Multiperspective Distortion Correction Using Collineations

Yuanyuan Ding and Jingyi Yu

Department of Computer and Information Sciences
University of Delaware
Newark, DE 19716, USA
{ding,yu}@eecis.udel.edu

Abstract. We present a new framework for correcting multiperspective distortions using collineations. A collineation describes the transformation between the images of a camera due to changes in sampling and image plane selection. We show that image distortions in many previous models of cameras can be effectively reduced via proper collineations. To correct distortions in a specific multiperspective camera, we develop an interactive system that allows users to select feature rays from the camera and position them at the desirable pixels. Our system then computes the optimal collineation to match the projections of these rays with the corresponding pixels. Experiments demonstrate that our system robustly corrects complex distortions without acquiring the scene geometry, and the resulting images appear nearly undistorted.

1 Introduction

A perspective image represents the spatial relationships of objects in a scene as they would appear from a single viewpoint. Recent developments have suggested that alternative multiperspective camera models [5,16] can combine what is seen from several viewpoints into a single image. These cameras provide potentially advantageous imaging systems for understanding the structure of observed scenes. However, they also exhibit multiperspective distortions such as the curving of lines, apparent stretching and shrinking, and duplicated projections of a single point [12,14].

In this paper, we present a new framework for correcting multiperspective distortions using collineations. A collineation describes the transformation between the images of a camera due to changes in sampling and image plane selection. We show that image distortions in many previous cameras can be effectively reduced via proper collineations. To correct distortions in a specific multiperspective camera, we develop an interactive system that allows users to select feature rays from the camera and position them at the desirable pixels. Our system then computes the optimal collineation to match the projections of these rays with the corresponding pixels.

Compared with classical distortion correction methods [12,2,11], our approach does not require prior knowledge on scene geometry and it can handle highly

Y. Yagi et al. (Eds.): ACCV 2007, Part I, LNCS 4843, pp. 95–105, 2007.

complex distortions. We demonstrate the effectiveness of our technique on various synthetic and real multiperspective images, including the General Linear Cameras [14], catadioptric mirrors, and reflected images from arbitrary mirror surfaces. Experiments show that our method is robust and reliable, thus the resulting images appear nearly undistorted.

2 Previous Work

In recent years, there has been a growing interest in designing multiperspective cameras which capture rays from different viewpoints in space. These multiperspective cameras include pushbroom cameras [5], which collect rays along parallel planes from points swept along a linear trajectory, the cross-slit cameras [8,16], which collect all rays passing through two lines, and the oblique cameras [7], in which each pair of rays are oblique. The recently proposed General Linear Cameras (GLC) uniformly model these multiperspective cameras as 2D linear manifolds of rays (Fig. 1). GLCs produce easily interpretable images, which are also amenable to stereo analysis [9]. However, these images exhibit multiperspective distortions [14].

In computer vision, image-warping has been commonly used to reduce distortions. Image-warping computes an explicit pixel-to-pixel mapping to warp the original image onto a nearly perspective image. For cameras that roughly maintain a single viewpoint [6], simple parametric functions are sufficient to eliminate perspective, radial, and tangential distortions [2,3]. However, for complex imaging systems, especially those exhibiting severe caustic distortions [12], the warping function is difficult to model and may not have a closed-form solution.

Image-based rendering algorithms have also been proposed to reduce image distortions [10,4]. There, the focus has been to estimate the scene structure from a single or multiple images. Swaminathan and Nayar [13] have shown that simple geometry proxies, such as the plane, sphere, and cylinder, are often sufficient to reduce caustic distortions on catadioptric mirrors, provided that the prior on scene structure is known.

We present a third approach based on multiperspective collineations. A collineation describes the transformation between the images of a camera due to changes in sampling and image plane selection. For many multiperspective cameras such as the pushbroom [5] and the cross-slit [8], collineations can be uniformly modeled using the recently proposed General Linear Cameras (GLC) [15].

2.1 GLC Collineation

In the GLC framework, every ray is parameterized by its intersections with the two parallel planes, where $[u, v]$ is the intersection with the first and $[s, t]$ the second, as shown in Fig. 1(a). This parametrization is often called a two-plane parametrization (2PP) [4,15]. We can reparameterize each ray by substituting $\sigma = s - u$ and $\tau = t - v$. In this paper, we will use this $[\sigma, \tau, u, v]$ parametrization to simplify our analysis. We also assume the default uv plane is at $z = 0$ and st plane at $z = 1$. Thus $[\sigma, \tau, 1]$ represents the direction of the ray.

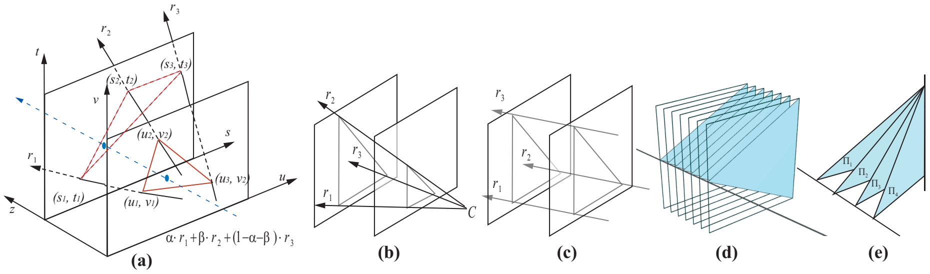

Fig. 1. General Linear Camera Models. (a) A GLC collects radiance along all possible affine combination of three rays. The rays are parameterized by their intersections with two parallel planes. The GLC model unifies many previous cameras, including the pinhole (b), the orthographic (c), the pushbroom (d), and the cross-slit (e).

A GLC is defined as the affine combination of three rays parameterized under 2PP:

$$r = \alpha[\sigma_1, \tau_1, u_1, v_1] + \beta[\sigma_2, \tau_2, u_2, v_2] + (1 - \alpha - \beta)[\sigma_3, \tau_3, u_3, v_3], \forall \alpha, \beta \tag{1}$$

Many well-known multiperspective cameras, such as pushbroom, cross-slit, linear oblique cameras are GLCs as shown in Fig. 1.

If we assume uv is the image plane, we can further choose three special rays with $[u, v]$ coordinates $[0, 0]$, $[1, 0]$, and $[0, 1]$ to form a canonical GLC as:

$$r[\sigma, \tau, u, v] = (1 - \alpha - \beta) \cdot [\sigma_1, \tau_1, 0, 0] + \alpha \cdot [\sigma_2, \tau_2, 1, 0] + \beta \cdot [\sigma_3, \tau_3, 0, 1] \tag{2}$$

It is easy to see that $\alpha = u$, $\beta = v$, and σ and τ are linear functions in u and v. Therefore, under the canonical form, every pixel $[u, v]$ maps to a ray $r(u, v)$ in the GLC.

A GLC collineation maps every ray $r(u, v)$ to a pixel $[i, j]$ on the image plane $\Pi[\dot{p}, \boldsymbol{d}_1, \boldsymbol{d}_2]$, where $\dot{p}$ specifies the origin and $\boldsymbol{d}_1$, and $\boldsymbol{d}_2$ specify the two spanning directions of Π. For every ray $r[\sigma, \tau, u, v]$, we can intersect r with Π to compute $[i, j]$:

$$[u, v, 0] + \lambda[\sigma, \tau, 1] = \dot{p} + i\boldsymbol{d}_1 + j\boldsymbol{d}_2 \tag{3}$$

Solving for i, j, and λ gives:

$$\begin{aligned} i &= \frac{(\tau d_2^z - d_2^y)(u - p_x) + (d_2^x - \sigma d_2^z)(v - p_y) - (\sigma d_2^y - \tau d_2^x) p_z}{\gamma} \\ j &= \frac{(d_1^y - \tau d_1^z)(u - p_x) + (\sigma d_1^z - d_1^x)(v - p_y) - (\tau d_1^x - \sigma d_1^y) p_z}{\gamma} \end{aligned} \tag{4}$$

where

$$\gamma = \begin{vmatrix} d_1^x & d_2^x & -\sigma \\ d_1^y & d_2^y & -\tau \\ d_1^z & d_2^z & -1 \end{vmatrix} \tag{5}$$

For a canonical GLC, since σ and τ are both linear functions in u and v, γ must be linear in u and v. Therefore, we can rewrite i and j as:

$$
\begin{aligned}
i &= \tfrac{a_1u^2+b_1uv+c_1v^2+d_1u+e_1v+f_1}{a_3u+b_3v+c_3} \\
j &= \tfrac{a_2u^2+b_2uv+c_2v^2+d_2u+e_2v+f_2}{a_3u+b_3v+c_3}
\end{aligned}
\tag{6}
$$

Thus, the collineation $\tilde{Col}_{\Pi}(u, v)$ of a GLC from the uv image plane to a new image plane Π is a quadratic rational function. Fig. 2 shows the images of a GLC under different collineations. It implies that image distortions may be reduced using a proper collineation.

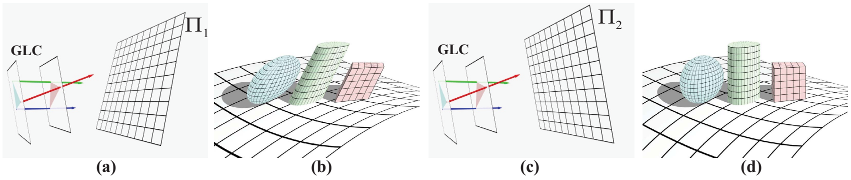

Fig. 2. The image of a cross-slit GLC (d) under collineation (c) appear much less distorted than the image (b) of the same camera under collineation (a)

3 Correct Distortions in GLCs

Given a specific GLC, our goal is to find the optimal collineation to minimize its distortions. Similar to previous approaches [12,11], we assume the rays captured by the camera are known. We have developed an interactive system to allow users to design their ideal undistorted images. Our system supports two modes. In the first mode, the user can select feature rays from the camera and position them at desirable pixels in the target images. In the second mode, the user can simply provide a reference perspective image. Our system then automatically matches the features points. Finally, the optimal collineation is estimated to fit the projections of the feature rays with the target pixels.

3.1 Interactive Distortion Correction

Given a canonical GLC, the user can first select n feature rays (blue crosses in Fig. 3(a)) from the source camera and then position them at desirable pixels (red crosses in Fig. 3(b)) on the target image. Denote $[u_k, v_k]$ as the uv coordinate of each selected ray r_k in the camera and $[i_k, j_k]$ as the desired pixel coordinate of r_k on the target image, we want to find the collineation $\Pi[\dot{p}, \boldsymbol{d}_1, \boldsymbol{d}_2]$ that maps $[u, v]$ as close to $[i, j]$ as possible. We formalize it as a least squares fitting problem:

$$\min_{\Pi} \sum_{k=1}^{n} ||\tilde{Col}_{\Pi}(u_k, v_k) - [i_k, j_k]||^2 \tag{7}$$

Since each collineation $\Pi[\dot{p}, \boldsymbol{d}_1, \boldsymbol{d}_2]$ has 9 variables, we need a minimal number of five ray-pixel pairs. This is not surprising because four pairs uniquely determine a projective transformation, a degenerate collineation in the case of perspective cameras.

Recall that the GLC collineations are quadratic rational functions. Thus, finding the optimal Π in Equation (7) requires using non-linear optimizations. To solve this problem, we use the Levenberg-Marquardt method. A common issue with the Levenberg-Marquardt method, however, is that the resulting optimum depends on the initial condition. To avoid getting trapped in a local minimum, we choose a near optimal initial condition by sampling different spanning directions of Π. We rewrite the spanning directions as:

$$\boldsymbol{d}_i = \eta_i \cdot [cos(\phi_i)cos(\theta_i), cos(\phi_i)sin(\theta_i), sin(\phi_i)], \quad i = 1, 2 \tag{8}$$

We sample several θ_1, θ_2, ϕ_1, and ϕ_2 and find the corresponding $\dot{p}$, η_1, and η_2 as the initial conditions. Finally, we choose the one with the minimum error. This preconditioned optimization robustly approximates a near optimal collineation that significantly reduces distortions as shown in Fig. 3(b).

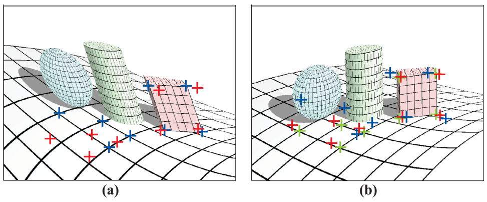

Fig. 3. Interactive Distortion Correction. (a) The user selects feature rays (blue crosses) and positions them at desirable pixels (red crosses). (b) shows the new image under the optimal collineation. The distortions are significantly reduced. The green crosses illustrate the final projections of the feature rays.

3.2 Automatic Distortion Correction

We also present a simple algorithm to automatically reduce distortions. Our method consists of two steps. First, the user provides a target perspective image that captures the same scene. Next, we automatically select the matched features between the source camera and the target image and compute the optimal collineation by minimizing Equation (7).

Recall that a GLC captures rays from different viewpoints in space and hence, its image may appear very different from a perspective image. To match the feature points, we use Scale Invariant Feature Transform (SIFT) to preprocess the two images. SIFT robustly handles image distortion and generates transformation-invariant features. We then perform global matching to find the potential matching pairs. Finally, we prune the outliers by using RANSAC with the homography model. To tolerate parallax, we use a loose inlier threshold of 20 pixels.

In Fig. 4, we show our automatic distortion correction results on various GLCs including the pushbroom, the cross-slit, and the pencil cameras. The user inputs

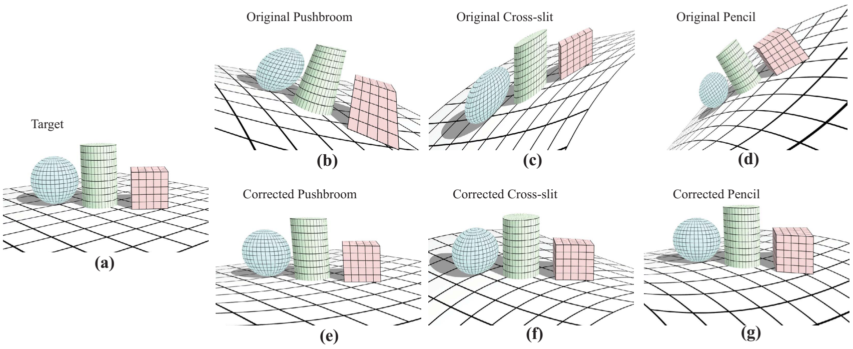

Fig. 4. Automatic Distortion Correction. (a) Perspective reference image; (b),(c), and(d) are distorted images captured from pushbroom camera, cross-slit camera, and pencil camera. (e), (f), and (g) are the distortion corrected results of (b), (c), and (d) using the automatic algorithm.

a perspective image (Fig. 4(a)) and the corrected GLC images appear nearly undistorted using the optimal collineations (bottom row of Fig. 4).

4 Correcting Distortions on Catadioptric Mirrors

Next, we show how to correct multiperspective distortions on catadioptric mirrors. Conventional catadioptric mirrors place a pinhole camera at the focus of a hyperbolic or parabolic surface to synthesize a different pinhole camera with a wider field of view [6]. When the camera moves off the focus, the reflection images exhibit complex caustic distortions that are generally difficult to correct [12].

We apply a similar algorithm using multiperspective collineations. Our method is based on the observation that, given any arbitrary multiperspective imaging system that captures smoothly varying set of rays, we can map the rays onto a 2D ray manifold in the 4D ray space. The characteristics of this imaging system, such as its projection, collineation, and image distortions can be analyzed by the 2-D tangent ray planes, i.e., the GLCs [14]. This implies that a patch on an arbitrary multiperspective image can be locally approximated as a GLC.

We first generalize the GLC collineation to arbitrary multiperspective imaging systems. Notice that not all rays in these systems can be parameterized as $[\sigma, \tau, u, v]$ (e.g., some rays may lie parallel to the parametrization plane). Thus, we use the origin $\dot{o}$ and the direction $\boldsymbol{l}$ to represent each ray r.

The collineation $\Pi[\dot{p}, \boldsymbol{d}_1, \boldsymbol{d}_2]$ maps $r[\dot{o}, \boldsymbol{l}]$ to a pixel $[i, j]$ as:

$$[o^x, o^y, o^z] + \lambda[l^x, l^y, l^z] = \dot{p} + i\boldsymbol{d}_1 + j\boldsymbol{d}_2 \tag{9}$$

Solving for i, j in Equation (9) gives:

$$\begin{aligned} i &= \frac{(l^y d_2^z - l^z d_2^y)(o^x - p^x) + (l^z d_2^x - l^x d_2^z)(o^y - p^y) + (l^x d_2^y - l^y d_2^x)(o^z - p^z)}{\gamma^*} \\ j &= \frac{(l^z d_1^y - l^y d_1^z)(o^x - p^x) + (l^x d_1^z - l^z d_1^x)(o^y - p^y) + (l^y d_1^x - l^x d_1^y)(o^z - p^z)}{\gamma^*} \end{aligned} \tag{10}$$

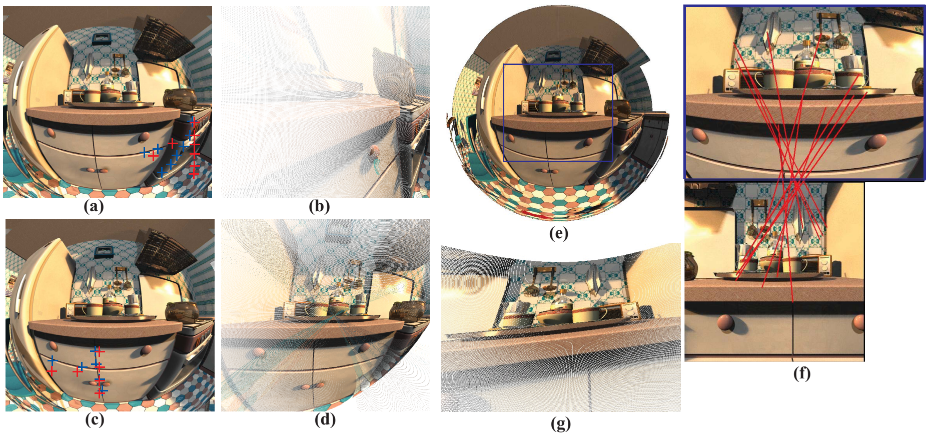

Fig. 5. Selecting different feature rays ((a) and (c)) produces different distortion correction results ((b) and (d)). (f) shows the automatic feature matching between a region (blue rectangle) on the spherical mirror and a perspective image. (g) is the final distortion corrected image. The holes are caused by the under-sampling of rays.

where

$$\gamma^* = \begin{vmatrix} d_1^x & d_2^x & -l^x \\ d_1^y & d_2^y & -l^y \\ d_1^z & d_2^z & -l^z \end{vmatrix} \tag{11}$$

We abbreviate Equation (10) as $[i,j] = \tilde{Col}_{\Pi}(\dot{o}, \boldsymbol{l})$.

The user then selects n feature rays from the catadioptric mirror and positions them at target pixels $[i_k, j_k]$, $k = 1 \ldots n$. Alternatively, they can provide a target perspective image (Fig. 5(f)) and our system will automatically establish feature correspondences using the SIFT-RANSAC algorithm. We then use the Levenberg-Marquardt method (equation (7)) with sampled initial conditions to find the optimal collineation $\tilde{Col}_{\Pi}$.

In the case of catadioptric mirrors, if the selected patch is too large, the resulting image may depend on which rays-pixel pairs are selected. In the kitchen scene example (Fig. 5(a)), selecting the rays from the right side of the spherical mirror produces different results than selecting the rays from the middle part, although distortions are reduced in both cases. This is because the rays inside the patch cannot be approximated as a single GLC model.

5 Results

We have experimented our system on various multiperspective images. We modify the PovRay [18] ray tracer to generate both GLC images and reflected images on catadioptric mirrors. Fig. 3 shows an image of a cross-slit camera in

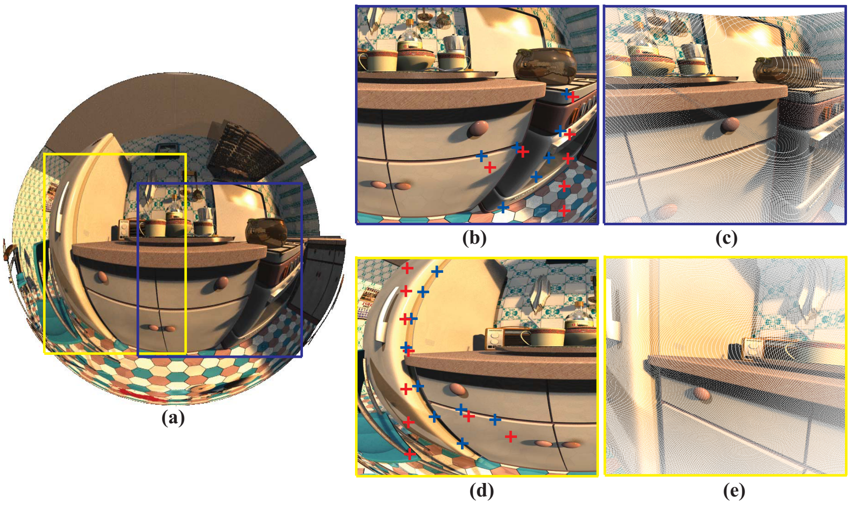

Fig. 6. Correcting distortions on a spherical mirror. The user selects separate regions on the sphere (a) to get (b) and (d). (c) and (e) are the resulting images by matching the selected features (blue) and target pixels (red) in (b) and (d) using collineations.

which the two slits form an acute angle. The user then selects feature rays (blue) from the GLC image and positions them at desirable pixels (red). Our system estimates the optimal collineation and re-renders the image under this collineation as shown in Fig. 3(b). The distortions in the resulting image are significantly reduced.

Next, we apply our algorithm to correct reflection distortions on a spherical mirror shown in Fig. 6. It has been shown [14] that more severe distortions occur near the boundary of the mirror than at the center. Our algorithm robustly corrects both distortions in the center region and near the boundary. In particular, our method is able to correct the highly curved silhouettes of the refrigerator (Fig. 6(d)). The resulting images are rendered by intersecting the rays inside the patch with the collineation plane, thus, containing holes due to the undersampling of rays.

Our algorithm can further correct highly complex distortions on arbitrary mirror surfaces. In Fig. 7, we render a reflective horse model of 48, 000 triangles at two different poses. Our system robustly corrects various distortions such as stretching, shrinking, and duplicated projections of scene points in the reflected image, and the resulting images appear nearly undistorted.

We have also experimented our automatic correction algorithm on both the GLC models and catadioptric mirrors. In Fig. 4, the user inputs a target perspective image 4(a) and our system automatically matches the feature points between the GLC and the target image. Even though the ray structures in the GLCs are significantly different from a pinhole camera, the corrected GLC images appear close to perspective. In Fig. 5(f), a perspective image of a kitchen scene is used to automatically correct distortions on a spherical mirror. This

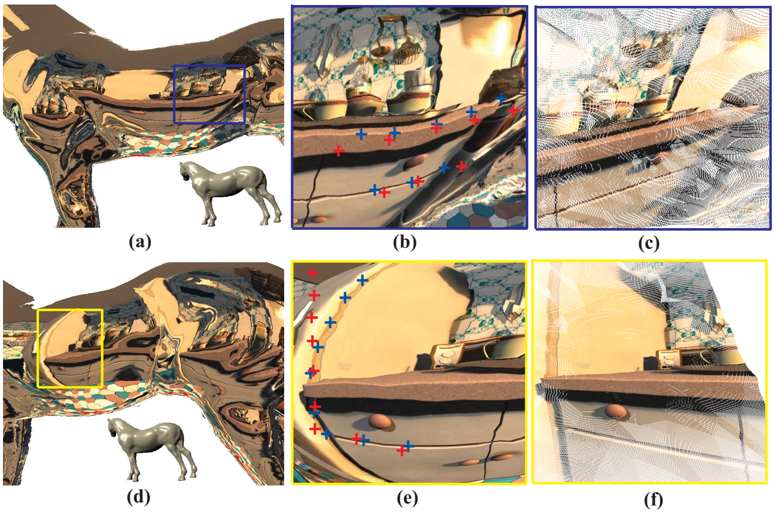

Fig. 7. Correcting complex distortions on a horse model. We render a reflective horse model under two different poses (a) and (d) and then select regions (b) and (e). (c) and (f) are the resulting images by matching the selected features (blue) and target pixels (red) in (b) and (e) using collineations.

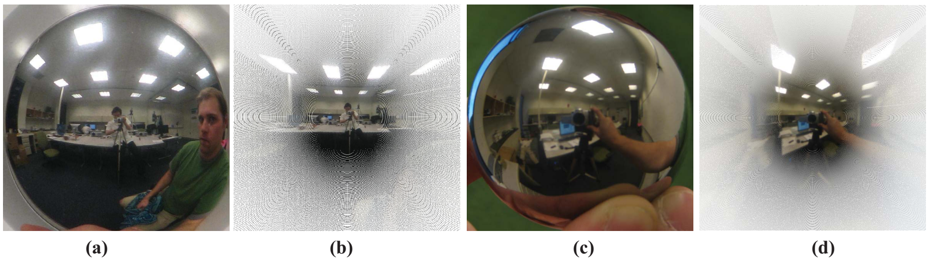

Fig. 8. Correcting reflection distortions. (a) and (c) are two captured reflected images on a mirror sphere. Our algorithm not only reduces multiperspective distortions but also synthesizes strong perspective effects (b) and (d).

implies that our collineation framework has the potential for benefiting automatic catadioptric calibrations.

Finally, we have applied our algorithm on real reflected images of a mirror sphere in a deep scene. We position the viewing camera far away from the sphere so that it can be approximated as an orthographic camera. We then calculate the corresponding reflected ray for each pixel and use our collineation algorithm to correct the distortions. Our system not only reduces multiperspective distortions but also synthesizes strong perspective effects as shown in Fig. 8.

6 Discussions and Conclusion

We have presented a new framework for correcting multiperspective distortions using collineations. We have shown that image distortions in many previous cameras can be effectively reduced via proper collineations. To find the optimal collineation for a specific multiperspective camera, we have developed an interactive system that allows users to select feature rays from the camera and position them at the desirable pixels. Our system then computes the optimal collineation to match the projections of these rays with the corresponding pixels. Experiments demonstrate that our system robustly corrects complex distortions without acquiring the scene geometry, and the resulting images appear nearly undistorted.

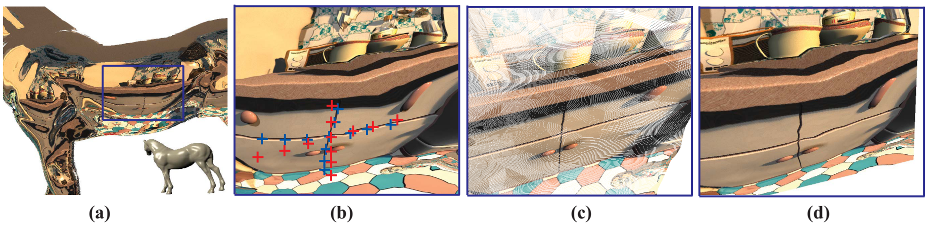

Fig. 9. Comparing collineations with the projective transformation. The user selects feature rays (blue) and target pixels (red). (c) is the result using the optimal collineation. (d) is the result using the optimal projective transformation.

It is important to note that a collineation computes the mapping from a ray to a pixel whereas image warping computes the mapping from a pixel to a pixel. One limitation of using collineations is that we cannot compute the inverse mapping from pixels to rays. Therefore, if the rays in the source camera are undersampled, e.g., in the case of a fixed-resolution image of the catadioptric mirrors, the collineation algorithm produces images with holes. As for future work, we plan to explore using image-based rendering algorithms such as the push-pull method [4] to fill in the holes in the ray space.

We have also compared our collineation method with the classical projective transformations. In Fig. 9, we select the same set of feature points (rays) from a reflected image on the horse model. Fig. 9(c) computes the optimal projective transformation and Fig. 9(d) computes the optimal collineation, both using the Levenberg-Marquardt method for fitting the feature points. The optimal collineation result is much less distorted and is highly consistent with the pinhole image while the projective transformation result remains distorted. This is because multiperspective collineation describes a much broader class of warping functions than the projective transformation.

Acknowledgement

This work has been supported by the National Science Foundation under grant NSF-MSPA-MCS-0625931.

References

1. Chahl, J., Srinivasan, M.: Reflective surfaces for panoramic imaging. Applied Optics 37(8), 8275–8285 (1997)
2. Chen, S.E.: QuickTime VR – An Image-Based Approach to Virtual Environment Navigation. Computer Graphcs 29, 29–38 (1995)
3. Derrien, S., Konolige, K.: Approximating a single viewpoint in panoramic imaging devices. International Conference on Robotics and Automation, 3932–3939 (2000)
4. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: 'The Lumigraph. SIGGRAPH 1996, 43–54 (1996)
5. Gupta, R., Hartley, R.I.: Linear Pushbroom Cameras. IEEE Trans. Pattern Analysis and Machine Intelligence 19(9), 963–975 (1997)
6. Nayar, S.K.: Catadioptric Omnidirectional Cameras. In: Proc. CVPR, pp. 482–488 (1997)
7. Pajdla, T.: Stereo with Oblique Cameras. Int'l J. Computer Vision 47(1/2/3), 161–170 (2002)
8. Pajdla, T.: Geometry of Two-Slit Camera, Research Report CTU–CMP–2002–02, March (2002)
9. Seitz, S., Kim, J.: The Space of All Stereo Images. In: Proc. ICCV, pp. 26–33 (July 2001)
10. Shum, H., He, L.: Rendering with concentric mosaics. Computer Graphcs 33, 299–306 (1999)
11. Stein, G.P.: Lens distortion calibration using point correspondences. In: Proc. CVPR, pp. 143–148 (June 1997)
12. Swaminathan, R., Grossberg, M.D., Nayar, S.K.: Caustics of Catadioptric Cameras. In: Proc. ICCV, pp. 2–9 (2001)
13. Swaminathan, R., Grossberg, M.D., Nayar, S.K.: A Perspective on Distortions. In: Proc. IEEE Computer Vision and Pattern Recognition, Wisconsin (June 2003)
14. Yu, J., McMillan, L.: 'Multiperspective Projection and Collineation. In: ICCV. 10th IEEE International Conference on Computer Vision, Beijing, P. R. China (October 2005)
15. Yu, J., McMillan, L.: Modelling Reflections via Multiperspective Imaging. In: Proc. IEEE Computer Vision and Pattern Recognition, San Diego (June 2005)
16. Zomet, A., Feldman, D., Peleg, S., Weinshall, D.: Mosaicing New Views: The Crossed-Slits Projection. IEEE Trans. on PAMI, 741–754 (2003)
17. Zorin, D., Barr, A.H.: Correction of Geometric Perceptual Distortions in Pictures. Computer Graphics 29, 257–264 (1995)
18. POV-Ray: The Persistence of Vision Raytracer, `http://www.povray.org/`