Edge-Based Markerless 3D Tracking of Rigid Objects

Javier Barandiarán, and Diego Borro
CEIT and Tecnun (University of Navarra)
Manuel de Lardizábal 15, 20018 San Sebastián, Spain
{jbarandiaran, dborro}@ceit.es

Abstract

In this paper we present a real-time 3D object tracking algorithm based on edges and using a single pre-calibrated camera. During the tracking process, the algorithm is continuously projecting the 3D model to the current frame by using the pose estimated in the previous frame. Once projected, some control points are generated along the visible edges of the object. The next pose is estimated by minimizing the distances between the control points and the edges detected in the image.

1. Introduction

The optical tracking of objects is a fundamental piece inside the complex field of artificial vision. In the 2D tracking case, the objective is to know the position of an object, in the image plane, in every single frame of a video sequence. In the 3D case, in addition, it is necessary to know the relative pose, position and orientation, between the object and the camera.

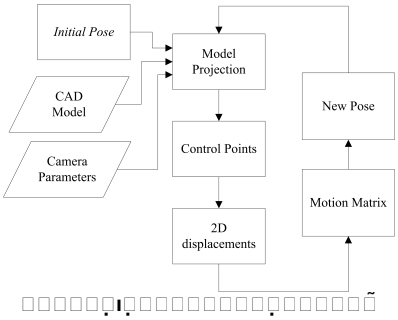
2. State of the Art

Many approximations have been realized to solve the problem of the3D tracking using computer vision. A recent survey [1] shows an excellent and very complete state of the art. There are mainly two kinds of methods: frame to frame i.e., the current pose of the object is calculated using the pose estimated in the previous frame [2]; and tracking by detection [3]. Basically, the last ones consist on detecting features of the objects in the images, matching them with the known features of a pre-built model and trying to calculate the pose using these relations.

The use of 3D information for tracking algorithms is widely extended because this knowledge makes the tracking process more robust [4]. These algorithms can use different information in order to match the 3D model. Some of them use the edges of the objects as features [2, 4, 5]. Others rely on the information extracted from pixels, using techniques like optical-flow [6]. Several works have also been done combining both methods [5] obtaining very good results.

3. Tracking method

The implemented model-based tracking system follows a recursive scheme frame by frame. The result of each iteration is a 3x4 transformation matrix V = [R | t], called the motion matrix, that transforms the pose calculated in the previous frame into the pose of the object in the new frame. In Figure 1, a scheme of the algorithm is shown.



The steps and data of the process are the following:

- Initial pose: This is done by the user and estimates the pose of the object in the first frame.
- Model projection: The projection of the model edges are calculated using the previous pose.
- Control points: Control points are generated along the visible edges of the model calculating their 3D coordinates in the model coordinates system and their 2D projections. A visibility test is done for control points using the OpenGL occlusion queries.
- 2D displacements: The distances between the projections of the control points and the edges detected in the image are calculated.
- Motion matrix: Using the data generated in the previous steps, the motion matrix is estimated. In next section, this step will be explained in more detail.
- New pose: The pose of the object is updated using the motion matrix.

4. Motion matrix estimation

With the 2D and 3D data generated with the previous steps the system is able to estimate the movement done by the object between the previous and the new frame. The objective of this step is to estimate the motion matrix V that transforms the previous pose S' into the new one S, i.e., $S = VS'$.

4.1. Motion matrix

The motion matrix computation can be reduced to 6 parameters using the motion vector representation:

$$W = [\ \omega x\ \omega y\ \omega z \mid tx\ ty\ tz\ ]^T$$

The rotation matrix R can be obtained from the vector ω=[ωx,ωy,ωz] and an angle $\theta = \|\boldsymbol{\omega}\|$ using the Rodrigues formula.

4.2. Estimation

The motion vector W can be estimated with a linear least-squares minimization process that minimizes the residual errors. The residuals errors are the absolute difference between a group of lineal equations (*n*x6 matrix A being *n* the number of control points), which unknowns are the components of the motion vector W, and the vector of distances l_i (these are the 2D displacements previously calculated).

$$W = \min \sum_i (A_i W - l_i)^2$$

However this is not a robust estimation, because the outliers have more influence that the inliers. In order to estimate a correct vector motion is necessary to reduce the number or the influence of the outliers in the minimization process. Several strategies can be used:

- Non-linear minimization: using the Levenberg-Marquardt method, the problem becomes an iterative minimization process trying to minimize the distance between the projections of the control points m_i and the edges b_i.

$$W = \min \sum_i \left(\text{dist}(m_i, b_i)^2 \right)$$

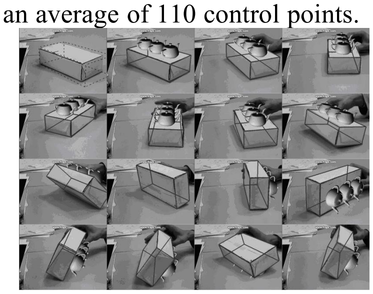
- Rejecting outliers: all the displacements corresponding to the control points of a model edge must be adjusted to the same line model. This line can be estimated and the outliers are detected and rejected based on its distance to this line.
- Multiple hypotheses: several edges or hypotheses found inside the search range are retained. During the minimization process for each control point m_i, one of its hypotheses b_{ij} is selected in order to use always the minimum distance.

$$W = \min \sum_i \left(\min_j \left(\text{dist}(m_i, b_{ij})^2 \right) \right)$$

- Robust estimator: the least-squares estimator can be replaced by a robust estimator, also called M-Estimator, as the Tukey estimator.

5. Results

The system has been tested with an Intel Pentium 2.8 Ghz PC with 1GB of RAM and an 30 fps IEEE-1394 camera with a resolution of 640x480. However, the tracker could be able of working at 50 fps with other cameras. Figure 3 shows a tracking sequence of a box with an average of 110 control points.



6. Conclusions

We have presented a real-time 3D tracking algorithm. It is easy to implement and fast even using a high number of control points. The benefits of the tracker consist especially in the lack of markers during the whole sequence of the images.

7. References

[1] Lepetit, V. and P. Fua, "Monocular Model-Based 3D Tracking of Rigid Objects: A Survey", *Foundations and Trends in Computer Graphics and Vision*, 1(1): Pp. 1-89. 2005.

[2] Drummond, T. and R. Cipolla, "Real-Time Visual Tracking of Complex Structures", *IEEE Transactions on pattern analysis and machine intelligence*, 24(7): Pp. 932-946. 2002.

[3] Gordon, I. and D.G. Lowe, "Scene Modelling, Recognition and Tracking with Invariant Image Features", *in Proceedings of IEEE/ACM Symposium on Mixed and Augmented Reality (ISMAR)*. Arlington, VA, USA. Pp. 110-119. 2004.

[4] Harris, C., Tracking With Rigid Objects, *in Active Vision*. 1993, MIT press. Pp. 59-73.

[5] Vacchetti, L., V. Lepetit, and P. Fua, "Combining edge and texture information for real-time accurate 3D camera tracking", *in Proceedings of IEEE/ACM Symposium on Mixed and Augmented Reality (ISMAR)*. Arlington, VA, USA. Pp. 48-57. 2004.

[6] Basu, S., I. Essa, and A. Pentland, "Motion regularization for model-based head tracking", *in Proceedings of International Conference on Pattern Recognition (ICPR)*. Vienna, Austria. Pp. 611. 1996.