REAL TIME TRACKING USING STEREO VISION FOR AUGMENTED REALITY

F.I. Cosco, F. Caruso, F. Bruno, M. Muzzupappa

Università della Calabria
Rende(CS) Italy

{ficosco, francesco.caruso, f.bruno, muzzupappa}@unical.it

Abstract

Correct tracking of the user position is crucial for an Augmented Reality (AR) application. For this reason, tracking became a very important research subject in the field of real-time AR. The most common solution to implement an AR system is based on the use of an Head Mounted Display (HMD) and two cameras for the stereoscopic video see-through visualization. Almost all the systems presented in literature employ one camera to obtain a marker based video-tracking. This technique is widespread because it offers sufficient precision and accuracy. This paper describes an innovative tracking solution based on a stereo vision approach which aims to use the information coming from both cameras in order to improve tracking accuracy. The proposed solution is also able to switch itself into a standard single camera visual tracking when the stereo pair correspondence is missing. Further, a comparison of the performance of both mono and stereo tracking is reported.

Keywords Augmented Reality, Stereo Tracking, Optical Tracking, Pose Estimation

1. INTRODUCTION

Augmented Reality (AR) systems enhances the perception of the real world by adding virtual objects within a real scene. An AR scene, in fact, is created by combining the real scene viewed by the user and a virtual scene generated by computers, augmenting the scene with additional information which can be useful to the user. In the last decades, a great variety of AR applications and services has been developed, both for entertainment and professional tasks. A comprehensive overview of AR could be found in [1] and [2].

In order to obtain a correct AR scene visualisation, a robust registration of the virtual objects within the real scene is needed. Without accurate registration, the illusion that the virtual objects exist in the real environment is severely compromised. The correct alignment of virtual objects is possible only after the estimation of the user pose within the scene. Virtual objects, in fact, must be rendered in accord with the user’s point of view, and any mis-registration will prevent the user from seeing the virtual and real scenes as fused together [3].

Pose-tracking is nowadays one of the most active research area. There are several pose-tracking methods, including mechanical, magnetic, inertial and optical. An overview of tracking systems is presented in [4]. Each method has its own advantages and disadvantages. Vision-based tracking is commonly employed for AR [5]. Vision methods can estimate camera pose directly from the same imagery observed by the user, without using a separate sensor or emitter attached to the environment. This has several advantages:

- tracking may occur relative to moving objects;
- tracking measurements made from the viewing position often minimize the visual alignment error;
- tracking accuracy varies in proportion to the visual size (or range) of the objects in the image [6].

However, vision also suffers from a notorious lack of robustness, end-to-end system delay, and high computational expense. Since vision sensors (cameras) nominally sample at video rates (30Hz), they are most appropriate for
measuring low-frequency pose variations. Rapid or abrupt camera rotations or motions can cause vision tracking failures or instabilities.

Several optical-based approaches for pose estimation work for coplanar points [7], some have been extended to use points and lines [8], and some work also for arbitrary 3D target point configurations [9]. Recent success has also been reported for online structure and motion estimation [10].

Several vision-based tracking systems are based on planar targets. [11][12][13][14]. Users of such systems observe that vision-based pose is not very precise, which results in significant jitter, and not very robust suffering from pose jumps and gross pose outliers.

2. RELATED WORKS

The marker–based optical tracking is, probably the most common approach in AR applications. This technique require the solution of three sub-problems:

1. camera(s) calibration in order to compensate for image distortion and retrieve the projection matrix;
2. pattern recognition of the marker within the image(s) and solution of the correspondence problem;
3. marker pose estimation with respect to the camera system coordinates (resection problem);

ARToolkit [11] is one of the most commonly used software library able to completely solve all of these problems. Fiala improved the marker detection in non uniform illumination scenarios and against accidental occlusion of the marker[16].

Abidi et al. [19] presented a pose estimation algorithm based upon invariant properties of quadrangular markers. Quan et al. [20] give a general solution for the N-point resection problem. Another widely employed method in computer vision has been presented by Zhang [22], in which both the camera calibration and resection problem are solved for the particular case of a planar marker. For the same problem, Kato et al. [11] presented a geometrical approach based upon the projection of the lines passing through marker sides.

Pinz et al. [17] solved the resection problem taking into account pose ambiguities due to perspective projection. Lucchese [18] presented a closed solution suitable for real time application, using an approach similar to Zhang, i.e. based upon the homography relationship between marker plane and camera plane.

In this work we present a new tracking methodology based upon a stereo generalization of the Zhang method (see section 4.2), that directly computes the pose matrix of the marker given the projections of the four vertices on both left and right cameras. A new calibration strategy is also defined based on the use of this resection algorithm. Moreover, we implemented also a single camera tracking solution, which is executed when the marker is visible by only one camera.

3. HARDWARE DESCRIPTION

The system has been created starting from an eMagin Z800 HMD. This visor has a pair of sharp OLED displays with native resolutions of 800x600 pixels each, and supporting 8 bits per pixel (16.67 million colors). The unit offers a 200:1 contrast ratio. Further, the visor is equipped with a gyroscope for the head-tracking.

The visor has been modified to mount two pointgrey flea2 1394b camera. These camera
has a 1/3” color CCD sensor, with a resolution of 1024x768 pixels and a 30Hz frame-rate.

Both cameras are equipped with a Pentax 16mm f/1.4 lens.

The computer is equipped with an Intel E6400 Core2 Duo CPU and nvidia 7900GS graphics adapter.

4. THE STEREO-TRACKING STRATEGY

As described in previous sections, many authors have already developed marker-based tracking applications. In this section we describe in detail the tracking strategy we implemented, and we summarize the mathematical framework upon which the single-camera tracker (see 4.1) and the stereo-camera tracker (see 4.2) have been developed.

First of all, a pattern recognition algorithm, based upon the well known Artoolkit library [21], is performed in order to find the marker position in both left and right images. Whenever the marker is found in the image, the pixel coordinates of each vertex are stored and organized in a precise order. The conventional order permits to solve also the so to simplify the overall procedure.

The system decides which type of tracking to execute (stereo or mono) according to the logic illustrated in Fig. 2. Only when the marker is visible in both left and right image, the stereo tracking is executed; otherwise, if it is visible by just one camera, the single camera tracking is executed. Nothing is done when marker is not found at all.

The tracking stability, whenever a switching between stereo and single tracking modes occurs, is granted by the accurate knowledge of the relative pose between the cameras. As a consequence, the overall accuracy of our tracking approach, is strongly dependent by the correctness in estimating geometrical (the relative position of each camera in respect of the other) and optical (focal length, aspect ratio, skew ratio, distortion parameters, etc.) properties of the system, more generally indicated as intrinsic and extrinsic camera parameters. For this reason, also a calibration procedure (see 4.3), has been developed.

4.1. Single camera tracking

According to the general pinhole camera model, the projection law can be expressed as

\[ p = M_p \cdot P \]  

(1)

where \( P = [X,Y,Z,1]^T \) and \( p = [hu, hv, h]^T \) are respectively a generic 3d point \( (X,Y,Z) \) and its projection \( (u,v) \) on the image plane, with \( h \) an arbitrary scale factor.

\( M_p \) is the projection matrix, that can be expressed also by:

\[ M_p = A[R_i \ T_i] \quad \text{with} \quad A = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \]  

(2)
where $A$ is the camera intrinsic matrix; $R_i$ is the 3x3 rotation matrix and $T_i$ is the translation vector that describe the position of the camera frame of reference with respect to the world one.

For a generic marker position, we can relate each vertex,

$$V = [V_x, V_y, V_z]^T$$

in the marker coordinate system to its projection,

$$v = [v_x, v_y]^T$$

expressed in pixel coordinates, as follows:

\[
\begin{bmatrix}
h \cdot v_x \\
h \cdot v_y \\
h
\end{bmatrix} = A [R_i \; T_i] \begin{bmatrix}
R \\
T \\
0 \; 1
\end{bmatrix} \begin{bmatrix}
V_x \\
V_y \\
V_z
\end{bmatrix}
\] (3)

in which $R$ is the 3x3 rotation matrix and $T$ is the translation vector that relate the marker coordinate system to the world coordinate system.

For a planar marker, the corresponding homography $H$ can be computed [22],

\[
\begin{bmatrix}
h \cdot v_x \\
h \cdot v_y \\
h
\end{bmatrix} = H \begin{bmatrix}
V_x \\
V_y \\
1
\end{bmatrix}
\] (4)

Introducing an auxiliary transformation, representing the relation between marker coordinates system and camera coordinates system, we can write

\[
\begin{bmatrix}
R \\
T
\end{bmatrix} = [R_i \; T_i]^{-1} \begin{bmatrix}
R^h \\
T^h
\end{bmatrix}
\] (5)

4.2. Multi-camera tracking

In this work we have fully developed a mathematical framework capable of managing the multi-view solution for the tracking problem of a planar marker. For each view we can rewrite (1) as

\[
p_{i,v} = M_{p,v} \begin{bmatrix}
x_{i,1} & \cdots & x_{i,3} & t_1 \\
\vdots & \ddots & \vdots & \vdots \\
x_{i,1} & \cdots & x_{i,3} & t_2 \\
0 & \cdots & 0 & 1
\end{bmatrix} P_{m_i}
\] (9)

where $i$ denotes the $i$-th point, and $v$ denotes the $v$-th view; $p_{i,v}$ is the projection, in homogeneous coordinates, of the $i$-th point, $P_{m_i}$ of the marker, on the $v$-th view.

Assuming $V_z=0$, (3) can be rewritten as:

\[
\begin{bmatrix}
h \cdot v_x \\
h \cdot v_y \\
h
\end{bmatrix} = A [R_i \; R^h] T^h \begin{bmatrix}
V_x \\
V_y \\
1
\end{bmatrix}
\] (6)

and by comparing (6) with (4) we find a simple way to compute the auxiliary rotation matrix, $R^h$, in respect to the homography $H$,

\[
[R_i \; R^h \; T^h] = A^{-1} H
\] (7)

At the end, we obtain

\[
R^h = \begin{bmatrix}
R_i^h & R_i^h & R_i^h \times R_i^h
\end{bmatrix}
\] (8)

where the normalization is needed to increase robustness of the overall computation.

Afterwards, to improve precision in translation computation [11], we compute $T^h$ by solving, in a least square sense (see multi-camera tracking), the over-constrained linear system which can be written considering the couple of equations resulting from the division of the first two lines in (6) by the last one, for each vertex of the marker. The pose of the marker can now be computed by replacing $T^h$ in (5).

Moreover, we also implemented an optimized version of this code that utilizes the pose just computed as the initial guess solution. We considered the Rodrigues parameterization of the rotation matrix [23], and so recomputed $T^h$, as described before, as a function of these parameters. Therefore, we minimized the resulting sum of the squared differences between measured and computed coordinates of all the projected vertices of the marker, applying a Levenberg-Marquardt standard algorithm.
we obtain the following generic linear equation in which only \( r_{k,h} \) and \( t_k \) are unknowns:

\[
\sum_{k=1}^{3} \sum_{h=1}^{3} [M_{p_{k,k}} - p_{v,v,j} \cdot M_{p_{k,k}}] (p_{m_{h,k}} \cdot r_{k,h} + t_k) = 0
\]

(12)

with \( i = 1..np \), \( v = 1..nv \), \( j = 1..2 \)

where \( np \) is the total number of points and \( nv \) is the total amount of views.

Finally, we obtain an over-constrained linear system, in the form

\[
A_{[N \times 9]} x_{[9 \times 1]} = b_{[N \times 1]}
\]

(13)

4.3. Stereo-Calibration Procedure

The main purpose of the calibration is to retrieve a consistent set of parameters to correctly model the projective geometry of the system, i.e. compute the projection matrix for each camera.

Let \( P_{\text{left}} \) and \( P_{\text{right}} \) the projection matrices of left and right camera, referring to (2) we can enforce, without loss of generality

\[
R_{\text{left}} = A_{\text{left}} [I \ 0] \quad P_{\text{right}} = A_{\text{right}} [R \ i]
\]

that represent one of the couples of projective matrices definable up to a projective transformation [25]. In this way, we have parameterized the system with a total amount of 16 parameter (5 intrinsic parameters for each camera, and 6 parameters to completely define their relative position).

The calibration problem can be reformulated as the optimization problem of finding the set of parameters minimizing the cost function

\[
Z = \sum_{p=1}^{N} (p_{i,j,p,c}^* - p_{i,j,p,c} \cdot s_{i,p,c})^2
\]

(15)

5. RESULTS

Both the previously described tracking techniques, stereo and mono, have been tested to highlight their benefits and drawbacks. Our tests aim, above all, to evaluate tracking accuracy and computational performances of each technique.

Firstly, we have numerically simulated the movement of the marker. In Fig. 3, marker trajectory is showed. The initial and final positions are represented by the black squares. We have simulated a spiral movement, filling almost the whole field of view of both cameras. In this way it is also possible to evaluate the influence of the distance on the overall accuracy.

![Fig. 3 Numerically simulated marker trajectory](image-url)
Assuming a focal length of 15 mm, and a 6 mm diagonal sensor dimension with a full resolution of 1024 x 768 pixel, the resulting intrinsic matrix, $K$, has been defined to simulate the projection from real coordinates into image coordinates.

\[
K = \begin{bmatrix}
3200 & 0 & 512 \\
0 & 3200 & 376 \\
0 & 0 & 1 
\end{bmatrix}
\]

Accordingly with the design dimensions of the system, the projection matrices have been defined as:

\[
MP_{left} = K \begin{bmatrix}
30 \\
0 \\
-30 \\
0 
\end{bmatrix}
\]

\[
MP_{right} = K \begin{bmatrix}
-30 \\
0 \\
30 \\
0 
\end{bmatrix}
\]

where $Ry(\pm0.06)$ identifies the sub-rotation matrix defined as a pure rotation of $\pm0.06$ radians (about $\pm3.4$ deg) against the y-axis. As a consequence, we obtain a global system of coordinates, placed in the middle of the cameras, representing the system of coordinate with respect to the pose of the marker is measured.

As a consequence, we obtain a global system of coordinates, placed in the middle of the cameras, representing the system of coordinate with respect to the pose of the marker is measured.

Once the numerical experiment has been performed, we noted that, theoretically, both our tracking techniques give back an exact solution, up to some numeric computation error.

Afterwards, we have introduced a random error in the vertices image coordinates of $\pm 5$ pixels, to simulate a realistic scenario, and analyze the robustness of the algorithms against noise.

In Fig. 4 is presented the translational error for the linear version of the algorithms. The error has been defined as the geometric distance between imposed and computed pose. The tracking accuracy decreases as the distance of the marker from the cameras increases, but the stereo tracking offers always a better performance (6 time better at 2m distance) than mono-tracking.

In Fig. 5 is depicted the translational error for the optimized version of the algorithms. The errors of both techniques are sensibly reduced, especially for mono-camera. However, comparing the linear and the optimized version, we can note that the linear stereo algorithm offers the same range of accuracy of the mono optimized one.

In Fig. 6 and Fig. 7 the rotational error is presented. It has been defined as the magnitude of the Rodrigues vector corresponding to the relative rotation between imposed and computed pose.

Differently than translational, the rotational accuracy of both techniques is comparable in both linear and optimized version of the
algorithms. In the optimized version we can put in evidence some sporadic loss of numeric stability, more evident for single camera technique.

As regards the computational time, stereo tracking is faster than mono in non-optimized algorithm and it is comparable in optimized algorithm. The following table shows the average computational times required for each version of the algorithms

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Linear</th>
<th>Optimized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mono</td>
<td>0.83</td>
<td>11.05</td>
</tr>
<tr>
<td>Stereo</td>
<td>0.58</td>
<td>12.2</td>
</tr>
</tbody>
</table>

Tab. 1 Average computational times

CONCLUSIONS

In this paper, an innovative marker-based tracking solution has been presented. The tracking algorithm is based on a stereo vision approach, but it can switch itself into a standard single camera tracking when the stereo pair correspondence is missing.

Some numerical test has been conducted to evaluate tracking accuracy and computational performances of the stereo technique compared with a standard single camera tracking solution. The stereo tracking offers always better performances, and it provides a more stable solution against noise in image segmentation. Further, it is faster to compute than the single-camera tracking algorithm.

Therefore, the presented tracking strategy could be a good alternative to the standard mono-camera tracking algorithms, especially for AR system employing two cameras.

In real-time applications, with no need of high precision, the linear version of stereo-tracking could give almost the same accuracy of a standard single camera algorithm, but it is about 19 time faster.

Moreover, for all the applications where an high tracking precision is desirable, it is possible to employ the optimized version of our tracking algorithm. The computation period of near 12 ms (~80 Hz) is comfortable compared to the refresh rate of commonly used camera (30 Hz)

However, an experimental test must be performed in order to fully confirm the good performance of our tracking approach.

REFERENCES