Pose Estimation from Reflections for Specular Surface Recovery

Miaomiao Liu[1], Kwan-Yee K. Wong[1], Zhenwen Dai[2], and Zhihu Chen[1]
[1]The University of Hong Kong, [2]Frankfurt Institute for Advanced Studies
[1]{mmliu, kykwong, zhchen}@cs.hku.hk, [2]dai@fias.uni-frankfurt.de

Abstract

This paper addresses the problem of estimating the poses of a reference plane in specular shape recovery. Unlike existing methods which require an extra mirror or an extra reference plane and camera, our proposed method recovers the poses of the reference plane directly from its reflections on the specular surface. By establishing reflection correspondences on the reference plane in three distinct poses, our method estimates the poses of the reference plane in two steps. First, by applying a colinearity constraint to the reflection correspondences, a simple closed-form solution is derived for recovering the poses of the reference plane relative to its initial pose. Second, by applying a ray incidence constraint to the incident rays formed by the reflection correspondences and the visual rays cast from the image, a closed-form solution is derived for recovering the poses of the reference plane relative to the camera. The shape of the specular surface then follows. Experimental results on both synthetic and real data are presented, which demonstrate the feasibility and accuracy of our proposed method.

1. Introduction

Despite the great advances in shape recovery for diffuse surfaces in the last few decades, specular surface recovery is still a challenging problem. Unlike a diffuse surface whose appearance is viewpoint independent, a specular surface does not have a unique appearance of its own but instead reflects its surrounding environment. Based on this special property, many researchers tried to recover the shape of a specular object by exploring the relation between its structure and its surrounding environment [2, 22, 14, 8]. Methods for specular surface recovery usually introduce motion to the surrounding environment and observe the changes in the reflections produced on the surface. Based on the assumptions made on the environment, existing state-of-art methods can be broadly classified into two approaches, namely shape from specular flow (SFSF) methods [15, 1, 6, 16] and shape from specular correspondences (SFSC) methods [11, 3, 4, 7, 17, 18, 13]. SFSF methods often assume an unknown distant environment under a known continuous motion. These methods often suffer from problems like tracking dense specular flows and solving partial differential equations (PDE). When the surrounding environment is known and close to the object, SFSC methods can be derived. Like SFSF methods, SFSC methods also assume the motion (which may be discrete though) of the environment being known a prior in order to infer surface shape from the observed reflections. Commonly, a reference plane with a known pattern is used as the known environment in SFSC methods. In order to produce a good view of its reflections on the specular surface, the reference plane is often placed side-by-side with the camera (see Fig. 1). This results in the camera not being able to see the reference plane directly, making the calibration of the setup non-trivial. Traditional methods calibrate the poses of the reference plane by introducing an extra reference plane in the field of view of the camera, and an extra camera looking at both reference planes. In [20], Sturm and Bonfort used a planar mirror to allow the camera to see the reference plane through reflection. The pose of the reference plane can be obtained by placing the auxiliary mirror in at least three dif-

(a) (b)

Figure 1. General setup for specular surface recovery. (a) A setup for recovering a specular spoon. The camera and the reference plane are put side-by-side. (b) Reflections produced on the spoon surface under two different positions of the reference plane.

ferent positions. Generally, multiple reference plane positions are needed for recovering a large area of the specular surface. Hence, how to compute the poses of the reference plane easily and automatically becomes an appealing problem. However, the literature becomes comparatively sparse when it comes to automatic pose estimation of the reference plane in specular surface recovery. Liu et al. proposed an automatic motion estimation method by constraining the motion of the reference plane to a pure translation with the assumption that the initial pose of the plane is known a prior [12]. Although they can achieve a simple closed-form solution for the motion estimation problem, their method cannot handle general motion and requires calibrating the initial pose of the reference plane.

In this paper, we consider the problem of estimating the poses of a reference plane in SFSC. Unlike existing methods which require an extra mirror or an extra reference plane and camera, our proposed method recovers the poses of the reference plane directly from its reflections on the specular surface. By establishing reflection correspondences on the reference plane in three distinct poses, our method estimates the poses of the reference in two steps. First, by applying a colinearity constraint to the reflection correspondences, a simple closed-form solution is derived for recovering the poses of the reference plane relative to its initial pose. Second, by applying a ray incidence constraint to the incident rays formed by the reflection correspondences and the visual rays cast from the image, a closed-form solution is derived for recovering the poses of the reference plane relative to the camera. The shape of the specular surface then follows.

The major contributions of this paper are

- The first approach, to the best of our knowledge, for recovering the poses of the reference plane relative to the camera directly from its reflections observed on the specular surface.
- A closed-form solution for recovering the poses of the reference plane relative to its initial pose by enforcing a colinearity constraint on the reflection correspondences.
- A closed-form solution for recovering the poses of the reference planes relative to the camera by enforcing a ray incidence constraint on the incident rays and visual rays, which simultaneously results in the shape of the specular surface.

The rest of the paper is organized as follows. Section 2 describes the physical configuration of the imaging system. Section 3 derives a closed-form solution for recovering the poses of the reference plane relative to its initial pose using a colinearity constraint. Section 4 derives a closed-form solution for recovering the poses of the reference plane relative to the camera by a ray incidence constraint. Section 5 describes implementation issues related to the extraction of the sparse reflection correspondences and recovery of the specular surface given the estimated poses of the reference plane. Experimental results on both synthetic and real data are presented in Section 6, followed by discussions and conclusions in Section 7.

2. Physical Configuration

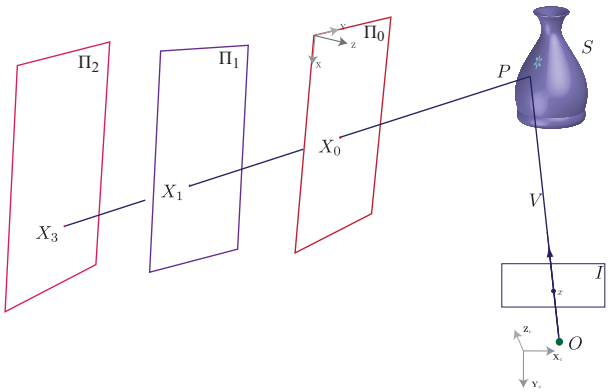

Figure 2. Setup used for specular surface recovery. A pinhole camera centered at O is viewing a specular object S which reflects a nearby reference plane Π_0 to the image I. Π_1 and Π_2 denote the reference plane at its two new poses after undergoing unknown rigid body motions. X_0, X_1, and X_2 are points on Π_0, Π_1 and Π_2 respectively which are reflected at a point P on S to the same image point x on I. They are referred to as *reflection correspondences* and determine the incident ray. V is defined as the visual ray for image point x.

Fig. 2 shows the setup used for specular surface recovery. Consider a pinhole camera centered at O observing the reflections of a moving reference plane on a specular object S. Let X_0 be a point on the plane Π_0 at its initial pose which is reflected by a point P on S to a point x on the image plane I. Suppose the reference plane undergoes an unknown rigid body motion, and let Π_1 and Π_2 denote the plane at its two new poses. Let X_1 be a point on Π_1 and X_2 be a point on Π_2 such that both X_1 and X_2 are also reflected by P to the same image point x on I. X_0, X_1 and X_2 are referred to as reflection correspondences for the image point x. Since reflection correspondences must lie on the same incident ray, it follows that they must be colinear in 3D space. This property will be used to derive a constraint for computing the poses of the moving reference plane relative to its initial pose (see Section 3). Note X_0, X_1, and X_2 define the incident ray and V denotes the visual ray constructed from the image point x. Since corresponding incident ray and visual ray must intersect at the specular surface, this

property will be exploited to determine poses of the reference planes relative to the camera (see Section 4) and the shape of the specular object (see Section 5).

3. Pose Estimation: Reference Plane

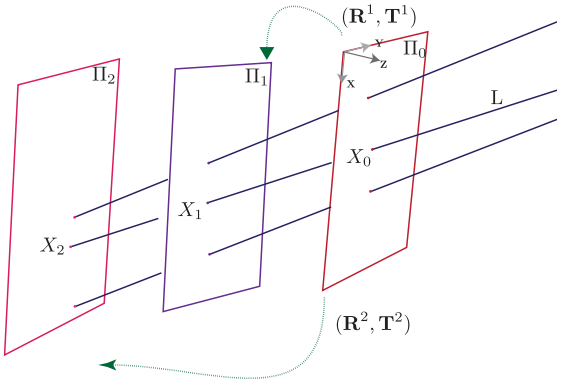

Figure 3. Three reference planes constraining all the incident rays. X_0, X_1 and X_2 are reflection correspondences defining the incident ray L. Assume the world coordinate system coincides with the local coordinate system on Π_0. $\mathbf{R}^1$ and $\mathbf{T}^1$ denote the relative pose between Π_0 and Π_1. $\mathbf{R}^2$ and $\mathbf{T}^2$ denote the relative pose between Π_0 and Π_2.

Referring to the setup as described in the previous section. Let the relative motions of (Π_0, Π_1) and (Π_0, Π_2) be denoted by $(\mathbf{R}^1, \mathbf{T}^1)$ and $(\mathbf{R}^2, \mathbf{T}^2)$ respectively, where $\mathbf{R}^i$ and $\mathbf{T}^i$, $i \in \{1, 2\}$, represent a rotation matrix and translation vector respectively. Further, let the plane-reference coordinates of X_i be $\mathbf{X}_i^{\mathrm{p}} = (x_i^{\mathrm{p}}, y_i^{\mathrm{p}}, 0)^{\mathrm{T}}$, where $i \in \{0, 1, 2\}$ (see Fig. 3). Their $3D$ coordinates, denoted by $\mathbf{X}_i = (x_i, y_i, z_i)^{\mathrm{T}}$, $i \in \{0, 1, 2\}$, with respect to the coordinate system of Π_0 can be written as

$$\begin{aligned} \mathbf{X}_0 &= \mathbf{X}_0^{\mathrm{p}} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}, && (1) \\ \mathbf{X}_1 &= \mathbf{R}^1 \mathbf{X}_1^{\mathrm{p}} + \mathbf{T}^1 \\ &= \mathbf{M} \bar{\mathbf{X}}_1^{\mathrm{p}}, && (2) \\ \mathbf{X}_2 &= \mathbf{R}^2 \mathbf{X}_2^{\mathrm{p}} + \mathbf{T}^2 \\ &= \mathbf{N} \bar{\mathbf{X}}_2^{\mathrm{p}}, && (3) \end{aligned}$$

where $\mathbf{M} = \left(\mathbf{R}_1^1, \mathbf{R}_2^1, \mathbf{T}^1\right)$, $\mathbf{N} = \left(\mathbf{R}_1^2, \mathbf{R}_2^2, \mathbf{T}^2\right)$, $\bar{\mathbf{X}}_i^{\mathrm{p}} = (x_i^{\mathrm{p}}, y_i^{\mathrm{p}}, 1)^{\mathrm{T}}$, and $\mathbf{R}_j^i$ denotes the jth column of $\mathbf{R}^i, i \in \{1, 2\}, j \in \{1, 2\}$. Note that the unknown motion parameters denoted by $(\mathbf{R}^1, \mathbf{T}^1)$ and $(\mathbf{R}^2, \mathbf{T}^2)$ are now embedded in $\mathbf{M}$ and $\mathbf{N}$, which contain 18 parameters in total. Since X_0, X_1 and X_2 are colinear, it follows that

$$\begin{aligned} \frac{x_1 - x_0}{x_2 - x_0} = \frac{y_1 - y_0}{y_2 - y_0} = \frac{z_1 - z_0}{z_2 - z_0}, \\ \frac{\mathbf{M}_1^{\mathrm{T}} \bar{\mathbf{X}}_1^{\mathrm{p}} - x_0}{\mathbf{N}_1^{\mathrm{T}} \bar{\mathbf{X}}_2^{\mathrm{p}} - x_0} = \frac{\mathbf{M}_2^{\mathrm{T}} \bar{\mathbf{X}}_1^{\mathrm{p}} - y_0}{\mathbf{N}_2^{\mathrm{T}} \bar{\mathbf{X}}_2^{\mathrm{p}} - y_0} = \frac{\mathbf{M}_3^{\mathrm{T}} \bar{\mathbf{X}}_1^{\mathrm{p}}}{\mathbf{N}_3^{\mathrm{T}} \bar{\mathbf{X}}_2^{\mathrm{p}}}, \end{aligned} \quad (4)$$

where $\mathbf{M}_i^{\mathrm{T}}$ and $\mathbf{N}_i^{\mathrm{T}}$ denote the ith row of M and N respectively. Eq. (4) gives two constraints as follows:

$$\begin{cases} (\bar{\mathbf{X}}_2^{\mathrm{p}})^{\mathrm{T}} \mathbf{A} \bar{\mathbf{X}}_1^{\mathrm{p}} - x_0 (\bar{\mathbf{X}}_2^{\mathrm{p}})^{\mathrm{T}} \mathbf{N}_3 + x_0 (\bar{\mathbf{X}}_1^{\mathrm{p}})^{\mathrm{T}} \mathbf{M}_3 = 0, \\ (\bar{\mathbf{X}}_2^{\mathrm{p}})^{\mathrm{T}} \mathbf{B} \bar{\mathbf{X}}_1^{\mathrm{p}} - y_0 (\bar{\mathbf{X}}_2^{\mathrm{p}})^{\mathrm{T}} \mathbf{N}_3 + y_0 (\bar{\mathbf{X}}_1^{\mathrm{p}})^{\mathrm{T}} \mathbf{M}_3 = 0, \end{cases} \quad (5)$$

where

$$\begin{aligned} \mathbf{A} &= \mathbf{N}_3 \mathbf{M}_1^{\mathrm{T}} - \mathbf{N}_1 \mathbf{M}_3^{\mathrm{T}}, && (6) \\ \mathbf{B} &= \mathbf{N}_3 \mathbf{M}_2^{\mathrm{T}} - \mathbf{N}_2 \mathbf{M}_3^{\mathrm{T}}. && (7) \end{aligned}$$

Furthermore, Eq. (5) can be written as $\mathbf{E}_1 \mathbf{W} = \mathbf{0}$, where

$$\begin{aligned} \mathbf{E}_1 &= \begin{pmatrix} (\bar{\mathbf{X}}_2^{\mathrm{p}})^{\mathrm{T}} \otimes (\bar{\mathbf{X}}_1^{\mathrm{p}})^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & -x_0^{\mathrm{p}} (\bar{\mathbf{X}}_2^{\mathrm{p}})^{\mathrm{T}} & -x_0^{\mathrm{p}} (\bar{\mathbf{X}}_1^{\mathrm{p}})^{\mathrm{T}} \\ \mathbf{0}^{\mathrm{T}} & (\bar{\mathbf{X}}_2^{\mathrm{p}})^{\mathrm{T}} \otimes (\bar{\mathbf{X}}_1^{\mathrm{p}})^{\mathrm{T}} & -y_0^{\mathrm{p}} (\bar{\mathbf{X}}_2^{\mathrm{p}})^{\mathrm{T}} & -y_0^{\mathrm{p}} (\bar{\mathbf{X}}_1^{\mathrm{p}})^{\mathrm{T}} \end{pmatrix}, \\ \mathbf{W} &= \begin{pmatrix} \mathbf{A}_1^{\mathrm{T}} & \mathbf{A}_2^{\mathrm{T}} & \mathbf{A}_3^{\mathrm{T}} & \mathbf{B}_1^{\mathrm{T}} & \mathbf{B}_2^{\mathrm{T}} & \mathbf{B}_3^{\mathrm{T}} & \mathbf{N}_3^{\mathrm{T}} & \mathbf{M}_3^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}}, \end{aligned} \quad (8)$$

and $\mathbf{A}_i^{\mathrm{T}}$ and $\mathbf{B}_i^{\mathrm{T}}$ denote the ith row for $\mathbf{A}$ and $\mathbf{B}$ respectively. The symbol $\otimes$ denotes kronecker tensor product [10]. If each element in $\mathbf{W}$ is considered as an independent variable, there are 24 unknowns in total. Since one incident ray will provide two constraints, at least 12 incident rays are needed to solve all the unknowns. Suppose we select m points $\bar{\mathbf{X}}_{ij}^{\mathrm{p}} = (x_{ij}^{\mathrm{p}}, y_{ij}^{\mathrm{p}}, 1)^{\mathrm{T}}$, where $0 \le i \le 2$ and $1 \le j \le m$, to solve the unknowns. We can formulate the problem of finding $\mathbf{W}$ as solving an over-constrained linear system:

$$\mathbf{E}\mathbf{W} = \mathbf{0}, \quad (9)$$

where $\mathbf{E}$ is defined as

$$\begin{pmatrix} (\bar{\mathbf{X}}_{21}^{\mathrm{p}})^{\mathrm{T}} \otimes (\bar{\mathbf{X}}_{11}^{\mathrm{p}})^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & -x_{01}^{\mathrm{p}} (\bar{\mathbf{X}}_{21}^{\mathrm{p}})^{\mathrm{T}} & -x_{01}^{\mathrm{p}} (\bar{\mathbf{X}}_{11}^{\mathrm{p}})^{\mathrm{T}} \\ \mathbf{0}^{\mathrm{T}} & (\bar{\mathbf{X}}_{21}^{\mathrm{p}})^{\mathrm{T}} \otimes (\bar{\mathbf{X}}_{11}^{\mathrm{p}})^{\mathrm{T}} & -y_{01}^{\mathrm{p}} (\bar{\mathbf{X}}_{21}^{\mathrm{p}})^{\mathrm{T}} & -y_{01}^{\mathrm{p}} (\bar{\mathbf{X}}_{11}^{\mathrm{p}})^{\mathrm{T}} \\ \vdots & \vdots & \vdots & \vdots \\ (\bar{\mathbf{X}}_{2m}^{\mathrm{p}})^{\mathrm{T}} \otimes (\bar{\mathbf{X}}_{1m}^{\mathrm{p}})^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & -x_{0m}^{\mathrm{p}} (\bar{\mathbf{X}}_{2m}^{\mathrm{p}})^{\mathrm{T}} & -x_{0m}^{\mathrm{p}} (\bar{\mathbf{X}}_{1m}^{\mathrm{p}})^{\mathrm{T}} \\ \mathbf{0}^{\mathrm{T}} & (\bar{\mathbf{X}}_{2m}^{\mathrm{p}})^{\mathrm{T}} \otimes (\bar{\mathbf{X}}_{1m}^{\mathrm{p}})^{\mathrm{T}} & -y_{0m}^{\mathrm{p}} (\bar{\mathbf{X}}_{2m}^{\mathrm{p}})^{\mathrm{T}} & -y_{0m}^{\mathrm{p}} (\bar{\mathbf{X}}_{1m}^{\mathrm{p}})^{\mathrm{T}} \end{pmatrix}.$$

Consider the structure of $\mathbf{E}$ resulted from the colinear constraint. Since the 21st column and 24th column of $\mathbf{E}$ are identical, the nullity of $\mathbf{E}$ is two for non-zero solutions. In order to solve it, we first apply SVD to get a solution space spanned by two solution basis vectors, $\mathbf{d}_1$ and $\mathbf{d}_2$. $\mathbf{W}$ is then parameterized as

$$\mathbf{W} = \alpha(\mathbf{d}_1 + \beta \mathbf{d}_2). \quad (10)$$

Now there are 26 unknowns in total. By combining Eq. (6), Eq. (7) Eq. (8), and Eq. (10), $\mathbf{M}$, $\mathbf{N}$, α and β will be involved in 18 bilinear and 6 linear equations by enforcing the element-wise equality. Furthermore, $rank(\mathbf{N}_3 \mathbf{M}_1^{\mathrm{T}}) \le 1$ and $rank(-\mathbf{N}_1 \mathbf{M}_3^{\mathrm{T}}) \le 1$ since $rank(\mathbf{N}_3 \mathbf{M}_1^{\mathrm{T}}) \le min(rank(\mathbf{N}_3), rank(\mathbf{M}_1)) = 1$. Thus,

$$\begin{aligned} rank(\mathbf{A}) &= rank(\mathbf{N}_3 \mathbf{M}_1^{\mathrm{T}} - \mathbf{N}_1 \mathbf{M}_3^{\mathrm{T}}) \\ &\le rank(\mathbf{N}_3 \mathbf{M}_1^{\mathrm{T}}) + rank(-\mathbf{N}_1 \mathbf{M}_3^{\mathrm{T}}) \\ &\le 2. \end{aligned} \quad (11)$$

We can also show $rank(\mathbf{B}) \leq \mathbf{2}$ in a similar manner. Therefore, not all of the obtained constraints are independent and new constraints should be applied for solving all the unknowns. Due to the fact that the first and second columns of $\mathbf{M}$ and $\mathbf{N}$ are the first two columns of $\mathbf{R}^1$ and $\mathbf{R}^2$ respectively, the orthonomality property will provide 6 more constraints, which lead to a closed-form solution for the unknown motion parameters and the two scale parameters. We use the symbolic Math Toolbox in Matlab to get an analytical solution for all the unknowns.

4. Pose Estimation: Camera

4.1. Line Incidence in Plücker Space

In order to formulate line intersections algebraically, we adopt the 6-vector Plücker line coordinates representation for lines in P^3 [9]. Given a line $\mathcal{L}$ defined by point $\mathbf{P} = (p_x, p_y, p_z, 1)^{\mathrm{T}}$ and point $\mathbf{Q} = (q_x, q_y, q_z, 1)^{\mathrm{T}}$, its Plücker line coordinates representation is given by

$$\mathcal{L} = \begin{pmatrix} l_0 \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \end{pmatrix} = \begin{pmatrix} p_x q_y - q_x p_y \\ p_x q_z - q_x p_z \\ p_x - q_x \\ p_y q_z - q_y p_z \\ q_y - p_y \\ p_z - q_z \end{pmatrix}. \tag{12}$$

With this notation, a line in P^3 is mapped to a homogenous 6-vector in the 5 dimensional Plücker line coordinates space. Suppose another line $\hat{\mathcal{L}}$ is the joins of points $\mathbf{G}$ and $\mathbf{H}$. Lines $\mathcal{L}$ and $\hat{\mathcal{L}}$ intersect if and only if

$$\begin{aligned} det(\mathbf{P}, \mathbf{Q}, \mathbf{G}, \mathbf{H}) &= l_0\hat{l}_5 + \hat{l}_0 l_5 + l_1\hat{l}_4 + \hat{l}_1 l_4 + l_2\hat{l}_3 + \hat{l}_2 l_3 \\ &= (\mathcal{L}|\hat{\mathcal{L}}) \\ &= 0. \end{aligned} \tag{13}$$

$det(\mathbf{P}, \mathbf{Q}, \mathbf{G}, \mathbf{H})$ represents the determinant for a matrix composed of vectors in the brackets. Eq. (13) gives an algebraic constraint for line intersection, and it will be used in the following section to derive a solution for the unknown poses of the reference plane relative to the camera.

4.2. Closed-form Solution

Consider the configuration in Fig. 4 and assume the world coordinate system coincides with the local coordinate system of the plane Π_0. Let the rigid body transformation from the camera coordinate system to the world coordinate system be described by the rotation matrix $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)^{\mathrm{T}}$ and translation vector $\mathbf{T} = (t_x, t_y, t_z)^{\mathrm{T}}$. Given the relative motion of the reference plane estimated in Section 3, $\mathbf{X}_0$, $\mathbf{X}_1$ and $\mathbf{X}_2$ are known and they define the incident ray $\mathcal{I}$. Moreover, the camera center, $\mathbf{T}$, and one point on the visual ray , $\mathbf{Rv} + \mathbf{T}$, defines the visual ray $\mathcal{V}$, where $\mathbf{v}$ represents the visual ray in the camera coordinate

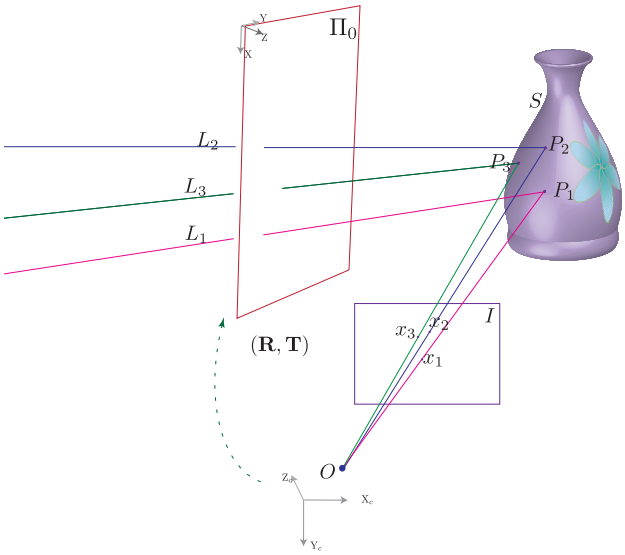

Figure 4. Line incidence constraint for solving the relative pose between the world coordinate system and the camera coordinate system. L_1, L_2 and L_3 are incident rays which will intersect their visual rays on the specular surface S at P_1, P_2, and P_3 respectively. Assume the world coordinate system coincides with the local coordinate system on Π_0. $\mathbf{R}$ and $\mathbf{T}$ denote the relative pose between Π_0 and the camera.

system. The Plücker line coordinates for $\mathcal{I}$ and $\mathcal{V}$ are given as follows

$$\mathcal{I} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} x_0 y_1 - x_1 y_0 \\ x_0 z_1 - x_1 z_0 \\ x_0 - x_1 \\ y_0 z_1 - y_1 z_0 \\ y_1 - y_0 \\ z_0 - z_1 \end{pmatrix}, \tag{14}$$

$$\mathcal{V} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} = \begin{pmatrix} t_x \mathbf{r}_2^{\mathrm{T}}\mathbf{v} - t_y \mathbf{r}_1^{\mathrm{T}}\mathbf{v} \\ t_x \mathbf{r}_3^{\mathrm{T}}\mathbf{v} - t_z \mathbf{r}_1^{\mathrm{T}}\mathbf{v} \\ -\mathbf{r}_1^{\mathrm{T}}\mathbf{v} \\ t_y \mathbf{r}_3^{\mathrm{T}} - t_z \mathbf{r}_2^{\mathrm{T}} \\ \mathbf{r}_2^{\mathrm{T}}\mathbf{v} \\ -\mathbf{r}_3^{\mathrm{T}}\mathbf{v} \end{pmatrix}. \tag{15}$$

According to Eq. (13), the intersection of $\mathcal{I}$ and $\mathcal{V}$ is defined as

$$a_0 b_5 + a_5 b_0 + a_1 b_4 + b_1 a_4 + a_2 b_3 + a_3 b_2 = 0. \tag{16}$$

It is noted that only line $\mathcal{V}$ involves the unknown parameters. Thus, we only substitute $\mathcal{V}$ in Eq. (16) and obtain the constraint as

$$\begin{aligned} &-a_0 \mathbf{v}^{\mathrm{T}}\mathbf{r}_3 + a_5 \mathbf{v}^{\mathrm{T}}(t_x \mathbf{r}_2 - t_y \mathbf{r}_1) \\ &+a_1 \mathbf{v}^{\mathrm{T}}\mathbf{r}_2 + a_4 \mathbf{v}^{\mathrm{T}}(t_x \mathbf{r}_3 - t_z \mathbf{r}_1) \\ &-a_3 \mathbf{v}^{\mathrm{T}}\mathbf{r}_1 + a_2 \mathbf{v}^{\mathrm{T}}(t_y \mathbf{r}_3 - t_z \mathbf{r}_2) = 0. \end{aligned} \tag{17}$$

Let

$$\begin{aligned}\mathbf{C}_1 &= t_x\mathbf{r}_2 - t_y\mathbf{r}_1,\\ \mathbf{C}_2 &= t_x\mathbf{r}_3 - t_z\mathbf{r}_1,\\ \mathbf{C}_3 &= t_y\mathbf{r}_3 - t_z\mathbf{r}_2.\end{aligned}$$

Eq. (17) can be rewritten as

$$\mathbf{KF} = \mathbf{0}, \tag{18}$$

where

$$\begin{aligned}\mathbf{K} &= (a_5\mathbf{v}^{\mathrm{T}}, a_4\mathbf{v}^{\mathrm{T}}, a_2\mathbf{v}^{\mathrm{T}}, -a_3\mathbf{v}^{\mathrm{T}}, a_1\mathbf{v}^{\mathrm{T}}, -a_0\mathbf{v}^{\mathrm{T}}),\\ \mathbf{F} &= (\mathbf{C}_1^{\mathrm{T}}, \mathbf{C}_2^{\mathrm{T}}, \mathbf{C}_3^{\mathrm{T}}, \mathbf{r}_1^{\mathrm{T}}, \mathbf{r}_2^{\mathrm{T}}, \mathbf{r}_3^{\mathrm{T}})^{\mathrm{T}}.\end{aligned} \tag{19}$$

Suppose $\mathcal{I}_{\mathrm{i}} = (a_{0i}, a_{1i}, a_{2i}, a_{3i}, a_{4i}, a_{5i})^{\mathrm{T}}$ and $\mathbf{v}_i$, $1 \le i \le n$, denote the incident rays and visual rays respectively. In order to solve Eq. (18), at least 18 incident rays are required. The problem of solving $\mathbf{R}$ and $\mathbf{T}$ is formulated as solving the following over-constrained linear equations

$$\begin{pmatrix} \hat{\mathcal{I}}_1 \otimes \mathbf{v}_1^{\mathrm{T}} \\ \hat{\mathcal{I}}_2 \otimes \mathbf{v}_2^{\mathrm{T}} \\ \vdots \\ \hat{\mathcal{I}}_n \otimes \mathbf{v}_n^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \mathbf{C}_3 \\ \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{pmatrix} = \mathbf{0}, \tag{20}$$

$$\begin{pmatrix} \mathbf{r}_2 & -\mathbf{r}_1 & \mathbf{0} \\ \mathbf{r}_3 & \mathbf{0} & -\mathbf{r}_1 \\ \mathbf{0} & \mathbf{r}_3 & -\mathbf{r}_2 \end{pmatrix} \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \mathbf{C}_3 \end{pmatrix}, \tag{21}$$

where $\hat{\mathcal{I}}_i = (a_{5i}, a_{4i}, a_{2i}, -a_{3i}, a_{1i}, -a_{0i})$. We apply SVD to get a solution space spanned by one basis vector $\mathbf{F} = \gamma\mathbf{e}$, which is up to an unknown scale γ. Since $\mathbf{F}$ includes the vectorized rotation matrix, γ can be solved by enforcing the orthogonality properties of the rotation matrix. However, due to noisy data, the obtained linear least square solution may not satisfy the orthogonality constraint for the rotation matrix. The strategy proposed in [21] is applied to approximate the noisy matrix $\mathbf{R}$ by a rotation matrix. $\mathbf{T}$ can then be obtained by solving Eq. (21).

5. Implementation Issues

5.1. Sparse Reflection Correspondences

We adopt the standard Gray code encoding strategy and use a standard computer monitor as the reference plane to display Gray code patterns, once original and once inverted [19]. The encoding patch unit is in square shape. The total number of images taken for each pose of the reference plane is therefore twice the binary resolution in each direction. As stated in [20], the resolution of the codes and the width of the lowest order stripes must be chosen according to the shape of the specular object and the resolution of the camera. Too high resolution codes tend to be blurred out and become unusable, whereas too coarse ones lack in precision. Generally, a patch of pixels in the camera image corresponds to a patch of 3D points with the same code on the reference plane. In order to get more accurate 3D reflection correspondences, we first extract from the reflection image of the encoding patch corners on the reference plane at its initial pose. Note that these extracted corner positions will not in general reflect encoding patch corners of the reference plane at the second pose. The coordinates of the pixels reflecting the encoding patch corners in the second pose are also extracted. Since we will not choose too coarse encoding resolution and the specular object is assumed to be smooth and locally planar, linear interpolation will be accurate enough for approximating the true reflection correspondences on the reference plane at the second pose. The reflection correspondences are extracted in a similar way for the reference plane at its third pose.

5.2. Shape Recovery for Specular Surface

Given the estimated motion of the reference plane, points on the specular surface can be recovered by ray triangulation between the visual rays and the corresponding incident rays (Fig. 2). However, due to the noise induced by the approximation described in the previous section, these rays may not intersect with each other in 3D space. This error is essentially caused by the inaccurate estimation of the incident rays. In this paper, we take the point closest to both the visual ray and the corresponding incident ray as an approximation to the surface point. Note that if the distance between the reference plane before and after its motion is large enough, the approximation of the correspondences on the reference plane will only lead to a small error. Further, if the angle between the incident ray and the visual ray is around $50°$, ray triangulation will give more accurate result. This angle measurement can be applied as a criteria to remove outliers.

6. Experimental Results

We validated our proposed method on both synthetic and real shiny mirror-like surfaces. For synthetic experiment, we used a parametric surface formed by two spheres as the specular object. A shiny spoon together with a sphere is used as the specular object in the real experiment. Experimental results with noise analysis and comparisons with the ground truth are detailed in the following sections.

6.1. Synthetic Experiment

We used two reflective spheres with parametric equations

$$\begin{cases} x_1^2 + y_1^2 + (z_1 - 10)^2 = 9 \ \text{ and} \\ (x_2 + 1)^2 + (y_2 + 0.5)^2 + (z_2 - 11)^2 = 25 \end{cases}$$

to represent a general surface and assumed no interreflection occurred on the surface. A reference plane displaying a set of Gray code patterns was placed at three different positions. A synthetic perspective camera with a focal length of $24mm$ was used to capture the reflections of the reference plane on the specular surface.

In practice, noises usually originate from inaccuracy in finding the reflection correspondences on the reference planes and the feature detection process. We carried out the noise analysis by adding $2D$ uniformly distributed noises to planar coordinates of the extracted reflection correspondences on the reference plane. The noise level ranges from 0 to 10 percent of the edge length of the encoding patch. We choose 4 different encoding patch size: $0.01cm, 0.04cm, 0.07cm$ and $0.1cm$. For each encoding patch size and noise level, 100 independent trials were performed. The motion estimation accuracy was evaluated by comparing the estimated rotation and translation with the ground truth data. The rotation estimation error was measured by the relative rotation angle (*rotErr*) between the estimated rotation and the ground truth. The translation estimation error was measured by translation scale error $trasErr = \frac{\|\mathbf{T_{est}} - \mathbf{T_{gt}}\|_2^2}{\|\mathbf{T_{gt}}\|_2^2}$, and translation direction error $tradErr = angle(\mathbf{T_{est}}, \mathbf{T_{gt}})$, where $\mathbf{T_{est}}$ and $\mathbf{T_{gt}}$ represent the estimated translation vector and the ground truth respectively, $\|\cdot\|_2$ denotes the l_2 norm, and $angle(\mathbf{T_{est}}, \mathbf{T_{gt}})$ denotes the intersection angle between $\mathbf{T_{est}}$ and $\mathbf{T_{gt}}$. Since the motion estimation algorithm proposed in this paper consists of two steps, the estimation errors were evaluated differently for the three different plane positions. Suppose the world coordinate system coincides with the coordinate system on Π_0. Π_1 and Π_2 represent two new positions of the moving reference plane. The estimated motion for Π_0 was evaluated by the relative pose between Π_0 and the camera. As for Π_1 and Π_2, their estimated poses were evaluated by their poses relative to Π_0.

The motion comparison results are shown in Fig. 5. It can be seen that the error for motion estimation among multiple reference plane positions varies almost linearly with noises, which shows that the colinearity properties for motion estimation in Section 3 are more robust to the change of the reflection correspondence errors. The estimated motion of Π_0 relative to the camera varies greatly with errors in the reflection correspondences. Therefore, the reflection correspondences should be estimated as accurate as possible for getting more accurate motions.

6.2. Real Experiment

We conducted a real experiment on two specular shiny surfaces, namely a *sphere* and a *spoon*. A Dell 17-inch LCD was used to display Gray code patterns. We used a CANON 40D camera equipped with a $24mm$ lens to capture the images of the moving reference plane reflected on the specular object. Another NIKON D3100 camera was used to view the reference plane directly for obtaining the ground truth data of the motion of the reference plane for comparison purpose. The intrinsic parameters of the two cameras were calibrated using [5]. The LCD screen that served as the reference plane was placed at three different positions. A set of 12 Gray code pattern images along with their 12 inverse images, with a resolution of 3888×2592, were reflected on the shiny surfaces and captured by the camera. The encoding patch was chosen according to the size of the object and the distance between the object and the reference plane, and it was in a square shape with a size of $0.59 \times 0.59cm^2$ for the current experiment. Reflection correspondences on the reference plane were then obtained from the encoding patch corners in the reflection images using interpolation as described in the previous section. Given the reflection correspondences, poses of the reference plane and the camera were obtained using the method detailed in Section 3 and Section 4. The motion estimation result was verified by displaying the sparse structure of the specular object (see Fig. 6). This qualitative evaluation can demonstrate the effectiveness of the proposed method. The proposed method was also evaluated quantitatively. We used the same error measures as defined in Section 6.1 for evaluating the accuracy of the estimated motion, and Table 1 shows the evaluation results.

Ref. Planes	*rotErr*[°]	*trasErr*[%]	*tradErr*[°]
Π_0	3.5221	2.7119	2.0052
Π_1	0.8789	0.7436	0.2627
Π_2	0.7482	0.2589	0.1705

Table 1. Motion estimation errors compared with ground truth for the real experiment.

7. Discussions and Conclusions

It is worthy of discussing the advantages and disadvantages of the proposed method. Our method can estimate the motion of the reference planes for specular surface recovery by only observing its reflections on the specular surface, which is different from the existing calibration method by using a mirror [20] or constraining the motion of the reference plane and knowing its initial pose [12].

In the proposed method, the motion estimation procedures will greatly depend on the accuracy of the extraction of reflection correspondences for pixel and scene points. Most of the current encoding strategies, if not all, cannot achieve one-to-one correspondences. In our current work, we only focus on motion estimation by *sparse reflection correspondences*. We use linear interpolation to obtain an approximation of the true correspondences on the reference plane, which implies that the surface should be smooth and locally planar. In order to achieve more accurate one-to-

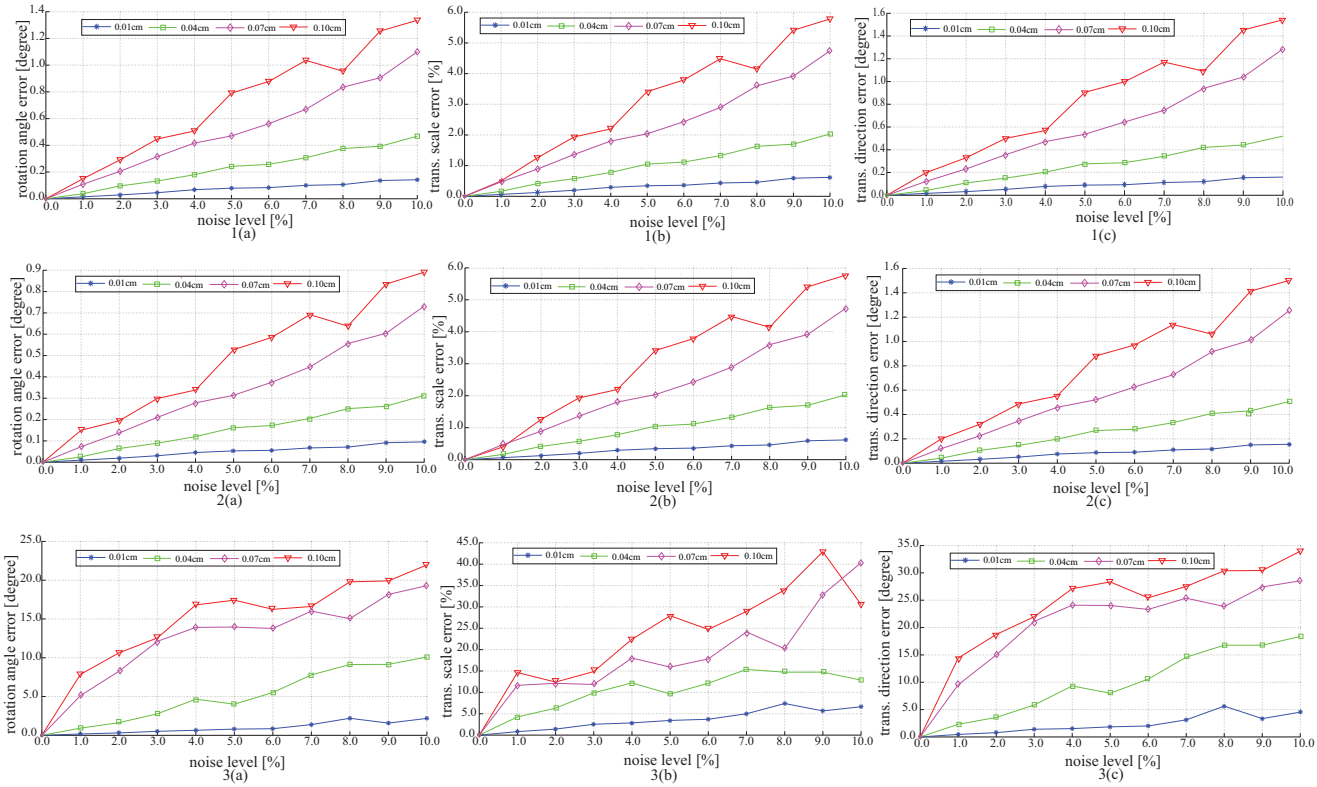

Figure 5. Motion estimation errors with respect to the choice of width of the encoding patch and different level of noises for the reflection correspondences. 1(a), 1(b), and 1(c) show the estimated motion errors for Π_1 relative to Π_0, where 1(a) shows the rotation estimation error defined by $rotErr$, 1(b) shows the translation scale errors defined by $trasErr$ and 1(c) shows the translation direction error defined by $tradErr$. 2(a), 2(b), and 2(c) show the estimated motion errors for Π_2 relative to Π_0. 3(a), 3(b), and 3(c) show the estimated motion errors for Π_0 relative to the camera.

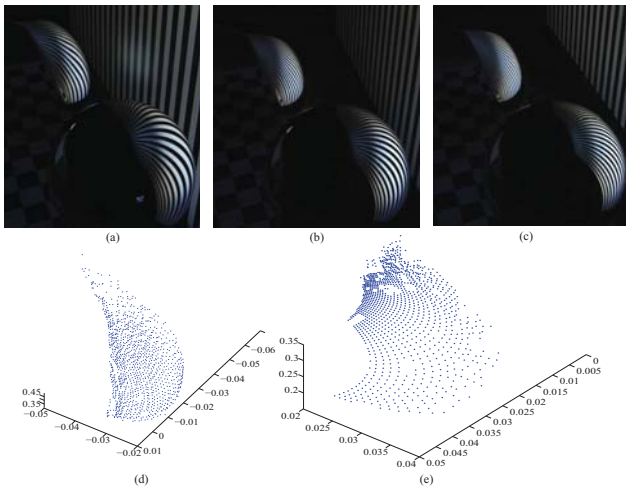

Figure 6. Structure recovery for specular object. (a), (b) and (c) show the reflections of the Gray code pattern under three different poses of the reference plane. (d) and (e) show the recovered sparse structure for the *spoon* and *sphere*. The shape can be well recovered by the proposed method.

one correspondences and sub-pixel accuracy, we will focus on dense shape recovery and formulate the correspondence problem in the framework of optimization in our future work.

Given the estimated motion, direct ray triangulation will lead to large error for 3D points with small intersection angles between the incident ray and visual ray. In the current work, the recovered 3D points with large errors are considered as outliers and removed. Nevertheless, if dense recovery is achieved, those points can be recovered by imposing the smoothness constraint from its neighbor points and normal information.

In this paper, our proposed method can be considered as *specular structure from motion*, which can automatically estimate the relative poses of the moving reference plane and the camera, as well as the structure of the specular object with sparse reflection correspondences between the pixels and scene points. In this regard, we believe that our proposed method will make the current specular surface recovery approach achieve a great step towards a more practical method. As for the future work, we will try to solve the calibration problems for specular surface recovery by using lines and try to derive the intrinsics of the camera from the

observed reflection information.

References

[1] Y. Adato, Y. Vasilyev, O. Ben-Shahar, and T. Zickler. Toward a theory of shape from specular flow. In *International Conference on Computer Vision*, pages 1–8, Brazil, Oct. 2007. 1

[2] A. Blake and G. Brelstaff. Geometry from specularities. In *International Conference on Computer Vision*, pages 394–403, 1988. 1

[3] T. Bonfort and P. Sturm. Voxel carving for specular surfaces. In *International Conference on Computer Vision*, pages 691–696, Nice, France, october 2003. 1

[4] T. Bonfort, P. Sturm, and P. Gargallo. General specular surface triangulation. In *Asian Conference on Computer Vision*, volume II, pages 872–881, Hyderabad,India, Jan. 2006. 1

[5] J.-Y. Bouguet. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`. 6

[6] G. D. Canas, Y. Vasilyev, Y. Adato, T. Zickler, S. Gortler, and O. Ben-Shahar. A linear formulation of shape from specular flow. In *International Conference on Computer Vision*, pages 191–198, Kyoto, Sep. 2009. 1

[7] M. Chen and J. Arvo. Theory and application of specular path perturbation. *ACM Transactions on Graphics*, 19(4):246–278, 2000. 1

[8] T. A. Fleming, R. W. and E. H. Adelson. Specular reflections and the perception of shape. *Journal of Vision*, 4(9):798–820, 2004. 1

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 4

[10] R. A. Horn. *Topics in Matrix Analysis*. Cambridge University Press New York, NY, USA, 1986. 3

[11] K. N. Kutulakos and E. Steger. A theory of refractive and specular 3d shape by light-path triangulation. *International Journal of Computer Vision*, 76(1):13–29, 2008. 1

[12] M. Liu, K.-Y. K. Wong, Z. Dai, and Z. Chen. Specular surface recovery from reflections of a planar pattern undergoing an unknown pure translation. In *Proc. Asian Conference on Computer Vision*, volume 2, pages 647–658, Queenstown, New Zealand, Nov. 2010. 2, 6

[13] D. Nehab, T. Weyrich, and S. Rusinkiewicz. Dense 3d reconstruction from specularity consistency. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, Alaska, Jun. 2008. 1

[14] M. Oren and S. K. Nayar. A theory of specular surface geometry. *International Journal of Computer Vision*, 24(2):105–124, 1997. 1

[15] S. Roth and M. J. Black. Specular flow and the recovery of surface structure. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1869–1876, NewYork, Jun. 2006. 1

[16] A. C. Sankaranarayanan, A. Veeraraghavan, O. Tuzel, and A. Agrawal. Specular surface reconstruction from sparse reflection correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1245–1252, San Francisco, Jun. 2010. 1

[17] S. Savarese and P. Perona. Local analysis for 3d reconstruction of specular surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 738–745, Los Alamitos, CA, USA, 2001. 1

[18] S. Savarese and P. Perona. Local analysis for 3d reconstruction of specular surfaces - part ii. In *European Conference on Computer Vision-Part II*, pages 759–774, London, UK, May 2002. 1

[19] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–202, 2003. 5

[20] P. Sturm and T. Bonfort. How to compute the pose of an object without a direct view. In *Asian Conference on Computer Vision*, pages 21–31, Jan. 2006. 1, 5, 6

[21] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000. 5

[22] A. Zisserman, P. Giblin, and A. Blake. The information available to a moving observer from specularities. *Image Vision Computing*, 7(1):38–42, 1989. 1