Registration of a Moving Rigid Object Using a Stereoscopic Vision Setup

Sébastien Gilbert and Robert Laganière

School of Information Technology and Engineering
University of Ottawa, Ottawa, Canada, K1N 6N5
sgilbert, laganier@site.uottawa.ca

Abstract

This paper addresses the problem of tridimensional registration of a moving rigid object. Matching, tracking and 3D reconstruction of feature points by a stereoscopic vision setup allows the computation of the homogeneous transformation matrix linking two consecutive scene captures. Robustness to errors is provided by the scene rigidity constraint. Accumulation of error is compensated through loop detection in the calculated camera positions.

1 Introduction

Tracking the 3D movement of a rigid object (or, alternatively, of a camera recording images of the object) has important applications in augmented reality systems, 3D modelling and robotics. Ideally, 3D tracking should be performed automatically and should be robust to noise.

A pair of cameras whose intrinsic and extrinsic calibration parameters are known forms a calibrated stereoscopic vision setup. It allows 3D reconstruction of matched points [1]. If the feature points at capture N are tracked in both images in capture $N+1$, the two clouds of 3D points can be registered [3], leading to the new position of the cameras. This idea is used to track the movements of the cameras with respect to a rigid object along a sequence. A correction scheme is proposed, that compensates for the accumulated error in the computed positions, exploiting the detection of loops in the movement.

2 Basic Tools

This section briefly presents the building blocks used in this paper.

2.1 Calibration

Calibration aims at computing the projection matrices of two cameras [1]. Let us assume that we have a set of n 3D points for which we know the global homogeneous coordinates $\vec{X}_i$. Each point, along with its corresponding image coordinates $\vec{u}_i$, allows to write:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_i = \lambda_i \begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_i \qquad (1)$$

Eliminating the λ_i and rearranging the expressions yields a pair of homogeneous linear equations in 12 unknowns, the entries of the projection matrix. Putting together the information of the n 3D points ($n \geq 6$) gives $2n$ homogeneous linear equations in 12 unknowns $p_{00}, p_{01}, ..., p_{23}$. This system can be solved up to a scale factor, through SVD. The quality of the computed projection matrix depends on the linearity of the camera model and the accuracy in the measured 3D 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 [1].

2.2 Matching

It is assumed that the two cameras are sufficiently close and parallel to each other to allow matching through correlation. The fundamental matrix is also available, since the stereo setup is calibrated. The matching procedure is performed in the following three steps:

1. Identify Harris corners in both images. Filter out the pixels whose corner strength is below a given threshold;

2. For each corner in the left image, identify the corners in the right image that are close enough to its epipolar line;

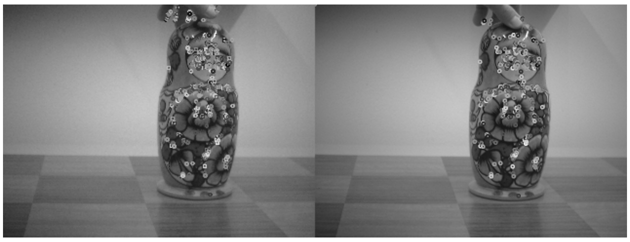

Figure 1. Pair of consecutive captures with tracked feature points

3. For each corner in the left image, compute the correlation of square windows centered on the corner of interest and each candidate corner in the right image. If the best correlation is above a given threshold, keep it as a match. Additional matching constraints can also be applied to improve the quality of the match set [5].

2.3 Tracking

The tracking function considers two images taken by the same camera at different instants, and a list of feature points to be tracked from instant N to instant $N + 1$. The tracking function must search in a disk whose center and radius are parameters. The mechanisms of identifying candidate corners and applying correlation is the same as described in Section 2.2. Figure 1 shows the result of the tracking algorithm, with corresponding feature points that were tracked.

Alternatively, the KL tracking algorithm [6] could have been used.

2.4 3D Reconstruction

Let us assume the projection matrices P_1 and P_2 of the cameras are known, and we want to compute the 3D location $\vec{X}$ of a feature point whose image coordinates in the two images, $\vec{u}_1$ and $\vec{u}_2$, are known. The projection equations have the form $\vec{u}_j = \lambda_j P_j \vec{X}$, $(j = 1, 2)$. They can be manipulated to yield 4 linear equations in 3 unknowns, X, Y and Z:

$$\begin{bmatrix} (p_{00} - up_{20})_1 & (p_{01} - up_{21})_1 & (p_{02} - up_{22})_1 \\ (p_{10} - vp_{20})_1 & (p_{11} - vp_{21})_1 & (p_{12} - vp_{22})_1 \\ (p_{00} - up_{20})_2 & (p_{01} - up_{21})_2 & (p_{02} - up_{22})_2 \\ (p_{10} - vp_{20})_2 & (p_{11} - vp_{21})_2 & (p_{12} - vp_{22})_2 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (up_{23} - p_{03})_1 \\ (vp_{23} - p_{13})_1 \\ (up_{23} - p_{03})_2 \\ (vp_{23} - p_{13})_2 \end{bmatrix} \quad (2)$$

This system can be solved through a least-square method.

3 Robust Registration

After having found matches and tracked the corresponding points in both sequences, two clouds of $3D$ points can be reconstructed. Based on the matches at instant N and their tracked correspondents at instant $N + 1$, these two clouds of $3D$ points can be registered to find the rigid motion of the object [3] (or, alternatively, the rigid motion of the stereo setup, when the reference frame is attached to the object). Unfortunately, one cannot use the raw data, since the false matches and the tracking errors will corrupt the result. Instead, it is necessary to incorporate a random sample consensus (RANSAC) algorithm [2] that will filter out the bad pairs of 3D points.

A minimum of 3 pairs of non-collinear 3D points are necessary to perform a 3D registration. As a consequence, the first step of the algorithm will consist in finding a trio of 3D matches.

3.1 Random Drawing of a Trio of 3D Matches

In order to make sure that a randomly drawn trio of 3D matches does not constitute a degenerate case (i.e. is not in a collinear configuration), two conditions must be imposed:

1. The distance between any two points in 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 since three collinear points that are located far apart would satisfy it, while the second item alone would allow a trio constituted of two points close from each other with a third point far away, such that the area of the triangle is sufficient.

Once both trios have been identified as being non-collinear, the rotation and the translation that best describe the rigid movement of the points can be computed [3]. This is a candidate registration $(R_{reg}, \vec{T}_{reg})$.

3.2 Count of the Number of Matches that Agree with the Candidate Registration

Given a candidate registration, a count of the number of agreeing matches can be performed. For each 3D match, if the distance between $\vec{X}|_{Ref1}$ and $R_{reg}\vec{X}|_{Ref2} + \vec{T}_{reg}$ is less than a maximum distance (a parameter), then this match is said to agree with the candidate registration. The whole procedure of Sections 3.1 and 3.2 is repeated several times. The number of trials can be set such that the probability of success at finding at least one trio of good matches is above a desired value [2]. The candidate registration having the highest number of agreeing matches is declared the best candidate registration.

3.3 Identification of the Good Matches and Final Registration

Finally, all the matches that agree with the best candidate registration are used to compute the final output registration:

$$Q_{reg} = \left[\begin{array}{cc} R_{reg} & \vec{T}_{reg} \\ \mathbf{0}^T & 1 \end{array} \right] \tag{3}$$

3.4 Computation of the New Positions of the Cameras

From the registration homogeneous transformation Q_{reg}, the new positions of the cameras can be computed:

$$Q_{C_{N+1}/W} = Q_{reg_N} Q_{C_N/W} \tag{4}$$

One of the main problems associated with such a technique is the accumulation of error, due to the fact that every new position is computed from the previous. It is assumed that no special target points that could allow recalibration are available on the object. Instead, one must rely on the knowledge of the approximate camera positions to identify points of view that were previously captured (loop detection). This information will be used to correct for the drift, each time the cameras pass by a location where they have been before.

4 Detection of Previously Viewed Locations

This procedure aims at identifying, in a sequence, camera positions that are close to their previous positions in an earlier image capture.

As pointed out in section 2.2, we won't address the situations of wide-baseline matching or tracking. This means that, in order to be able to match images captured at non-consecutive instants, two conditions must be met:

1. The Z-axes of the two views must be near parallel;

2. The distance between the center of projection of the views must be sufficiently small.

In reality, regarding the first item, it is not sufficient that the Z-axes be near parallel, it is also necessary to have the Y- (or the X-) axis nearly parallel for the correlation technique to work. Nevertheless, we can relax this constraint since our knowledge of the approximate camera positions will allow us to rotate the images around their Z-axes in such a way that they are sufficiently aligned.

The distance between the center of projection of the views is directly calculated from the length of the translational vector going from one center to the other. In order to calculate the maximum distance that we can afford, 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 closer the object of interest will be to the cameras, the smaller the tolerance on the distance between the views will be, since the tracking algorithm is obviously not scale invariant.

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:

$$\cos(\theta) = \left(Q_{C_M/W} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right) \cdot \left(Q_{C_N/W} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right) \tag{5}$$

where $Q_{C_M/W}$ and $Q_{C_N/W}$ are the homogeneous transformation matrices linking a camera at capture M and at capture N with respect to the world reference frame (attached to the object).

The angle between the Z-axes of the left camera at capture M and N need not be the same as the equivalent for the right camera. 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 the right camera. 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 having been previously captured, the drift of the later view can be compensated for. 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.

4.1 Identification of the Rotation Angle Around the Z-Axis

As discussed previously, a pair of similar views must have their Z-axes nearly parallel, but they can have a wide angular difference around their Z-axes. Since 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. We will be searching for the rotation that must be applied to the image of the later view, such that it is as aligned as possible with the earlier view.

The rotation matrix linking the later view N with the earlier view M is R_{C_N/C_M}. It is known approximately. We will aim at minimizing the angle between the Y-axes of the two views by applying a rotation around the Z-axis of the second view. To do so, we post-multiply to R_{C_N/C_M} the matrix of a pure rotation around the Z-axis of the later view. This is the rotation matrix linking the reference frame of the

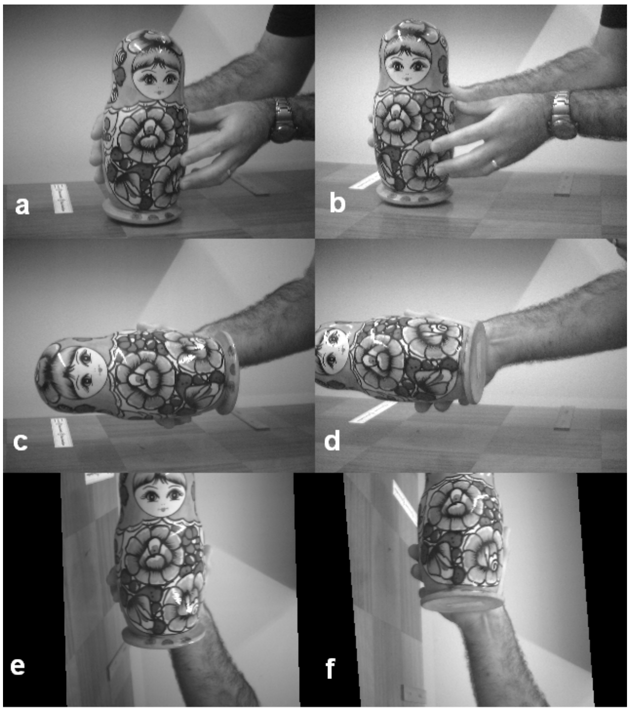

Figure 2. (a) Initial left image; (b) Initial right image; (c) Left image after 17 registrations; (d) Right image after 17 registrations; (e) Optimally rotated left image; (f) Optimally rotated right image. These two transformed images can now be matched with images (a) and (b)

earlier view with the reference frame of the later view, arbitrarily rotated by an angle α around its Z-axis. We then compute the scalar product of unit vectors parallel to the two Y-axes, as expressed in the earlier view's reference frame. The optimal angle is such that this scalar product is maximized. Let us state the result:

Let r_{ij} be the element (i, j) of the rotation matrix linking the later view N with the earlier view M, R_{C_N/C_M}.

If $r_{10}sin(\arctan(-\frac{r_{10}}{r_{11}})) < r_{11}cos(\arctan(-\frac{r_{10}}{r_{11}}))$, then:

$$\alpha_Y = \arctan(-\frac{r_{10}}{r_{11}}) \qquad (6)$$

else:

$$\alpha_Y = \arctan(-\frac{r_{10}}{r_{11}}) + \pi \qquad (7)$$

The angle α_Y is the rotation angle that must be applied around the principal point (known since it is part of the intrinsic calibration parameters) of an image at view N, such that the Y-axes of the camera at view N and at view M are as parallel as possible. It must be stressed that one could have decided equivalently to align the X-axes instead and would have then obtained a slightly different mathematical expression.

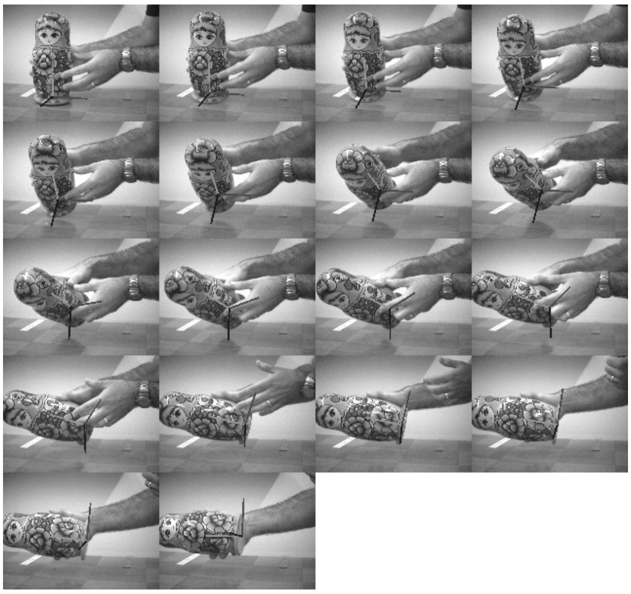

Figure 3. Russian Headstock sequence, as seen by the right camera, augmented with its attached reference frame

Given a pair of stereo images at view N that is identified as being close to an earlier pair of images at view M, the optimal angles that must be applied to the later images are not necessarily the same for the left and right cameras.

4.2 Correction of the Later Camera Positions

Once the later views have been optimally rotated, tracking can take place on feature points from the earlier capture to the later rotated images. Of course, the tracked feature points must be de-rotated prior to 3D reconstruction. The robust registration algorithm can then be applied between the two clouds of 3D points, and the later camera positions corrected accordingly.

5 Experimental Results

Figure 2 shows the first pair of views of a sequence, the 18^{th} pair of views and the rotated 18^{th} images such that tracking is possible with the first images. The error was corrected at view 18 through tracking of matched points from the initial views to the rotated 18^{th} views. The error correction matrices were then uniformly distributed along the sequence. Figure 3 shows the *Russian Headstock* sequence, augmented with its attached reference frame.

The computed locations of the cameras can be used to

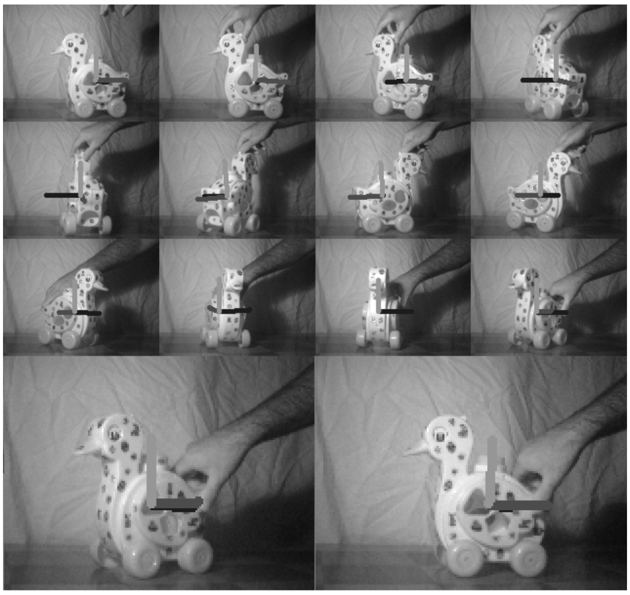

Figure 4. Samples of the Duck sequence, as seen by the left camera, augmented with its attached reference frame

build a volumetric representation of the object, through shape-from-silhouette [4]. Figure 4 shows the *Duck* sequence, augmented with its attached reference frame and Figure 5 shows the model obtained by silhouette intersection of 82 images. The model contains approximately 12600 voxels, each having dimensions of 5 mm × 5 mm × 5 mm. The presence of the hand in the images did not pose a problem here since the registration algorithm is robust. Matches on the hand surface were filtered out, as their reconstructions were not moving rigidly with respect to the surface of the object. Regarding model building, both the hand and its shadow were considered part of the silhouette in each individual image, but since they were constantly moving with respect to the reference system, they were progressively eliminated by the silhouette intersection, leaving only the object rigid body.

6 Conclusion

In this paper, we addressed the problem of 3D 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 3D 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.

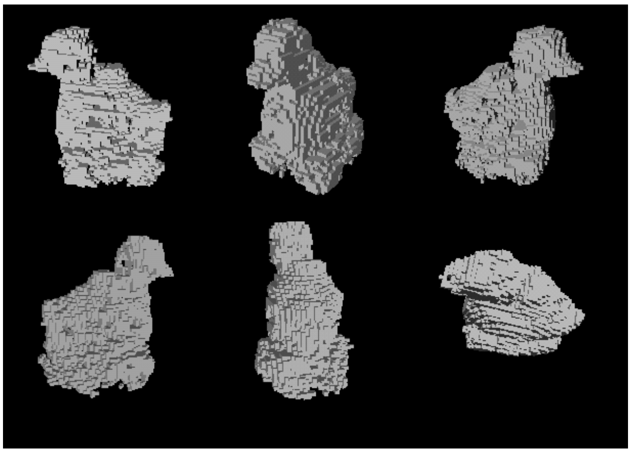

Figure 5. A few views of the Duck model

References

[1] Trucco E., Verri A. 1998. Introductory Techniques for 3-D Computer Vision, Prentice-Hall (eds)

[2] Fischler Martin A., Bolles Robert C. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, in *Communications of the ACM*, vol. 24, no. 6, June 1981, pp 381-395

[3] Arun K.S., Huang T.S., Blostein S.D. “Least-Squares Fitting of Two 3-D Point Sets”, in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, Sept. 1987

[4] Laurentini Aldo “The Visual Hull Concept for Silhouette-Based Image Understanding”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, February 1994, pp 150-162

[5] Vincent Étienne, Laganière Robert “Matching Feature Points in Stereo Pairs: A Comparative Study of Some Matching Strategies”, in *Machine Graphics and Vision*, vol. 10, no. 3, 2001, pp 237-259

[6] Lucas B., Kanade T. “An Iterative Image Registration Technique with an Application to Stereo Vision”, in *Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, 1981, pp. 674-679