MVA2000 IAPR Workshop on Machine Vision Applications, Nov. 28–30, 2000, The University of Tokyo, Japan

12—3 Recovering Camera Motion from Image Sequence Based on Registration of Silhouette Cones: Shape from Silhouette Using a Mobile Camera with a Gyro Sensor

Takayuki Okatani and Koichiro Deguchi
Graduate School of Information Sciences, Tohoku University
{okatani, kodeg}@fractal.is.tohoku.ac.jp

Abstract

A method for reconstructing shape of an object from its silhouette using a mobile camera to which a gyro sensor is attached is proposed. In order to determine unknown camera positions at which images are taken, the pose information of the camera derived from the attached gyro sensor as well as silhouette of the object are used. An algorithm for computing the camera positions by an iterative process of registering silhouettes associated with viewpoints is shown. After the computation of the camera positions, the object shape is reconstructed based on the usual shape from silhouette algorithm. The camera is mobile, and thus it can take silhouette images of the object from arbitrary directions. This enables us to avoid incorrect shape reconstruction due to restricted viewing angles, which often occurs in conventional shape from silhouette methods. Several experimental results are shown.

1 Introduction

Many methods have been developed for reconstructing 3D shape of an object from its silhouette based on the idea of shape from silhouette [1, 2, 3, 4]. One of them is the method that uses a turntable for rotating a target object; multiple images of the object viewed from different angles is obtained by the turntable and a fixed camera. Since the relative position and pose between the camera and the object are reliably obtained, this method produces comparatively good results without special devices and has already been commercially available.

A major restriction in the shape from silhouette methods is the inability to reconstruct the shape of a concave part of the object surface. It is basically impossible to reconstruct a concave shape by the shape from silhouette methods. However, it often occurs that even a convex part of the object shape is incorrectly recovered because of self-occlusion. This occurs when the viewing directions are restricted and are not enough. In this case, the resulting shape is partially true shape and partially incorrect shape that is similar to the convex hull of the true shape, and therefore this is called the visual hull effect. The method that uses a turntable often suffers from this effect.

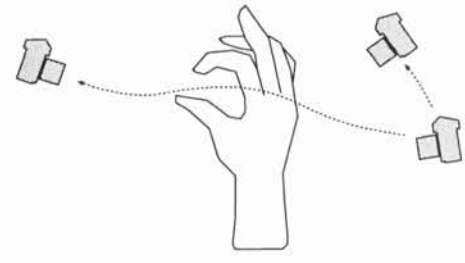

Figure 1: In order to avoid self-occlusion and to reconstruct correct shape by the shape from silhouette method, it is important to take images from various directions.

To avoid this problem, the viewpoints must be changed arbitrarily in order to take images of the object from arbitrary directions; see Fig.1. This is possible by taking images using a mobile camera. In that case, however, the position and pose of the camera at each viewpoint are unknown and thus they have to be precisely estimated in order to reconstruct the object's shape based on the shape from silhouette algorithm. Niem et al. proposed a method that uses a calibration pattern placed on the floor and under the object. The images of the object and of the calibration pattern is taken at the same time, and the camera parameters are estimated based on the idea of structure from motion [1]. (The extrinsic parameters as well as intrinsic ones are estimated.) In their method, processes of extracting characteristic points and establishing their correspondences across the images are required. Furthermore, the necessity of the calibration pattern can restrict the applicability of the methods.

From these points of view, this paper proposes a novel method that uses a mobile camera to which a gyro sensor is attached. The pose of the camera, which is one of the two extrinsic camera parameters, is obtained from the attached gyro sensor. By calibrating the relation between the output of the gyro sensor and the true camera pose in advance, the camera pose is assumed to be precisely known. Then, the camera position, which is another ex-

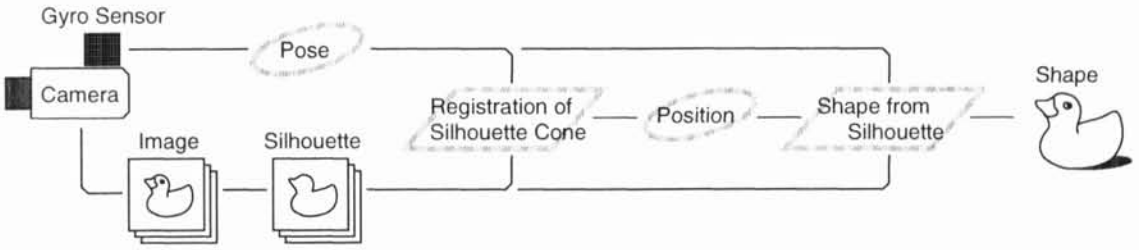

Figure 2: Data flow diagram for overall system. Inputs are object's images and pose of the camera at each viewpoint.

trinsic parameter, is determined by an iterative process of registering all silhouette cones associated with viewpoints, which are obtained by back-projecting the silhouette onto the 3D space. Once the camera position at each viewpoint is determined, the 3D shape of the object is reconstructed based on the usual shape from silhouette algorithm. The proposed method should be appealing because the method can treat large objects such as an automobile and a building, which are impossible to move and put on a turntable; it can treat even objects that are fixed on the ground or large objects such that a calibration pattern of an appropriate scale cannot be prepared.

2 Registration of silhouette cones

Figure 2 shows a graph showing the flow of the data from image acquisition to shape reconstruction in the overall system. As described, the pose of the camera at each viewpoint is obtained from the gyro sensor attached to the camera. Using the camera poses as well as the object's silhouettes in the images, the position of the camera at each viewpoint is computed. Then its 3D shape is reconstructed based on the known algorithm of shape from silhouette. The process of computing the camera positions that is written as "registration of silhouette cone" in Fig.2 is the main theme of this paper.

We assume perspective projection and known intrinsic parameters of the camera. As shown in Fig.3, silhouette is an outline of the object on the image plane. By back-projecting the silhouette onto the 3D space, we have a cone whose vertex is at the projection center and that osculates the surface of the object, called the silhouette cone. We have as many silhouette cones as the images. We "register" these silhouette cones so that they overlap each other, as shown in Fig.3. In this way, we determine the camera position of each viewpoint. (Note that the absolute size of the object cannot be determined in our situation.)

Let n be the number of images and i denote each of the viewpoints $(i = 1, \ldots, n)$. The transformation from the world coordinates to each camera coordinates is written by

$$\mathbf{x}_i = R_i \mathbf{x} + \mathbf{t}_i. \tag{1}$$

Since R_i, the camera pose for each viewpoint, is known

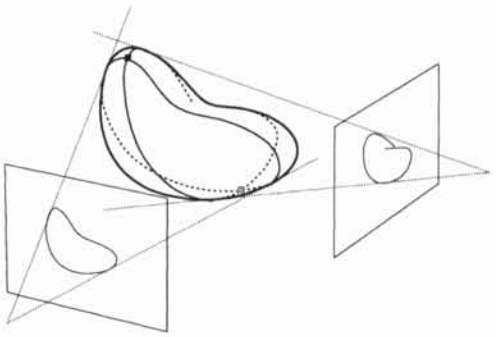

Figure 3: Illustration of silhouette cones.

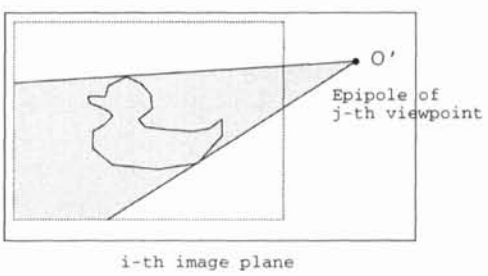

Figure 4: When the positions of the ith and jth viewpoints are correct, the ith silhouette should be inscribed to the projection of the jth silhouette cone on the ith image plane.

from the gyro sensor, we are to determine $\mathbf{t}_i$ for $i = 1, \ldots, n$.

As described, the determination of $\mathbf{t}_i$ is done by the registration of the silhouette cones. We carry it out on the image plane. The projection of the jth silhouette cone onto the ith image plane usually forms a region enclosed by two straight lines; see Fig.4. The two lines cross at the epipole of the jth viewpoint. When $\mathbf{t}_i$ and $\mathbf{t}_j$ are correct, the ith silhouette should be inscribed to the two lines of the jth silhouette, as shown in Fig.4. Based on this fact, we determine the positions $\mathbf{t}_i$ $(i = 1, \ldots, n)$ so that the silhouette is inscribed to every two lines formed by the silhouette cone of other viewpoint.

To determine $\mathbf{t}_i$ in this way, iterative computation is necessary, since when $\mathbf{t}_i$ is changed, projections of the silhouette cones onto the image plane will change. The iterative algorithm takes the following steps:

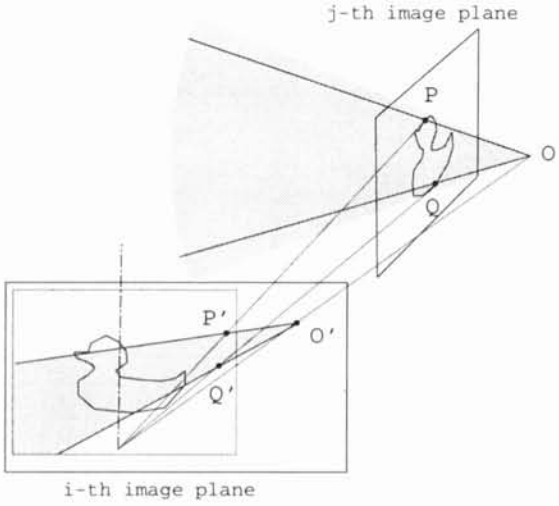

Figure 5: Illustration of P and Q; see text.

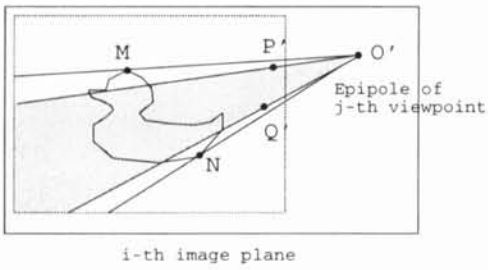

Figure 6: Illustration of M and N; see text.

1. Initialize $\mathbf{t}_i$ $(i = 1, \ldots, n)$.
2. For all i and j, choose two points on the jth silhouette curve such that their corresponding two lines on the ith image plane form the boundary lines of the projection of the jth silhouette cone, as shown in Fig.5. Let P_{ij} and Q_{ij} denote these points.
3. For all i and j, choose two points on the ith silhouette curve such that the line connecting each of them to epipole of the jth viewpoint is outermost on the ith image plane, as shown in Fig.6. Let M_{ij} and N_{ij} denote these points.
4. Update $\mathbf{t}_i$ so that the two lines connecting the epipole of the jth viewpoint to P'_{ij} and Q'_{ij}, which are the projection of P_{ij} and Q_{ij} onto the ith image plane, pass through M_{ij} and N_{ij} on the ith image, respectively.
5. If $\mathbf{t}_i$ converges, then stop. Otherwise, go to Step 2.

In Step 2 and 3, $\mathbf{t}_i$ is used for choosing the point sets (P_{ij}, Q_{ij}) and (M_{ij}, N_{ij}). In Step 4, $\mathbf{t}_i$ is recomputed using the chosen point sets. The recomputation is done by solving a linear minimization problem. The detail is as follows.

The condition for the line $O'_{ij}P'_{ij}$ to pass through M_{ij} is given by

$$J_{ij}^{(1)} \equiv f(-p_2 + p_3 m_2)t_1 + f(p_1 - p_3 m_1)t_2 + [p_2(m_1 - u_0) - p_1(m_2 - v_0) + p_3(u_0 m_2 - m_1 v_0)]t_3 = 0, \tag{2}$$

where f is focal length of the lens, $[p_1, p_2, p_3]$ is the vector $O'_{ij}P'_{ij}$ represented in the ith coordinates system, $[m_1, m_2]$ is the image coordinates of M_{ij}, and $[t_1, t_2, t_3]$ is the coordinates of the epipole O'_{ij} represented in the ith coordinate system; $[t_1, t_2, t_3]$ is given by

$$[t_1, t_2, t_3]^\top = -R_i R_j^{-1}\mathbf{t}_j + \mathbf{t}_i. \tag{3}$$

Thus, $J_{ij}^{(1)} = 0$ is a linear equation for $\mathbf{t}_i$ and $\mathbf{t}_j$. The condition for the line $O'_{ij}Q'_{ij}$ to pass through N_{ij} is given by a similar equation; we denote it by $J_{ij}^{(2)} = 0$. In the process of iteration, all of these equations cannot be zero, since the points P_{ij} and Q_{ij} etc. are possibly different from the points that would be chosen under the true arrangement of the viewpoints. Furthermore even after a sufficient number of iteration, they cannot be zero due to noises derived from the gyro sensor, the quantization error of the silhouette, and so on. Thus, at each step, we compute $\mathbf{t}_i$ minimizing

$$J = \sum_i \sum_{j \neq i} [(J_{ij}^{(1)})^2 + (J_{ij}^{(2)})^2]. \tag{4}$$

If noises were successfully modeled, we would have more preferable minimization. This will be studied in the future work.

For each of n viewpoints and each of other $n-1$ viewpoints, there are two constraints on $\mathbf{t}_i$ that are derived from two lines, $O'_{ij}P'_{ij}$ and $O'_{ij}Q'_{ij}$. Hence, the number of equations is $2n(n-1)$. There can be cases where the projection of the silhouette cone does not form a region enclosed by two lines, depending on relative poses and positions of viewpoints. Let the number of such cases be k. Then, the number of equations is $2n(n-1) - k$.

On the other hand, unknowns are $\mathbf{t}_i$ for $i = 1, \ldots, n$. Applying overall translation to $\mathbf{t}_i$ would result in the same solution. To avoid this ambiguity, we fix the coordinates of the first viewpoint as $\mathbf{t}_1 = [0,0,0]^\top$. Thus the number of unknowns is $3(n-1)$. We define a vector by putting them in order:

$$\mathbf{T} = [\mathbf{t}_2^\top, \mathbf{t}_3^\top, \ldots, \mathbf{t}_n^\top]^\top. \tag{5}$$

Then, the above minimization problem can be at least formally written as

$$X\mathbf{T} = 0, \tag{6}$$

where X is a $(2n(n-1) - k) \times 3(n-1)$ matrix. To avoid the described ambiguity of scale, we constrain $\mathbf{t}_i$ as $|\mathbf{T}| = 1$. As a result, $\mathbf{T}$ is given by the normalized eigenvector associated with the minimum eigenvalue of the matrix $X^\top X$.

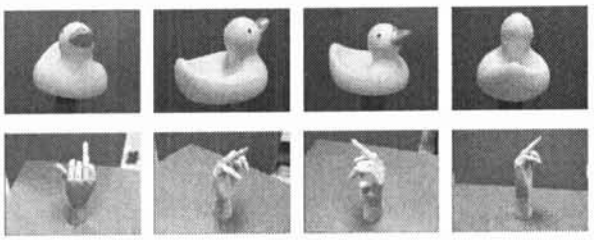

Figure 7: Image sequences used for the experiment. Upper: Rubber duck (4 out of 33 images). Lower: Hand model (4 out of 28 images).

3 Experimental result

We tested the proposed algorithm using a CCD camera to which a gyro sensor (Datatec GU-3011) is attached. Starting from any initial value chosen randomly, the algorithm always converged after at most 7 iterations. Table 1 shows the iteration count that the algorithm took until it converged for the image sequence shown in the upper row of Fig.7. The initial values for $\mathbf{t}_i$ were altered randomly for each trial, and 100 trials were carried out.

Two objects, a rubber duck and a hand model, were used for experiments. Figure 7 shows selected four images in the image sequences. The number of the images is 33 for the duck and 28 for the hand. The camera positions for them are recovered by the algorithm described in the last section. Using the recovered camera positions, the shapes of these objects are reconstructed from the silhouette based on the existing shape from silhouette algorithm. Figure 8 shows the reconstructed shapes. For the duck, it can be seen that the overall shape is correctly reconstructed. For the hand, except a few visual hull effect occurs in the bases of a few fingers, most of the result is satisfactory; even a small space between the thumb and the forefinger is correctly reconstructed.

4 Summary

We have shown the algorithm that determines the positions of the camera using the object's silhouette on the image when the pose of the camera is known. The camera pose is obtained by a gyro sensor attached to the camera. The method does not extract feature points or establish their correspondences. Experimental results show that the shape reconstruction is possible with reasonable accuracy.

In the method, accuracy of extracting the object's silhouette gives compound effects on accuracy of the final shape reconstruction. For other conventional methods, e.g., using a turntable, the silhouette accuracy actually affects the reconstruction accuracy, too. In our method, however, it first affects the accuracy of estimating the camera position, and then affects that of the shape reconstruction. Therefore, a precise extraction of silhouette is more important in our method. In a future work, we will study these effects. Furthermore, development of a minimization criterion that considers error of the camera pose or quantization error of the image is also a future work.

Table 1: Result of the algorithm's convergence.

Iteration count	1	2	3	4	5	6	7	8
Times of 100 trials	0	0	0	0	20	52	28	0

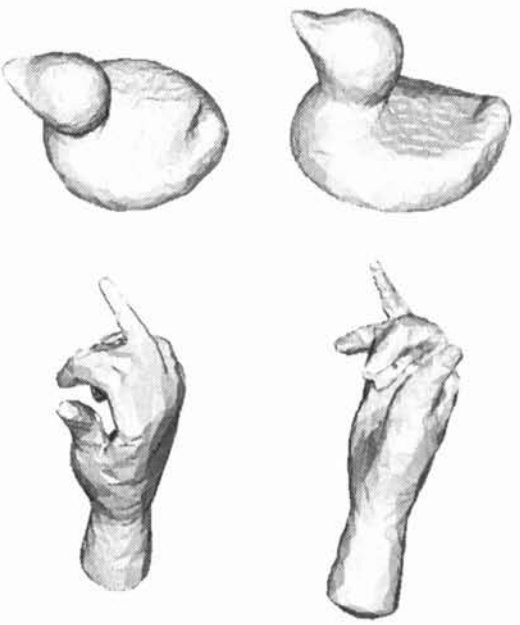

Figure 8: Reconstructed 3D shapes.

References

[1] W. Niem and J. Wingbermühle. Automatic reconstruction of 3D objects using a mobile monoscopic camera. In *Proceedings of the International Conference on Recent Advances in 3D Imaging and Modelling*, 1997.

[2] W. Niem and R. Buschmann. Automatic modelling of 3D natural objects from multiple views. In *European workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production*, 1994.

[3] A. W. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3D model construction for turn-table sequences. In L. Van Gool and R. Koch, Editors, *Structure and Motion from Multiple Images in Large-Scale Environments*, Lecture Notes in Compute Science, Springer, 1998.

[4] R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics, and Image Processing*, 1993.