-z_0}{d_y}v + \frac{z_0}{d_x}s - \frac{z_0}{d_y}t + \frac{y_0}{d_y} - \frac{x_0}{d_x} = 0 \\ [s-u, t-v, 1] \cdot [d_x, d_y, 0]^T = 0 \end{cases} \qquad (13)$$

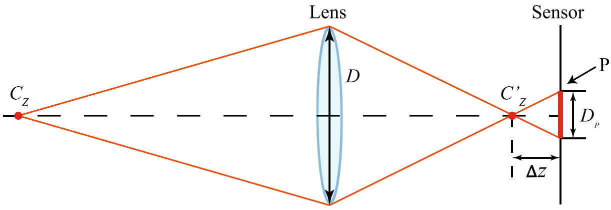


Fig. 3 The thin lens maps a 3D scene point $\dot{C}$ to $\dot{C}'$. By moving the sensor Δz away from $z = C'_z$, $\dot{C}'$ spreads onto a PSF with size D_p

where the first constraint is the parallel slit constraint and the second corresponds to the parallel sweeping planes, both linear. In practice, a pushbroom image can be synthesized by moving a perspective camera along a linear path and assembling the same column of each perspective image as shown in Fig. 4(a) and (b).

Another popular class of non-pinhole cameras are the XSlit cameras. An XSlit camera has two oblique (neither parallel nor coplanar) slits in 3D space. The camera collects rays that simultaneously pass through the two slits and projects them onto an image plane. If we choose the parametrization plane parallel to both slits, rays in an XSlit camera will then satisfy two parallel slit constraints, i.e., two linear constraints. Similar to pushbroom images, XSlit images can also be synthesized using images captured by a moving pinhole camera. Zomet et al. [60] generated XSlit images by stitching linearly varying columns across a row of pinhole images, as shown in Fig. 4(c) and (d).

3.2 General non-pinhole cameras

The analysis above reveals that pinhole, orthographic, pushbroom, and XSlit all correspond to 2D manifolds in the ray space (since they are subject to two linear ray constraints). This is not surprising as a general imaging process entails mapping 3D geometry onto a 2D manifold of rays, i.e., each pixel $[x, y]$ maps to a ray. Therefore a general non-pinhole camera can be viewed as a 2D ray manifold Σ:

$$\Sigma(x, y) = \big[u(x, y), v(x, y), s(x, y), t(x, y)\big] \qquad (14)$$

To analyze ray geometry, one then approximate the local behavior of the rays by computing the tangent plane about any specified ray r. The tangent plane can be expressed as two spanning vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ by taking the partial derivatives of $[u, v, s, t]$:

$$\mathbf{d}_1 = [u_x, v_x, s_x, t_x], \qquad \mathbf{d}_2 = [u_y, v_y, s_y, t_y] \qquad (15)$$

This is analogous to modeling a curved 2D surface using local tangent planes. A local ray tangent plane hence can then be modeled by three generator rays: r, $r + \mathbf{d}_1$ and $r + \mathbf{d}_2$.

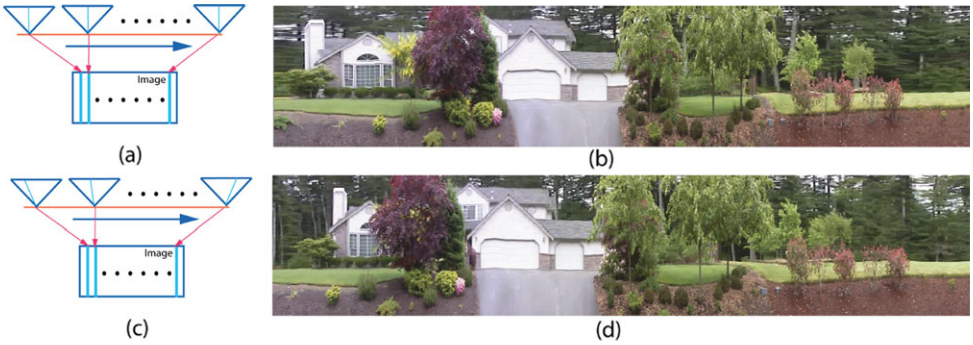


Fig. 4 Pushbroom and XSlit. (**a**) The stationary column sampling routine for synthesizing a pushbroom panorama (**b**). (**c**) The linearly varying column sampling routine for synthesizing an XSlit panorama (**d**) (courtesy of Steve Seitz)

Table 1 Characterizing general linear cameras by characteristic equation

Characteristic equation	2 Solutions	1 Solution	0 Solution	∞ Solutions
$A \neq 0$	XSlit	Pencil/Pinhole[a]	Bilinear	$\varnothing$
$A = 0$	$\varnothing$	Pushbroom	Twisted/Ortho.[a]	EPI

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

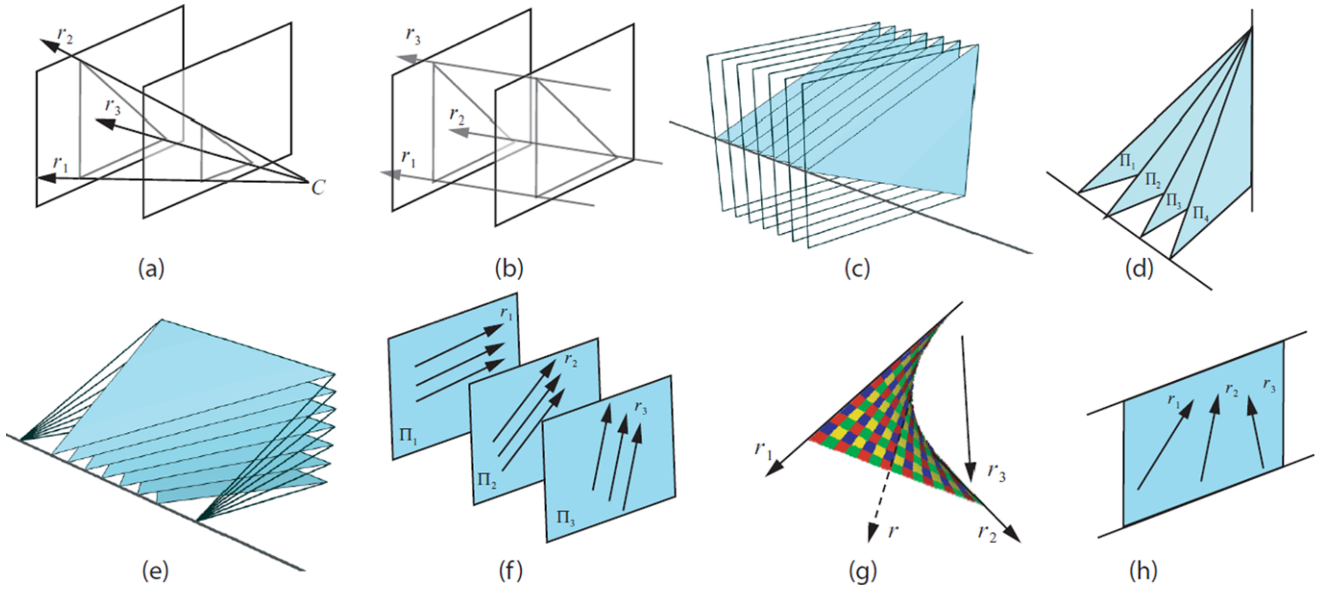


Fig. 5 General Linear Camera Models. (**a**) A pinhole camera. (**b**) An orthographic camera. (**c**) A pushbroom. (**d**) An XSlit camera. (**e**) A pencil camera. (**f**) A twisted orthographic camera. (**g**) A bilinear camera. (**h**) An EPI camera. See Sect. 3.2.1 for detailed discussions on each GLC

3.2.1 General linear cameras (GLC)

To study ray geometry of local ray tangent plane, Yu and McMillan [55] developed a new camera model called the *General Linear Camera* (*GLC*). GLCs are 2D planar ray manifolds which can apparently describe the traditional pinhole, orthographic, pushbroom, and XSlit cameras.

A GLC is defined as the affine combination of three generator rays $r_i = [u_i, v_i, s_i, t_i]$, $i = 1, 2, 3$:

$$r = \alpha[u_1, v_1, s_1, t_1] + \beta[u_2, v_2, s_2, t_2] + (1 - \alpha - \beta)[u_3, v_3, s_3, t_3] \tag{16}$$

For example, in the ray tangent plane analysis, the three ray generators are chosen as r, $r + \mathbf{d}_1$ and $r + \mathbf{d}_2$.

To determine the type of the non-pinhole camera for any GLC specification, they further derived a ray characteristic equation that computes how many singularities (lines or points) that all rays in the GLC can pass through:

$$\begin{vmatrix} \lambda \cdot s_1 + (1-\lambda) \cdot u_1 & \lambda \cdot t_1 + (1-\lambda) \cdot v_1 & 1 \\ \lambda \cdot s_2 + (1-\lambda) \cdot u_2 & \lambda \cdot t_2 + (1-\lambda) \cdot v_2 & 1 \\ \lambda \cdot s_3 + (1-\lambda) \cdot u_3 & \lambda \cdot t_3 + (1-\lambda) \cdot v_3 & 1 \end{vmatrix} = 0 \tag{17}$$

Equation (17) yields a quadratic equation of the form $A\lambda^2 + B\lambda + C = 0$ where

$$A = \begin{vmatrix} s_1 - u_1 & t_1 - v_1 & 1 \\ s_2 - u_2 & t_2 - v_2 & 1 \\ s_3 - u_3 & t_3 - v_3 & 1 \end{vmatrix}, \quad C = \begin{vmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ u_3 & v_3 & 1 \end{vmatrix},$$
$$B = \begin{vmatrix} s_1 - u_1 & v_1 & 1 \\ s_2 - u_2 & v_2 & 1 \\ s_3 - u_3 & v_3 & 1 \end{vmatrix} - \begin{vmatrix} t_1 - v_1 & u_1 & 1 \\ t_2 - v_2 & u_2 & 1 \\ t_3 - v_3 & u_3 & 1 \end{vmatrix} \tag{18}$$

An edge parallel condition is defined to check if all three pairs of the corresponding edges of the $u - v$ and $s - t$ triangles formed by the generator rays are parallel:

$$\frac{s_i - s_j}{t_i - t_j} = \frac{u_i - u_j}{v_i - v_j} \quad i, j = 1, 2, 3 \text{ and } i \neq j \tag{19}$$

Given three generator rays, its GLC type can be determined by the A coefficient and the discriminant $\Delta = B^2 - 4AC$ of its characteristic equation and the edge parallel condition, shown in Table 1. Yu and McMillan [55] have shown that there are precisely eight types of GLC as shown in Fig. 5: in a pinhole camera, all rays pass through a single point; in an orthographic camera, all rays are parallel;

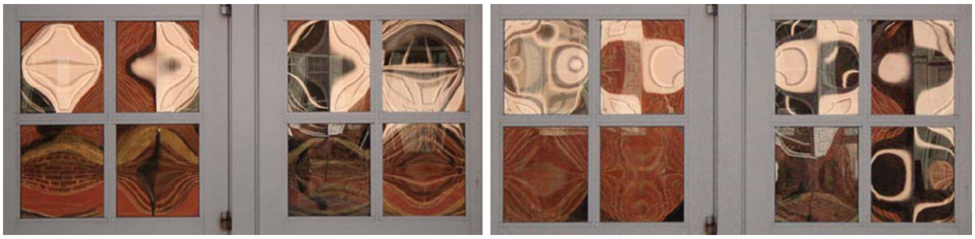


Fig. 6 Curved line images on specular window surfaces: the two images of 0.5 m × 1 m near-flat windows are captured from 15 m away. Images of straight lines far away form interesting conic patterns

Table 2 Conic types observed in general linear cameras

GLC type	Pinhole	Ortho.	XSlit	Pushbroom	Pencil	Twisted	Bilinear
Determinant	$\varDelta = 0$	$\varDelta = 0$	$\varDelta > 0$	$\varDelta > 0$	$\varDelta = 0$	$\varDelta = 0$	$\varDelta < 0$
Conic Type	Line	Line	Hyperbolae	Hyperbolae	Parabola	Parabola	Ellipse

in a pushbroom camera [16], all rays lie on a set of parallel planes and pass through a line; in an XSlit camera [60], all rays pass through two non-coplanar lines; in a pencil camera, all coplanar rays originate from a point on a line and lie on a specific plane through the line; in a twisted orthographic camera, all rays lie on parallel twisted planes and no rays intersect; in a bilinear camera [32], no two rays are coplanar and no two rays intersect; and in an EPI camera, all rays lie on a 2D plane.

To find the projection of a 3D point in a GLC, one can combine the GLC constraints with pinhole constraints. For example, considering an XSlit camera that obeys two parallel slit constraints (Eq. (5)) derived in Sect. 2.1.3. Rays passing through the 3D point obeys another two pinhole linear constraints (Eq. (2)). We therefore can uniquely determine the ray from the XSlit that passes through the 3D point. To calculate the projection of a 3D line in the XSlit camera, one can compute the projection of each point on the line. Ding et al. [12] show that line projections can only be lines or conics, as shown in Fig. 6. The complete classification of conics that can be observed by each type of GLC is enumerated in Table 2.

3.2.2 Case study 1: reflection on curved mirrors

To demonstrate how to use the GLC analysis to model general non-pinhole cameras, let us look at a special non-pinhole camera, a catadioptric camera that combines a pinhole camera and a curved mirror. Given the camera position and the mirror surface, we can map each reflected ray into the ray space as $[u, v, s, t]$. Assuming the 3D surface is of form $z(x, y)$, the reflected ray in $[u, v, s, t]$ can be computed via the reflection constraint:

$$r = i - 2(\hat{n} \cdot i)\hat{n} \tag{20}$$

where $r = [r_x, r_y, r_z]$ is the reflected ray; i is the incident ray and $\hat{n}$ is unit normal that can be computed by normalizing $[-z_x, -z_y, 1]$. If we choose $\varPi_{uv}$ $(z = 0)$ to contain the surface point $(x, y, z(x, y))$ and tangential to the surface and set $\varPi_{st}$ as $z = 1$, we obtain $[u, v, s, t]$ of the reflected ray as

$$[u, v, s, t] = \left[x - z \cdot \frac{r_x}{r_z}, y - z \cdot \frac{r_y}{r_z}, x - (z - 1) \cdot \frac{r_x}{r_z}, \right.$$
$$\left. y - (z - 1) \cdot \frac{r_y}{r_z}\right] \tag{21}$$

All variables r, z, u, v, s and t are functions in x and y, hence, the set of reflection rays from the 3D surface form a 2D parametric ray manifold in x and y. We can then use the tangent GLC analysis (Sect. 3.2.1) to determine the type of local non-pinhole camera model. Using this approach, Yu and McMillan [56] have shown that all local reflections observed by a pinhole or an orthographic camera can be characterized as only be XSlit, pushbroom, pinhole, or orthographic, all are special cases of XSlit. The two slits correspond to the two reflection caustic surfaces and provide a special set of rulings on the surface. These rulings determine which rays lie on the local tangent GLC and the local distortions seen in the reflection.

3.2.3 Case study 2: 3D surfaces

It is also possible to convert a 3D surface to a 2D ray manifold. Yu et al. [58] proposed a normal-ray model to represent surfaces that locally parameterize the surface about its normal based on focal surface approximation, as shown in Fig. 7(a)–(d). Given a smooth surface $S(x, y)$, at each vertex $\dot{P}$, we orient the local frame to align $z = 0$ with the tangent plane at $\dot{P}$. We further assume $\dot{P}$ is the origin of $z = 0$ plane and set $\varPi_{uv}$, $\varPi_{st}$ at $z = 0$ and $z = 1$, respectively. Under this parametrization, normal rays can be mapped as $n = [u, v, s, t]$. The tangent plane can then be represented by a GLC with three rays: n, $n + n_x$ and $n + n_y$.

By using the GLC analysis, one can compute the two slits for each normal ray GLC from the characteristic equation. Yu et al. [58] have shown that the two slits are perpendicular

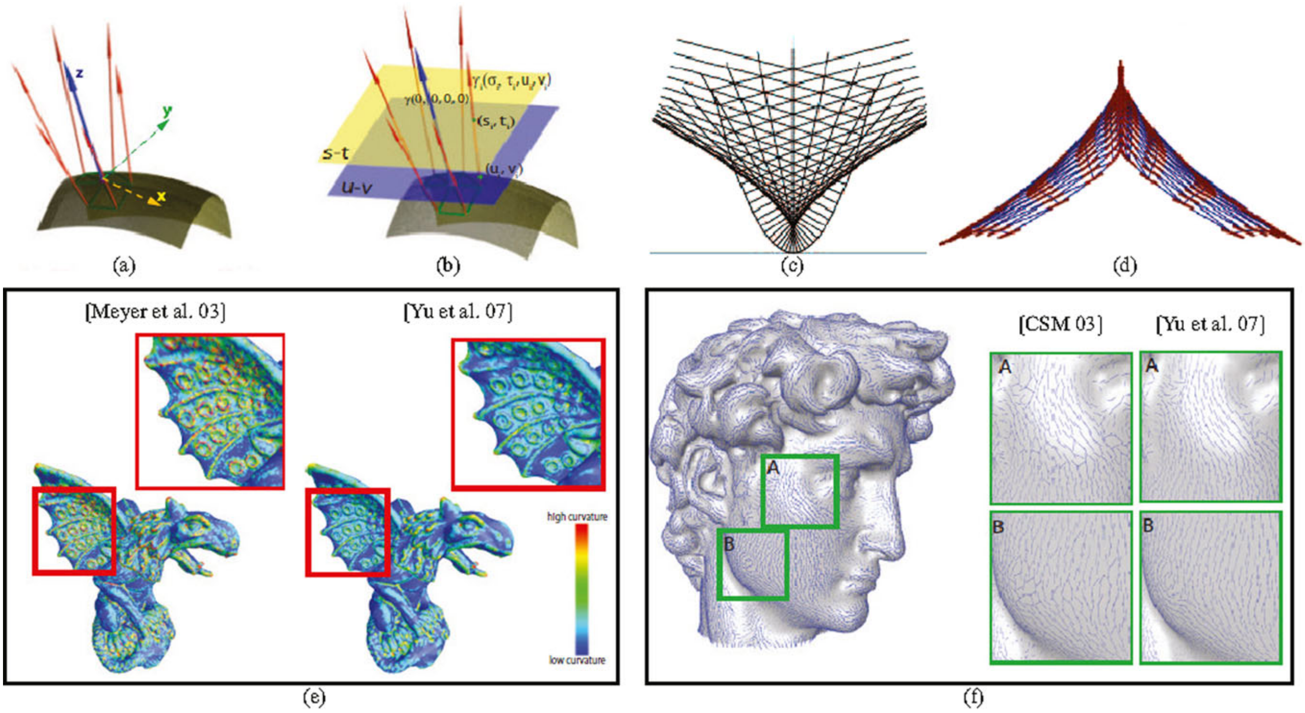


Fig. 7 Estimating focal meshes using the normal ray model. (**a**) We orient the local frame to align Π_{uv} ($z=0$) with the surface tangent plane at $\dot{P}$. (**b**) We choose the second plane Π_{st} to be $z=1$. Each neighboring normal ray can be parameterized by as $[u, v, s, t]$. (**c**) Focal surfaces (curve) formed by the foci of the normal rays of a parabolic surface. (**d**) Neighboring normal rays are constrained by the slits (*red*) that rule the focal surfaces (*blue*). (**e**) The color-coded mean curvature image illustrates that the normal ray model (*right*) is less sensitive to mesh connectivity than [28] (*left*), especially shown on the wings of the gargoyle model. (**f**) *Left*: the estimated min principal curvature direction using normal ray model. *Right*: compare the normal ray model with [7] on different parts of the model

to each other and rule the focal surfaces. Swept by the loci of the principal curvatures' radii, the focal surfaces encapsulate many useful geometric properties of the corresponding actual surface. For example, normals of the actual surface are tangent to both focal surfaces and the normal of each focal surface indicates a principle direction of the corresponding point on the original surface. In fact, each slit is tangential to its corresponding focal surface. Since the two focal surfaces are perpendicular to each other, one slit is parallel to the normal of the focal plane that the other slit corresponds to. Therefore, the two slits give us the principal directions of the original surface. Besides, the depths of the slits/focal surfaces, computed as the roots of the characteristic equation, represent the principle curvatures of the surface. In Fig. 7(e) and (f), we shows two results of estimated mean curvature and min principle curvature using normal ray model, comparing to the Voronoi-edge algorithm [7, 28].

4 Non-pinhole camera through the thin lens

Next, we review how ray geometry transforms through the thin lens. A typical example is reflections observed by a camera with a wide aperture.

4.1 GLC through a thin lens

Ding et al. [13] studied the transformation for GLCs through a thin lens. To reiterate, recall that a GLC defined by the affine combination of three generator rays and the thin lens operator $L(\circ)$ is a linear operator, therefore the affinity of the GLC will be preserved under the thin lens operator, i.e.,

$$\begin{aligned} L(r) &= L\big(\alpha r_1 + \beta r_2 + (1-\alpha-\beta) r_3\big) \\ &= \alpha L(r_1) + \beta L(r_2) + (1-\alpha-\beta) L(r_3) \end{aligned} \tag{22}$$

Equation (22) reveals that the exit rays form a new GLC where the three new generator rays are $L(r_1)$, $L(r_2)$, and $L(r_3)$.

4.2 Slit-direction duality

To determine the type of exit GLCs transformed by thin lens, it is important to investigate the duality between slits and directions, which can be derived by applying TLO (Eq. (9)) on the slit ray constraint (parallel or non-parallel).

If the slit does not lie on the focal plane Π_{L-} ($z=-f$) of the lens at the world side, we consider the incident rays in the following two cases:

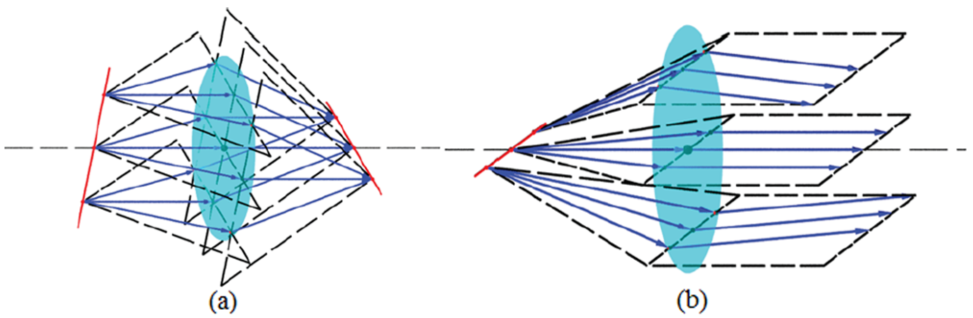


Fig. 8 Slit-Direction Duality. (**a**) A pushbroom camera with a slit at focal length transforms to another pushbroom camera with a different slit at focal length. (**b**) A pencil camera transforms to twisted orthographic camera

Table 3 General linear camera transformations through a thin lens

Incident GLC	*Exit GLC*
XSlit: all rays passing through two slits l_1 and l_2	l_1 or l_2 lies on Π_{L-}: ***Pushbroom*** Neither slits lies on Π_{L-}: ***XSlit***
Pushbroom: collect rays parallel to some plane Π and passing through a slit l	l lies on Π_{L-}: ***Pushbroom*** l does not lie on Π_{L-}: ***XSlit***
Pinhole: all rays passing through COP $\dot{C}$	$\dot{C}$ lies on Π_{L-}: ***Orthographic*** $\dot{C}$ does not lie on Π_{L-}: ***Pinhole***
Pencil: collect rays on a set of non-parallel planes that share a line l	l lies on Π_{L-}: ***Twisted Orthographic*** l does not lie on Π_{L-}: ***Pencil***
Bilinear	***Bilinear***
Orthographic	***Pinhole***
Twisted Orthographic	***Pencil***

(1) The slit parallel to Π_{L-} is parameterized by a point $[P_x, P_y, P_z]$ and direction $[d_x, d_y, 0]$. Since $P_z \neq -f$ one can combine the parallel slit ray constraint (Eq. (5)) with TLO as

$$\frac{1-P'_z}{d_x}u' - \frac{1-P'_z}{d_y}v' + \frac{P'_z}{d_x}s' - \frac{P'_z}{d_y}t' + \frac{P'_y}{d_y} - \frac{P'_x}{d_x} = 0 \quad (23)$$

where $[P'_x, P'_y, P'_z] = \frac{f}{f+P_z}[P_x, P_y, P_z]$. Therefore all exit rays pass through a new slit parameterized by $[P'_x, P'_y, P'_z]$ and its direction $[d_x, d_y, 0]$.

(2) The slit is not parallel to Π_{L-}. In this case, one can apply the TLO on the bilinear non-parallel slit ray constraint (Eq. (7)) to obtain a new constraint for $[u', v', s', t']$ as

$$\frac{u'-u_0}{v'-v_0} = \frac{s'-s_0+\frac{u_0}{f}}{t'-t_0+\frac{v_0}{f}} \quad (24)$$

which is also bilinear constraint that indicates that all exit rays will pass through another slit $[u_0, v_0, s_0 - \frac{u_0}{f}, t_0 - \frac{v_0}{f}]$.

These derivations show that all rays that pass through a slit not lying on Π_{L-} will be mapped to exit rays through a new slit at the other side of the lens. We call it the Slit-Slit Duality(see Fig. 8(a)).

If the slit lies on Π_{L-} ($P_z = -f$), the linear slit constraint can be reformulated as the following by applying TLO:

$$\left[s'-u', t'-v', 1\right] \cdot \left[-f d_y, f d_x, P_y d_x - P_x d_y\right]^T = 0 \quad (25)$$

It reveals that all exit rays are orthogonal to a vector $\mathbf{n} = [-f d_y, f d_x, P_y d_x - P_x d_y]$, which is the normal direction of the plane formed by the slit and the lens optical center. Therefore, for rays passing through a slit lying on Π_{L-}, exit rays correspond to directions. We call it the Slit-Direction Duality, as shown in Fig. 8(b).

As a reciprocity of the analysis for incident rays through a slit lying on Π_{L-}, for those rays parallel to some plane Π through the lens optical center, all exit rays will pass through a slit parallel to 2PP and lying on the focal plane Π_{L+} of the lens at the sensor side. Furthermore, the slit can be found by intersecting Π with Π_{L+}. This is called the Direction-Slit Duality. The complete GLC transformations are listed in Table 3.

4.3 Case study 3: defocus analysis in catadioptric cameras

Based on the GLC-TLO analysis, Ding et al. [13] showcased using the theory for characterizing and compensating catadioptric defocusing. They use the Ray Spread Function (RSF) to describe how a general set of incident rays spread to pixels on the sensor. The classical PSF is a special case of the RSF when the incident rays are from a pinhole camera.

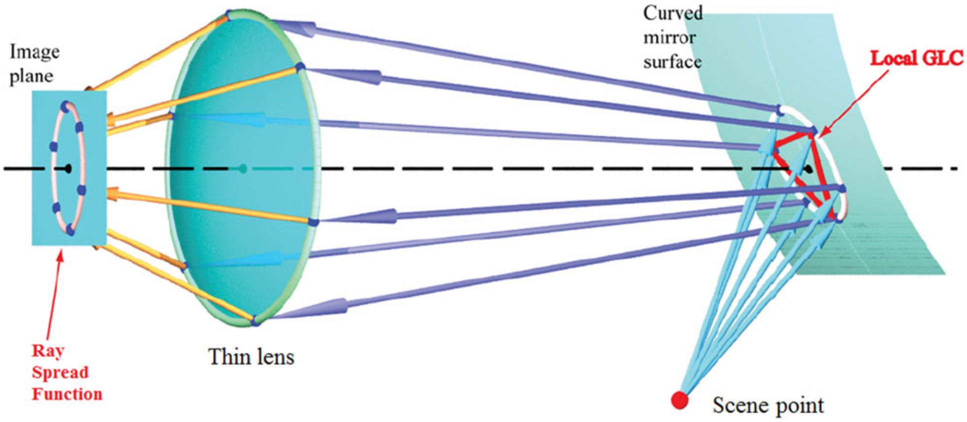


Fig. 9 The formation of RSF in a catadioptric imaging system: light from a scene point is reflected off the mirror, truncated by the thin lens aperture, and finally received by the sensor forming the RSF

Assume a scene point $\dot{P}$ and a curved mirror surface $z(x, y)$, the RSF of $\dot{P}$ is formed by rays emitting from $\dot{P}$ reflected off the mirror, then transmitted through the lens and finally received by the sensor, as shown in Fig. 9. They then formulate the RSF by composing the thin lens operator $L(\circ)$, the aperture operator $A(\circ)$, and the reflection operator $R(z(x, y), \dot{P})$. Notice the order between $L(\circ)$ and $A(\circ)$ is interchangeable:

$$\begin{aligned} RSF(\dot{P}) &:= K(s, t) \\ &= L\Big(A\Big(R\big(z(x, y), \dot{P}\big)\Big)\Big) \\ &= A\Big(L\Big(R\big(z(x, y), \dot{P}\big)\Big)\Big) \\ &= A\Big(L\big(u(s, t), v(s, t)\big)\Big) \\ &= \begin{cases} nil, & G(u(s, t), v(s, t)) > 0 \\ L(u(s, t), v(s, t)), & G(u(s, t), v(s, t)) \le 0 \end{cases} \end{aligned} \tag{26}$$

where $u(s, t)$ and $v(s, t)$ are determined by the mirror geometry, and $G(u, v) = u^2 + v^2 - \frac{D}{2}^2$ corresponds to a circular aperture of diameter D.

Using the reflection analysis in Sect. 3.2.2, one can decompose each local reflection patch as an XSlit camera. It is particulary useful to analyze the RSF of an XSlit GLC. According to the GLC-TLO transformation, the exit GLC is also an XSlit with two slits l_1 and l_2 lying on $z = \lambda_1$ and $z = \lambda_2$, respectively. To simplify the analysis, let us consider the special case when the two slits are orthogonal to each other. One can further rotate the coordinate system such that the slit directions are aligned with the u, v axis. The two slits constraints can then be rewritten as

$$\begin{cases} (1 - \lambda_1)u' + \lambda_1 s' = 0 \\ (1 - \lambda_2)v' + \lambda_2 t' = 0 \end{cases} \Longrightarrow \begin{cases} u' = \frac{s'}{1-\frac{1}{\lambda_1}} \\ v' = \frac{t'}{1-\frac{1}{\lambda_2}} \end{cases} \tag{27}$$

Substituting Eq. (27) into the aperture constraint $G(u, v)$, we have

$$\left(\frac{s'}{\frac{1}{\lambda_1} - 1}\right)^2 + \left(\frac{t'}{\frac{1}{\lambda_2} - 1}\right)^2 \le \left(\frac{D}{2}\right)^2 \tag{28}$$

Equation (28) indicates that the RSF of a GLC is of elliptical shape and the major and minor radii of the ellipse are $|\frac{1}{\lambda_1} - 1|$ and $|\frac{1}{\lambda_2} - 1|$. The specific shape and orientation of the ellipse varies along with the depths of the two slits λ_1 and λ_2. Various cases for different λ_1 and λ_2 combinations are now enumerated and shown in Fig. 10:

1. When $\frac{2\lambda_1\lambda_2}{\lambda_2+\lambda_1} > 1$, the major radius is $|\frac{1}{\lambda_2} - 1| \cdot \frac{D}{2}$, and the major axis is parallel to the direction of the slit l_1.
2. When $\lambda_1 = 1$, the minor radius $|\frac{1}{\lambda_1} - 1| \cdot \frac{D}{2} = 0$, i.e., the RSF degenerates into a line segment parallel to slit l_1, and the length of the line segment is $|\frac{1}{\lambda_2} - 1| \cdot \frac{D}{2}$.
3. When $\frac{2\lambda_1\lambda_2}{\lambda_2+\lambda_1} = 1$, the RSF shape degenerates into a circular disk, and the radius of the circle is $|\frac{\lambda_1-\lambda_2}{\lambda_1+\lambda_2}| \cdot \frac{D}{2}$.
4. When $\frac{2\lambda_1\lambda_2}{\lambda_2+\lambda_1} < 1$, the major radius is $|\frac{1}{\lambda_1} - 1| \cdot \frac{D}{2}$, and the major axis is parallel to the direction of the slit l_2.
5. When $\lambda_2 = 1$, the minor radius $|\frac{1}{\lambda_2} - 1| \cdot \frac{D}{2} = 0$, i.e., the RSF degenerates into a line segment parallel to the slit l_2, and the length of the segment is $|\frac{1}{\lambda_1} - 1| \cdot \frac{D}{2}$.

The analysis reveals that the RSF caused by a 3D point in a catadioptric mirror can only be an ellipse, a circle, or a line segment. Furthermore, the shape of the RSF depends on the location of the scene point, size of the aperture and the camera's focus setting.

5 Applications of synthetic non-pinhole cameras

Although many of the non-pinhole cameras discussed above are synthetic models, they have broad applications in graphics and vision.

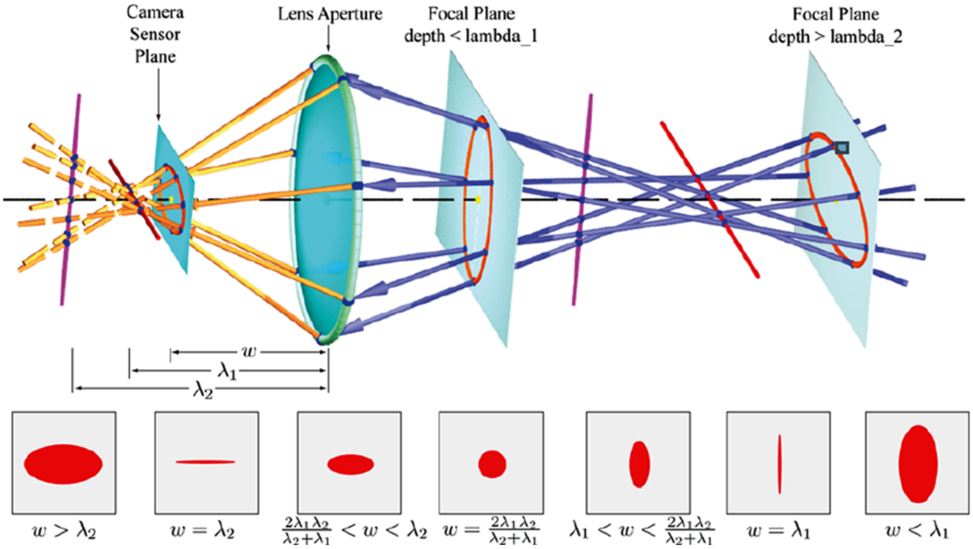

Camera Sensor Plane
Lens Aperture
Focal Plane depth < lambda_1
Focal Plane depth > lambda_2
w
λ_1
λ_2
$w > \lambda_2$
$w = \lambda_2$
$\frac{2\lambda_1\lambda_2}{\lambda_2+\lambda_1} < w < \lambda_2$
$w = \frac{2\lambda_1\lambda_2}{\lambda_2+\lambda_1}$
$\lambda_1 < w < \frac{2\lambda_1\lambda_2}{\lambda_2+\lambda_1}$
$w = \lambda_1$
$w < \lambda_1$

Fig. 10 Various RSF Shapes: the RSF shape depends on the distance w between the aperture and the sensor planes

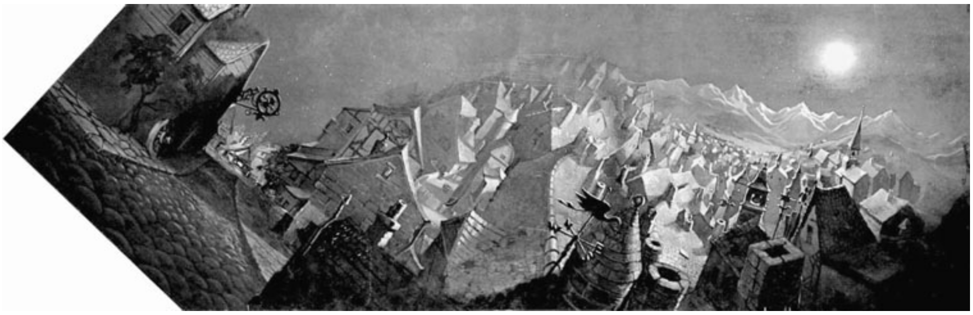


Fig. 11 Multi-perspective panorama from Disney's Pinocchio (courtesy of Disney)

5.1 Synthesizing panoramas

A non-pinhole camera can combine patches from multiple pinhole cameras into a single image to overcome the FoV limits. In image-based rendering, pushbroom and XSlit panoramas can be synthesized by translating a pinhole camera along the image plane and then stitching specific columns from each perspective image. Pushbroom image assembles the same column [41] whereas the XSlit linearly varies the column [60]. The synthesized panoramas can exhibit image distortions such as apparent stretching and shrinking, and even duplicated projections of a single point [44, 56]. To alleviate the distortions, Agarwala et al. [3] constructed panorama using arbitrarily shaped regions of the source images taken by a pinhole camera moving along a straight path, instead of selecting simple strips. The region shape in each perspective image is carefully chosen by using Markov Random Field (MRF) optimization based on various properties that the panorama desires. Instead of translating the camera planarly, Shum and Szeliski [42] created panorama on a cylindrical manifold by panning a pinhole camera around its optical center. They project the perspective images to a common cylinder to combine the final panorama. Peleg et al. [35] proposed a mosaicing method for more general camera motion. They first determine the projection manifolds according to the camera motion and then warp the source images onto the manifolds to stitch the panorama.

Non-pinhole camera models are also widely used for creating computer generated panoramas. The 1940 Disney animation Pinocchio [36] opens with a virtual camera flying over a small village. Instead of traditional panning, the camera rotates at the same time, creating astonishing 3D effect via 2D painting. In fact, the shot was made by drawing a panoramic view with "warped perspective" as shown in Fig. 11 and then showing only a small clip at a time. Wood et al. [53] proposed to create similar cel animation effect from 3D models. They combined elements of multiple pinhole strips into a single image using a semi-automatic image registration process. Their method relies on optimization techniques as well as optical flow and blending transitions between views.

Popescu et al. [37] proposed the graph camera for generating a single panoramic image that simultaneously captures/renders regions of interest of a 3D scene from different perspectives. Conceptually, the graph camera is a combination of different pinhole cameras that sample the scene. A non-perspective panorama can then be generated

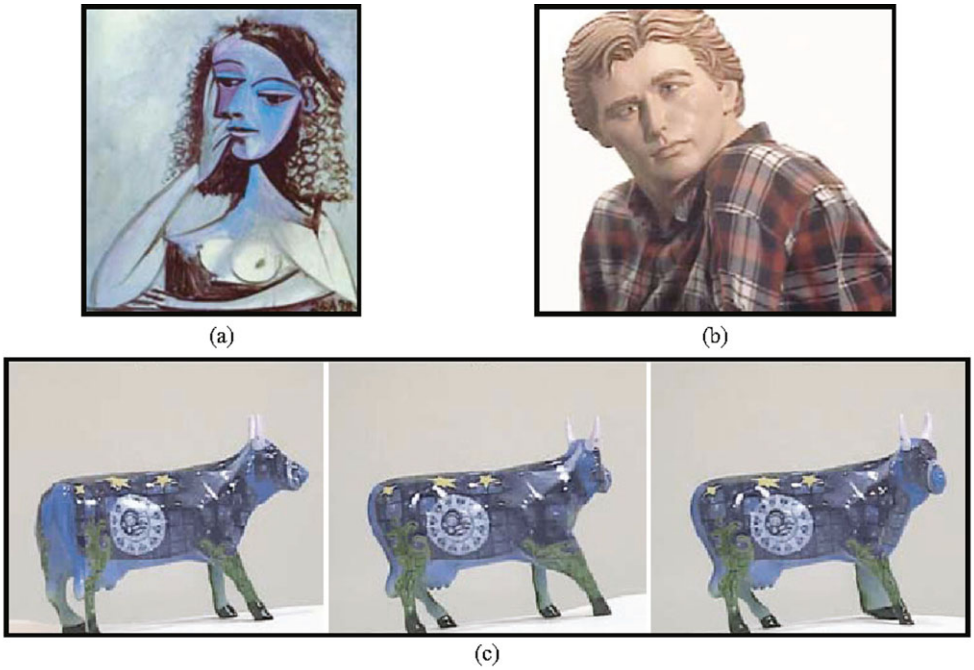


Fig. 12 Non-Perspective Images. (**a**) Nusch Eluard by Pablo Picasso. (**b**) A multi-perspective image rendered using GLC framework [20]. (**c**) Extracted images from a faux-animation generated by [20]. The source images were acquired by rotating a ceramic figure on a turntable. Multi-perspective renderings were used to turn the head and hind quarters of the figure in a fake image-based animation

by elaborately stitching the boundary of multiple pinhole images. The viewing continuity with minimum redundancy is achieved through a sequence of pinhole frustum bending, splitting and merging. The panoramic rendering can then be used in 3D scene exploration, summarization and visualization.

5.2 Non-photorealistic rendering

Rendering perspectives from multiple viewpoints can be combined in ways other than panoramas. By making subtle changes in viewing direction across the imaging plane it is possible to depict more of 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 non-perspective images. Figure 12(a) and (b) compare one of Picasso's paintings with an image synthesized using the GLC framework [20]. Starting from a simple layout, it achieves similar multi-perspective effects. 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. Figure 12(c) show three frames from a synthesized animation, each of which corresponds to a multi-perspective image rendered from a 3D light field. Zomet [60] used a similar approach by using a single XSlit camera to achieve rotation effects.

Mei et al. [27] defined an occlusion camera that can sample visible surfaces as well as occluded ones in the reference view, to allow re-rendering new views with correct occlusions. Their occlusion camera bends the rays towards a central axis (the pole) to sample the hidden surfaces in the reference view. A 3D radial distortion centered at the pole allows the occlusion camera to see around occluders along the pole. Such distortion pulls out hidden samples according to their depth: the larger the depth, the larger the sample will be displaced. Therefore, samples that are on the same ray in a conventional perspective camera are separated to different locations in the distorted occlusion camera image according to their depth. In this way, hidden samples that are close to the silhouette becomes visible in the occlusion camera reference image.

Hsu et al. [18] recently proposed a multi-scale rendering framework that can render objects smoothly at multiple levels of details in a single image. They set up a sequence of pinhole cameras to render objects at different scales of interest and use a user-specified mask to determine regions to be displayed in each view. The final multi-scale image is rendered by reprojecting the images of multi-scale cameras to the one with the largest scale and use Bezier curve-based non-linear ray casting to ensure coherent transition between each scale view. Their technique can achieve focus plus context visualization and is useful in scientific visualization and artistic rendering.

5.3 Stereo and 3D reconstruction

Traditional stereo matching algorithms for pinhole cameras have also been extended to non-pinhole geometry. Seitz [41] and Pajdla [33] independently studied all possible non-pinhole camera pairs that can have epipolar geometry. Their work suggests that only three varieties of epipo-

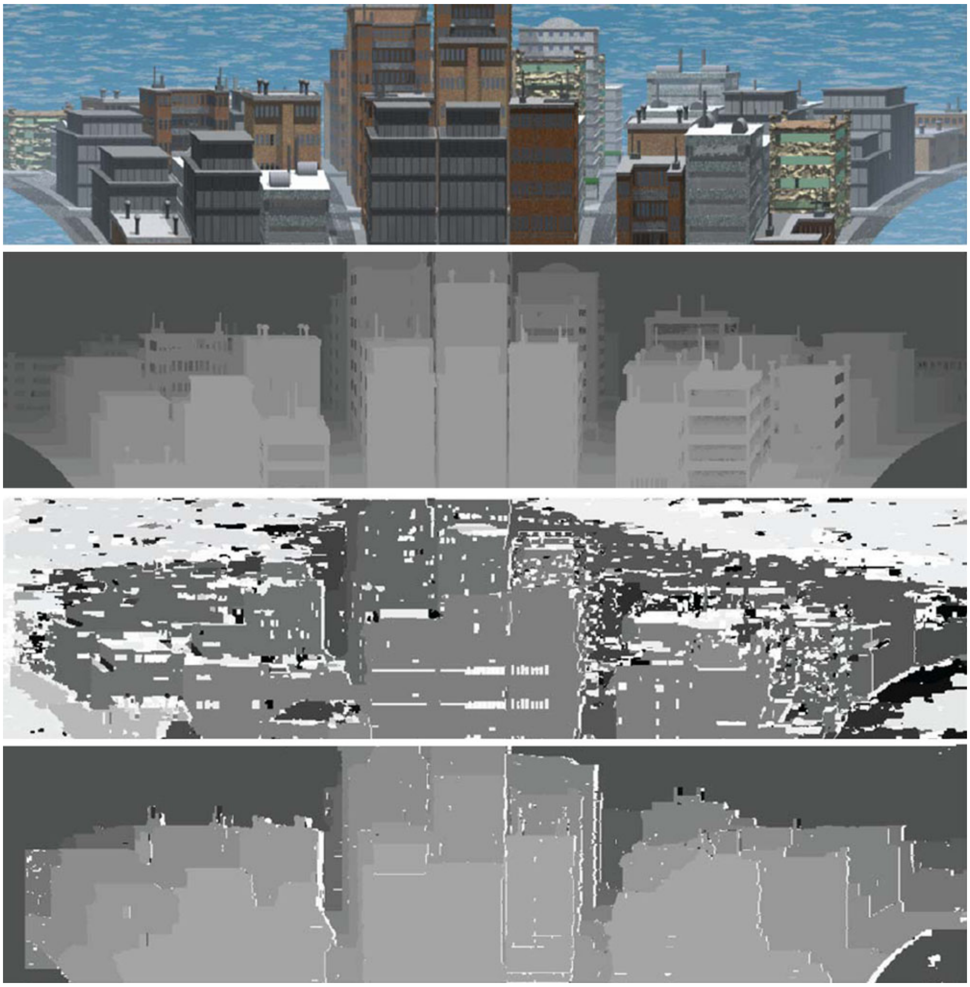
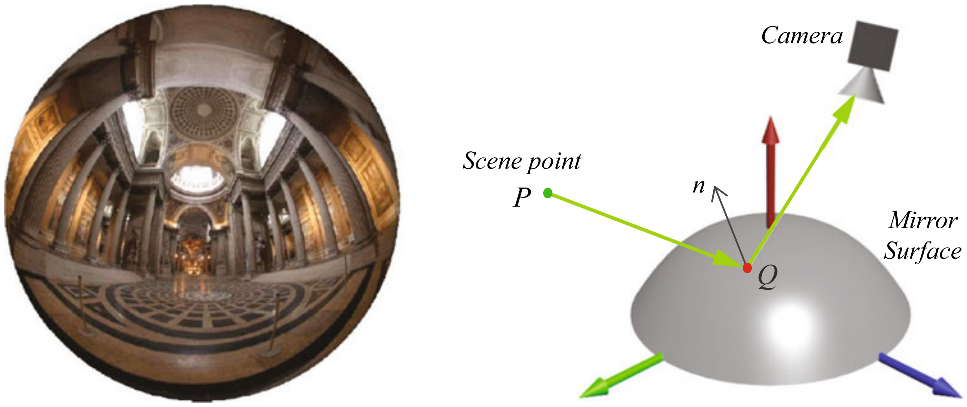


Fig. 13 Epsilon stereo matching on two XSlit cameras. *From top to bottom*: (**a**) shows one of the two XSlit images; (**b**) shows the ground truth depth map; (**c**) shows the recovered disparity map by treating the two images as a stereo pair and applying the graph cut algorithm; (**d**) shows the horizontal disparity map recovered by the epsilon stereo mapping algorithm

lar geometry exist: planes, hyperboloids, and hyperbolic-paraboloids, all corresponding to doubly ruled surfaces. Peleg et al. [34] stitched the same column of images from a rotating pinhole camera to form a circular pushbroom. They then fused two oblique circular pushbrooms to synthesize a stereo panorama. Feldman et al. [14] proved that a pair of XSlit cameras can have valid epipolar geometry if they share a slit or the slits intersect in four pairwise distinct points.

However, Seitz and Pajdla's results also reveal that very few varieties of multi-perspective stereo pairs exist. Ding and Yu [8] introduced a new near stereo model, which they call epsilon stereo pairs. An epsilon stereo pair consists of two non-pinhole images with a slight vertical parallax. They have shown that many non-pinhole camera pairs that do not satisfy the stereo constraint can still form epsilon stereo pairs. They then have introduced a new ray-space warping algorithm to minimize stereo inconsistencies in an epsilon pair using non-pinhole collineations (homograph) which makes epsilon stereo model a promising tool for synthesizing close-to-stereo fusions from many non-stereo pairs, as shown in Fig. 13. Most recently, Kim et al. [21] presented a method for generating near stereoscopic view by cutting the light field. They compute the stereoscopy as the optimal cut through the light field under the depth budget, maximum disparity gradient, and desired stereoscopic baseline.

A special class of non-pinhole cameras are reflective and refractive surfaces. One can then view the surface reconstruction problem as the camera calibration problem. Ding et al. [9, 12] proposed a shape-from-distortion framework for recovering specular (reflective/refractive) surfaces by analyzing the local reflection GLCs and curved line images. In [9], they focused on recovering a special type of surface: near-flat surfaces such as the windows and relatively flat water surfaces. Such surfaces are difficult to model because lower-order surface attributes provide little information. They divide the specular surface into piecewise triangles and estimate each local reflection GLCs for recovering high-order surface properties such as curvatures. In [12], the authors have further shown how to analyze the curving of lines to recover the GLC parameters and then the surface attributes.



(a)

(b)

Fig. 14 (**a**) A typical catadioptric image with wide FoV. (**b**) Forward Projection: Given a scene point P, the mirror surface and the camera, find its projection in the viewing camera after reflection. It is crucial to find the reflection point on the mirror surface Q

6 Real non-pinhole imaging systems

In previous sections, we have discussed many conceptual non-pinhole camera models. In this section, we discuss a number of real non-pinhole cameras that are constructed by modifying a commodity camera using special optical units.

6.1 General catadioptric cameras

As mentioned above, the most commonly used class of "real" non-pinhole/multi-perspective cameras are catadioptric cameras. These cameras put a regular pinhole camera in front of a curved mirror for acquiring images with a much wider FoV, as shown in Fig. 14(a). A large FoV can benefit many applications such as video surveillance, autonomous navigation, obstacle avoidance and panoramic image acquisition.

The core problem in catadioptric cameras is to solve for forward projection, i.e., given the view camera (a pinhole), the curved mirror, and a 3D scene point, how to find the projection of the point in the view camera, as shown in Fig. 14(b). To resolve this problem, it is crucial to find the reflection point on the mirror in order to trace the ray path from the scene point to the CoP of the pinhole camera. This is a classical inverse problem and for complex catadioptric systems with multiple viewpoints, closed-form solution does not exist. We review recent attempts to address the forward projection problem using ray geometry analysis.

Centric catadioptric cameras The simplest catadioptric cameras are designed to maintain a single viewpoint, i.e., all the projection rays intersect at one common point (the effective viewpoint), in order to generate perspectively correct images from sections of the acquired image. Such systems are commonly referred to *centric* catadioptric cameras. Since all projection rays from scene points form a same pinhole camera about the effective viewpoint before reflection, we can easily resolve the forward projection problem by projecting the 3D point in the virtual pinhole camera.

Nayar and Baker [5, 30] analyzed all possible classes of centric catadioptric systems. They derived a fixed viewpoint constraint that requires all projection rays passing through the effective pinhole of the camera (after reflection) would have passed through the effective viewpoint before reflected by the mirror surface. Since the mirror is rotationally symmetric, one can then consider this problem in 2D by taking a slice across the central axis. Assuming that the effective viewpoint is at origin $[0,0]$; the effective pinhole is at $[0,c]$ and the mirror surface is of form $z(r)=z(x,y)$, where $r=\sqrt{x^2+y^2}$, the constraint can then be written as a quadratic first-order ordinary differential equation:

$$r(c-2z)\left(\frac{dz}{dr}\right)^2-2\left(r^2+cz-z^2\right)\frac{dz}{dr}+r(2z-c)=0 \quad (29)$$

The solution to Eq. (29) reveals that only 3D mirrors swept by conic sections around its central axis can satisfy the fixed viewpoint constraint, therefore, maintaining a single viewpoint. They have further shown two practical setups of centric catadioptric cameras: (1) positioning a pinhole camera at the focal point of a hyperboloidal mirror; and (2) orienting an orthographic camera (realized by using a tele-lens) towards the rotational axis of a paraboloidal mirror. Both designs, however, require highly accurate alignment and precise assembly of the optical components.

Non-centric catadioptric cameras Relaxing the single viewpoint constraint allows more general but *non-centric* catadioptric cameras. In a non-centric catadioptric camera, the loci of virtual viewpoints form the caustic surfaces of the mirror. The centric catadioptric camera is a special case with its caustic being a point. Swaminathan et al. [44–46] proposed to use the envelop of these reflection rays for computing the caustic surface. Yu and McMillan [56] instead decompose the mirror surface into piecewise triangle patches and model each reflection patch as a GLC, as shown in Sect. 3.2.2. Recall that local reflection ray geometry observed by a pinhole or an orthographic camera can only

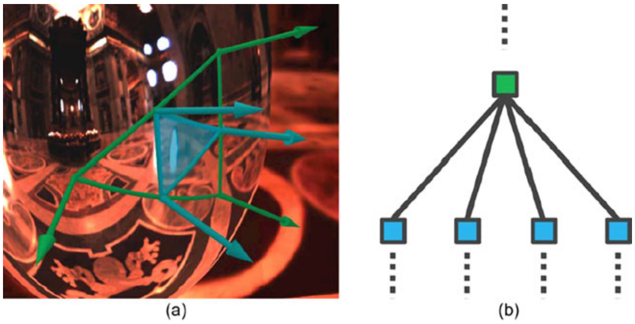


Fig. 15 Solving for forward projection using GLC decomposition. (**a**) We can decompose a curved mirror image using piecewise GLCs. (**b**) A multi-resolution hierarchy can be created for querying the image of a 3D point

```
Algorithm 6.1: MIRROR_GLCFORWARDPROJECTION($glc$, $\dot{P}$)

procedure GETRAY(const GLC &$glc$, const Point3D &$\dot{P}$)
  $p[u, v] \leftarrow glc$.Project($\dot{P}$);
  if $p[u, v] \notin glc.triangle$
    then {return (false)
  if isLeaf($glc$)
    then {return ($p[u, v]$)
          bNotFind $\leftarrow$ true;
          while bNotFind
               $x \leftarrow glc.subGLCs$.getNext();
            do $q[u_1, v_1] \leftarrow x$.Project($\dot{P}$);
    else       if $q[u_1, v_1] \in x.triangle$
                 then {bNotFind $\leftarrow$ false;
          return (GetRay($x$, $\dot{P}$))
```

be one of the four types of GLC: XSlit, pushbroom, pinhole, or orthographic, all can be viewed as special cases of XSlit cameras: when the two slits intersect, it transforms into a pinhole camera; when one of the slits goes to infinity, the XSlit transforms into a pushbroom; and when both slits go to infinity, it transforms into an orthographic camera.

6.2 Solutions to the forward projection problem

GLC approximation The key advantage of using this GLC approximation is that it provides a closed-form solution to the forward projection problem: one can decompose the mirror into piecewise GLCs, project the 3D point into each GLC, and verify if the projection location lies inside the GLC [57]. The result apparently is an approximation to the real solution and the accuracy depends on the fineness of triangulation. To improve both efficiency and accuracy, they have further developed a dynamic tessellation scheme similar to the Level-of-Detail (LoD) technique in computer graphics. They first tessellate the reflection surface using a coarse set of GLCs and then perform standard 1-to-4 subdivision and store the subdivision in a quad tree as shown in Fig. 15. To forward project a 3D point $\dot{P}$ to the camera, they start from the top level GLCs and compute the image of $\dot{P}$'s projection. They then determine which GLC contains the final projection and repeat the search on its children GLCs. The search stops until it reaches the leaf nodes. The detailed forward projection steps are shown in Algorithm 6.1.

Axial cameras The forward projection problem can also be addressed using special catadioptric cameras such as the axial camera. The axial camera is an intermediate class of cameras that lies between centric and non-centric ones. In an axial camera, all the projection rays are constrained to pass through a common axis but not a 3D point. One such model is a rotationally symmetric mirror with a pinhole camera viewing from its rotation axis, as shown in Fig. 16(a).

Axial cameras are easier to construct than the centric catadioptric ones. For example, in a centric hyperbolic catadioptric camera, the optical center of the view camera has to be placed precisely at the mirror's foci whereas in an axial camera the optical center can be placed anywhere on the mirror axis to satisfy the axial geometry. The fact that all reflection rays passing through the rotation axis reveals that local GLCs will map all reflection patches to a group of XSlit cameras that share a common slit, i.e., the rotation axis. Ramalingam et al. [38] proposed a generic calibration algorithm for axial cameras by computing projection rays for each pixel constrained by the mirror axis. Agrawal et al. [4] further provided an analytical solution for forward projection for axial cameras. Given the viewpoint and a mirror, they compute the light path from a scene point to the viewing camera by solving a closed-form high-order forward projection equation. Conceptually, this can be done by exhaustively computing the projection for each centric ring of the virtual camera. For spherical mirror, they derived that the projection equation is reduced to 4th degree. This closed-form solution can be used to effectively compute the epipolar geometry to accelerate catadioptric stereo matching and to compose multiple axial camera images for forming a perspective one [47].

Another special class of axial cameras is the radial camera proposed by Kuthirummal and Nayar [22]. Their goal is to strategically capture the scene from multiple viewpoints within a single image. A radial camera consists of a conventional camera looking through a hollow rotationally symmetric mirror polished on the inside, as shown in Fig. 16(b). The FoV of the camera is folded inwards and consequently the scene is captured both directly and from virtual viewpoints after reflection by the mirror, as shown in Fig. 16(c). 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 special axial camera that has a circular locus of virtual viewpoints. Similar to the regular axial camera, closed-form solution can be derived for computing the forward projection. Further, this camera has the same epipolar

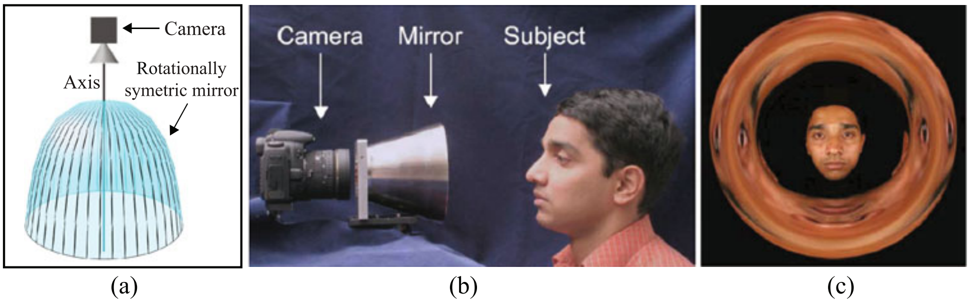


Fig. 16 Two Examples of Axial Camera. (**a**) Rotationally symmetric mirror with a viewing camera lying on the rotation axis. (**b**) A radial catadioptric camera can capture the same 3D point from different perspectives in a single image. (**c**) A multi-perspective image captured by (**b**) (courtesy of Shree Nayar)

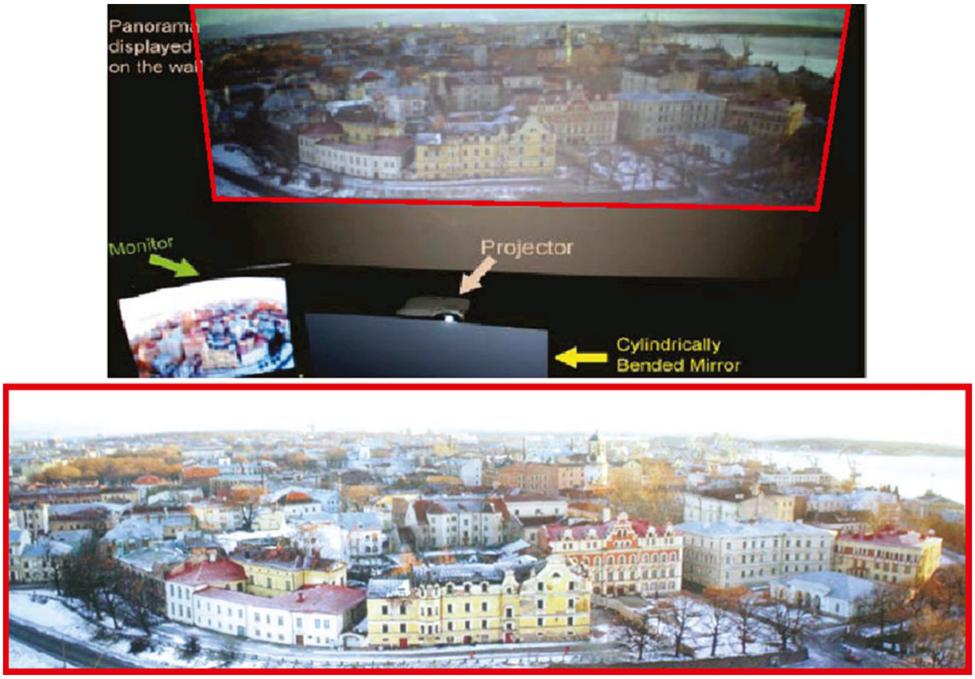


Fig. 17 *Top*: A panoramic catadioptric projector system that combines a regular projector with curved plastic mirror. *Bottom*: The final projection uses the projector's full resolution (1024 × 768) and displayed on a 16 m × 3 m wall

geometry as the cyclographs [41] and therefore can be effectively used for omni-directional 3D reconstruction, acquiring 3D textures, sampling and estimating the surface reflectance properties such as the Bidirectional Reflectance Distribution Functions (BRDF).

6.3 Catadioptric projectors

Finally, one can replace the viewing pinhole camera with a projector. Ding et al. [10] proposed the catadioptric projector by combining a digital commodity projector with specially shaped reflectors to achieve an unprecedented level of flexibility in aspect ratio, size, and FoV, as shown in Fig. 17. Their system assumes unknown reflector geometry and does not require accurate alignment between the projector and the optical units. They then use the inverse light transport technique to correct geometric distortions and scattering.

The main difference between the catadioptric camera and catadioptric projector is that the camera uses a near-zero aperture whereas the projector requires a wide aperture to achieve bright projections. However, the wide aperture may cause severe defocus blurs. Due to the non-pinhole nature of reflection rays, the defocus blurs are much more complicated, e.g., the blur kernels are spatial-varying and non-circular shaped. Therefore, traditional image preconditioning algorithms are not directly applicable.

The analysis in Sect. 4.3 shows that the catadioptric defocus blur can range from an ellipse to a line segment, depending on the aperture setting and the projector focal depth. To compensate for defocus blurs, Ding et al. [10] adopt a hardware solution: they change the shape of the aperture to reduce the average size of the defocus blur kernel. Conceptually, one can use a very small aperture to emulate pinhole-type projection. However, small apertures block a large amount of light and produce dark projections. Their solution is then to find the appropriate aperture shape that can effectively reduce the blurs without sacrificing the brightness in projection. In their approach, they first estimate the blur kernel by projecting a dotted pattern onto the wall and fit an ellipse to each captured dot. They then compute the average major and minor radii across all dots as a' and b'. Using the analysis in Sect. 4.3, they prove that the major and

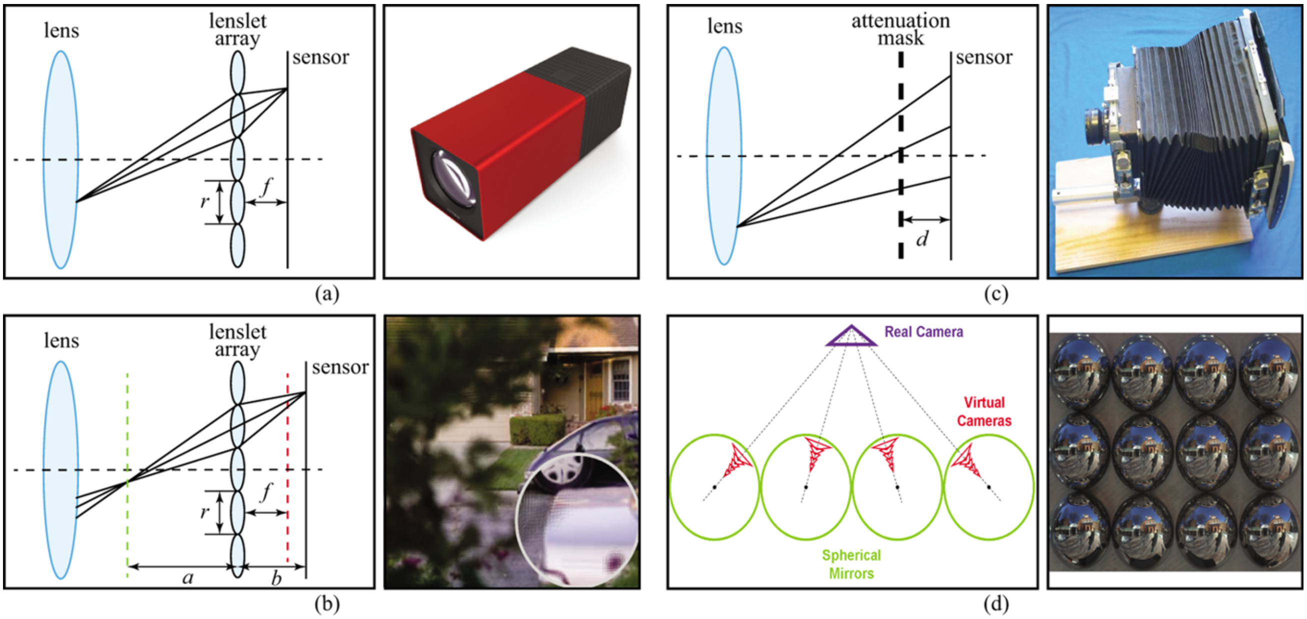


Fig. 18 Different light field camera designs. Lenslet-based light field camera places the lenslet array on the image plane of the main lens. (**a**) In Lytro [25], the sensor is located at the focal plane of the microlenses. (**b**) In Lumsdaine et al. [24], the microlenses focus on the sensor to trade angular resolution for spatial resolution. (**c**) The heterodyne light field camera puts a narrowband 2D cosine mask near the sensor (courtesy of Ramesh Raskar). (**d**) Catadioptric light field camera uses a view camera facing an array of mirrors (courtesy of Yuichi Taguchi)

minor radii a and b of the optimal aperture should produce a circular shaped defocus kernel. They have shown that the optimal aperture should be an ellipse with $a = \frac{D}{2}\sqrt{\frac{a'}{b'}}$ and $b = \frac{D}{2}\sqrt{\frac{b'}{a'}}$, where D is the diameter of the actual aperture.

6.4 4D ray sampler: light field cameras

The most general non-pinhole should be able to sample the complete 4D ray space and then reconfigure the rays at will. This requires using the generalized optics that treats each optical element as a 4D ray-bender that modifies the rays in a light field [2, 24, 31]. The collected ray bundles can then be regrouped into separate measurements of the plenoptic function [1, 26]. The most straightforward scheme is to move a camera along a 2D path to sample the 4D ray space [19, 23]. Although this method is simple and easy to implement, it is only suitable for acquiring static scenes. Wilburn et al. [52] instead built a camera array to capture the light field. Constructing such a light field camera array, however, is extremely time and effort consuming and requires substantial amount of engineering. The latest developments are the light field cameras.

Lenslet based light field camera Recent advances in optics manufacturing has enabled the light field to be captured using a single camera in one shot. Ng [31] designed a handheld plenoptic camera to record the light field within a single shot by placing a lenslet array in front of the camera sensor to separate converging rays. Each microlens focuses at the main aperture plane. Since the size of the main lens is several magnitude larger than the lenslet, it can be treated as infinity to the lenslet. The sensor is placed at the focal plane of the lenslet array for simplification. In Ng's design, the F-numbers of the main lens and each microlens are matched to avoid cross-talk among microlens images. By parameterizing the in-lens light field with a 2PP of Π_{uv} at the main aperture and Π_{st} at the lenslet array, the acquired ray space is uniformly sampled. This design has led to the commercial light field camera, Lytro [25], as shown in Fig. 18(a).

Lumsdaine et al. [24] introduced a slightly different design by focusing the lenslet array on a virtual plane inside camera. In this case each microlens image will capture more spatial samples but less angular samples on the focused virtual plane. This design is capable of producing higher resolution results when focusing near the sampled image plane. However, the lower angular resolution leads to more severe ringing artifacts at the out-of-focus regions, as shown in Fig. 18(b).

Mask based light field camera Instead of using a lenslet array to separate light arriving at the same pixel from different directions, Veeraraghavan et al. [49] used a non-refractive patterned attenuation mask to modulate the light field in the frequency domain. By placing the mask on the light path between the lens and sensor, it attenuates light from different directions accordingly, as shown in Fig. 18(c). Considering the process in the frequency domain, we can view it as heterodyning the incoming light field. The attenuation mask needs to be reversible to ensure that demodulation can be performed. To recover the light field, they first transform the captured 2D image into Fourier domain and then rearrange the tiles of the 2D Fourier transform into 4D space. Finally, the light field of the scene is computed by taking the inverse 4D Fourier transform. Further, they can insert the mask at different location along the optical path of the camera to achieve dynamic frequency modulation. However, the mask partially blocks out the incoming light and greatly reduces light efficiency.

Mirror based light field camera It is also possible to acquire the light field using a catadioptric mirror array, as shown in Fig. 18(d). Unger et al. [48] combined a high resolution tele-lens camera and an array of spherical mirrors to capture the incident light field. The use of mirror arrays instead of lenslet arrays has its advantages: it avoids chromatic aberrations caused by refraction, it does not require elaborate calibration between the lenslet array and the sensor, it captures images at a wide FoV, and it is less expensive and reconfigurable. The disadvantages are two-fold. First, each mirror image is non-pinhole and therefore requires conducting forward projection for associating the reflection rays with 3D points. Second, the sampled the light field is non-uniform.

Two notable examples of these systems are the spherical mirror arrays by Ding et al. [11] and Taguchi et al. [47]. In [11], the authors applied the GLC-based forward projection (Sect. 6.2) on multi-view space carving for reconstructing the 3D scene. Taguchi et al. [47] developed both a mirror array and a refractive sphere array and applied the axial camera formulation (Sect. 6.2) to compute the closed forward projection. They have shown various applications including distortion correction and light field rendering.

Light field probes Analogous to catadioptric cameras vs. catadioptric projectors, the duality to the light field camera is the light field probe, i.e., replacing the sensor with a projector. The real light field probe has been implemented by using a backlight, diffuser, pattern, and a single or array of lenslet, as shown in Fig. 19. Similar to Lytro, the pattern is placed at the focal plane of the lenslet array to simulate an array of projectors projecting towards infinity. The light field probe is apparently a multi-view display. It is also particularly useful for acquiring transparent surfaces. Notice that the light field probe can directly estimate ray–ray correspondences as the view camera can associates each pixel to a ray. Ye et al. [54] used a single lens probe (a Bokode [29]) for recovering dynamic fluid surfaces. They presented a robust feature matching algorithm based on the Active Appearance Model (AAM) to robustly establishing ray–ray correspondences. The ray–ray correspondences then directly provide the surface normal and they derive a new angular-domain surface integration scheme to recover the surface from the normal field. Wetzstein et al. [50, 51] also used the light field probe for reconstructing complex transparent objects. They encode both spatial and angular by using specially designed color pattern. Specifically, they use gradients of different color channels (red and blue) to encode the 2D incident ray direction and the green channel to encode the 1D (vertical) spatial location of the pattern. The second (horizontal) spatial location can be recovered through geometric constraints. Their approach is able to achieve highly accurate ray–ray correspondences for reconstructing surface normals of complex static objects.

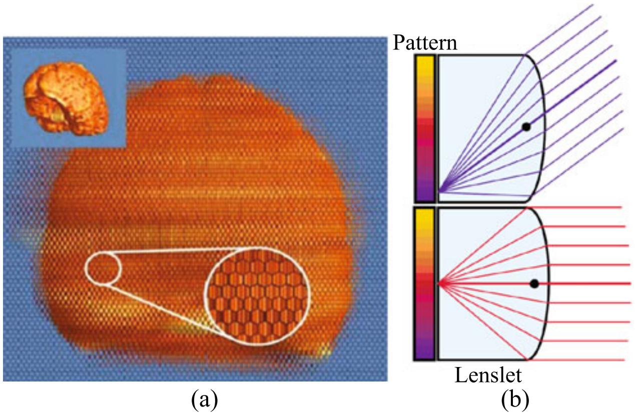


Fig. 19 Light Field Probes. (**a**) A light field probe combines a lenslet array and a special projection pattern. (**b**) Similar to Lytro, each lenslet acts as a view-dependent pixel

7 Future directions

There are several directions for future research related to non-pinhole cameras.

The ray geometry theory may lead to new acquisition devices for many image-based rendering (IBR) and computational photography applications. For example, it will be useful to design specially curved mirrors that can efficiently capture the light field. The pinhole-based and mirror-based light field cameras have provided one way to sample the ray space. The current spherical mirror array suffers from artifacts such as non-uniform sampling and image distortions. Alternatively, special-shaped mirrors may be able to more evenly sample the ray space, e.g., via a different type of ray subspace (e.g., using the GLC-type mirrors).

In addition to non-pinhole cameras, one can potentially develop non-pinhole light sources by replacing the viewing camera in a catadioptric camera with a point light source.

Previous image-based relighting, surface reflectance sampling, and shape recovering algorithms are restricted by geometric constraints of the light source. By strategically devising a different type of lighting condition, one can improve the way for measuring and sampling the radiance off the surface. In addition, a non-pinhole light source will cast special-shaped shadows. In particular, the shadow of a 3D line segment can be a curve under a non-pinhole light source. This may lead to the development of new shape-from-shadow algorithms which determine the depth of the object by analyzing the shape of the shadow at the silhouettes.

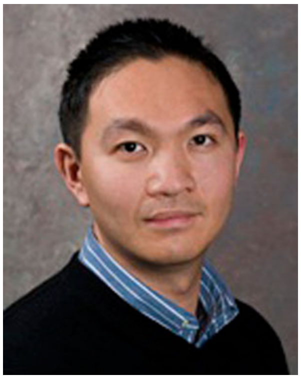
Finally, it is possible to develop a new theoretical framework based on computational and differential geometry to characterize and catalog the structures of ray space. For example, it can be highly useful to model the algebraic ray subspaces (e.g., ray simplices) and analyze how ray geometries are related to specific types of non-pinhole distortions. Further, by correlating the geometric attributes of the reflector/refractor surface with these distortions, we can explore novel shape-from-caustics, shape-from-distortion, or depth-from-defocus algorithms for recovering highly complex specular surfaces.

References

1. Adelson, E.H., Bergen, J.R.: The plenoptic function and the elements of early vision. In: Computational Models of Visual Processing, pp. 3–20 (1991)
2. Adelson, E., Wang, J.: Single lens stereo with a plenoptic camera. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 99–106 (1992)
3. Agarwala, A., Agrawala, M., Cohen, M., Salesin, D., Szeliski, R.: Photographing long scenes with multi-viewpoint panoramas. In: ACM SIGGRAPH, pp. 853–861 (2006)
4. Agrawal, A., Taguchi, Y., Ramalingam, S.: Analytical forward projection for axial non-central dioptric and catadioptric cameras. In: Proceedings of the 11th European Conference on Computer Vision, pp. 129–143 (2010)
5. Baker, S., Nayar, S.K.: A theory of single-viewpoint catadioptric image formation. Int. J. Comput. Vis. **35**(2), 1–22 (1999)
6. Buehler, C., Bosse, M., McMillan, L., Gortler, S., Cohen, M.: Unstructured lumigraph rendering. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’01, pp. 425–432 (2001)
7. Cohen-Steiner, D., Morvan, J.-M.: Restricted Delaunay triangulations and normal cycle. In: Proceedings of the Nineteenth Annual Symposium on Computational Geometry, pp. 312–321 (2003)
8. Ding, Y., Yu, J.: Epsilon stereo pairs. In: Proceedings of the British Machine Vision Conference, September, pp. 10–13 (2007)
9. Ding, Y., Yu, J.: Recovering shape characteristics on near-flat specular surfaces. In: Computer Vision and Pattern Recognition, June (2008)
10. Ding, Y., Xiao, J., Tan, K.-H., Yu, J.: Catadioptric projectors. In: Computer Vision and Pattern Recognition, June, pp. 2528–2535 (2009)
11. Ding, Y., Yu, J., Sturm, P.: Multiperspective stereo matching and volumetric reconstruction. In: Proceedings. 12th IEEE International Conference on Computer Vision, Oct., pp. 1827–1834 (2009)
12. Ding, Y., Yu, J., Sturm, P.: Recovering specular surfaces using curved line images. In: Computer Vision and Pattern Recognition, June, pp. 2326–2333 (2009)
13. Ding, Y., Xiao, J., Yu, J.: A theory of multi-perspective defocusing. In: Computer Vision and Pattern Recognition, June (2011)
14. Feldman, D., Pajdla, T., Weinshall, D.: On the epipolar geometry of the crossed-slits projection. In: Proceedings. Ninth IEEE International Conference on Computer Vision, Oct. (2003)
15. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The lumigraph. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’96, pp. 43–54 (1996)
16. Gupta, R., Hartley, R.I.: Linear pushbroom cameras. IEEE Trans. Pattern Anal. Mach. Intell. **19**(9), 963–975 (1997)
17. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2003)
18. Hsu, W.-H., Ma, K.-L., Correa, C.: A rendering framework for multiscale views of 3d models. In: Proceedings of the SIGGRAPH Asia Conference, pp. 131:1–131:10 (2011)
19. Isaksen, A., McMillan, L., Gortler, S.J.: Dynamically reparameterized light fields. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, pp. 297–306 (2000)
20. Jingyi, Y., Leonard, M.: A framework for multiperspective rendering. In: Proceedings of Rendering Techniques, Eurographics Symposium on Rendering (2004)
21. Kim, C., Hornung, A., Heinzle, S., Matusik, W., Gross, M.: Multi-perspective stereoscopy from light fields. ACM Trans. Graph. **30**(6), 190:1–190:10 (2011)
22. Kuthirummal, S., Nayar, S.K.: Multiview radial catadioptric imaging for scene capture. In: ACM SIGGRAPH 2006, pp. 916–923 (2006)
23. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’96, pp. 31–42 (1996)
24. Lumsdaine, A., Georgiev, T.: The focused plenoptic camera. In: Proceedings. IEEE International Conference on Computational Photography, pp. 1–8 (2009)
25. Lytro. www.lytro.com
26. McMillan, L., Bishop, G.: Plenoptic modeling: an image-based rendering system. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, pp. 39–46 (1995)
27. Mei, C., Popescu, V., Sacks, E.: The occlusion camera. Comput. Graph. Forum **24**, 335–342 (2005)
28. Meyer, M., Desbrun, M., Schröder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. In: Proc. Visisualization and Mathematics, pp. 35–57 (2002)
29. Mohan, A., Woo, G., Hiura, S., Smithwick, Q., Raskar, R.: Bokode: imperceptible visual tags for camera based interaction from a distance. In: ACM SIGGRAPH (2009)
30. Nayar, S.: Catadioptric omnidirectional camera. In: Computer Vision and Pattern Recognition, June, pp. 482–488 (1997)
31. Ng, R.: Fourier slice photography. In: ACM SIGGRAPH 2005 Papers, pp. 735–744 (2005)
32. Pajdla, T.: Stereo with oblique cameras. In: IEEE Workshop on Stereo and Multi-Baseline Vision, pp. 85–91 (2001)
33. Pajdla, T.: Geometry of two-slit camera. Research Report CTU-CMP-2002-02
34. Peleg, S., Ben-Ezra, M.: Stereo panorama with a single camera. In: Computer Vision and Pattern Recognition (1999)
35. Peleg, S., Rousso, B., Rav-Acha, A., Zomet, A.: Mosaicing on adaptive manifolds. IEEE Trans. Pattern Anal. Mach. Intell. **22**(10), 1144–1154 (2000)
36. W.D. Productions: Pinocchio, 1940. Movie
37. Popescu, V., Rosen, P., Adamo-Villani, N.: The graph camera. In: ACM SIGGRAPH Asia (2009)

38. Ramalingam, S., Sturm, P., Lodha, S.K.: Theory and calibration for axial cameras. In: Asian Conference on Computer Vision, vol. 1, pp. 704–713 (2006)
39. Raskar, R., Tumblin, J., Mohan, A., Agrawal, A., Li, Y.: Computational photography, pp. 1–20. Eurographics Association (2006)
40. Rucker, R.: The Fourth Dimension: Toward a Geometry of Higher Reality. Houghton Mifflin, Boston (1984)
41. Seitz, S.M., Kim, J.: The space of all stereo images. Int. J. Comput. Vis. **48**(1), 21–38 (2002)
42. Shum, H.-Y., Szeliski, R.: Construction of panoramic image mosaics with global and local alignment. Int. J. Comput. Vis. **48**, 151–152 (2002)
43. Soler, C., Subr, K., Durand, F., Holzschuch, N., Sillion, F.: Fourier depth of field. ACM Trans. Graph. **28**, 1–12 (2009)
44. Swaminathan, R., Grossberg, M., Nayar, S.: Caustics of catadioptric cameras. In: Proceedings. Eighth IEEE International Conference on Computer Vision, vol. 2, pp. 2–9 (2001)
45. Swaminathan, R., Grossberg, M.D., Nayar, S.K.: A perspective on distortions. In: Computer Vision and Pattern Recognition, pp. 594–601 (2003)
46. Swaminathan, R., Grossberg, M.D., Nayar, S.K.: Non-single viewpoint catadioptric cameras: geometry and analysis. Int. J. Comput. Vis. **66**(3), 211–229 (2006)
47. Taguchi, Y., Agrawal, A., Veeraraghavan, A., Ramalingam, S., Raskar, R.: Axial-cones: modeling spherical catadioptric cameras for wide-angle light field rendering. In: ACM SIGGRAPH Asia, pp. 172:1–172:8 (2010)
48. Unger, J., Wenger, A., Hawkins, T., Gardner, A., Debevec, P.: Capturing and rendering with incident light fields. In: EGRW, pp. 141–149 (2003)
49. Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A., Tumblin, J.: Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. In: ACM SIGGRAPH (2007)
50. Wetzstein, G., Raskar, R., Heidrich, W.: Hand-held schlieren photography with light field probes. In: Proceedings. IEEE International Conference on Computational Photography, April, pp. 1–8 (2011)
51. Wetzstein, G., Roodnick, D., Raskar, R., Heidrich, W.: Refractive shape from light field distortion. In: Proceedings. 13th IEEE International Conference on Computer Vision (2011)
52. Wilburn, B., Joshi, N., Vaish, V., Talvala, E.-V., Antunez, E., Barth, A., Adams, A., Horowitz, M., Levoy, M.: High performance imaging using large camera arrays. In: ACM SIGGRAPH, pp. 765–776 (2005)
53. Wood, D.N., Finkelstein, A., Hughes, J.F., Thayer, C.E., Salesin, D.H.: Multiperspective panoramas for cel animation. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, pp. 243–250 (1997)
54. Ye, J., Ji, Y., Li, F., Yu, J.: Angular domain reconstruction of dynamic 3d fluid surfaces. In: Computer Vision and Pattern Recognition (2012)
55. Yu, J., McMillan, L.: General linear cameras. In: ECCV (2004)
56. Yu, J., McMillan, L.: Modelling reflections via multiperspective imaging. In: Computer Vision and Pattern Recognition, pp. 117–124 (2005)
57. Yu, J., McMillan, L.: Multiperspective projection and collineation. In: Proceedings. 10th IEEE International Conference on Computer Vision (2005)
58. Yu, J., Yin, X., Gu, X., McMillan, L., Gortler, S.: Focal surfaces of discrete geometry. In: Proceedings of the Fifth Eurographics Symposium on Geometry Processing, pp. 23–32 (2007)
59. Yu, J., McMillan, L., Sturm, P.: Multi-perspective modelling, rendering and imaging. Comput. Graph. Forum **29**(1), 227–246 (2010)
60. Zomet, A., Feldman, D., Peleg, S., Weinshall, D.: Mosaicing new views: the crossed-slits projection. IEEE Trans. Pattern Anal. Mach. Intell. **25**(6), 741–754 (2003)

Jinwei Ye received her B.E. degree from the Department of Electrical Engineering, Huazhong University of Science and Technology in 2009. She is now a Ph.D. student at the Department of Computer and Information Sciences, University of Delaware. Her research interests include computational photography, and ray geometry.



Jingyi Yu is an associate professor at Computer and Information Science Department at the University of Delaware. He received his B.S. from Caltech in 2000 and M.S. and Ph.D. degree in EECS from MIT in 2005. His research interests span a range of topics in computer graphics, computer vision, and image processing, including computational photography, medical imaging, nonconventional optics and camera design, tracking and surveillance, and graphics hardware.