How to Compute the Pose of an Object without a Direct View?

Peter Sturm, Thomas Bonfort

How to Compute the Pose of an Object without a Direct View?

Peter Sturm and Thomas Bonfort

INRIA Rhône-Alpes, 38330 Montbonnot St Martin, France
{Peter.Sturm, Thomas.Bonfort}@inrialpes.fr

Abstract. We consider the task of computing the pose of an object relative to a camera, for the case where the camera has no direct view of the object. This problem was encountered in work on vision-based inspection of specular or shiny surfaces, that is often based on analyzing images of calibration grids or other objects, reflected in such a surface. A natural setup consists thus of a camera and a calibration grid, put side-by-side, i.e. without the camera having a direct view of the grid. A straightforward idea for computing the pose is to place planar mirrors such that the camera sees the calibration grid's reflection. In this paper, we consider this idea, describe geometrical properties of the setup and propose a practical algorithm for the pose computation.

1 Introduction

Consider a calibration grid or any other known object, planar or not, and a camera. We would like to determine their relative pose, but for the case where **the camera does not see the object directly**. This is an unusual setting, but it is quite natural for the task of reconstructing specular or shiny surfaces, as explained in the following. Modeling of specular or shiny surfaces is an important application in inspection of industrial parts, especially in the car manufacturing industry (control of wind shields and bodywork) but also in the control of optical lenses or mirrors, glasses of watches etc. Vision-based reconstruction of specular surfaces is usually based on acquiring images of known patterns or light sources, reflected in the surface to be reconstructed [3, 4, 6, 7, 11, 14].

It is thus rather natural to place the camera and pattern such that the camera does not have a direct view of the latter, or at most sees a small part of it. We have proposed practical approaches for the reconstruction of specular surfaces where such an arrangement is indeed used. The question of how to compute the pose of an object without a direct view is thus important for us and in addition scientifically appealing.

Our initial solution consisted in attaching the pattern rigidly to the camera and to move the two to a few locations. During this, the camera acquired images of a calibration grid, and a secondary camera (static) acquired images of our pattern. With this input, the pattern's 3D trajectory was computed as well as the main camera's one. By registering the two trajectories into a common coordinate frame, along the lines of [2] and of the classical hand-eye calibration problem, we finally computed the pose of the pattern relative to the main camera. This approach was found to be too cumbersome in practice. A second camera is required and especially, having to move the camera–pattern pair is not desirable, as we currently use an LCD monitor to produce the pattern(s).

We are thus aiming at a lighter procedure. A natural idea is to proceed as follows: place a planar mirror in different positions in front of the camera such that the pattern's reflection is seen, and acquire images. The question arises if this input is sufficient to solve our pose problem, and if yes, how many positions of the planar mirror are required? We show in this paper that our pose problem can be solved up to 1 degree of freedom from two positions, and can be fully solved from three or more positions.

2 Background

2.1 Camera model

We consider perspective projection as camera model. The projection of 3D points is modeled by a 3×4 projection matrix $\mathsf{P} = \mathsf{K}\mathsf{R}\left(\mathtt{I}|-\mathbf{t}\right)$, where K is the usual 3×3 calibration matrix with the camera's intrinsic parameters, and the orthogonal matrix R and the vector $\mathbf{t}$ represent camera orientation and position. For simplicity, we assume that the camera is calibrated, i.e. that K is known (this will be relaxed later). We thus directly work with geometric image coordinates, i.e. consider that 3D points $\mathbf{Q}$ are projected to image points $\mathbf{q}$ via the canonical projection matrix $\mathbf{q} \sim \mathsf{R}\left(\mathtt{I}|-\mathbf{t}\right)\mathbf{Q}$. 2D and 3D points are expressed in homogeneous coordinates and $\sim$ means equality of vectors or matrices, up to scale.

2.2 Pose computation

A classical task of photogrammetry and computer vision is to compute the pose of a calibrated camera, relative to an object of known structure. In this work, we use planar reference objects. There exist many algorithms for the planar pose problem; we use [10].

2.3 Reflections in planes

Consider a plane $\mathit{\Pi} = \left(\mathbf{n}^\mathsf{T}, d\right)^\mathsf{T}$ in 3-space, i.e. consisting of points satisfying the equation $n_1X + n_2Y + n_3Z + d = 0$. In the following, we will always suppose that the plane's normal vector is of unit norm. The reflection in $\mathit{\Pi}$ can be represented by the following transformation matrix:

$$\mathsf{S} = \begin{pmatrix} \mathtt{I} - 2\mathbf{n}\mathbf{n}^\mathsf{T} & -2d\mathbf{n} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}$$

Let us denote the upper left 3×3 matrix of S by $\bar{\mathsf{S}}$. It is an orthogonal matrix, with determinant -1 (whereas a rotation matrix has determinant $+1$). Further, it has $+1$ as double eigenvalue and -1 as single eigenvalue. The plane normal $\mathbf{n}$ is an eigenvector of $\bar{\mathsf{S}}$ to the eigenvalue -1. Note also that $\mathsf{S}^{-1} = \mathsf{S}$.

2.4 Planar motion and fixed-axis rotation

Planar motion usually means a translation in some direction, followed by a rotation about an axis that is orthogonal to the translation direction. Such a motion can always be expressed as just a rotation about an axis that is parallel to the above rotation axis; we thus prefer to call such motions *fixed-axis rotations*.

It is easy to show that any euclidian transformation that preserves some line point-by-point, is a fixed-axis rotation, whose axis is that line.

Let the axis be represented by its direction vector $\mathbf{D}$ and a footpoint $\mathbf{A}$ such that $\mathbf{A} + \lambda\mathbf{D}$ represents the points (in non-homogeneous coordinates) on the axis. Any finite point on the axis can serve as footpoint; we always choose the one that is "orthogonal" to $\mathbf{D}$: $\mathbf{A}^\mathsf{T}\mathbf{D} = 0$. This is the point on the axis that is closest to the origin.

Let α be the angle of rotation and R be the rotation matrix representing rotation by α about $\mathbf{D}$. Then, the 4×4 matrix representing the complete fixed-axis rotation, is:

$$\mathsf{T} = \begin{pmatrix} \mathtt{I} & \mathbf{A} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} \mathsf{R} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} \mathtt{I} & -\mathbf{A} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} = \begin{pmatrix} \mathsf{R} & \mathbf{A} - \mathsf{R}\mathbf{A} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}$$

2.5 Reflection in two planes

Consider successive reflections in two planes. It can be shown that this is a fixed-axis rotation, with the intersection line of the two planes as rotation axis: the transformation preserves the intersection line of the two planes point-by-point, and thus is a fixed-axis rotation.

Further, the rotation angle is twice the angle between the two planes. This is also easy to see: let the transformation be the sequence $\mathsf{S}_2\mathsf{S}_1$ of reflections in two planes. Let us apply this transformation to the point at infinity $(\mathbf{n}_1^\mathsf{T}, 0)^\mathsf{T}$, i.e. the normal direction of the first plane. This is a fixed point of S_1, hence the transformation gives the point's reflection in S_2. The angle between the original point at infinity, and the transformed one, i.e. the fixed-axis rotation angle, is thus twice the angle between the original point at infinity and the second reflection plane. Hence, as said above, the sequence of reflections in two planes is a fixed-axis rotation, whose angle is twice the angle between the planes.

2.6 Horopter

The horopter of a stereo system is the set of 3D points that project to points with identical coordinates in the two cameras. Let P_1 and P_2 be the two cameras' projection matrices. The horopter thus consists of all 3D points $\mathbf{Q}$ with $\mathsf{P}_1\mathbf{Q} \sim \mathsf{P}_2\mathbf{Q}$. This is in general a quartic curve. If the two cameras have identical calibration and are separated by a fixed-axis rotation, then the horopter "degenerates" into the union of a straight line and a circle: the motion's rotation axis and the circle in the motion plane that contains the two optical centers and that cuts the rotation axis [5].

3 Outline of the Proposed Approach

We consider a camera and an object in fixed position, put a planar mirror in the scene in n different positions, and take an image for each of those. We suppose that the camera sees the object's reflection in each image. We further suppose that the object's structure is known and that correspondences between object and image points can be obtained.

In the first step of our approach, the views of the reflected object are treated as if they were direct views. We may thus compute a camera pose, from the

camera's calibration and the given point correspondences. This will actually give the pose of a "virtual" camera that is the reflection of the true camera, in the planar mirror (cf. figure 1). Overall, we thus get the pose of n virtual cameras, relative to the object.

In the second step, we try to infer the positions of the planar mirrors. The underlying constraint is that reflecting the virtual cameras in mirrors with the correct positions, will lead to n identical cameras – the true one. We show how the mirror positions can be computed using the above notions of horopter and fixed-axis rotation. We further show that for $n = 2$, the problem can be solved up to 1 degree of freedom, and that with $n > 2$ a unique solution can be found. These steps are described in the following sections.

4 Computing Pose of Virtual Cameras

In the following, we adopt the object's coordinate system as our reference system, in which the pose of true and virtual cameras will be expressed. Let the pose of the true camera be represented by the projection matrix

$$\mathsf{R}\left(\mathtt{I}|-\mathbf{t}\right)$$

Consider now a planar mirror, defined by the plane

$$\mathit{\Pi} = \begin{pmatrix}\mathbf{n}\\ d\end{pmatrix}$$

Object points $\mathbf{Q}$ are projected into the true camera as follows:

$$\mathbf{q} \sim \mathsf{R}\left(\mathtt{I}|-\mathbf{t}\right)\begin{pmatrix}\mathtt{I}-2\mathbf{n}\mathbf{n}^\mathsf{T} & -2d\mathbf{n}\\ \mathbf{0}^\mathsf{T} & 1\end{pmatrix}\mathbf{Q}$$

From point correspondences $(\mathbf{Q}, \mathbf{q})$, we can run any pose computation algorithm and compute the projection matrix of the virtual camera:

$$\mathsf{P} = \mathsf{R}\left(\mathtt{I}|-\mathbf{t}\right)\begin{pmatrix}\mathtt{I}-2\mathbf{n}\mathbf{n}^\mathsf{T} & -2d\mathbf{n}\\ \mathbf{0}^\mathsf{T} & 1\end{pmatrix} = \mathsf{R}\left(\mathtt{I}-2\mathbf{n}\mathbf{n}^\mathsf{T}\right)\left(\mathtt{I}\;2d\mathbf{n}-\left(\mathtt{I}-2\mathbf{n}\mathbf{n}^\mathsf{T}\right)\mathbf{t}\right) \quad (1)$$

One issue needs to be considered: pose algorithms for perspective cameras compute a pose consisting of a rotation and a translation component, whereas the above projection matrix contains a reflection part. What the pose computation will compute is thus a rotation matrix R' and a camera position $\mathbf{t}'$, with:

$$\mathsf{P} \sim \underbrace{\mathsf{R}\left(2\mathbf{n}\mathbf{n}^\mathsf{T}-\mathtt{I}\right)}_{\mathsf{R}'}\left(\mathtt{I}\;-\mathbf{t}'\right) \quad \text{with} \quad -\mathbf{t}' = 2d\mathbf{n}-\left(\mathtt{I}-2\mathbf{n}\mathbf{n}^\mathsf{T}\right)\mathbf{t}$$

Our input for the following steps is thus a set of n projection matrices P_i (we drop the ' above the R_i and $\mathbf{t}_i$):

$$\mathsf{P}_i = \mathsf{R}_i\left(\mathtt{I}\;-\mathbf{t}_i\right)$$

The basic constraint for solving our pose problem is the following (cf. §3): we try to compute n planes $\mathit{\Pi}_i$ and associated reflection matrices S_i such that

$$\forall i, j : \mathsf{P}_i\mathsf{S}_i \sim \mathsf{P}_j\mathsf{S}_j$$

If there is a unique solution for the set of planes, then any $\mathsf{P}_i\mathsf{S}_i$ gives the pose of the true camera. In the above equation, we may actually replace the equality up to scale by a component-wise equality, since the determinants of the left 3×3 submatrices of the $\mathsf{P}_i\mathsf{S}_i$ are all equal to -1. Hence, our constraint becomes:

$$\forall i, j : \mathsf{P}_i\mathsf{S}_i = \mathsf{P}_j\mathsf{S}_j$$

5 Two Mirror Positions

In this section, we investigate what can be done from just two mirror positions. Our basic constraint is:

$$\mathsf{P}_1\mathsf{S}_1 = \mathsf{P}_2\mathsf{S}_2$$

Instead of directly trying to compute the reflections S_1 and S_2, we first concentrate on:

$$\mathsf{P}_1 = \mathsf{P}_2\mathsf{S}_2\mathsf{S}_1^{-1} = \mathsf{P}_2\mathsf{S}_2\mathsf{S}_1$$

We have seen above that the sequence of two reflections gives a fixed-axis rotation. Let us thus compute R and $\mathbf{t}$ in the following euclidian transformation between the two virtual cameras:

$$\mathsf{P}_1 = \mathsf{P}_2 \begin{pmatrix} \mathsf{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}$$

We get $\mathsf{R} = \mathsf{R}_2^\mathsf{T}\mathsf{R}_1$ and $\mathbf{t} = \mathbf{t}_2 - \mathsf{R}_2^\mathsf{T}\mathsf{R}_1\mathbf{t}_1$. In the following, we analyze what R and $\mathbf{t}$ reveal about the individual reflections S_1 and S_2.

Let α be the rotation angle of R. We already know that it equals twice the angle between the two mirror planes. Further, we want to compute the fixed axis (the intersection of the two mirror planes). Let us represent it by its direction $\mathbf{D}$ and a footpoint $\mathbf{A}$, cf. §2.4. The direction $\mathbf{D}$ is identical with the rotation axis of R and can for example be computed as its eigenvector to the eigenvalue $+1$. Let $\mathbf{D}_1$ and $\mathbf{D}_2$ be an orthonormal basis of the complement of $\mathbf{D}$, such that:

$$\mathsf{R}\begin{pmatrix}\mathbf{D}_1 & \mathbf{D}_2\end{pmatrix} = \begin{pmatrix}\mathbf{D}_1 & \mathbf{D}_2\end{pmatrix}\begin{pmatrix}\cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha\end{pmatrix}$$

As for the footpoint $\mathbf{A}$, we compute it as follows. Since we want $\mathbf{A}$ to be "orthogonal" to $\mathbf{D}$, we can parameterize it by two scalars a_1 and a_2:

$$\mathbf{A} = \begin{pmatrix}\mathbf{D}_1 & \mathbf{D}_2\end{pmatrix}\begin{pmatrix}a_1 \\ a_2\end{pmatrix}$$

The translation part of the fixed-axis rotation would thus be (cf. §2.4):

$$\mathbf{A} - \mathsf{R}\mathbf{A} = \begin{pmatrix}\mathbf{D}_1 & \mathbf{D}_2\end{pmatrix}\begin{pmatrix}1-\cos\alpha & \sin\alpha \\ -\sin\alpha & 1-\cos\alpha\end{pmatrix}\begin{pmatrix}a_1 \\ a_2\end{pmatrix}$$

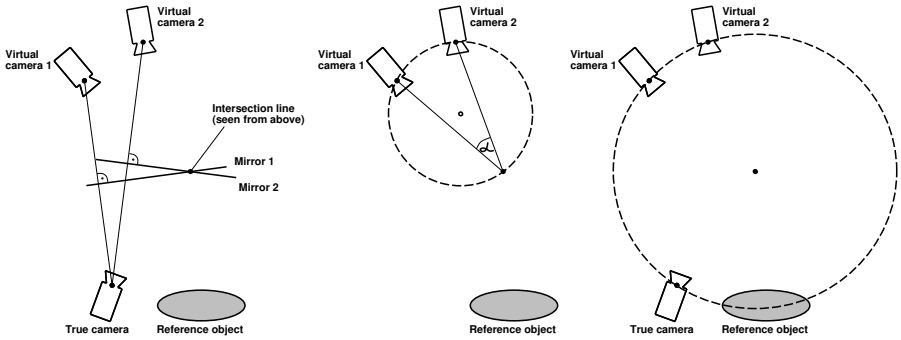

Fig. 1. Illustration of the case of two planar mirrors. **Left:** the virtual cameras are the reflections of the true one in the planar mirrors. **Middle:** the horopter curve of the two virtual cameras is the union of the shown circle and the axis of the fixed-axis rotation, i.e. the mirror planes' intersection axis. Further shown is the angle α of the fixed-axis rotation. **Right:** the true camera pose can be recovered up to one degree of freedom. The reconstructed camera position is only constrained to lie on the shown circle.

In the absence of noise, this would be equal to $\mathbf{t}$. However, with noise, the computed R and $\mathbf{t}$ will in general not exactly represent a fixed-axis rotation. We thus determine a_1 and a_2 that minimize the L_2 norm of:

$$\mathbf{t} - \begin{pmatrix}\mathbf{D}_1 & \mathbf{D}_2\end{pmatrix} \begin{pmatrix} 1-\cos\alpha & \sin\alpha \\ -\sin\alpha & 1-\cos\alpha \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

This is a linear least squares problem, with the following closed-form solution:

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & -\cot\frac{\alpha}{2} \\ \cot\frac{\alpha}{2} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{D}_1^\mathsf{T} \\ \mathbf{D}_2^\mathsf{T} \end{pmatrix} \mathbf{t}$$

So far, we have computed the axis and angle of the fixed-axis rotation being the sequence of S_2 and S_1. What does this tell us about the mirror planes $\mathit{\Pi}_1$ and $\mathit{\Pi}_2$? The axis being the planes' intersection line, we know that both planes must contain it; this determines each plane up to a rotation about the axis. Further, we know the angle between the planes ($\alpha/2$). In addition, not explained here in more detail, we know the "ordering" of the two planes, i.e. the second plane has always to be on the same side of the first (in terms of rotation about their intersection line). Overall, we thus have computed the two mirror planes up to a single unknown. It can be shown (not done due to lack of space) that this can not be reduced further with only two planes.

A geometric illustration of the situation is given in figure 1. For simplicity, we show the scene as seen from the direction of the mirror planes' intersection line. On the right, the ambiguity in the inferred pose of the true camera is shown: its position can lie anywhere on the circle that is centered in the fixed-axis rotation axis, is "orthogonal" to the latter and passes through the two virtual camera positions. Let us call this circle the *pose circle*. For every possible camera position on the pose circle though, the camera's orientation is uniquely defined.

All possible poses for the true camera can be parameterized by an angle β as follows. Any plane containing the axis of the fixed-axis rotation, can be written

$$\Pi \sim \begin{pmatrix} \mathbf{D}_1 & \mathbf{D}_2 \\ -a_1 & -a_2 \end{pmatrix} \begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix}$$

for some β. We can thus parameterize the possible poses of the true camera by β, by reflecting any of the virtual ones, say the first, in the family of planes Π.

6 Three or More Mirror Positions

With three or more mirror positions, our pose problem will in general be solvable. Different approaches are possible. One could for example use the solution of the previous section for all available pairs of mirror positions. The problem could be geometrically expressed as one of finding the common point of a set of circles in 3-space (the circles as sketched in the right part of figure 1). A few special cases need to be discussed:

- Consider the case where the planar mirror is rotated about some axis contained in the mirror plane. In that case, the fixed-axis rotations for pairs of mirror positions will all have the same axis, and the resulting pose circles will all be identical. The pose of the true camera will remain ambiguous. Note that this case also refers to mirror positions that are parallel to each other (this can be seen as rotating the mirror about a line at infinity).
- The case where the mirror moves in such a way that it remains tangent to some cylinder. This implies that the lines of intersection for pairs of positions, will be parallel to one another. Hence, all pose circles lie on the same plane but since we have three or more of them, there will be a single common point: the position of the true camera.
- If there are intersection lines for pairs of mirror positions that are not parallel, then there are pose circles with different supporting planes. It can be shown that there will be pose circles in at least three supporting planes with different normals. Consequently, the set of pose circles can only have a single common point (the intersection point of all supporting planes), meaning that again, the pose problem can be solved.

In the following, we present a less geometrical method for combining results from pairs of mirror positions. Consider mirror position i. Using §5, we can compute the intersection lines of the mirror in that position, with any other position. The plane at position i has to contain all these lines, and can thus be uniquely computed from two or more lines, unless all of them are identical (cf. the above discussion). In the presence of noise, there will not be a plane that exactly contains all these lines, and we perform a fitting procedure, as follows. First, perform a least-squares fit to the direction vectors of all available lines. This will be used as the plane's normal vector. Then, compute the plane position (the scalar d appearing elsewhere in this paper) that minimizes the sum of squared distances to the available footpoints. This method fails if all lines are parallel. An alternative procedure for that case is simple to devise though.

Once mirror plane positions are computed, we reflect the virtual cameras in the corresponding planes, to obtain the true camera's projection matrix. The complete method is summarized in the next section.

7 Complete Approach

1. For each mirror position, compute the pose of the virtual camera, relative to the object.
2. For each pair of mirror positions, compute the fixed-axis rotation between the virtual cameras.
3. For each mirror position, compute the mirror plane by fitting it to the associated axes of fixed-axis rotations.
4. Compute the true camera's projection matrix by reflecting any virtual camera in the associated plane.
5. Do a non-linear bundle adjustment: minimize the sum of squared reprojection errors, over the pose of the true camera and the positions of the mirror plane. This is implemented in the usual sparse manner [12].

For conciseness, we did not mention that in practice we also compute intrinsic camera parameters during this procedure: in the first step, we also calibrate the camera. In practice, we use a planar reference object; we thus use the method of [13, 9] to calibrate the camera from the reflected views (the reflection does not alter the intrinsic parameters), prior to computing pose in step 1. Further, the bundle adjustment in the last step also optimizes intrinsic camera parameters.

8 Experiments

8.1 Setup

We use an LCD monitor as reference object, considering it effectively as a planar surface. A structured light type approach [8] is used to get correspondences between the screen and the image plane: for each position of the planar mirror, we actually take a set of images, with the screen displaying a sequence of different patterns (cf. figure 2). Patterns are designed such that for each pixel in the image plane, we can directly compute the matching "point" on the screen, from the sequence of black-and-white (dark-and-light) greylevels received at the pixel.

8.2 Surface Reconstruction

We tested our method on real images, cf. figure 2. It was difficult to evaluate the estimated pose, so we evaluate it indirectly as follows. In [1], we describe an approach for the reconstruction of general specular surfaces from two images of the reference object's reflections. Here, to perform a quantitative evaluation, we reconstruct a planar specular surface (a hard drive platter), without making use of the planarity information for the reconstruction. Images are taken with a fixed pose of the camera and the specular surface (cf. figure 3), but with two different positions of the LCD monitor. Each of the two positions is estimated using the approach presented in this paper, by placing planar mirrors in the scene and making use of the knowledge of planarity.

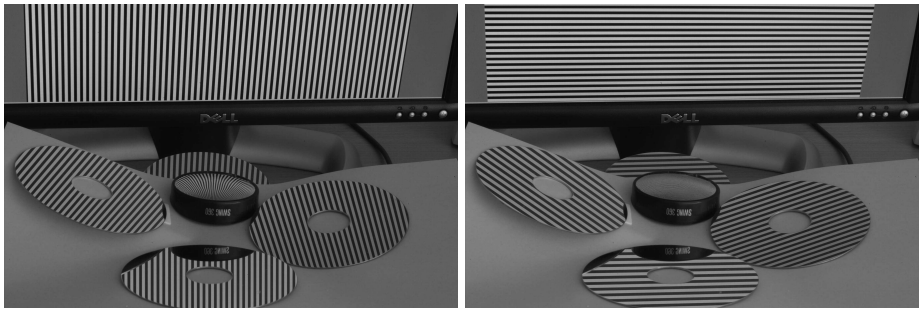

Fig. 2. Two images of our setup. Four planar mirrors (hard disk platters) are placed simultaneously in the scene. The object in the middle is a curved mirror, which was not used for the experiments reported here. The LCD monitor is partly visible in the image, but only its reflections are used to compute camera pose.

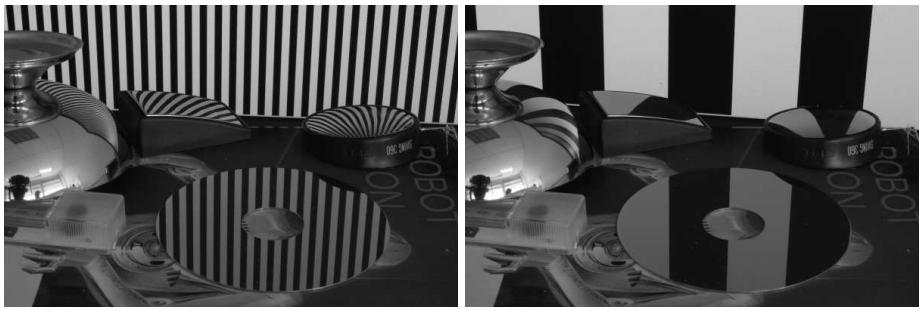

Fig. 3. Two of the images used for the reconstruction of the planar hard drive platter.

The specular surface is reconstructed as a dense point cloud [1], to which we fit a plane (linear least squares fitting without outlier removal). Over 98% of the roughly 525,000 computed points were less than 0.2 mm away from the computed plane and 64% less than 0.1 mm. The approximate diameter of the reconstructed part of the platter was 80 mm, resulting in a 0.3% relative error in the reconstruction. Refer to figure 4 for the histogram of point-plane distances.

9 Conclusions

We have addressed the problem of computing the pose of an object relative to a camera, without any direct view of the object. This problem has to our knowledge not been studied yet. A theoretical study and a practical algorithm have been provided, making use of planar mirrors put in unknown positions in the scene. It was shown that with three mirror positions or more, the problem can in general be solved.

Although rather specific, this problem is very relevant for our work on specular surface reconstruction, which like many similar works uses setups where the camera has not direct view of the reference object.

The method was shown to work with real images, although by an indirect evaluation via a specular surface reconstruction method. A more in-depth eval-

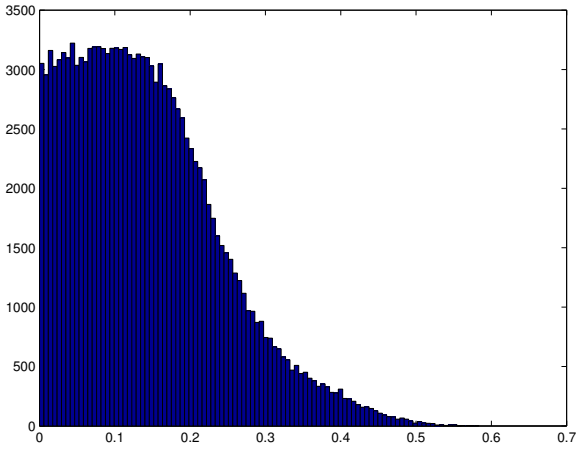

Fig. 4. Point-plane distance. Histogram of the distance of each point to the linear least squares fitted plane (in mm).

uation using simulated data should be done, but it seems to be reasonable to assume that the performances will be similar to those of calibration and pose estimation of a camera from several images of a planar calibration grid [13, 9] (the number of parameters and the geometries of the problems are similar).

References

1. T. Bonfort, P. Sturm, P. Gargallo. General Specular Surface Triangulation. ACCV, 2006.
2. Y. Caspi, M. Irani. Alignment of Non-Overlapping Sequences. ICCV, 76-83, 2001.
3. M.A. Halstead, B.A. Barsky, S.A. Klein, R.B. Mandell. Reconstructing Curved Surfaces from Specular Reflection Patterns Using Spline Surface Fitting of Normals. SIGGRAPH, 335-342, 1996.
4. S. Kammel, F. Puente León. Deflectometric measurement of specular surfaces. IEEE Instrumentation and Measurement Technology Conference, 531-536, 2005.
5. S. Maybank. Theory of Reconstruction from Image Motion. Springer Verlag, 1993.
6. M. Oren, S.K. Nayar. A Theory of Specular Surface Geometry. IJCV, 24(2), 1996.
7. S. Savarese, M. Chen, P. Perona. Recovering local shape of a mirror surface from reflection of a regular grid. ECCV, 2004.
8. D. Scharstein, R. Szeliski. High-accuracy stereo depth maps using structured light. CVPR, 195-202, 2003.
9. P. Sturm, S. Maybank. On Plane-Based Camera Calibration. CVPR, 432-437, 1999.
10. P. Sturm. Algorithms for Plane-Based Pose Estimation. CVPR, 706-711, 2000.
11. M. Tarini, H. Lensch, M. Goesele, H.-P. Seidel. 3D acquisition of mirroring objects. Research Report MPI-I-2003-4-001, Max-Planck-Institut für Informatik, 2003.
12. B. Triggs, P.F. McLauchlan, R.I. Hartley, A. Fitzgibbon. Bundle Ajustment – A Modern Synthesis. Vision Algorithms, 298-372, 1999.
13. Z. Zhang. A flexible new technique for camera calibration. PAMI, 22(11), 2000.
14. J.Y. Zheng, A. Murata. Acquiring 3D object models from specular motion using circular lights illumination. ICCV, 1101-1108, 1998.