Eurographics Symposium on Rendering (2004)
A. Keller, H. W. Jensen (Editors)

A Framework for Multiperspective Rendering

J. Yu[1,2] and L. McMillan[2]

[1]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology
[2]Department of Computer Science, The University of North Carolina at Chapel Hill

Abstract

We present a framework for the direct rendering of multiperspective images. We treat multiperspective imaging systems as devices for capturing smoothly varying set of rays, and we show that under an appropriate parametrization, multiperspective images can be characterized as continuous manifolds in ray space. We use a recently introduced class of General Linear Cameras (GLC), which describe all 2D linear subspaces of rays, as primitives for constructing multiperspective images. We show GLCs when constrained by an appropriate set of rules, can be laid out to tile the image plane and, hence, generate arbitrary multiperspective renderings. Our framework can easily render a broad class of multiperspective images, such as multiperspective panoramas, neocubist style renderings, and faux-animations from still-life scenes. We also show a method to minimize distortions in multiperspective images by uniformly sampling rays on a sampling plane even when they do not share a common origin.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.6 [Computer Graphics]: Methodologies and Techniques

1. Introduction

A perspective rendering represents the spatial relationships of objects in a scene as they would appear from a single viewpoint. In contrast, a multiperspective rendering combines what is seen from several viewpoints into a single image. Despite their incongruity of view, effective multiperspective images are still able to preserve spatial coherence. More importantly multiperspective 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.

Multiperspective rendering techniques are frequently employed by artists to depict more than can be seen from any specific point. Classic examples include the visual paradoxes of Escher, and the Cubism of Picasso and Braque. Multiperspective images have also been used as backdrops in cel animation to effectively depict camera motion in a single panorama [WFH*97]. Multiperspective images have also received attention from the computer vision community for analyzing structure revealed via motion [Sei01, PBEP01]. Finally, in our everyday lives, we experience multiperspective images when observing reflections off of curved surfaces. However, only perspective rendering is widely supported in three-dimensional computer graphics. This is partly due to the inherent difficulty of specifying the projection process necessary for mapping 3 dimensions down to 2 for multiperspective renderings.

Figure 1: *A neocubism styled multiperspective image synthesized using our framework.*

In this paper, we present a framework for the direct rendering of multiperspective computer graphics images. Moreover, the images produced by our method are coherent. Specifically, they are C_0 continuous within a space of rays. Our methods are enabled by a recently introduced class of General Linear Camera (GLC) models, which are, in a sense, atoms for the constructing multiperspective images [YM04]. These atomic cameras, when constrained by an appropriate set of rules, can be laid out to tile the image plane, thereby generating arbitrary multiperspective renderings. In fact, to a first order, our framework can describe any multiperspective image that is consistent with a smoothly varying set of rays. The contributions of our multiperspective rendering framework are:

- A piecewise linear approximation of any multiperspective image using a set of 8 atomic GLC camera models.
- A set of rules for tiling the image plane with GLC models.
- Techniques for inferring intermediate GLC cameras for transitions between specific GLCs (in overlapping regions).
- A method for uniformly sampling rays in the image plane, even when they do not share a common origin.

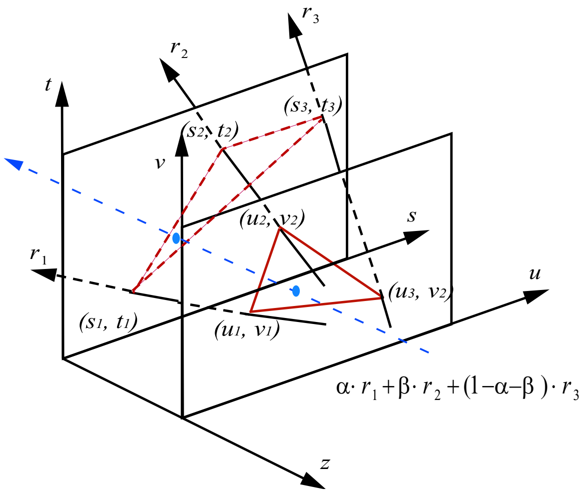

Figure 2: *A General Linear Camera Model collects radiance along all possible affine combination of three rays. The rays are parameterized by their intersections with 2 parallel planes.*

2. Previous Work

Computer generated multiperspective 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. The concentric mosaics of [SH99] and [PBEP01] are another type of multiperspective image that is useful for exploring captured environments. Durand [Dur02] suggests that multiperspective projections can also be an interactive process and uses them as an example to distinguish between picture generation and user interaction.

Agrawala et al [AZM00] developed a system for rendering multiperspective images from three-dimensional models based on spatially varying projections. They point out that the most difficult part of multiperspective rendering using spatially varying projections is resolving occlusion. This difficulty can be relaxed if the transition between projections vary smoothly. Hanson and Wernert [HW98] addressed the occlusion problem by densely interpolating the path of the ray.

Multiperspective camera models have also been explored in the computer vision community. Common examples include cross-slit [ZFPW03] and pushbroom cameras [GH97]. [ZFPW03] have shown how to generate interesting multiperspective images and animations by slicing video sequences. Multiperspective images have also been analyzed for there potential for exhibiting coherent parallax [Sei01], which is necessary for stereo fusion and analyzing 3D structure.

The MCOP (multiple center of projection) images of Rademacher [RB98] are another example of unstructured multiperspective images. They are closely related to images generated 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. Glassner [Gla00] described a camera construction for collecting rays along the camera path. In his approach, rays are specified by the two surfaces with common parameterizations. Our approach has similarities, but we can guarantee continuity and uniqueness properties, as well as provide intuitive user controls.

Our methods focus on the direct rendering of multiperspective images from either image-based or traditional 3D models. We treat the problem of multiperspective rendering as one of specifying and sampling a smooth space of rays.

3. The General Linear Camera Model

Recently a new camera model has been developed called the General Linear Camera (GLC) [YM04]. This single model describes typical pinhole and orthographic cameras, as well as many commonly studied multiperspective cameras including push-broom and cross-slit cameras. GLCs also include many lesser known multiperspective cameras, such as the pencil, twisted orthographic, EPI, and bilinear cameras, as is shown in Figure 3.

A General Linear Camera (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 Π_{image}, as is shown in Figure 2. A GLC collects radiance measurements

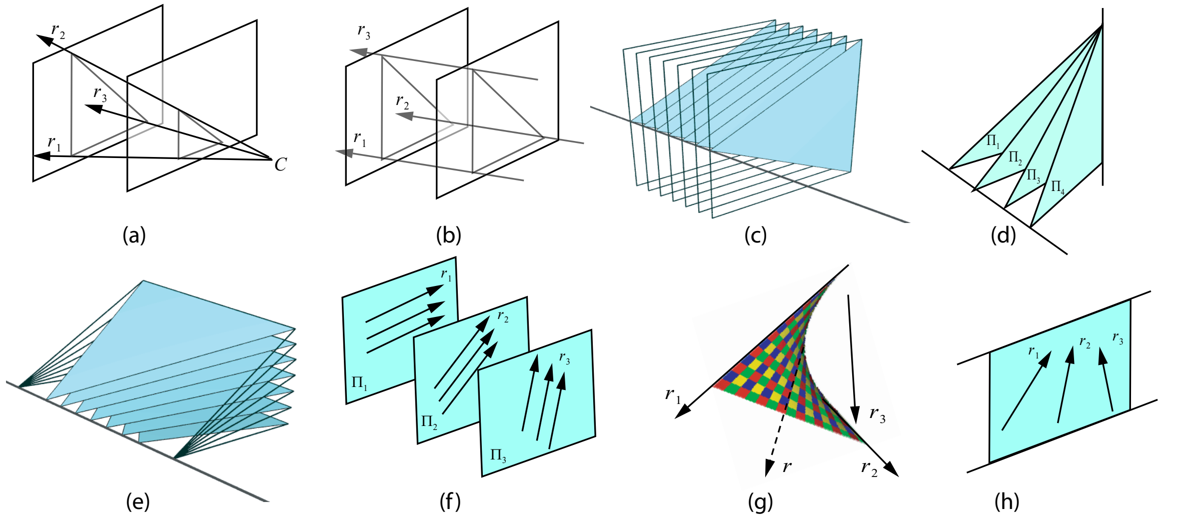

Figure 3: *General Linear Camera Models. (a) In a pinhole camera, all rays pass through a single point. (b) In an orthographic camera, all rays are parallel. (c) In a pushbroom, all rays lie on a set of parallel planes and pass through a line. (d) In a cross slit camera, all rays pass through two non-coplanar 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 an bilinear camera, no two rays are coplanar and no two rays intersect. (h) In an EPI camera, all rays lie on a 2D plane.*

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:

$$r = \alpha r_1 + \beta r_2 + (1-\alpha-\beta) r_3$$

GLCs model all 2-dimensional linear subspaces in the 4-dimensional "ray space" imposed by a two-plane parametrization. Moreover, these 2D subspaces of rays form images (perspective if pinhole, multiperspective if not). An important characteristic of a GLC is that any general point in 3D space has a unique mapping to a ray. This is because under (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), in general 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. GLCs, thus, are particularly useful for imaging systems since each 3D point has a unique mapping to the image plane.

In [YM04], a pair of quadratic characteristic equations is given for GLCs.

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

where

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

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

A second 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.

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

The number of solutions to the 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 3. The derivation and proofs of these characterizations can be found in [YM04].

4. Piecewise Multiperspective Image Plane

While the GLC model includes many multiperspective camera types, it is still not sufficient to describe a general multiperspective image. Multiperspective GLCs respect specific linear geometric structures. For instance, they describe specific sets of rays that all pass through either a specific point,

Table 1: *Characterize General Linear Cameras by Characteristic Equation*

Characteristic Equation	2 Solution	1 Solution	0 Solution	∞ Solution
$A \neq 0$	XSlit	Pencil/Pinhole†	Bilinear	Ø
$A = 0$	Ø	Pushbroom	Twisted/Ortho.†	EPI

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

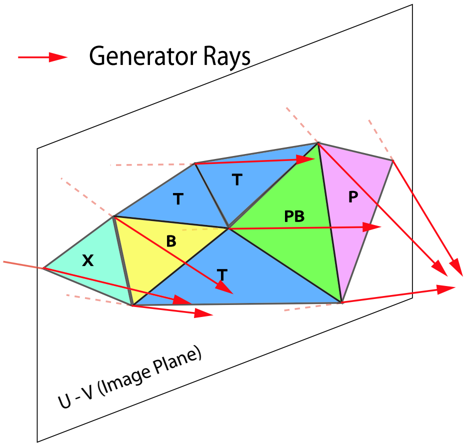

Figure 4: *A tessellation of the image plane with triangles that correspond to different types of general linear cameras. This model is equivalent to a 2D piecewise planar manifold embedded in 4D.*

Table 2: *Adjacency Tables of General Linear Cameras*

Posssible Adjacency	P	O	PB	X	PN	T	B
pinhole (P)	N	N	Y	Y	Y	N	N
orthographic (O)	N	N	Y	N	N	Y	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

line, pair of lines, or lie on a ruled bilinear surface. Multiperspective rendering, as practiced by the masters, follow a far less restrictive set of rules. Specifically, they only attempt to preserve spatial continuity in the image plane, which can be accomplished by range of smooth, yet nonlinear, variations of viewpoint.

We exploit a different aspect of GLCs in order to assure spatial continuity without restricting the viewpoints to fall on specific linear structures. This characteristic of GLCs relates to the fact that a smooth variation in sample rays on the image plane is necessary to provide spatial continuity in rendering. Under a 2PP rays map to points in 4D. Therefore, any smooth 2D manifold embedded in 4D will correspond to rays of a multiperspective image. Imposing a parametrization over the manifold enables sampling and the subsequent rendering. Under this interpretation, GLCs describe all planar, or flat, manifolds in 4D. While this is a limited set of potential manifolds, it is sufficient to construct a piecewise linear approximation and to describe all the tangent planes of any given manifold. In other words, any piecewise planar tessellation of a "ray-space" manifold corresponds a specific collection of GLCs, much like a polygonal model of a curved surface.

An equivalent interpretation of the 4D points in 2PP is an image plane with an attached ray, much like a generator of a GLC, as is shown in Figure 2. Therefore, a general class of multiperspective images can be described by a triangulation of the image plane along with generator rays attached to each vertex as shown in Figure 4. As the tessellation of the image plane increases, this model can approximate arbitrarily smooth 2D manifolds, and hence render arbitrary multiperspective images.

4.1. GLC Adjacency Table

When the triangles and generators of a given tessellation are analyzed according to their type, as determined by the characteristic equation (1), one finds that there are specific rules (constraints) for tiling the image plane using GLCs.

Since each triangle on the image plane corresponds to a general linear camera, adjacent triangles sharing a common edge represent two GLC cameras that share two rays. This imposes a constraint on possible pairs of adjacent general linear cameras. 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 pinhole while no two rays will intersect in a bilinear camera as shown in Figure 3.

In Table 2, we show all possible adjacency relationships between general linear cameras. Triangulations of the image

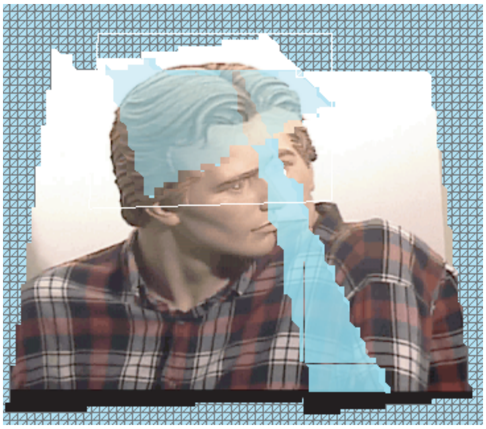

Figure 5: *Desirable image fragments from different GLCs are overlayed on the image plane.*

plane into GLCs must satisfy these adjacency constrains in order 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, multiperspective 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 multiperspective image which looks locally like a pinhole camera everywhere. However, there do exist multiperspective images which look locally like pushbroom or cross-slit images everywhere. In fact, multiperspective panoramas for cel animations are good examples of these type of multiperspective images.

4.2. Image Layout

While any triangulation of the image plane with generator rays describes a multiperspective rendering, it is, in our experience, not a very intuitive specification. In practice, we employ a multiperspective image design method that is inspired by the automatic layout method described by Wood [WFH*97], but with user guidance. The technique works as follows. 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, so long as the blend preserves affine combinations. The end result is that corresponding rays are interpolated such that the transition is smooth.

We provide an interface that allows users to select from a palette of general linear camera images. These images can

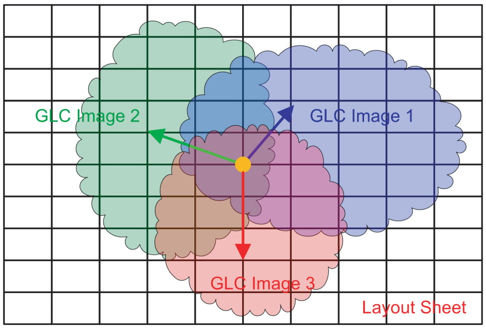

Figure 6: *Ray coordinate of each vertex in the overlapped region is computed using Algorithm 1.*

be rendered directly from either a 3D model or a light field. Both 3D linear and 4D light fields can be used. The user can inspect and crop interesting regions from GLC images and then lay them out on the grid as desired. This process is shown in Figure 5.

We have found it very useful to use irregular crops, which we call *image fragments*, to focus on the desired scene elements. When an image fragment of a general linear camera is placed onto the image plane, we first assign pixel coordinates to the fragment. We then compute the affine coordinates of all vertexes of the GLC triangulates inside the fragment and use the same weight to compute their (s,t,u,v) ray coordinates.

Our system allows the user to perform any transformation on image fragments that maintains affine relationships including translation, scaling and rotation. Each time a transformation is performed, we simply recompute the pixel coordinate of the fragment and update the ray coordinates for all vertexes of GLCs inside fragment.

Any triangle on the image plane whose three vertex-ray coordinates have been associated with a generator will form a GLC that can be rendered by ray tracing or querying into a light field. The image plane, therefore, embodies a piecewise linear multiperspective manifold and maintains C_0 continuity.

4.3. Ray Blending

When multiple GLC image fragments are placed onto the image plane, the (s,t,u,v) ray coordinates of points lying in overlapped regions need to be computed in order to maintain a smooth transition from one general linear camera to another. To do so, we find the ray coordinates of each point in the overlapping region. We then compute the shortest distance from the point to each of the image fragments, i.e., the distance to the closet point in the corresponding fragment that is not in the overlapped region, as is shown in Figure 6. We then use this distance as a weighing function to blend ray coordinates. Finally, we render all triangles as general linear cameras. Notice a triangle whose three vertexes come from a common reference image can be directly copied from the image fragment, thus saving processing. The complete algorithm is shown algorithm 1.

Algorithm 1 Compute Ray Coordinates r_{blend}

for each image fragment *img* **do**
  for each grid point $P(i,j)$ in *img* **do**
    calculate the $r_{img}(s,t,u,v)$ coordinate of $P(i,j)$ in *img*
    add r_{img} to P's ray list $List_r$
    add *img* to P's image list $List_{img}$
  end for
end for
for each grid point $P(i,j)$ **do**
  if P's image list $List_{img}$ is not empty **then**
    for each image fragment *img* in $List_{img}$ **do**
      compute the distance dis_{img} from $P(i,j)$ to the closest point in *img* that is not overlapped
      compute the weighing factor $weight_{img}$ in term of dis_{img}
    end for
    normalize weighing factor
    $r_{blend} = (0,0,0,0)$
    for each image fragment *img* in $List_{img}$ **do**
      $r_{blend} = r_{blend} + r_{img} * weight_{img}$
    end for
  end if
end for

5. Multiperspective Sampling

With regard to sampling, the GLC model is considerably different than the specification of traditional camera models. A GLC determines only a set of rays seen, it does not uniquely specify how these rays are distributed across the image plane.

Image plane sampling requires a consistent parametrization between GLCs of the multiperspective manifold. We achieve this parametrization by specifying a sampling plane with uniform grid and defining the corresponding manifold parametrization as a mapping from the sampling plane to 2PP, as is shown in Figure 7(c).

To understand our sampling approach it is helpful to tease apart a traditional pinhole camera into its separate components. A pinhole camera is specified by a center-of-projection, the point where all rays originate, and a sampling plane which is determined by an image origin and two vectors spanning the plane as shown in Figures 7(a). Integer multiples of the spanning vectors determine the sampling grid. The center-of-projection is the GLC component of the model, whereas the sampling plane and the grid implied by the spanning vectors is the sampling component of this model. The process of orienting the image plane in space has the effect of projectively warping the image without changing the set of rays seen by the camera, and is commonly referred to as *homography*.

Therefore, we can modify the sampling of a GLC by similarly transforming the sampling plane. However, we need to guarantee that each grid point on the sampling plane can be seen only once. Fortunately, all GLCs maintain this property except for at singular points for the particular GLC (COP of pinhole, two slits of cross-slit, all points of an EPI, and etc) [YM04]. In Figure 7(b), we show an example of the sampling scheme for a cross-slit camera. We call changes in the image due to transformations of the image plane *perspective distortion*, whereas we call distortions introduced by smooth changes in the center of projection *projective distortion*.

5.1. Perspective and Projective Distortions

Once a generator ray is established for each vertex of our multiperspective image, we can render each pixel on the image plane by either ray tracing or querying a light field. Since pixels are uniformly sampled on image plane, rays are sampled uniformly over all the GLCs of the multiperspective manifold. Such uniform sampling, done independent of the projection, leads to noticeable projective distortions. For example, consider two pinhole cameras (not adjacent to each other) from a common multiperspective image, with congruent triangle regions, but whose centers-of-projection are at different depths. Imposing a common image plane has the effect that the camera with larger depth will have a smaller view of the scene than the closer viewpoint looking though an image-plane triangle of the same area. The mixing of very different fields of views leads to significant projective distortions in the final image. Our goal in sampling is to introduce subtle perspective distortions in order to smooth the transitions of projective distortion. An example of an image with an abrupt projective distortion is shown in Figure 8(a) and its correction is shown in 8(b).

5.2. Resampling

We provide a simple algorithm to correct projective distortions by smoothly varying the sampling rate across the general linear cameras. We assume rays on each GLC are specified using a sampling plane, as is shown in Figure 7 and the sampling plane itself is uniformly sampled. When the sampling plane is close to the actual scene geometry, sampling the GLC rays uniformly will reduce the projective distortion effects. Thus, when a 3D model of the scene is available, we can simply align the sampling plane close to the real geometry. When a light field is used, we can estimate the depth of the scene in terms of disparities, and align the sampling plane closer to the geometries using a method similar to the focal plane used in dynamically reparametrized lightfield rendering[IMG00].

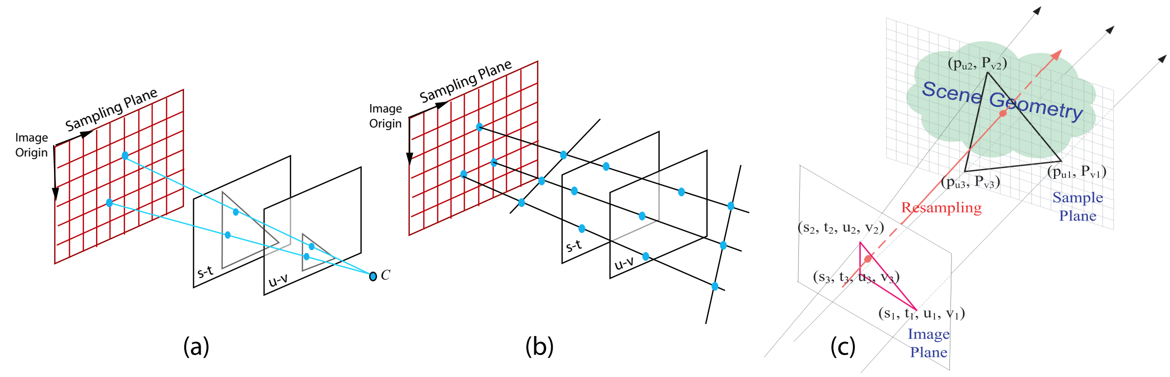

Figure 7: *(a) Sampling of a pinhole camera (homography). (b) Sampling of a cross-slit camera. (c) Sampling of an arbitrary multiperspective image is achieved by assigning a sampling plane to each GLC.*

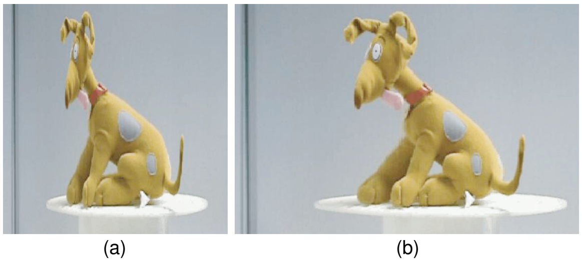

Figure 8: *(a) A multiperspective image before sampling correction; (b) A multiperspective image after sampling correction using sampling planes is applied. In the original stuffed toy, the head is oriented such that it looks at the tail. This multiperspective image appears like a plausible perspective image with a different pose of the toy.*

Figure 8 compares multiperspective images of a stuffed toy rendered from a 360 degree turntable sequence. When we compose a different perspective of the head and tail of the stuffed toy to the body, we observe foreshortening artifacts in Figure 8(a), due to projective distortion. By estimating the depth of the head and the detail, we can then correct this distortion by resampling, as is shown in Figure 8(b).

6. Results

6.1. Multiperspective Panoramas

Panoramas attempt to combine a series of views into a single image that is viewed in pieces. Sets of these triangles can be constrained to match the pinhole views while bands of transition GLCs can be used to fuse the images into a single multiperspective panorama. A practical advantage of our approach over that of Woods et al is that there is no need to compute optical flow or blend the overlap regions between images. Our images can be as sharp as the source material. Alternatively the images can even be rendered directly.

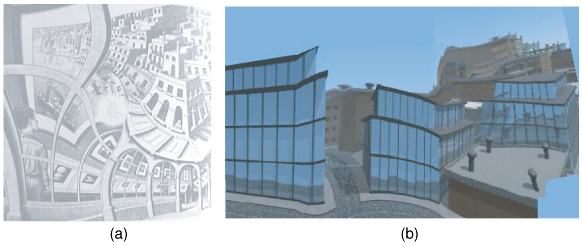

Figure 9: *(a) A painting by M.C Escher; (b) our synthesized multiperspective image. Notice the similarity between the curved outlines of the building in both images.*

In Figure 10, we show a multiperspective panorama synthesized from virtual flight path over a city. We also show the layout of the camera types on the right, where blue represents pinholes, red represents cross-slit and green represents pushbrooms. Although many adjacent cross-slit cameras are labelled as the same camera types, because of our smooth blending, they are different cross-slit cameras.

6.2. Artistic Multiperspective

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 Escher are examples of such multiperspective images. In Figure 9, we compare Escher with our synthesized image. Our framework starts from simple layout and achieves similar multiperspective effects. In Figure 11, we show the multiperspective view of the teapot by overlaying image pieces from significantly different perspectives. In the limit, we are able to show close to 360 degree view of the teapot, reminiscent of an MCOP image [RB98].

6.3. Multiperspective Faux-Animation

It is possible to use multiperspective rendering to create fake or faux-animations from still-life scenes. This is particularly useful for animating image based models. In Figure 12, we show three frames from a synthesized animation. By using perspective correction, we can achieve plausible motions. Zomet [ZFPW03] used similar approach by using single cross-slit camera (one of our general linear camera) to achieve rotation effects.

7. Discussion and Conclusions

We have presented a framework for the direct rendering of multiperspective images by treating multiperspective imaging systems as continuous manifolds in ray space and approximating them using piecewise planar sections, where each facet is a GLC. Multiperspective renderings can be easily achieved by laying out different pieces of GLCs on an image plane. Using our framework, we can achieve a broad class of multiperspective images. Our analysis reveals multiperspective images are constrained by neighboring tangent planes and hence cannot be arbitrary combinations of GLCs. We have also shown multiperspective images have both perspective and projective distortions, and by applying a dynamic sampling plane, projective distortions can be significantly reduced.

References

[AZM00] AGRAWALA M., ZORIN D., MUNZNER T.: Artistic multiprojection rendering. In *Eurographics Rendering Workshop 2000* (2000).

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

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

[GGSC96] GORTLER S. J., GRZESZCZUK R., SZELISKI R., COHEN M. F.: The lumigraph. In *Proc. ACM SIGGRAPH '96* (1996), Computer Graphics Proceedings, Annual Conference Series, ACM, ACM Press / ACM SIGGRAPH, pp. 43–54.

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

[Gla00] GLASSNER A.: Cubism and cameras: Free-form optics for computer graphics. *Microsoft Research MSR-TR-2000-05* (2000).

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

[IMG00] ISAKSEN A., MCMILLAN L., GORTLER S.: Dynamically reparametrized light fields. In *Proc. ACM SIGGRAPH '00* (2000), Computer Graphics Proceedings, Annual Conference Series, ACM, ACM Press / ACM SIGGRAPH, pp. 297–306.

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proc. ACM SIGGRAPH '96* (1996), Computer Graphics Proceedings, Annual Conference Series, ACM, ACM Press / ACM SIGGRAPH, pp. 31–42.

[PBEP01] PELEG S., BEN-EZRA M., PRITCH Y.: Omnistereo: Panoramic stereo imaging. *IEEE Trans. Pattern Analysis and Machine Intelligence 23*, 3 (2001), 279–290.

[RB98] RADEMACHER P., BISHOP G.: Multiple-center-of-projection images. In *Proc. ACM SIGGRAPH '98* (1998), Computer Graphics Proceedings, Annual Conference Series, ACM, ACM Press / ACM SIGGRAPH, pp. 199–206.

[Sei01] SEITZ S. M.: The space of all stereo images. *International Journal of Computer Vision '01* (2001), 26–33.

[SH99] SHUM H. Y., HE L. W.: Rendering with concentric mosaics. In *Proc. ACM SIGGRAPH '99* (1999), Computer Graphics Proceedings, Annual Conference Series, ACM, ACM Press / ACM SIGGRAPH, pp. 299–306.

[WFH*97] WOOD D., FINKELSTEIN A., HUGHES J., THAYER C., SALESIN D.: Multiperspective panoramas for cel animation. In *Proc. ACM SIGGRAPH '97* (1997), Computer Graphics Proceedings, Annual Conference Series, ACM, ACM Press / ACM SIGGRAPH, pp. 243–250.

[YM04] YU J., MCMILLAN L.: General linear cameras. In *ECCV (2)* (2004), pp. 14–27.

[ZFPW03] ZOMET A., FELDMAN D., PELEG S., WEINSHALL D.: Mosaicing new views: The crossed-slits projection. *IEEE Trans. on PAMI* (2003), 741–754.