Recovering Specular Surfaces Using Curved Line Images

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

Peter Sturm
Perception Laboratory
INRIA Grenoble - Rhône-Alpes
Montbonnot 38334, France
Peter.Sturm@inrialpes.fr

Abstract

We present a new shape-from-distortion framework for recovering specular (reflective/refractive) surfaces. While most existing approaches rely on accurate correspondences between 2D pixels and 3D points, we focus on analyzing the curved images of 3D lines which we call curved line images or CLIs. Our approach models CLIs of local reflections or refractions using the recently proposed general linear cameras (GLCs)[23]. We first characterize all possible CLIs in a GLC. We show that a 3D line will appear as a conic in any GLC. For a fixed GLC, the conic type is invariant to the position and orientation of the line and is determined by the GLC parameters. Furthermore, CLIs under single reflection/refraction can only be lines or hyperbolas. Based on our new theory, we develop efficient algorithms to use multiple CLIs to recover the GLC camera parameters. We then apply the curvature-GLC theory to derive the Gaussian and mean curvatures from the GLC intrinsics. This leads to a complete distortion-based reconstruction framework. Unlike conventional correspondence-based approaches that are sensitive to image distortions, our approach benefits from the CLI distortions. Finally, we demonstrate applying our framework for recovering curvature fields on both synthetic and real specular surfaces.

1. Introduction

Recovering curved 3D specular surfaces such as mirrors and fluid surfaces is one of the few remaining open problems in computer vision. Many existing methods have adopted a correspondence-based approach: by establishing correspondences between observed pixels and three dimensional points, it is possible to recover the light path [8], the surface normal [9, 11], and even the surface curvatures [5]. However, robustly finding and tracking point correspondences over time is a very challenging task. Images of a 3D point can be stretched, sheared, and even duplicated due to reflection and refraction distortions as shown in Figure 1. One particularly interesting class of distortions
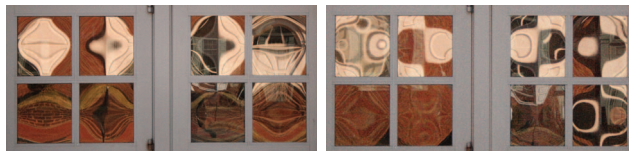

Figure 1. Two images of 0.5m×1m near-flat window captured from 15m away. Images of straight lines far away form interesting conic patterns.

are the curving of 3D lines. We call these images curved line images or CLIs. In general, a 3D line maps to a 2D curve on non-planar specular surfaces. However, characterizing all possible CLIs on the surface is difficult, even with known surface geometry. This is mainly because reflections and refractions are highly non-linear phenomena and closed-form solutions for projecting a 3D line onto a general curved specular surfaces do not exist. Therefore, CLI analysis has been mainly restricted to mirrors of simple shapes [2, 4]. Its inverse problem, i.e., recovering specular surfaces from CLIs, is even more difficult since it is unclear how CLI distortions are related to surface geometry.

In this paper, we present a new CLI-based specular surface reconstruction framework. Our approach uses the recently proposed general linear camera model or GLCs [23] to model local reflections and refractions. We first develop a new theory to characterize all possible CLIs in a GLC. We derive a closed-form solution for projecting a line onto a GLC and we show that the corresponding CLI is either a line or a conic. The type of the conic depends completely on the GLC camera intrinsic parameters and is invariant to the 3D line position and orientation. For example, ellipse and hyperbola CLIs cannot co-exist in the same GLC. Furthermore, we show that CLIs under single reflection/refraction can only be lines or hyperbolas.

Based on our new theory, we develop an efficient algorithm to use multiple CLIs to recover the GLC intrinsic parameters. We then apply the curvature-GLC theory to derive the Gaussian and mean curvatures from the GLC intrinsics. This leads to a complete distortion-based specular surface reconstruction framework. Unlike conventional

1

correspondence-based approaches that are sensitive to distortions, our approach benefits from distortions in CLIs. Finally, we demonstrate applying our framework for recovering curvature fields of both synthetic and real specular surfaces.

2. Previous Work

Despite recent advances in imaging technologies and algorithmic developments, robust reconstruction of specular surfaces from images remains a challenging problem in computer vision. Many existing approaches rely on accurate estimation of point-pixel correspondences to recover the surface geometry or ray geometry. For example, it is a common practice to place a known checkerboard pattern near the specular surface and correlate the checkerboard corners with the captured image pixels.

To recover the specular surface from correspondences, single-image-based methods have followed a shape-from-distortion approach by analyzing how the pattern gets distorted. To eliminate ambiguity, additional constraints are often introduced to model the distortions, including the planarity assumption [9, 13, 5], surface smoothness prior [18], surface integrability constraints[21], and the use of special optics [25, 22]. It is also possible to use stereo [17] or multi-view cameras [3] to resolve the ambiguity, e.g., by establishing point correspondences across the cameras. Morris and Kutulakos introduced the notion of refractive disparity [11] analogous to classical stereo disparity and developed the scatter trace framework [12] for recovering complex and dynamic specular surfaces. Adato et al. introduced the notion of specular flow and they have shown that recovering specular surfaces is equivalent to solving a non-linear PDE defined by the specular flow [1].

The most challenging step in both single-camera and multi-camera-based approaches is to robustly establish point-to-pixel or pixel-to-pixel correspondences. To track dynamic fluid surfaces over time, Morris and Kutulakos [11] used a non-distortion assumption for the initial configuration and applied optical flow to propagate the correspondence from the previous frame. This requires the frame rate to be high enough to avoid motion blur and does not consider complicated surface conditions. In general, state-of-the-art matching algorithms can easily fail due to radiometric inconsistency, calibration errors, and surface perturbations.

An alternative solution is to use images of higher-order geometric primitives such as 3D lines. However, since reflections and refractions behave like a non-perspective camera [24], a line can map to an arbitrarily complex curve. Finding closed-form CLIs, thus, is restricted to non-perspective cameras that follow special ray structures. For example, Zomet et al. [26] derived a line projection equation for a cross-slit camera using $3 \times 4 \times 4$ tensors. Feldman et al. [6] have further shown that the rays passing through
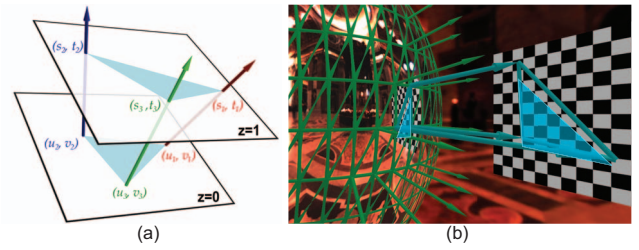

Figure 2. (a) A General Linear Camera is described by 3 generator rays parametrized under two parallel planes (2PP). (b) We decompose a reflection image of a curved mirror into piecewise GLCs (green triangles), and the green arrows represent the reflection ray directions.

a line in the cross-slit camera lie on a double-ruled surface and its intersection with the image plane is a conic. For general non-perspective cameras, projecting a 3D line onto the image plane is a typical inverse ray-tracing problem and does not have a closed-form solution. Furthermore, it is unclear how the CLIs are related to ray structures or specular surface geometries.

3. Analyzing CLIs via GLCs

To model CLIs on general specular surfaces, we first decompose a reflection/refration image into piecewise *linear* multi-perspective cameras, as shown in Figure 2(b). In most of our examples, we pick three pixels on the image plane and find their corresponding reflected/refracted ray coordinates. We then treat each ray-triplet as a primitive camera and study its CLIs.

3.1. General Linear Cameras

Our analysis is based on the recently proposed general linear camera or GLCs [23]. In GLCs, every ray is parameterized by its intersections with the two parallel planes, where $[s, t]$ is the intersection with the first and $[u, v]$ the second, as shown in Figure 2(a). This parametrization is often called a two-plane parametrization (2PP) [10]. In this paper, we further substitute $\sigma = s - u$ and $\tau = t - v$ and use the $[\sigma, \tau, u, v]$ parametrization to represent rays. We also assume the uv plane is the default image plane and is at $z = 0$ while the st plane is at $z = 1$, thus $[\sigma, \tau, 1]$ represents the direction of the ray.

A GLC collects rays that are affine combinations of three generator rays:

$$GLC = \{r : r[\sigma, \tau, u, v] = (1 - \alpha - \beta) \cdot [\sigma_1, \tau_1, u_1, v_1]$$
$$+ \alpha \cdot [\sigma_2, \tau_2, u_2, v_2] + \beta \cdot [\sigma_3, \tau_3, u_3, v_3], \forall \alpha, \beta\}(1)$$

It has been shown that most well-known multi-perspective cameras, such as push-broom, cross-slit, and linear oblique cameras are GLCs [15]. In fact, GLCs provide a first order approximation of any multi-perspective camera. Therefore, we use GLCs to model local reflections and refractions and analyze how surface geometry is related to the CLIs in the local GLC.

To identify the GLC type, [23] proposed to examine whether the three generator rays will simultaneously pass through a line in 3D space. They suggest sweeping planes parallel to the 2PP (i.e., $z = \lambda$). They then compute the area formed by the three intersection points and test if the area is zero as:

$$\begin{vmatrix} u_1 + \lambda \cdot \sigma_1 & v_1 + \lambda \cdot \tau_1 & 1 \\ u_2 + \lambda \cdot \sigma_2 & v_2 + \lambda \cdot \tau_2 & 1 \\ u_3 + \lambda \cdot \sigma_3 & v_3 + \lambda \cdot \tau_3 & 1 \end{vmatrix} = 0 \qquad (2)$$

Equation (2) yields to a quadratic equation in λ:

$$\tilde{A}\lambda^2 + \tilde{B}\lambda + \tilde{C} = 0 \qquad (3)$$

where

$$\tilde{A} = \begin{vmatrix} \sigma_1 & \tau_1 & 1 \\ \sigma_2 & \tau_2 & 1 \\ \sigma_3 & \tau_3 & 1 \end{vmatrix}, \tilde{B} = \begin{vmatrix} \sigma_1 & v_1 & 1 \\ \sigma_2 & v_2 & 1 \\ \sigma_3 & v_3 & 1 \end{vmatrix} - \begin{vmatrix} \tau_1 & u_1 & 1 \\ \tau_2 & u_2 & 1 \\ \tau_3 & u_3 & 1 \end{vmatrix}, \tilde{C} = \begin{vmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ u_3 & v_3 & 1 \end{vmatrix}$$

Equation (2) is called the GLC characteristic equation. The coefficients $\tilde{A}, \tilde{B}$, and $\tilde{C}$[1] and the discriminant $\tilde{\Delta} = \tilde{B}^2 - 4\tilde{A}\tilde{C}$ can be used to determine the GLC type [23]. In fact, it is possible to use these parameters to derive the Gaussian and mean curvatures of the specular surface when using GLCs to model local reflections and refractions (Section 5.3).

3.2. Line Projection Equation

Next, we show how to project a 3D line into a GLC. To further simplify our analysis, we pick three special generator rays that originate from $[0,0]$, $[1,0]$, and $[0,1]$ from the uv plane. Thus the GLC can be represented as:

$$r[\sigma, \tau, u, v] = (1 - \alpha - \beta) \cdot [\sigma_1, \tau_1, 0, 0] \\ + \alpha \cdot [\sigma_2, \tau_2, 1, 0] + \beta \cdot [\sigma_3, \tau_3, 0, 1] \qquad (4)$$

It is easy to see that $\alpha = u$ and $\beta = v$ under this simplification. Therefore, every pixel $[u, v]$ directly maps to a ray. We can further substitute Equation (4) into the characteristic equation (2) as:

$$\tilde{A} = \sigma_1(\tau_2 - \tau_3) + \sigma_2(\tau_3 - \tau_1) + \sigma_3(\tau_1 - \tau_2) \\ \tilde{B} = (\tau_3 - \tau_1) - (\sigma_1 - \sigma_2), \quad \tilde{C} = 1 \qquad (5)$$

Next, we derive the line projection equation using the simplified GLC. If l is parallel to the uv plane, we can parameterize l using a point $[x_0, y_0, z_0]$ on the line and the direction $[d^x, d^y, 0]$ of the line. All rays passing through l satisfy

$$[u, v, 0] + \lambda_1[\sigma, \tau, 1] = [x_0, y_0, z_0] + \lambda_2[d^x, d^y, 0] \qquad (6)$$

By eliminating λ_1 and λ_2, we obtain the following $u - v$ constraint:

$$(u + z_0\sigma - x_0)d^y - (v + z_0\tau - y_0)d^x = 0 \qquad (7)$$

This reveals that the image of l is a line in the GLC image and its slope is related to the direction of l.

---

[1] In this paper, we use $\tilde{A}, \tilde{B}, \tilde{C}$ to distinguish from the conic coefficients A, B, C.
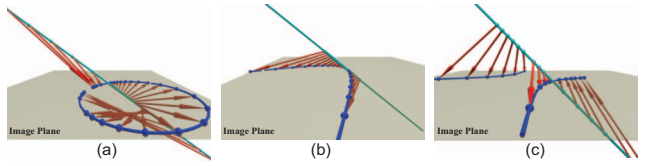

Figure 3. Projection of lines using different GLC parameter conditions: (a) $\tilde{B}^2 - 4\tilde{A}\tilde{C} < 0$. (b) $\tilde{B}^2 - 4\tilde{A}\tilde{C} = 0$. (c) $\tilde{B}^2 - 4\tilde{A}\tilde{C} > 0$.

If l is not parallel to the uv plane, then l will intersect the uv plane at $[u_0, v_0, 0]$ and will have direction $[\sigma_0, \tau_0, 1]$. All rays passing through l satisfy:

$$[u, v, 0] + \lambda_1[\sigma, \tau, 1] = [u_0, v_0, 0] + \lambda_2[\sigma_0, \tau_0, 1]$$

Eliminating λ_1 and λ_2, we have:

$$(u - u_0)(\tau - \tau_0) - (v - v_0)(\sigma - \sigma_0) = 0$$

We then replace σ, τ with u, v using Equation (4) as:

$$(u - u_0)((1 - u - v)\tau_1 + u\tau_2 + v\tau_3 - \tau_0) \\ -(v - v_0)((1 - u - v)\sigma_1 + u\sigma_2 + v\sigma_3 - \sigma_0) = 0 \qquad (8)$$

This reveals that the CLI of l is a conic (Figure 3) in the GLC and it has the form:

$$Au^2 + Buv + Cv^2 + Du + Ev + F = 0 \qquad (9)$$

where

$$A = \tau_2 - \tau_1, \quad C = \sigma_1 - \sigma_3 \\ B = (\tau_3 - \tau_1) + (\sigma_1 - \sigma_2) \\ D = \tau_1 - \tau_0 + (\tau_1 - \tau_2)u_0 + (\sigma_2 - \sigma_1)v_0 \qquad (10) \\ E = \sigma_0 - \sigma_1 + (\tau_1 - \tau_3)u_0 + (\sigma_3 - \sigma_1)v_0 \\ F = (\tau_0 - \tau_1)u_0 + (\sigma_1 - \sigma_0)v_0$$

We call Equation (10) the canonical conic CLI. Our goal is to first determine what conic types can be observed in a GLC and then derive the GLC intrinsic parameters from multiple CLIs.

4. Characterizing Conic Types in the GLC

Lemma 1. *If a 3D line l is not parallel to 2PP, its CLI in the GLC is a line iff the GLC is a pinhole or orthographic camera.*

We refer the reviewers to Lemma 1 in [23] for the proof. Lemma 1 implies that for general GLCs that do not follow a pinhole structure (e.g., local reflections and refractions on curved specular surface), the CLI of a 3D line that is not parallel to the uv plane must be a conic. For the rest of the paper, we will focus on characterizing these conic CLIs.

Recall that the type of a conic represented in Equation (9) can be determined by its discriminant:

$$J = B^2 - 4AC \qquad (11)$$

If $J < 0$, the conic is an ellipse. If $J = 0$, the conic is a parabola. If $J > 0$, the conic is a hyperbola. Equation (10)
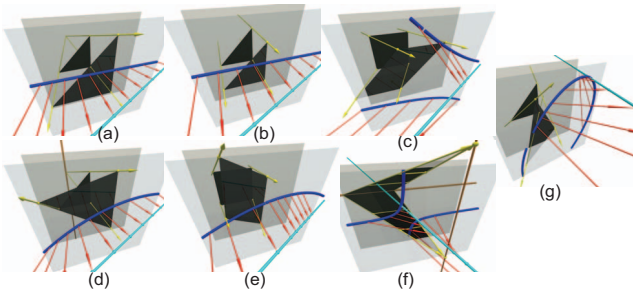
Figure 4. Each GLC image captures a unique curving pattern: lines project to lines in pinhole (a) and orthographic cameras (b), to hyperbolas in pushbroom (c) and cross-slit (f) cameras, to parabolas in pencil (d) and twisted orthographic cameras (e), and to ellipses in bilinear (g) cameras.

shows that the A, B, and C coefficients are uniquely determined by the GLC generator rays $\sigma_1, \sigma_2, \sigma_3, \tau_1, \tau_2, \tau_3$ and are independent of l. Thus, the sign of the discriminant J of the conic is also independent of the location and orientation of l. Therefore we must have:

Lemma 2. *The CLIs of all 3D lines that are not parallel to the image plane, their conic coefficients A, B, and C are identical (up to a scale) and their discriminant J have the same sign in a fixed GLC. They must correspond to the same conic type.*

Lemma 2 reveals that no two different types of conic CLIs can co-exist in the same GLC.

Next, we show how the conic type is related to the GLC intrinsic parameters. We can substitute the conic coefficients A, B, C with the GLC generator rays into Equation (10) and the conic discriminant becomes:

$$J = ((\tau_3 - \tau_1) + (\sigma_1 - \sigma_2))^2 - 4(\tau_2 - \tau_1)(\sigma_1 - \sigma_2) \quad (12)$$

We can also compute the discriminant $\tilde{\Delta}$ of the GLC Characteristic Equation (2) as:

$$\tilde{\Delta} = ((\tau_3 - \tau_1) - (\sigma_1 - \sigma_2))^2 \\ - 4(\sigma_1(\tau_2 - \tau_3) + \sigma_2(\tau_3 - \tau_1) + \sigma_3(\tau_1 - \tau_2)) \quad (13)$$

It is easy to verify that:

$$J = B^2 - 4AC = \tilde{B}^2 - 4\tilde{A}\tilde{C} = \tilde{\Delta} \quad (14)$$

Corollary 1. *All CLIs are conics in a GLC. The conics are of the same type and the type can be determined by the GLC characteristic equation.*

Figure 4 and Table 1 show the complete classification of conics that can be observed by each type of GLC. A special class of GLCs are local reflections and refractions. In [24], Yu and McMillan have shown that, for single reflections under a pinhole or orthographic viewing camera, local GLCs can only be one of the four types: cross-slit, pushbroom, pinhole, or orthographic. No pencil, twisted orthographic,

or bilinear cameras could be observed. This is mainly because the differential geometry property of the specular surface prohibits the rays from twisting. In fact, the two slits of the local crossed-slit cameras rule the caustic surfaces of the reflections rays [20].

Their analysis implies that the CLIs of the local reflection GLCs can only be hyperbolas (in the case of cross-slit and pushbroom) and lines (in the case of orthographic and pinhole). Furthermore, if the CLI of local reflections is a line, we can further conclude that either the line is parallel to the surface patch (by Lemma 1) or the local GLC is a pinhole or orthographic. A similar analysis holds for single refractions when the refraction rays form caustic surfaces.

It is important to note that the GLCs are approximations to local reflections and refractions and the quality of this approximation depends on the smoothness of the geometry. For example, although our analysis reveals that CLIs in local reflections can only be hyperbolas or lines, we often observe ellipse-shaped contours. Because we approximate all rays in a large window using a single GLC, it will correspond to a bilinear GLC and, by Corollary 1, its CLI is an ellipse as shown in Figure 1. We refine our approximation by using smaller patches of rays. As a result, the refined GLCs correspond to cross-slit cameras and the corresponding contour segments are close to hyperbolas.

5. Recovering the GLC from Multiple CLIs

In this section, we address the inverse CLI problem: given multiple 3D lines and their corresponding CLIs in a GLC, how can we recover the GLC parameters? Recall that a canonical GLC is defined by 6 parameters, i.e., $\sigma_1, \tau_1, \sigma_2, \tau_2, \sigma_3, \tau_3$. Since the conic CLI constraint also has 6 equations (Equation (10)), it may seem that we can recover the GLC parameters from a single conic. We start with proving that a single CLI is not sufficient to fully recover the GLC.

Corollary 2. *One conic CLI of a known 3D line is not sufficient to recover the GLC.*

Proof. Assume that the 3D line is not parallel to the image plane, we then parameterize the line by its intersections with 2PP as $[\sigma_0, \tau_0, u_0, v_0]$. The conic constraints (10) can be rewritten as:

$$MS = T \quad (15)$$

where

$$M = \begin{pmatrix} 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ -v_0 & v_0 & 0 & u_0+1 & -u_0 & 0 \\ -1-v_0 & 0 & v_0 & u_0 & 0 & -u_0 \\ v_0 & 0 & 0 & -u_0 & 0 & 0 \end{pmatrix},$$

$$S = (\sigma_1, \sigma_2, \sigma_3, \tau_1, \tau_2, \tau_3)^T, \quad (16)$$

$$T = (A, B, C, D + \tau_0, E - \sigma_0, F - u_0\tau_0 + \sigma_0 v_0)^T$$

Table 1. Conic Types Observed in General Linear Cameras

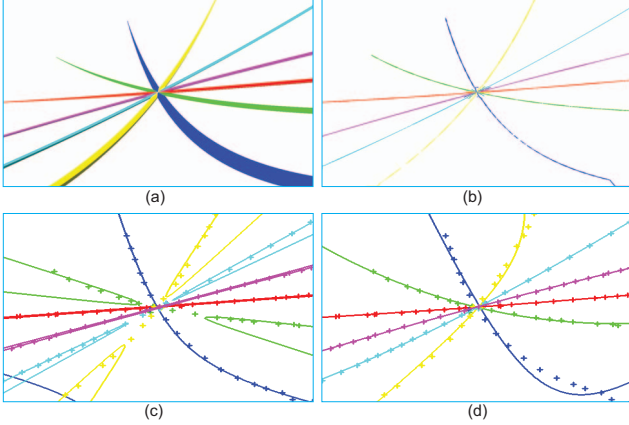| GLC Type	Pinhole	Ortho.	XSlit	Pushbroom	Pencil	Twisted	Bilinear
Conic Type	Line	Line	Hyperbolae	Hyperbolae	Parabola	Parabola	Ellipse
Determinant	$\Delta = 0$	$\Delta = 0$	$\Delta > 0$	$\Delta > 0$	$\Delta = 0$	$\Delta = 0$	$\Delta < 0$

Figure 5. Conic fitting. (a) shows the 6 CLIs in a cross-slit camera. (b) shows the recovered skeletons of the conic images. (c) shows the resulting conics by separately fitting each curve. (d) shows the group fitting results.

We can further verify that the determinant of M is zero and the rank of M is 5. This implies that there are only 5 independent equations in (10). Hence, we cannot uniquely determine the GLC parameters S from a single CLI even if the 3D line coordinates and conic parameters are known. Furthermore, the conic coefficients A to F are unique up to a scale. Therefore, we need to solve for the scaling factor. This leads to 7 unknowns and 5 linearly independent equations. ∎

Corollary 2 shows that we need at least two conic CLIs of known 3D lines in order to fully recover the GLC.

5.1. Conic Fitting

Next, we show how to use multiple conic CLIs to recover the GLC parameters. As the first step, we recover the conic coefficients. Recall that multi-perspective distortions in the GLC can not only curve the line but also stretch and even duplicate the line. As a result, the conic image can exhibit uneven thickness as shown in Figure 5(a). To resolve this issue, we set out to find the medial axis of the conic image. We first use Chamfer Distance Transform (CDT) to compute the distance map and then apply the Laplacian operator to obtain the edge strength map. Finally, we threshold the edge strength map to get the skeleton of the original image. In our experiments, we find this simple procedure can robustly capture smooth medial curves even if the thickness of the curve is highly uneven.

Once we recover the medial curves, we implement a simple method for simultaneously recovering the conic coeffi-

cients for all curves. Recall that for the first conic curve, we have 6 unknowns (A_1 to F_1). For each additional curve i, by Lemma 2, we have $A_i = A, B_i = B, C_i = C$. Therefore, each new conic curve brings 3 new unknowns (D_i, E_i, F_i). For n conic curves, we have a total of $3n + 3$ unknowns.

To solve these unknowns, we select m_i samples on each conic curve i. Denote each pixel sample has coordinate $[u_{ij}, v_{ij}], 0 < j \leq m_i$, we can form the problem of finding the conic coefficients as an over-constrained linear system:

$$\begin{pmatrix} u_{11}^2 & u_{11}v_{11} & v_{11}^2 & u_{11} & v_{11} & 1 \cdots 0 & 0 & 0 \\ \vdots & & & \cdots & & & \vdots \\ u_{1m_1}^2 & u_{1m_1}v_{1m_1} & v_{1m_1}^2 & u_{1m_1} & v_{1m_1} & 1 \cdots 0 & 0 & 0 \\ \vdots & & & \cdots & & & \vdots \\ u_{n1}^2 & u_{n1}v_{n1} & v_{n1}^2 & 0 & 0 & 0 \cdots u_{n1} & v_{n1} & 1 \\ \vdots & & & \cdots & & & \vdots \\ u_{nm_n}^2 & u_{nm_n}v_{nm_n} & v_{nm_n}^2 & 0 & 0 & 0 \cdots u_{nm_n} & v_{nm_n} & 1 \end{pmatrix} \begin{pmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \\ E_1 \\ F_1 \\ D_2 \\ E_2 \\ \vdots \\ F_n \end{pmatrix} = 0$$

To solve it, we apply SVD and obtain the conic coefficients for all the curves. We have also implemented more sophisticated schemes [7] by extending the non-linear optimization [19] and forcing that quadratic coefficients of the hyperbolas are identical. In our experiments, they usually give just slight improvements over SVD. This is mainly because our conic curves are strongly correlated by sharing the same A, B, C coefficients.

Figure 5 compares the conic fitting results using the separate and the group fitting schemes. Separately fitting each individual conic produces results that deviate drastically from the ground truth (Figure 5(c)), e.g., the yellow curve should be one branch of the hyperbola yet the separate fitting scheme recovers it as two separate branches. The global fitting scheme, however, is able to accurately recover the conics as shown in Figure 5(d).

5.2. Recovering the GLC Parameters

To estimate the GLC parameters, we assume each 3D line L_i intersects the uv plane and can be parameterized as $[\sigma_{0i}, \tau_{0i}, u_{0i}, v_{0i}]$. Our goal is to recover the GLC parameters from the line coordinates L_i and the recovered conic coefficients A, B, C, D_i, E_i, F_i.

Notice that the coefficients of all conics can be uniformly scaled yet still correspond to the same conic curves. Therefore, we also need to recover the single scaling factor ξ. This adds up to 7 unknowns (ξ and the GLC parameters
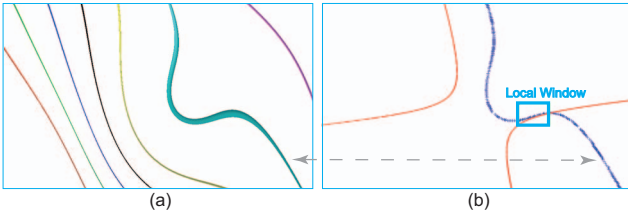
Figure 6. Conic Fitting on CLIs. (a) shows 6 CLIs rendered on a reflective surface. (b) show the estimated conic (red) for fitting the medial curve (blue) in the local window.

$\sigma_k, \tau_k, k = 1, 2, 3$). To solve for these unknowns, we reuse the conic constraints (Equations (10)), where each conic provides 6 linear constraints. Since the A, B, C coefficients are the same for all conic curves by Lemma 2, we have a total of $3n + 3$ equations for n conic curves. We can stack them as the following linear system:

$$\begin{pmatrix} 0 & 0 & 0 & 1 & & -1 & 0 & A \\ -1 & 1 & 0 & 1 & & 0 & -1 & B \\ -1 & 0 & 1 & 0 & & 0 & 0 & C \\ v_{01} & -v_{01} & 0 & -u_{01} & -1 & u_{01} & 0 & D_1 \\ 1+v_{01} & 0 & -v_{0i} & -u_{0i} & 0 & u_{0i} & E_1 \\ -v_{01} & 0 & 0 & u_{01} & & 0 & 0 & F_1 \\ v_{02} & -v_{02} & 0 & -u_{02} & -1 & u_{02} & 0 & D_2 \\ \vdots & & & \cdots & & & & \vdots \\ -v_{0n} & 0 & 0 & u_{0n} & & 0 & 0 & F_n \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_1 \\ \tau_2 \\ \tau_3 \\ \xi \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\tau_{01} \\ \sigma_{01} \\ \phi_{01} \\ -\tau_{02} \\ \vdots \\ \phi_{0n} \end{pmatrix}$$

where $\phi_{0i} = u_{0i}\tau_{0i} - \sigma_{0i}v_{0i}, i = 1 \cdots, n$. We then apply SVD to find the GLC coefficients.

In theory, two conics are sufficient to fully recover the GLC parameters, even if their corresponding 3D lines have the same origin or direction. In practice, we use 10 to 20 curves to improve the robustness and accuracy of our estimation. Furthermore, although we individually solve for the conic coefficients and the GLC parameters, it is possible to combine the two steps into a single procedure. We choose to treat them separately because the quality of conic fitting relies on how densely we sample the pixels on the curves while the accuracy of GLC parameter fitting depends on the number of conic curves. By separating the two parts, we can effectively locate the cause of inaccuracy when the estimated GLC parameters become unreasonable.

5.3. Estimating Specular Surfaces from GLCs

Finally, we can reapply the GLC-curvature theorem for recovering higher order differential geometry attributes of specular surfaces. Ding and Yu [5] have shown that, for near-flat specular surfaces, second-order differential geometries such as curvatures are more important than surface positions or normals. They have further shown that, by modeling local reflections/refractions as GLCs, one can directly derive the Gaussian and mean curvatures from the GLC intrinsic parameters.

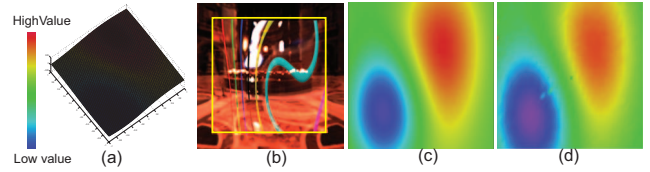Specifically, when viewing a reflective surface using a

Figure 7. Recovering a synthetic mirror surface. (a) shows the surface hight field map, (b) shows a typical ray-traced CLI image, (c) and (d) show the ground truth and recovered Mean curvature respectively.

nearly orthographic camera, we can derive the Gaussian K and mean curvature H from the GLC characteristics as,

$$K = \frac{\tilde{A}}{4}, \qquad H = \frac{\tilde{B}}{4} \qquad (17)$$

In the case of refractions, we have

$$A = (\frac{m-1}{m})^2(z_{xx}z_{yy} - z_{xy}^2) = (\frac{m-1}{m})^2 K$$
$$B = \frac{(m-1)}{m}(z_{xx} + z_{yy}) = \frac{2(m-1)}{m}H \qquad (18)$$

where m is the refraction index.

Our CLI-based approach is more robust than the correspondence-based methods. In [5], Ding and Yu used a color-coded checkerboard pattern to establish point-to-pixel correspondences. They then estimated the ray coordinates from the correspondences and fit a GLC. In the presence of strong distortions, it can be very difficult to find accurate correspondences. Our approach, on the other hand, only requires knowing which curve corresponds to which 3D line. This can be easily achieved by using color-coded lines.

6. Experiment Results

We have validated our CLI-based approach for recovering both synthetic and real specular surfaces. For synthetic reflective and refractive surfaces, we use the POV-Ray Ray Tracer to render multiple curved line images from an orthographic camera [16]. For real mirror surfaces, we construct a calibrated two-camera system. The first camera captures the positions and orientation of 3D lines. The second camera uses a telelens to simulate an orthographic viewing camera towards the specular surfaces.

6.1. Synthetic Surfaces

We first validate our methods on known parametric surfaces. The reflective surface is a combination of Gaussians:
$1.5e^{-1.38(x-0.3)^2-0.62(y-0.5)^2} + e^{-2(x+0.5)^2-1.02(y+0.5)^2}$
In Figure 7, we show the color-coded height field and the ground truth mean curvature field. To recover the surface using the curve line images, we use a total number of 160 lines. Since our CLI-based approach assumes that the 3D lines are not parallel to the surface, we randomly pick the orientation of the lines and only use the ones whose normalized direction d satisfies $\|\hat{d} \cdot \hat{n}\| > 0.75$, where $\hat{n}$ is the

normal of the surface. To be more realistic, we model each line as a cylinder so that its image on the reflective surface will be stretched and distorted, as shown in Figure 6. We render each CLI at an 800×600 resolution.

We partition the rendered CLIs into uniform windows, each covering 30×30 pixels. For each window, we discard the line whose image falls outside the window. On average, each window consists of 18 curves. We then estimate the conic coefficients by simultaneously fitting conics to all medial curve segments using the SVD method (Section 5.1). In Figure 6(b), we plot the medial curve in blue and the recovered conic in red. The conic corresponds to a hyperbola that has two branches. The lower branch fits well to the medial curve inside the window. Finally, we use the recovered conic coefficients to estimate the GLC parameters and then apply the GLC-curvature equation to approximate the curvatures in each window and finally interpolate the estimated curvature field. In Figure 7(c) and 7(d), we compare the ground truth and recovered mean curvatures. Our distortion-based approach robustly recovers the curvatures despite strong image distortions.

In Figure 8, we apply a similar method on a refractive surface:

$$- \cos[1.8((x - 3.2)^2 + (y - 8.0)^2)^{1/2}$$
$$\cdot (3 - \log(1.1 + 0.4(x - 3.2)^2 + 0.4(y - 8.0)^2))]$$

We assume the surface has a refractive index of 1.33. We use the same method as in the reflection case for generating the 3D lines. We use a total of about 200 lines. We also use the same window size as in the reflection case and recover the refraction GLC parameters for each window. We then apply Equation (18) to estimate both the Gaussian and mean curvature fields on the refraction surface. Compared with the ground truth, our method faithfully captures the curvature characteristics.

6.2. Real Surfaces

We have also experimented our approach on real reflection surfaces. We emulate a complex reflector by adhering a transparent paper to a black board. We then apply our method for recovering the geometry of the paper, as shown in Figure 9. To generate 3D lines, we place an LCD monitor next to the board with approximately 60 degree angle and display lines with width of about 15 pixels. We place a CANON Rebel XTi SLR camera equipped with a $300mm$ telelens in front of the paper and set its viewing direction approximately perpendicular to the paper surface so that it satisfies the near-flat condition for applying the GLC-curvature theorem [5].

We use another CANON PowerShot SD800 SI digital camera to calibrate the position of the transparent paper and the LCD display so that we can directly compute the 3D position and orientation of the lines on the display. To accelerate our acquisition, we display multiple lines on the screen
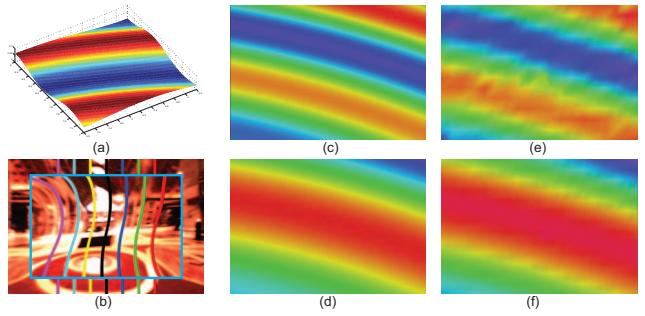

Figure 8. Recovering a synthetic refractive surface. (a) shows the surface height field map, (b) shows a ray-traced CLI image, (c) and (d) show the ground truth Gaussian and mean curvature, (e) and (f) show the Gaussian and mean curvature recovered using our algorithm.
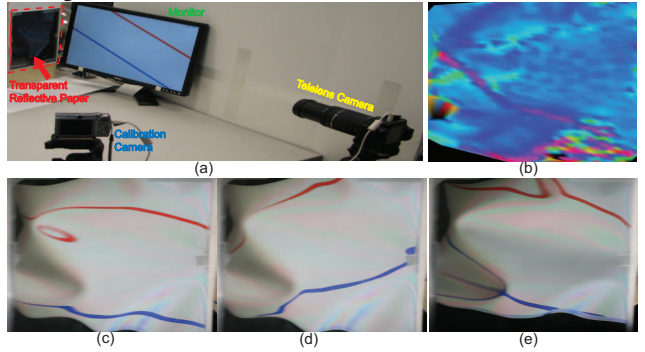

Figure 9. (a) System setup. (b) the recovered mean curvature. (c),(d), and (e) show 3 captured curved line images on the reflective transparent paper using the telelens camera.

using different colors so that we can easily establish line-curve correspondences. We choose 5 different poses for the LCD display and at each pose, we dynamically change the orientation of lines and display a total of 70 different lines. Hence, we capture 350 CLI images, each with a resolution of 1936×1288.

In our experiment, we find that our captured CLIs exhibit chromatic aberrations due to refraction. Therefore, at every LCD display pose, we first capture an ambient image and subtract it from each captured CLI. We uniformly partition the specular surface into 70×70 pixel patches. We then apply the method described in Section 5 to detect and fit conics, and finally compute GLC parameters for every patch. In Figure 9(b), we show the recovered mean curvature field of the transparent paper. Notice that the recovered mean curvatures are closely related to the degree of distortions on the CLI images.

7. Limitations and Future Work

The main limitation of our approach is that we require capturing a large number of CLIs to accurately recover the specular surface. One possible solution is to integrate our approach with the classical point-based methods, e.g., applying the CLI-based method only when it is difficult to es-

tablish pixel-point correspondences. In addition, the recovered curvature fields from the CLIs are noisier than the ones from the correspondence-based results [5]. Notice that our framework estimates ray geometry and curvature at each pixel whereas the correspondence-based approaches estimate these attributes at a sparse set of feature points and then interpolate them over the whole surface. Therefore, special filters consistent with the underlying differential geometry may be applied to smooth the estimated curvature fields from CLIs. Finally, the resulting Gaussian and Mean curvatures may also be used to recover lower-order surface geometry, e.g., by Poisson surface completion, provided that we can capture the surface boundary.

Although we have shown that our method is able to robustly recover synthetic refractive surfaces, it can be difficult to directly apply our framework to capture real refractive surfaces such as fluids in practice. The methods by [5] and [11] place a checkerboard pattern beneath the fluid surface and the pattern is nearly parallel to the fluid surface. In our case, we require that 3D lines intersect the surface, and therefore, similar setups are not directly applicable to our framework. Furthermore, our method requires capturing a large number of 3D lines for accurate estimation. For dynamic fluid surfaces, it is difficult to simultaneously capture and distinguish multiple 3D lines using a single camera. In the future, we plan to explore using multiple cameras or even a camera array for capturing CLIs, where each camera only recovers a sampled curvature field using a subset of 3D lines. The final curvature samples can then be integrated using surface completion algorithms.

There are many other types of distortions besides CLIs. Two special classes of distortions that we plan to explore are the shearing and stretching distortions. Reflections and refractions are obviously non-conformal: a right-angled triangle maps to a sheared triangle in the reflection/refraction GLC. Therefore, we plan to combine conformal geometry and ray caustic analysis to quantitatively characterize these distortions and use them for recovering the GLC intrinsics and surface geometry attributes.

Acknowledgement

References

[1] Y. Adato, Y. Yasilyev, O. Ben-Shahar, and T. Zickler, Toward a Theory of Shape from Specular Flow, In Proc. ICCV' 2007

[2] S. Baker and S.K. Nayar, A theory of catadioptric image formation. In Proc. ICCV' 1998, pages 35–42.

[3] T. Bonfort and P. Sturm. Voxel carving for specular surfaces. In ICCV' 03, pages 591–596.

[4] V. Caglioti, P. Taddei, G. Boracchi, S. Gasparin and, A. Gius, Single-Image Calibration of Off-Axis Catadioptric Cameras Using Lines, In OMNIVIS' 2007.

[5] Y. Ding and J. Yu, Recovering Shape Characteristics on Near-flat Specular Surfaces. In CVPR' 2008.

[6] D. Feldman, T. Pajdla, and D. Weinshall: On the Epipolar Geometry of the Crossed-Slits Projection. In Proc. 9th Int. Conf. on Computer Vision (2003).

[7] Fitzgibbon, A., Pilu, M., Fisher, R.B.: Direct Least Square Fitting of Ellipses. IEEE TPAMI 21(5), 476–480 (1999)

[8] K. N. Kutulakos and E. Steger. A theory of refractive and specular 3d shape by light-path triangulation. In Int. J. Comput. Vision, pages 13–29, 2008.

[9] K. Ikeuchi. Determining surface orientations of specular surfaces by using the photometric stereo method. In IEEE - PAMI 1981, pages 661-669.

[10] M. Levoy and P. Hanrahan, Light Field Rendering. In Proc. SIGGRAPH' 1996.

[11] N. J. W. Morris and K. N. Kutulakos. Dynamic refraction stereo. In ICCV' 05, pages 1573–1580.

[12] N. J. W. Morris and K. N. Kutulakos, Reconstructing the Surface of Inhomogeneous Transparent Scenes by Scatter Trace Photography. In Proc. 11th Int. Conf. Computer Vision (ICCV), 2007.

[13] H. Murase. Surface shape reconstruction of an undulating transparent object. In Radiometry, pages 213–217, 1992.

[14] M. Oren and S. K. Nayar, A Theory of Specular Surface Geometry. In IJCV 1997, pages 105–124.

[15] T. Pajdla: Stereo with Oblique Cameras. Int'l J. Computer Vision, vol. 47, nos. 1/2/3 (2002) 161–170.

[16] POV-Ray. http://www.povray.org/

[17] A. C. Sanderson, L. E. Weiss, and S. K. Nayar. Structured highlight inspection of specular surfaces. IEEE Trans. PAMI 1988, 10(1):44–55.

[18] S. Savarese and P. Perona. Local analysis for 3d reconstruction of specular surfaces - part ii. In ECCV' 2002.

[19] P. Sturm and P. Gargallo, Conic Fitting Using the Geometric Distance, In Proc. ACCV 2007.

[20] R. Swaminathan, M. Grossberg, and S. Nayar. Caustics of catadioptric cameras. ICCV, vol.2, pages 2–9, 2001.

[21] M. Tarini, H. P. A. Lensch, M. Goesele, and H.-P. Seidel. 3d acquisition of mirroring objects using striped patterns. In Graphical Models, 67(4):233–259, 2005.

[22] J. Wang and K. J. Dana. A novel approach for texture shape recovery. In ICCV' 03, pages 1374–1380.

[23] J. Yu and L. McMillan. General linear cameras. In ECCV, pages 14–27, 2004.

[24] J. Yu and L. McMillan. Modelling reflections via multiperspective imaging. In Proc. IEEE CVPR, June 2005.

[25] X. Zhang and C.S.Cox. Measuring the two-dimensional structure of a wavy water surface optically: a surface gradient detector. In Experiments in Fluids, pp. 225–237, '94

[26] A. Zomet, D. Feldman, S. Peleg, and D.Weinshall: Mosaicing New Views: The Crossed-Slits Projection. IEEE Trans. on PAMI (2003)