Robust Object Pose Estimation From Feature-Based Stereo

Robert Laganière, *Member, IEEE*, Sébastien Gilbert, and Gerhard Roth, *Senior Member, IEEE*

***Abstract*—This paper addresses the problem of computing the three-dimensional (3-D) path of a moving rigid object using a calibrated stereoscopic vision setup. The proposed system begins by detecting feature points on the moving object. By tracking these points over time, it produces clouds of 3-D points that can be registered, thus giving information about the underlying camera motion. A novel correction scheme that compensates for the accumulated error in the computed positions by automatic detection of loop-back points in the movement of the object is also proposed. An application to object modeling is presented in which a handheld object is moved in front of a camera and is reconstructed using silhouette intersection.**

***Index Terms*—Camera calibration, feature matching, feature tracking, pose estimation, shape-from-silhouette, stereovision, three-dimensional (3-D) reconstruction.**

I. Introduction

One of the main challenges inherent in using images from a large number of viewpoints is the issue of camera pose estimation. For three-dimensional (3-D) reconstruction to be possible, the location and orientation of the cameras at different capture instants must be accurately known. Several applications can benefit from the knowledge of the position of a camera with respect to some rigid reference frame. Among them are virtual or augmented reality systems, scene reconstruction, object modeling, and robotics. In a video sequence in which a camera is moving inside a fixed environment, keeping track of the camera/object's respective positions can be challenging. In the case of a moving camera, a workable solution consists of installing calibration targets, precisely registered with respect to a global reference frame. By having them visible inside the scene, it becomes possible to compute the camera position as the camera moves with respect to the global reference frame [5]. Alternately, in an object-based solution, a computer-aided design (CAD) model of the observed objects can be created, and by registering that model to the observations, the spatial relation between the scene and the camera can be determined [1]. In practice, however, these solutions are not always feasible. It is therefore desirable to develop a method to compute the camera motion in an unprepared scene for which no *a priori* knowledge is available.

In 3-D reconstruction, bundle adjustment is a widely accepted approach [2], and commercial software tools are now available. The technique most often relies on a human operator who has to supply the matches since there is typically a small number of widely separated views. Bundle adjustment is used for a wide spectrum of applications, such as accident reconstruction, animation and graphics, archaeology, forensics, engineering, and architecture. The main drawback of bundle adjustment is its instability. In many situations, the algorithm will fail to converge to an accurate solution. To overcome this problem, it is recommended that the user starts with a small subset of available pictures and a small number of feature points that can be seen in many pictures. Once the algorithm succeeded in converging to a first reasonable solution, additional intermediate pictures and more feature points can be added to improve the accuracy. This lack of robustness can be related to the iterative nature of bundle adjustment, which implies initial estimates of the camera positions. If these estimates are very far from the actual solution, the algorithm may fail to converge.

This problem is amplified when one wants to automate the whole process. Matches between narrowly separated views can be found automatically through correlation. Unfortunately, nothing can guarantee that the matches will all be good. Bad matches will definitely negatively affect any structure and motion estimation process. While the use of more widely separated views would help to improve the accuracy of the reconstruction, the matching process would become much more difficult and error prone.

The method proposed in this paper aims at resolving these issues using a calibrated stereoscopic vision setup. This system is observing a rigid object in motion on which feature points are detected. Because they are seen by a stereo setup, these points can be 3-D reconstructed when they are matched. By tracking these points over time, the resulting clouds of 3-D points can be registered, thus giving information about the underlying camera motion. This is the idea that is exploited in this paper to robustly keep track of the camera/object's relative motion along a sequence. Camera position computation from reconstructed points has been used in the past but was generally limited to few images, long image sequences posing the problem of error accumulation error. The scheme proposed here overcomes this problem because 1) it includes a novel correction scheme that compensates for the accumulated error in the computed positions, and 2) it exploits the automatic detection of loop-back points in the movement of the object. An application to object modeling is presented in which a handheld object is

Manuscript received June 15, 2005; revised April 6, 2006.

R. Laganière is with the VIVA Research Lab, School of Information Technology and Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: laganier@site.uottawa.ca).

S. Gilbert is with the VIVA Research Lab, School of Information Technology and Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada, and also with Dalsa Semiconducteur, Bromont, QC J2L 1S7, Canada.

G. Roth is with the Computational Video Group, Institute for Information Technology, National Research Council of Canada, Ottawa, ON K1A 0R6, Canada.

Digital Object Identifier 10.1109/TIM.2006.876521

moved in front of a camera and is reconstructed using silhouette intersection.

The path of a binocular or trinocular stereoscopic setup is computed in [4]. In this approach, points are matched at each camera location. In addition, points in one view are tracked from image to image. The method uses trilinear tensors and/or fundamental matrix constraints for robust tracking and matching over views. The computed transformations are then cascaded to place them in a common coordinate frame. To overcome the problem of error accumulation, it is proposed to add an extra step where the final 3-D transformation for all cameras would be computed simultaneously.

In [7], the goal is to compute the registration of two consecutive scene captures along with the extrinsic calibration parameters of the stereo setup and the 3-D location of a minimum of four matched and tracked feature points. The essential matrix of the stereo setup is calculated from the eight correspondences given by the four feature points in both captures; nonlinear methods are used to enforce its constraints. It is decomposed to retrieve the extrinsic calibration parameters up to a scale factor of the translation vector. At this point, 3-D reconstruction can be applied to the feature points, yielding two clouds of a minimum of four 3-D points. The registration between the two 3-D point clouds can then be calculated. It differs from the proposed method in that they do not compute the extrinsic calibration parameters of the stereo setup prior to the computation of the registration. As a consequence, the matching process cannot be guided by the epipolar constraint. No experimental results along a sequence were shown to display the accumulation of error.

The method in [20] tracks points in each view of a stereo rig. It introduced binocular matching constraints. Camera motion is recovered from the left and right temporal fundamental matrices. Stereo correspondences are then inferred by combining stereo geometry and motion correspondences through projective mapping. A similar approach is presented in [8]; stereoscopic vision and shape-from-motion are combined in an attempt to exploit the strengths of both approaches, i.e., accurate 3-D reconstruction for stereo and easy feature tracking for visual motion. The result is a 3-D reconstruction of feature points and the camera motion in two separate steps. In this paper, the experiments were limited to short sequences where the viewpoints do not change dramatically from the first to the last capture.

Stereo and motion correspondences are computed simultaneously in [24]. They defined a coarse-to-fine algorithm in which local surface parameters and rigid-body motion parameters are iteratively estimated. They were able to extract local range information from a sequence of a few stereo images.

In [6], self-calibration and Euclidean reconstruction from a stereo rig are achieved through a stratified approach that proceeds by upgrading a projective reconstruction to affine and to metric reconstruction. The results are shown for images taken before and after a single rigid motion.

A strategy based on active stereo is proposed in [11] for simultaneous localization and mapping in a robotic application. As the robot moves inside the scene, the stereo head is actively moved to select the feature measurement that will best improve the current robot position estimation. The solution is based on the definition of measurement and motion models predicted using a Kalman filter.

Finally, a 3-D model acquisition system is proposed in [12], in which an object is also freely rotated in front of a camera. The method, however, uses a structured-light rangefinder to extract the required 3-D information. The images are then geometrically aligned using an iterative closest point (ICP)-like algorithm that identifies the best rigid body transformation.

The rest of this paper is organized as follows. Section II reviews the concepts of 3-D reconstruction. Section III presents the used feature matching and tracking strategy. Section IV discusses the problem of previously visited locations while Section V is concerned with accumulated error correction. Finally, Section VI presents the experimental results and Section VII the conclusion.

II. Stereo Reconstruction

Three-dimensional reconstruction requires the computation of the Euclidean coordinates of image features from the visual data observed in multiple views. Stereoscopic vision involves the use of two cameras for which there is a fixed rigid transformation between them.

A. Calibration

Stereo calibration aims at computing the projection matrices of the two cameras. When a set of 3-D points at precisely known locations is available, the projection matrices can be obtained straightforwardly. Indeed, a point $\mathbf{X}_i$ and its corresponding image coordinates $\mathbf{x}_i$ satisfy the relationship

$$\mathbf{x} = P\mathbf{X} = K[R|T]\mathbf{X}$$
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (1)$$

Eliminating λ and rearranging the expressions yield a pair of homogeneous linear equations in 12 unknowns, which are the entries of the projection matrix. Putting together the information of n 3-D points $(n \geq 6)$ gives $2n$ homogeneous linear equations in 12 unknowns: $p_{00}, p_{01}, \ldots, p_{23}$. This system can be solved up to a scale factor through singular value decomposition (SVD). The quality of the computed projection matrix depends on the linearity of the camera model and the accuracy in the measured 3-D location of the points. Once the projection matrices are computed for both cameras, they can be decomposed to retrieve their intrinsic and extrinsic calibration parameters [3].

In practice, however, it is more convenient to use a planar configuration object with known metric pattern. By exploiting the homographic constraints that exists between the calibration plane and the corresponding images, it is possible to build a linear system of equations. This is the approach proposed in [16] and [17], where several views of the planar calibration pattern are used to calibrate a single camera. The procedure has

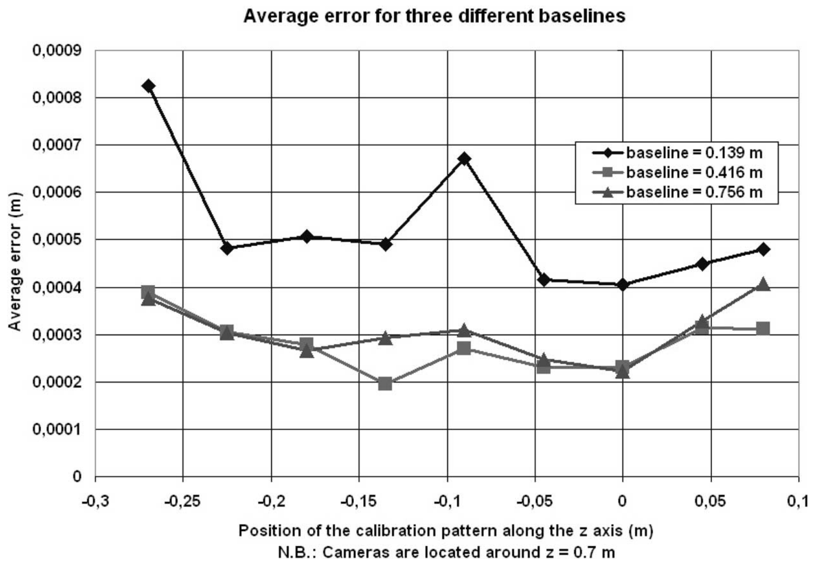


Fig. 1. Average reconstruction error for three different baselines.

been extended to the case of a stereo setting in [19]. The method makes use of the additional constraint provided by the stereo rig configuration. The constraint is also expressed in the form of a homography: the one that links the image of the calibration plane as a function of the rigid transformation between the cameras of the stereo head, i.e., [18]

$$H = K \left[R - \frac{Tn^\top}{d} \right] K^{-1}. \tag{2}$$

Here, n is the normal of the calibration, and d is the distance from the camera to the plane.

To determine the optimal stereo configuration of the cameras, we performed an experiment in which we used three stereo setups with different baselines (0.139, 0.416, and 0.756 m). The angles between the z-axes of the two cameras were adjusted in such a way that a given working volume was preserved, resulting in angles of 0.112, 0.463, and 1.05 rad, respectively. A checkerboard calibration pattern was used, allowing easy detection of its feature points with subpixel resolution. The position of the calibration pattern with respect to the table was measured with a ruler. This procedure provides the ground truth value of the feature point position, with an estimated accuracy of 0.3 mm.

Fig. 1 shows the reconstruction error ($|\vec{x}_{\text{calculated}} - \vec{x}_{\text{measured}}|$) averaged over the 20 feature points of a calibration pattern as a function of the z-position of the calibration pattern for three different baselines. It can be observed that the reconstruction error is higher for the stereo setup with the smallest baseline, as expected. No significant difference can be observed by comparing the results of the stereo setups with baselines of 0.416 and 0.756 m. Because matching is facilitated when the baseline is shorter, we can conclude that there is no need to increase the baseline of our stereo setup above 0.4 m as it does not seem to provide any significant improvement in reconstruction accuracy and would make the matching process more difficult.

B. Three-Dimensional Point Reconstruction

Once the stereo rig is calibrated, and consequently the projection matrices P_1 and P_2 are known, it is possible to compute the 3-D position of any point seen by the two cameras. Using the projective relation in (1), the 3-D location $\mathbf{X}$ of a feature point whose image coordinates in the two images are $\mathbf{x}_1$ and $\mathbf{x}_2$ can be obtained by solving the system of four linear equations in three unknowns as

$$\begin{bmatrix} (p_{00} - xp_{20})_1 & (p_{01} - xp_{21})_1 & (p_{02} - xp_{22})_1 \\ (p_{10} - yp_{20})_1 & (p_{11} - yp_{21})_1 & (p_{12} - yp_{22})_1 \\ (p_{00} - xp_{20})_2 & (p_{01} - xp_{21})_2 & (p_{02} - xp_{22})_2 \\ (p_{10} - yp_{20})_2 & (p_{11} - yp_{21})_2 & (p_{12} - yp_{22})_2 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (xp_{23} - p_{03})_1 \\ (yp_{23} - p_{13})_1 \\ (xp_{23} - p_{03})_2 \\ (yp_{23} - p_{13})_2 \end{bmatrix}. \tag{3}$$

This system can be solved through a least square method. Even if this approach, involving the minimization of algebraic quantities, works well in practice, a geometric triangulation formulation is often preferred. The method finds the 3-D point that minimizes its 3-D distance with two noncrossing lines in space. In other words, it returns the middle point of the segment perpendicular to both rays.

Fig. 2 shows the geometry of two cameras projecting the images $\mathbf{x}_1$ and $\mathbf{x}_2$ of the 3-D point $\mathbf{X}$. In an ideal situation, the extension of the lines $\overrightarrow{O_1 K_1^{-1} \mathbf{x}_1}$ and $\overrightarrow{O_2 K_2^{-1} \mathbf{x}_2}$ should cross each other in space at the location of the projected 3-D point $\mathbf{X}$. In reality, the two lines may not cross. The best solution is therefore to search for the point $\mathbf{X}$ that is the middle of the segment perpendicular to both lines. From Fig. 2, we have

$$\mathbf{X}_1 = \frac{1}{\lambda_1} K_1^{-1} \mathbf{x}_1 \tag{4}$$

$$\mathbf{X}_2 = \frac{1}{\lambda_2} K_2^{-1} \mathbf{x}_2. \tag{5}$$

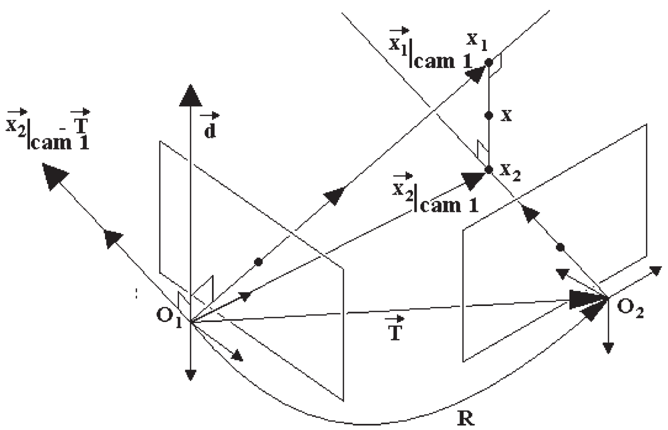


Fig. 2. Geometry of the triangulation procedure.

Expressing point $\mathbf{X}_2$ in the reference frame of the first camera gives

$$\begin{aligned}\mathbf{X}_2|_{\text{cam1}} &= R\mathbf{X}_2 + T\\ &= \frac{1}{\lambda_2} RK_2^{-1}\mathbf{x}_2 + T.\end{aligned} \qquad (6)$$

Let us now define the vector $\vec{d}$ that is proportional to the cross product of $\mathbf{X}_1|_{\text{cam1}}$ and $(\mathbf{X}_2|_{\text{cam1}} - T)$

$$\begin{aligned}\vec{d} &\equiv \lambda_1\lambda_2\mathbf{X}_1|_{\text{cam1}} \times (\mathbf{X}_2|_{\text{cam1}} - T)\\ &= K_1^{-1}\mathbf{x}_1 \times RK_2^{-1}\mathbf{x}_2.\end{aligned} \qquad (7)$$

The vector $\vec{d}$ is therefore parallel to the vector $\overrightarrow{\mathbf{X}_1\mathbf{X}_2}$. Let us now define three scalars a, b, and c such that the path $O_1\mathbf{X}_1\mathbf{X}_2O_2O_1$ forms a closed loop

$$aK_1^{-1}\mathbf{x}_1 + b\vec{d} + cRK_2^{-1}\mathbf{x}_2 - T = 0 \qquad (8)$$

$$aK_1^{-1}\mathbf{x}_1 + b\left[K_1^{-1}\mathbf{x}_1 \times RK_2^{-1}\mathbf{x}_2\right] + cRK_2^{-1}\mathbf{x}_2 = T. \qquad (9)$$

Equation (9) provides three linear equations in three unknowns: a, b, and c. Once this system is solved for a given match ($\mathbf{x_1}$, $\mathbf{x_2}$), the location of the point $\mathbf{X}$ can be calculated as

$$\mathbf{X}|_{\text{cam1}} = aK_1^{-1}\mathbf{x}_1 + \frac{1}{2}b\left[K_1^{-1}\mathbf{x}_1 \times RK_2^{-1}\mathbf{x}_2\right]. \qquad (10)$$

III. Feature-Based Stereokineopsis

To keep track of the position of an object that moves in front of a set of cameras, we used a match-and-track paradigm. At one instant, feature points are detected and matched across views. The matched points are then independently tracked in each view until a new matching process is initiated. Proceeding this way, we benefit from both the accuracy of the reconstruction provided by stereo-matching and the reliability and efficiency of image tracking. Robustness of the process to the unavoidable presence of outliers is ensured by the 3-D registration procedure that is applied between each stereo-matching phase. Note that matching is typically applied every X frames, while tracking is performed at full frame rate. In fact, the matching rate is determined by the frequency at which a given application requires new object position data. The following subsections detail each of these steps. Another important difficulty related to sequential pose estimation approaches concerns the error accumulation problem; this aspect is addressed in Sections IV and V.

A. Tracking

Feature point tracking is achieved using the Intel OpenCV implementation of the Lucas–Kanade tracker [15]. It is an accurate and robust tracker that can run at several frames per second. To reduce the computational load, the tracker uses a pyramid of resolutions in the computation of displacement vectors.

When Harris corners are used, the tracker performs reliably over quite long sequences. However, it is unavoidable to have some false tracks occurring. Occlusion boundaries, for instance, are particularly problematic as they tend to produce moving corners on the image. Consequently, even when starting with an exact match set, the independent tracking of the features in each view will most probably cause the introduction of some false matches; therefore, the 3-D registration process has to be robust to outliers.

B. Matching

Because the epipolar geometry is available through a preliminary calibration phase, it is used to guide the matching of features. Thus, only points from the second image that lie close to the epipolar line of a point in the first image are considered as possible matches.

Feature comparison is done using variance normalized correlation (VNC), which is designed to produce reliable results over a wide range of viewing conditions. VNC is defined for a candidate match ($\mathbf{x}_1$, $\mathbf{x}_2$) as

$$\text{VNC}(\mathbf{x}_1, \mathbf{x}_2) = \frac{\sum_{\mathbf{k}_1,\mathbf{k}_2}\left[I_1(\mathbf{k}_1) - \overline{I_1(\mathbf{x}_1)}\right]\left[I_2(\mathbf{k}_2) - \overline{I_2(\mathbf{x}_2)}\right]}{N\sqrt{\sigma_{I_1}^2(\mathbf{x}_1)\sigma_{I_2}^2(\mathbf{x}_2)}} \qquad (11)$$

where the sum is taken over the points $\mathbf{k}_1$ and $\mathbf{k}_2$ in the neighborhoods of $\mathbf{x}_1$ and $\mathbf{x}_2$ and where $\overline{I(\mathbf{x})}$ and $\sigma_I^2(\mathbf{x})$ are, respectively, the mean and the variance of the pixel intensities over the neighborhoods.

The point pairs found through correlation along the epipolar lines are not necessarily accurate correspondences. This is why additional matching constraints must be applied. In particular, the uniqueness and symmetry constraints have been shown to be simple and advantageous [14]. Uniqueness requires that only the best match in the second image be kept for a given point in the first image. Symmetry requires that the point in the first image also be the best match for the other point.

Finally, to prune additional mismatches that might still be present, the disparity gradient is used, as in [21]. The disparity gradient is a measure of the compatibility of matched points. For two pairs ($\mathbf{x}_1$, $\mathbf{x}_2$) and ($\mathbf{y}_1$, $\mathbf{y}_2$) having disparities $d(\mathbf{x}_1, \mathbf{x}_2)$ and $d(\mathbf{y}_1, \mathbf{y}_2)$, respectively, the cyclopean separation

$d_{cs}(\mathbf{x}_1, \mathbf{x}_2; \mathbf{y}_1, \mathbf{y}_2)$ is the vector joining the midpoints of the line segments $\overline{\mathbf{x}_1\mathbf{x}_2}$ and $\overline{\mathbf{y}_1\mathbf{y}_2}$, and their disparity gradient is defined as

$$\Delta d(\mathbf{x}_1\mathbf{x}_2; \mathbf{y}_1\mathbf{y}_2) = \frac{|d(\mathbf{x}_1, \mathbf{x}_2) - d(\mathbf{y}_1, \mathbf{y}_2)|}{|d_{cs}(\mathbf{x}_1, \mathbf{x}_2; \mathbf{y}_1, \mathbf{y}_2)|}. \tag{12}$$

This compatibility measure is used in a constraint that accepts pairs that share disparity gradients below some threshold value, with at least two of their three closest neighbors. This eliminates false matches as long as they are not surrounded by other similar false matches.

C. Robust Registration

After having found matches and tracked the corresponding points in both sequences, 3-D points can be reconstructed. Based on the matches at a given instant M and their tracked correspondents at a latter instant N, the resulting two clouds of 3-D points can be registered to find the rigid motion of the object (or, reciprocally, the rigid motion of the stereo setup when the reference frame is attached to the object). Unfortunately, in identifying the motion complying with the observations, one cannot simply use the complete data set because the false matches, and the tracking errors will corrupt the result. Instead, it is necessary to introduce a random sample consensus (RANSAC) algorithm [9] to filter out the corrupted pairs of 3-D points.

A minimum of three pairs of noncollinear 3-D points is necessary to compute an unambiguous 3-D registration. As a consequence, the first step of the algorithm will consist of finding a triplet of 3-D matches.

To make sure that a randomly selected triplet of 3-D matches does not constitute a degenerate case (i.e., is not in a collinear configuration), two conditions are imposed.

1) The distance between any two points of the trio must be greater than a given minimum.
2) The area defined by the three points must be greater than a given minimum.

The first item alone is not sufficient because three collinear points that are located far apart would satisfy it, while the second item alone would allow a triplet constituting of two points close from each other with a third point far away, such that the area of the triangle is sufficient.

Once corresponding triplets have been identified as being noncollinear, the rotation and the translation that best describe the rigid movement of the points can be computed. This now constitutes a candidate registration $(R_{N/M}, T_{N/M})$.

Given such a candidate registration, a count of the number of supporting matches can be obtained as follows. For each 3-D match, if the distance between $\mathbf{X}_N$ and $R_{N/M}\mathbf{X}_M + T_{N/M}$ is less than the maximum distance, then this match is said to agree with or support the candidate registration. This procedure is repeated several times with the number of trials set so that the probability of success is above a desired value. The candidate registration having the highest number of supporting matches is declared the best candidate registration.

Finally, all the matches that support the best candidate registration are used to compute the final output registration through least square fitting of the two sets of points [10], i.e.,

$$Q_{N/M} = \begin{bmatrix} R_{N/M} & T_{N/M} \\ 0^T & 1 \end{bmatrix} \tag{13}$$

From the computed homogeneous transformation $Q_{N/M}$, the new world coordinates of the cameras can be computed as

$$Q_N = Q_{N/M} Q_M. \tag{14}$$

The main problem associated with this technique resides in the accumulation of errors, because every new position is computed from the previous one. Because it is assumed that no special target points that could allow recalibration are available on the object, the only information that can be used here is the knowledge of the approximate camera positions that will allow us to identify points of view that were previously captured. This is the information that will be used here to correct for the drift each time the cameras pass by a location where they have been before.

IV. Detection of Previously Visited Locations

The goal of this procedure is to take a sequence of camera positions and identify those that are close to their previous positions in an earlier image capture. Whenever such a loop-back situation is detected, a connection between earlier and later views becomes possible. Once this is done, a registration between the two positions and correction of the accumulated error can be undertaken.

Because the tracking algorithm described in Section III-A is used to match the two extreme views of a detected loop, it is necessary that these views be separated by a relatively short baseline; this requirement can be expressed by the following two conditions.

1) The z-axes of the two views must be nearly parallel.
2) The distance between the center of projection of the views must be sufficiently small.

At first sight, these conditions might appear to be insufficient for loop-back detection to work properly as this test is not completely rotationally invariant. Indeed, it is also necessary that the y-axes (or the x-axes) be nearly parallel for correct neighborhood matching. Nevertheless, we can relax this constraint since our knowledge of the approximate camera positions will allow us to derotate the images around their z-axes in such a way that they become adequately aligned.

A. Detection of Close Views

The distance between the center of projection of the views is directly calculated from the length of the vector going from one center to the other. To calculate the maximum distance we can tolerate, we must take into consideration the fact that the two views may be collinear along their parallel z-axes (i.e., one view may be in front of the other), resulting in a scale difference between the two images.

The angle between the z-axes of two views can be computed through a scalar product of unit vectors parallel to the z-axes of the two cameras, as expressed in the world reference frame

$$\hat{k}_M = Q_M \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{15}$$

$$\hat{k}_N = Q_N \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{16}$$

$$\cos(\theta) = \hat{k}_M \cdot \hat{k}_N. \tag{17}$$

In a sequence, the minimal angle (or distance) with respect to a given frame may not happen at the same frame for the left and right cameras. When trying to identify the best capture to be matched with an earlier capture, we must find a compromise between the two cameras.

Whenever a view is detected as being close to a previously captured view, the drift of the later view can be compensated. Of course, it is assumed that the earlier the view, the better the accuracy, since its location has been computed from a smaller number of cascaded transformations. This is discussed in Section V.

B. Identification of the Rotation Angle Around the z-Axis

As discussed previously, two views are similar if their z-axes are nearly parallel; they can, however, have a wide angular difference around their z-axes. Because the tracking algorithm is not rotation invariant, this situation could prevent the identification of correspondences. We can overcome this difficulty by making use of the knowledge we have of the approximate positions of the camera; that is, we can determine the rotation that must be applied to the images of the later view such that it is as aligned as possible with the earlier view.

In the first approach, we will aim at minimizing the angle between the y-axes of two views by applying a rotation around the z-axis of the second view. Let us state the result:

Let r_{ij} be the element (i, j) of the rotation matrix linking the view N with the view M, i.e., $R_{N/M}$. If $r_{10} \sin(\arctan(-r_{10}/r_{11})) < r_{11} \cos(\arctan(-r_{10}/r_{11}))$, then

$$\alpha_Y = \arctan\left(-\frac{r_{10}}{r_{11}}\right) \tag{18}$$

else

$$\alpha_Y = \arctan\left(-\frac{r_{10}}{r_{11}}\right) + \pi. \tag{19}$$

Proof: The rotation component of a reference system built with a pure rotation α around the z-axis of the second reference system is

$$R_{N/M}R_{(\alpha,0,0)} = \begin{bmatrix} r_{00}\cos(\alpha)+r_{01}\sin(\alpha) & -r_{00}\sin(\alpha)+r_{01}\cos(\alpha) & r_{02} \\ r_{10}\cos(\alpha)+r_{11}\sin(\alpha) & -r_{10}\sin(\alpha)+r_{11}\cos(\alpha) & r_{12} \\ r_{20}\cos(\alpha)+r_{21}\sin(\alpha) & -r_{20}\sin(\alpha)+r_{21}\cos(\alpha) & r_{22} \end{bmatrix}. \tag{20}$$

A unit vector oriented along the y-axis of the camera N will be expressed in the reference frame of camera 1 as

$$\hat{j}_N|_M = R_{N/M}R_{(\alpha,0,0)} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{21}$$

$$\overset{(20)}{=} \begin{bmatrix} -r_{00}\sin(\alpha)+r_{01}\cos(\alpha) \\ -r_{10}\sin(\alpha)+r_{11}\cos(\alpha) \\ -r_{20}\sin(\alpha)+r_{21}\cos(\alpha) \end{bmatrix}. \tag{22}$$

We aim at maximizing the scalar product between the y-axes of the two cameras

$$\hat{j}_N|_M \cdot \hat{j}_M|_M = \hat{j}_N|_M \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = -r_{10}\sin(\alpha) + r_{11}\cos(\alpha). \tag{23}$$

To maximize the scalar product (23), we pose its first derivative with respect to α equal to 0 and its second derivative negative to

$$\begin{aligned} \frac{\partial}{\partial\alpha}(\hat{j}_N|_M \cdot \hat{j}_M|_M) &= \frac{\partial}{\partial\alpha}(-r_{10}\sin(\alpha) + r_{11}\cos(\alpha)) \\ &= -r_{10}\cos(\alpha) - r_{11}\sin(\alpha) \\ &= 0 \end{aligned} \tag{24}$$

$$\begin{aligned} \frac{\partial^2}{\partial\alpha^2}(\hat{j}_N|_M \cdot \hat{j}_M|_M) &= \frac{\partial}{\partial\alpha}(-r_{10}\cos(\alpha) - r_{11}\sin(\alpha)) \\ &= r_{10}\sin(\alpha) - r_{11}\cos(\alpha) \\ &< 0. \end{aligned} \tag{25}$$

Together, constraints (24) and (25) yield (18) and (19).

Alternatively, one can aim at minimizing the angle between the x-axes of the two cameras by applying a rotation α_X around the z-axis of the later view. It can be shown that, in the case where the z-axes are perfectly aligned, the two angles α_Y and α_X are equal (the two reference frames can be made to coincide). In the general case where the z-axes are not perfectly parallel, the optimal angles α_Y and α_X will not be equal. The optimal angle α_X that will minimize the angle between the x-axes is given by the following relations. If $-r_{00}\cos(\arctan(r_{01}/r_{00})) < r_{01}\sin(\arctan(r_{01}/r_{00}))$, then

$$\alpha_X = \arctan\left(\frac{r_{01}}{r_{00}}\right) \tag{26}$$

else

$$\alpha_X = \arctan\left(\frac{r_{01}}{r_{00}}\right) + \pi. \tag{27}$$

The proof is similar to the one given for α_Y. ■

Because there is no *a priori* reason to believe that it is more important to align the x-axes nor the y-axes, we will use a rotation angle that is the average value of α_X and α_Y. The center of the rotation that must be applied to the image is the principal point of the camera. Fig. 3 shows an example where the described image rectification scheme is applied on an image

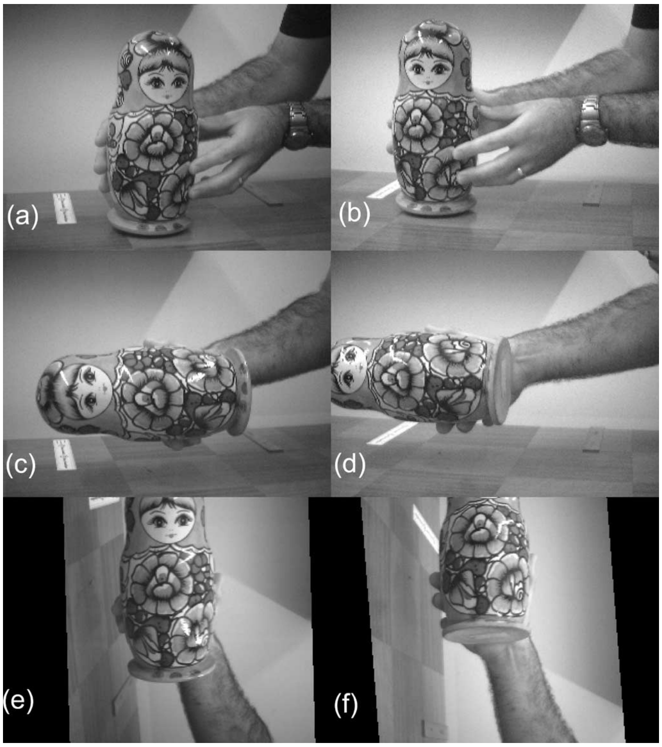


Fig. 3. (a) Initial left image. (b) Initial right image. (c) Left image after 17 registrations. (d) Right image after 17 registrations. (e) Optimally rotated left image. (f) Optimally rotated right image. These two transformed images can now be matched with images (a) and (b).

pair to make it similar to a previously captured pair. This result is further discussed in Section VI.

V. Accumulated Error Correction

Let us assume the view N has been identified as being close to the earlier view M. Then, it is possible to compute a correction matrix that can be premultiplied to the initially computed location of the view N, i.e.,

$$Q'_{N(\text{corrected})} \equiv Q_{\text{correction},N} Q_N. \tag{28}$$

The problem we would like to address now is how to correct the intermediate views. We assume there is a high level of confidence in the knowledge of the location of the view M (and therefore in the corrected location of view N). The goal is therefore to correct the intermediate views using $Q_{\text{correction},N}$.

Let us assume that the drift in the calculated location of the views was uniformly distributed over all the registration steps. Furthermore, let us assume that the individual registration steps along with the error in the registration had small rotation components. Let us model the uniform error in the following way, rewriting (14) with the introduction of Q_{error}, the unit error transformation matrix that happened at every registration, i.e.,

$$\begin{aligned} Q_n &= Q_{\text{error}} Q_{n,n-1} Q_{n-1} \\ &= \left(\prod_{i=n-1}^{M} Q_{\text{error}} Q_i \right) Q_M \end{aligned} \tag{29}$$

with $(M < n < N)$. Under the assumption of small rotation components of $\{Q_i\}$ and Q_{error}, we can commute the matrices in such a way that we gather the error matrices to the left and take them out of the product, that is

$$Q_n = Q_{\text{error}}^{n-M} \left(\prod_{i=n-1}^{M} Q_i \right) Q_M. \tag{30}$$

The correction matrix is assumed to annihilate the error of the view N. Therefore

$$Q_{\text{correction},N} = \left(Q_{\text{error}}^{N-M} \right)^{-1} = Q_{\text{error}}^{M-N} \tag{31}$$

$$Q_{\text{error}} = Q_{\text{correction},N}^{\frac{1}{M-N}}. \tag{32}$$

The correction matrix at view n will have to annihilate the error at view n, i.e.,

$$\begin{aligned} Q_{\text{correction},n} &= \left(Q_{\text{error}}^{n-M} \right)^{-1} = Q_{\text{error}}^{M-n} \\ &\overset{(32)}{=} Q_{\text{correction},N}^{\frac{M-n}{M-N}} \\ &= Q_{\text{correction},N}^{\frac{n-M}{N-M}}. \end{aligned} \tag{33}$$

Equation (33) gives the correction matrix that must be premultiplied to the calculated location of a camera at view n, given the correction matrix at view N, under the assumption of uniform distribution of the error along the registration steps, and under the assumption of small rotation components of both the registration matrices and the error transformation matrix.

It should be noted that the uniform distribution of the correction can be extended to the case of nonuniform distribution of the error. One can have some information about which registration steps contributed most to the overall error, according to some suspicion level. In this scheme, the ratio $n - M/N - M$ in (33) would be replaced by some factor α. This factor would increase monotonously between 0 (for view M) and 1 (for view N) according to the suspicion level of each registration step. For instance, the number of 3-D pairs of points that were used in the robust registration procedure could be used as a measure of the suspicion level (a high number of 3-D pairs corresponding to a low suspicion level). In all cases, the correction matrix takes care of the average component of the error.

VI. Experimental Validation of the Error Correction Scheme

In the first experiment, the movement of a Russian headstock was recorded by the stereo setup. Four images of this sequence have been presented in Fig. 3.

The projection matrices of the cameras at each position have been computed by matching, tracking, and 3-D registration of reconstructed points. The estimated angles between the z-axes and distances between the centers of projection, with respect to the first capture, have been plotted in Fig. 4. The minimum angle and distance happen at Capture 19 for the left camera.

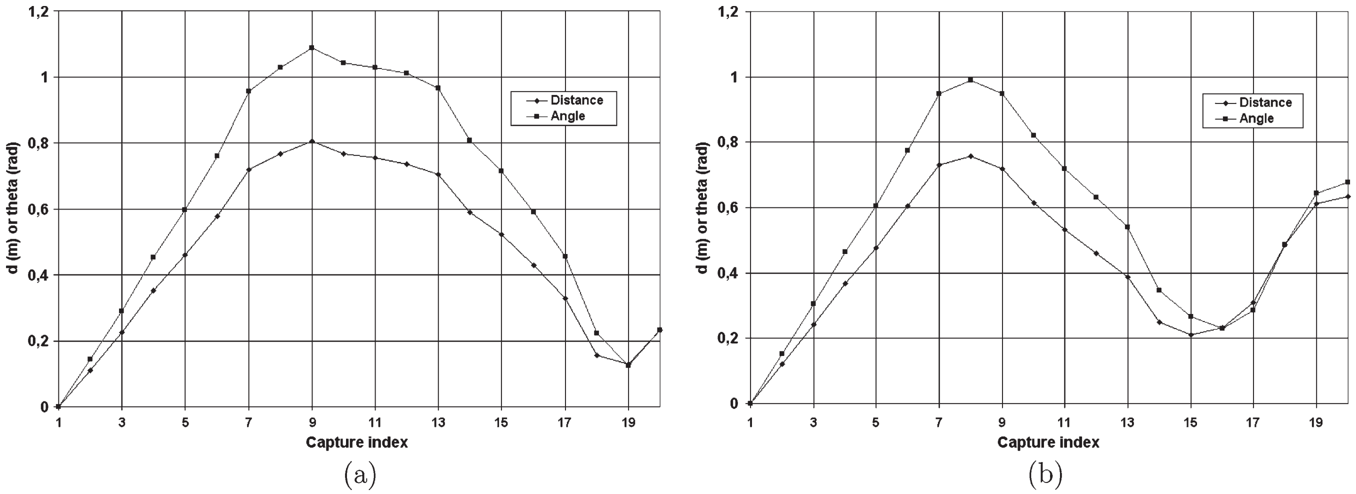


Fig. 4. (a) Angle and distance of the left camera with respect to the first capture. (b) Angle and distance of the right camera with respect to the first capture.

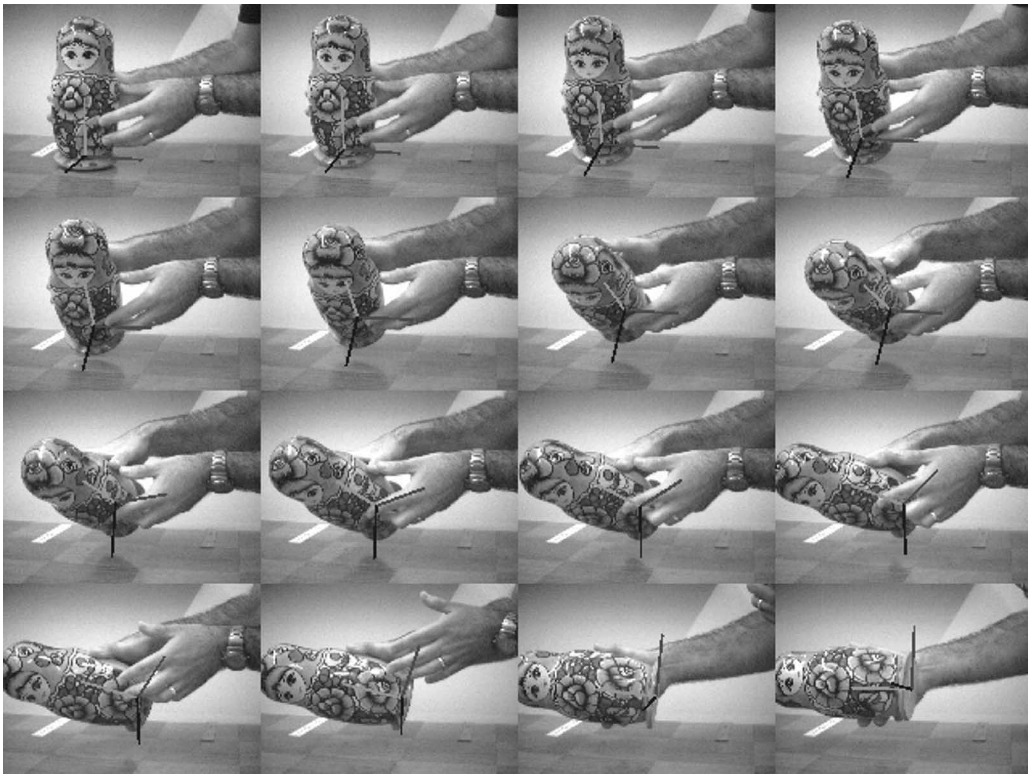


Fig. 5. Russian headstock sequence recorded by the right camera augmented with an attached reference frame (after recorrection).

For the right camera, the minimum distance happens at Capture 15, while the minimum angle happens at Capture 16. For the sake of illustration, we will correct the error at Capture 18, but it probably could have been done for Captures 16 to 19.

The rotation angles around the z-axes of the left and right cameras were estimated to be -1.54 and -1.49 rad, respectively. Fig. 3 shows the first pair of views of a sequence, the 18th pair of views, and the rotated 18th images such that tracking is possible with the first images. Error was corrected at view 18 through tracking of matched points from the initial views to the rotated 18th views. The error correction matrices were then uniformly distributed along the sequence. Fig. 5 shows the Russian headstock sequence, augmented with an attached reference frame projected on each image after recorrection of the camera positions. The natural movement of the augmented reference frame confirms the validity of the corrected projection matrices.

The second set of experiments tries to evaluate the quality of the 3-D information obtained. To this end, a bundle adjustment program was used. Bundle adjustment is an iterative method of computing the camera pose and the 3-D location of feature points given a set of matches from different cameras and the intrinsic calibration parameters of the cameras. The problem is to minimize the sum of the Euclidean-squared distance between

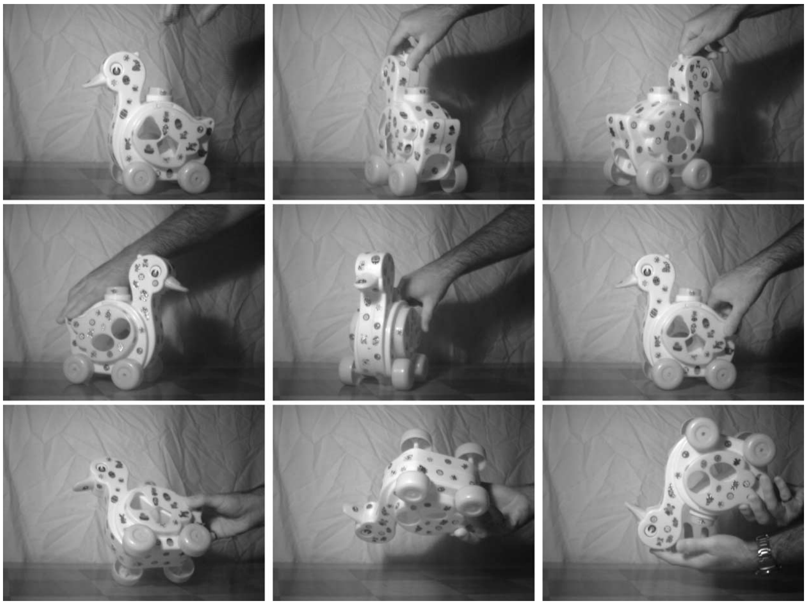


Fig. 6. Some images of the duck sequence recorded by the left camera (frames 1, 11, 17, 24, 32, 40, 49, 59, and 64).

the reprojection of the 3-D points and their corresponding image points [22], [23] by varying the cameras and the 3-D points locations, i.e.,

$$\min\left[\sum_i\sum_j\left\|\mathbf{x}_i^j - P_i\mathbf{X}_i\right\|^2\right] \tag{34}$$

where P_i are the projection matrices, and $\mathbf{x}_i^j$ is the image of the jth point at view i.

Bundle adjustment is difficult to automate because it is very sensitive to false matches and initial camera pose estimates, but when it converges correctly, bundle adjustment gives an optimal solution, i.e., the 3-D configuration that best explains the observations. For this reason, the bundle adjustment solution will be used here to generate the "ground truth" in our experiments to validate our error correction scheme.

Most commercial implementations of bundle adjustment rely on the manual selection of matches. PhotoModeler[1] is a commercial software that implements bundle adjustment from a set of manually identified matches in a set of images; it has been used to process a subset of the Duck sequence (see Fig. 6). Fourteen images were manually matched with 68 feature points. The returned camera positions are displayed in Fig. 7 along with the 3-D location of the selected feature points. Because a good match set was used and the bundle software was run with judiciously chosen optimization parameters, we were able to obtain a small reprojection error. This solution can therefore be safely considered to be accurate.

The camera path forms two loops, namely 1) 360° in the $x-y$ plane and 2) a half-turn under the duck. These loops allow us to apply our error correction scheme. Following the procedure defined in Section IV, frames 1 and 40 were detected as the two extreme views of a looping sequence (and similarly for 17 and 64). These were then matched and their 3-D point sets registered.

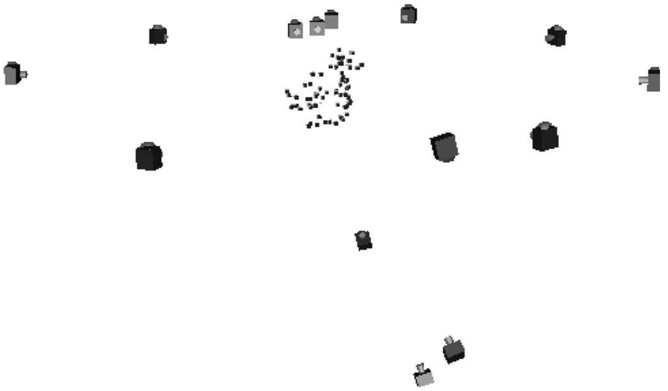
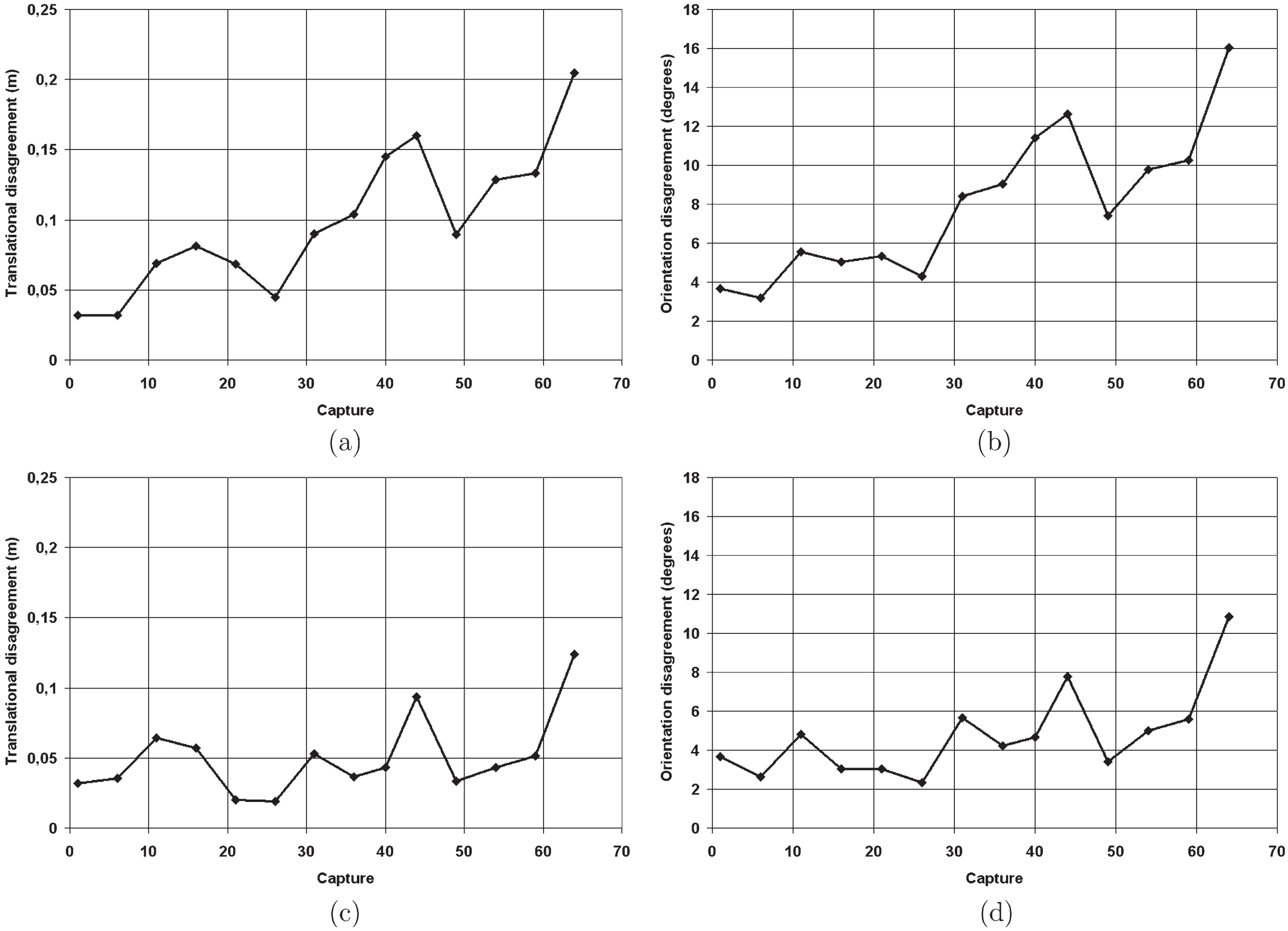


Fig. 7. Positions of the left camera in the duck sequence as computed by PhotoModeler.

Fig. 8(a) and (b) shows the disagreement (in the position and orientation of the cameras) between PhotoModeler and the proposed method without error correction. The position disagreement is the distance between the computed centers of projection. The orientation disagreement is the angle between the z-axes of the camera reference frames. As expected, the magnitude of the disagreement increases with the number of registrations, as the proposed method accumulates error. The PhotoModeler project had matches between the first and the last image, allowing for a closed-loop configuration and thus preventing error accumulation.

It is worth noticing the fact that, as opposed to the proposed method, bundle adjustment does not grant any special status to the first capture. It can be adjusted like every other camera position. In contrast, the proposed method gives a higher level of confidence in the earlier captures. The discrepancy between

[1] EOS Systems Technology (www.photomodeler.com).



(a)

(b)

(c)

(d)

Fig. 8. (a) Position disagreement between bundle adjustment and the proposed method without error correction. (b) z-axis orientation disagreement between bundle adjustment and the proposed method without error correction. (c) Position disagreement between bundle adjustment and the proposed method after error correction. (d) z-axis orientation disagreement between bundle adjustment and the proposed method after error correction.

the two methods at the first capture is most probably related to errors in the bundle adjustment solution.

Fig. 8(c) and (d) shows the disagreement between PhotoModeler and the proposed method after the two passes of error correction through uniform distribution of the correction matrix. It can be seen that the disagreement magnitude increases a lot more slowly with the number of registrations, as compared with Fig. 8(a) and (b), indicating that error correction provided an improvement in the projection matrices.

The computed locations of the cameras can be used to build a volumetric representation of the object through shape-from-silhouette [13]. Fig. 9 shows the model obtained by silhouette intersection of 82 images. The model contains approximately 12600 voxels, each having dimensions of $5 \times 5 \times 5$ mm that roughly correspond to the precision of the system. Some inaccuracies in the model are visible (in the wheels for example), but the resulting 3-D shape is clearly consistent with the real object. The presence of the hand in the images did not pose a problem here because view registration is robustly computed. Indeed, matches on the hand surface were filtered out, as their reconstructions were not moving rigidly with respect to the surface of the object. As can be seen, our error correction scheme gave object pose information of sufficient accuracy to obtain a model of relatively good accuracy. Object modeling is therefore achieved in a very convenient way by simply moving an object in front of a stereo setup in a totally unconstrained manner.

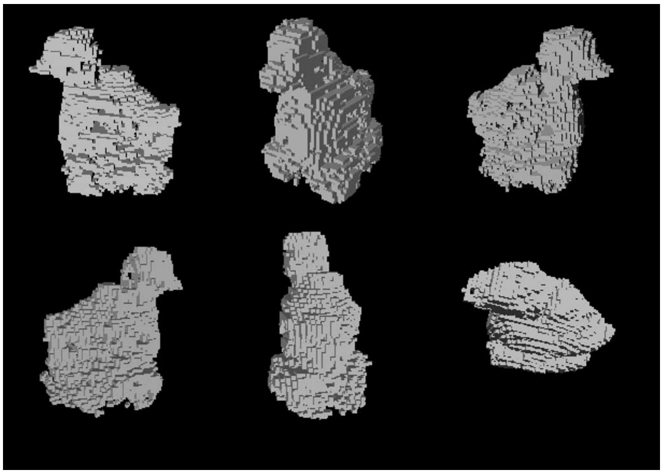


Fig. 9. Views of the duck 3-D model obtained from a sequence of 82 images (each square voxel having a dimension of 5 mm).

VII. Conclusion

In this paper, we addressed the problem of 3-D registration of a rigid object moving in front of two cameras, which is equivalent to the problem of camera pose estimation. We used a calibrated stereoscopic vision setup to track the camera positions along sequences of a moving rigid object. We proposed a robust 3-D registration procedure that exploits the rigidity of the scene to automatically filter out the reconstructed points originating from false matches and errors in feature tracking. An error correction scheme was introduced, which takes advantage of loops in the movement of the cameras to compensate for the accumulated error. Through experimental results, we showed the validity of the obtained projection matrices and that their accuracy was sufficient for tasks such as model building or scene augmentation.

References

[1] L. Vacchetti, V. Lepetit, and P. Fua, "Fusing online and offline information for stable 3D tracking in real-time," in *Proc. Int. Conf. Comput. Vis. and Pattern Recog.*, 2003, pp. 241–248.

[2] A. W. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences," in *Proc. Eur. Conf. Comput. Vis.*, 1998, pp. 311–326.

[3] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[4] G. Roth, "Computing camera positions from a multi-camera head," in *Proc. IEEE 3rd Int. Conf. 3-D Digital Imag. and Modeling*, 2001, pp. 135–142.

[5] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. IEEE and ACM Int. Workshop Augmented Reality*, 1999, pp. 85–94.

[6] R. Horaud and G. Csurka, "Self-calibration and Euclidean reconstruction using motions of a stereo rig," in *Proc. Int. Conf. Comput. Vis.*, 1998, pp. 96–103.

[7] Z. Zhengyou, "Motion and structure of four points from one motion of a stereo rig with unknown extrinsic parameters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 12, pp. 1222–1227, Dec. 1995.

[8] H. Pui-Kuen and C. Ronald, "Stereo-motion with stereo and motion in complement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 2, pp. 215–220, Feb. 2000.

[9] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[10] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, Sep. 1987.

[11] A. J. Davison and D. W. Murray, "Simultaneous localisation and map-building using active vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 865–880, 2002.

[12] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-time 3D model acquisition," *ACM Trans. SIGGRAPH 2002)*, vol. 21, no. 3, pp. 438–446, Jul. 2002.

[13] L. Aldo, "The visual hull concept for silhouette-based image understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 2, pp. 150–162, Feb. 1994.

[14] V. Étienne and R. Laganière, "Matching feature points in stereo pairs: A comparative study of some matching strategies," *Machine Graphics and Vision*, vol. 10, no. 3, pp. 237–259, 2001.

[15] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th IJCAI*, 1981, pp. 674–679.

[16] Z. Zhengyou, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recog.*, 1999, vol. 1, pp. 666–679.

[17] P. Sturm and S. Maybank, "On plane-based camera calibration: A general algorithm, singularities, applications," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recog.*, 1999, vol. 1, pp. 432–437.

[18] R. Y. Tsai and S. Maybank, "Estimating three-dimensional motion parameters of a rigid planar patch," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 30, no. 4, pp. 525–534, 1982.

[19] H. Malm and A. Heyden, "Stereo head calibration from a planar object," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Patt. Recog.*, Dec. 2001, vol. 2, pp. II-657–II-662.

[20] F. Dornaika and R. Chung, "Cooperative stereo-motion: Matching and reconstruction," *Computer Vision and Image Understanding*, vol. 79, no. 3, pp. 408–427, 2000.

[21] R. Klette, K. Schluns, and A. Koschan, *Computer Vision: Three-Dimensional Data From Images*. New York: Springer-Verlag, 1996.

[22] A. Bartoli, "A unified framework for quasi-linear bundle adjustment," in *Proc. 16th Int. Conf. Pattern Recog.*, 2002, vol. 2, pp. 560–563.

[23] Y. Han, "Relations between bundle-adjustment and epipolar-geometry-based approaches, and their applications to efficient structure from motion," *Real-Time Imaging 10*, vol. 10, no. 1, pp. 389–402, Feb. 2004.

[24] K. Hanna and N. Okamoto, "Combining stereo and motion analysis for direct estimation of scene structure," in *Proc. 4th Int. Conf. Comput. Vision*, 1993, pp. 357–365.

Robert Laganière (M'97) received the B.Ing. degree in electrical engineering from Ecole Polytechnique de Montreal, Montreal, QC, Canada, in 1987 and the Ph.D. degree from INRS-Telecommunications, Montreal, in 1996.

His research interests are in computer vision and image processing with applications to augmented reality, visual surveillance, 3-D reconstruction, and image-based rendering. He is also the coauthor of a book on object-oriented software development published by McGraw-Hill.

Sébastien Gilbert received the degree in physical engineering from Université Laval, Québec, QC, Canada, in 1996 and the M.S. degree in electrical engineering from the University of Ottawa, Ottawa, ON, Canada, in 2005.

From 1996 to 2002, he was with Alcatel Optronics Canada, where he led development projects of fiber-optic components involving fiber Bragg gratings and fused fiber couplers. Since 2005, he has been the CCD Project Manager for Dalsa Semiconducteur, Bromont, QC, where he leads projects in CCD process development. His research interests are fiber-optic components simulation and 3-D machine vision.

Gerhard Roth (M'87–S'87–A'88–SM'91) received the B.Math. degree (Hons) in computer science from the University of Waterloo, Waterloo, ON, Canada, the M.S. degree in computer science from Carleton University, Ottawa, ON, and the Ph.D. degree in electrical engineering from McGill University, Montreal, QC, Canada.

He is the Group Leader of the Computational Video Group, Institute for Information Technology, National Research Council of Canada, Ottawa, working in the fields of perceptual audio/video user interfaces, augmented reality, broadband visual communication, and building models from sensor data. He has done research in evolutionary algorithms, robust statistics applied to computer vision, and in building 3-D models from sensor data. His current research activities are in the fields of augmented reality and projective vision.