Hybrid tracking system for robust fiducials registration in augmented reality

Madjid Maidi, Fakhreddine Ababsa, Malik Mallem, Marius Preda

Hybrid tracking system for robust fiducials registration in augmented reality

Madjid Maidi · Fakhreddine Ababsa · Malik Mallem · Marius Preda

Abstract An effective augmented reality system requires an accurate registration of virtual graphics on real images. In this work, we developed a multi-modal tracking architecture for object identification and occlusion handling. Our approach combines several sensors and techniques to overcome the environment changes. This architecture is composed of a first coded targets registration module based on a hybrid algorithm of pose estimation. To manage partial target occlusions, a second module based on a robust method for feature points tracking is developed. The latest component of the system is the hybrid tracking module. This multi-sensors part handles total target occlusions issue. Experiments with the multi-modal system proved the effectiveness of the proposed tracking approach and occlusion handling in augmented reality applications.

Keywords Augmented reality · Computer vision · Real-time tracking · Hybrid tracking · Multi-sensors systems

1 Introduction

1.1 Background

This work aims to develop a multi-modal tracking architecture to handle target occlusions in augmented reality (AR) applications. Most existing systems enable fully visible target tracking and fail in presence of occlusions. Many studies were conducted in this domain to detect and identify fiducial targets using visual tracking methods. However, these methods are limited in robustness and computing capabilities due to the algorithm complexity, accuracy and environment changes. These reasons motivated us to propose and develop a novel tracking approach to tackle these problematics.

The proposed solution has to improve the vision-based tracking systems by building a multi-modal architecture of tracking and occlusion management. This architecture includes a registration module based on a coded target approach and a hybrid camera pose computation. The second module relies upon a robust estimation method to track feature point when targets are partially occluded. The third module is a multi-sensors system intended to overcome total target occlusion using the kinematic data provided the inertial measurement unit.

This work contributes to handle some limitations related to AR systems working in real environments. We have developed a new detection and identification method to track coded targets, and a novel pose estimator algorithm is implemented to determine with accuracy the camera localization. In addition, our system enables partial and total occlusion handling. The robust tracker and the vision-inertial tracker allow target tracking in case of partial or total occlusions of the fiducial markers under various environment conditions.

1.2 Literature review

Accuracy and robustness are prerequisites for developing an effective AR system. In literature, several AR systems were developed to track the user's viewpoint. Technologies based on active targets are used in various registration applications in indoor environments. However, these applications require power sources which limits their use in specific areas

M. Maidi (✉) · M. Preda
Département ARTEMIS, Télécom SudParis/Institut Mines-Télécom, 9 Rue Charles Fourier, 91000 Évry, France
e-mail: maidi.madjid@gmail.com

F. Ababsa · M. Mallem
Laboratoire IBISC, Université d'Évry Val d'Essonne, 40 Rue du Pelvoux, 91000 Évry, France

away from interferences and disturbances. Other systems use passive targets represented by vision systems and require significant processing resources. Several marker-based identification methods were developed in literature. We will present an overview of the most known techniques described in existing works.

ARToolKit [12] is a marker-based tracking system used in AR applications. Thanks to its robustness performance, it is used in many AR and computer vision systems. ARToolKit includes several models of 2D fiducial markers. However, the performance of the marker detection should be improved to cope with image correlation uncertainties. ARToolKit tracking consists in comparing markers with recorded fiducials within a database.

CyberCode was proposed by Rekimoto and Ayatsuka [25]. This system uses visual coded targets, and several operations are required to detect and extract the targets from the image and estimate the camera pose. The CyberCode algorithm consists, mainly, in finding the guide bar of patterns to retrieve corners. Then, the code is computed, and this code, called the CyberCode, identifies the target and allows to track it in image sequences. Afterward, the pose is determined using constraints relating the target corners in the image and their coordinates in the real-world. The CyberCode is composed of 33 bits, which makes approximately 8 billion possibilities to defining distinct codes.

In 2002, the Intersense company [21] developed its own system of coded targets. This system is based on circular targets. Although it is not the first system based on this kind of fiducials (Cho and Neumann [5] developed a similar system in 1998), it encountered a great success since the processing module operates in real-time and it is implemented on an embedded system including a camera and an inertial measurement unit. The codes are defined with 15 bits, which makes at all 32,768 possibilities. The circular fiducials represent only a single feature point, and so, it is necessary to have several landmarks to compute the camera pose. Intersense system uses at least 4 targets to estimate the pose.

Fiala [8] proposed a system based on ARToolKit called ARTag. ARTag is a marker system that uses digital coding to get a very low false positive and intermarker confusion rate with a small marker size. The system employs an edge linking method to handle lighting variations. The author created a series of 2002 single markers coded on 36 bits. ARToolKit carries out a correlation calculation of the gray level image following 4 positions of the target, where ARTag uses coded targets to obtain a very low error rate for identification. Moreover, this method allows fiducial identification in presence of occlusions.

The multi-sensors systems are usually used to improve vision-based systems. These hybrid systems exploit the complementarity of sensors to compensate errors of each device. However, many difficulties can be encountered in experimentation, in particular, the multi-sensor calibration, the data fusion and the error management.

The estimation of camera pose is an important step to determine the user's viewpoint in AR applications. In literature, various visual tracking methods were developed. Stricker et al. [26] have presented an interactive AR application to solve occlusion problem. Occlusions are managed by locating the user hand and subtracting the background. This approach is feasible in case of homogeneous background with the assumption of static camera. Naimark and Foxlin [21] implemented a hybrid vision-inertial self-tracker system, which operates in various real-world lighting conditions. The aim is to extract coded fiducials in presence of very nonuniform lighting. Comport et al. [6] integrated a M-estimator into a visual control law via an iteratively re-weighted least squares implementation. The implementation showed that the method is robust to occlusion, changes in illumination and miss-tracking. Naimark and Foxlin [22] presented a technique based on active targets using amplitude modulation codes instead of binary codes. Such a system provides high precision with compact targets and operates in a wide range of viewing angles under various luminosity conditions. Maidi et al. [17,18] presented a robust fiducials tracking method for AR systems. A generic algorithm for object detection and feature points extraction is developed to identify targets in real-time. The authors proposed a tracking method based on RANSAC algorithm to deal with target occlusion. Gabriele and Didier [3] presented a new visual-inertial tracking device for augmented and virtual reality applications. The authors provide an evaluation of different models and developed a markerless tracking approach. The solution relies on a 3D model of the scene to predict the appearances of the features by rendering the model using the prediction data of the sensor fusion filter. High stability and accuracy were demonstrated using the developed system.

Recently, the telecommunication market is revolutionized by mobile phones and mobile applications. With high-performance processing units and powerful graphic processors, more evolved applications could be deployed on these devices. Thousands of applications are available on Web portals and can be downloaded from application stores.

In literature, many works are interested in mobile AR applications. Today, applications on mobile phones or PDA are marketed for general public [20]. Numerous studies have been conducted in this domain, Takacs et al. [27] built a mobile AR system which makes correspondence of referenced images to a database located on server using detection algorithms based on local descriptors. By directing the camera toward the object of interest, the tracking system provides information and services on its location. The SURF method [2] was applied to images captured from the phone camera. The system is intended for tourists to serve them as a guide during their visits. Chen et al. [4] realized a recognition

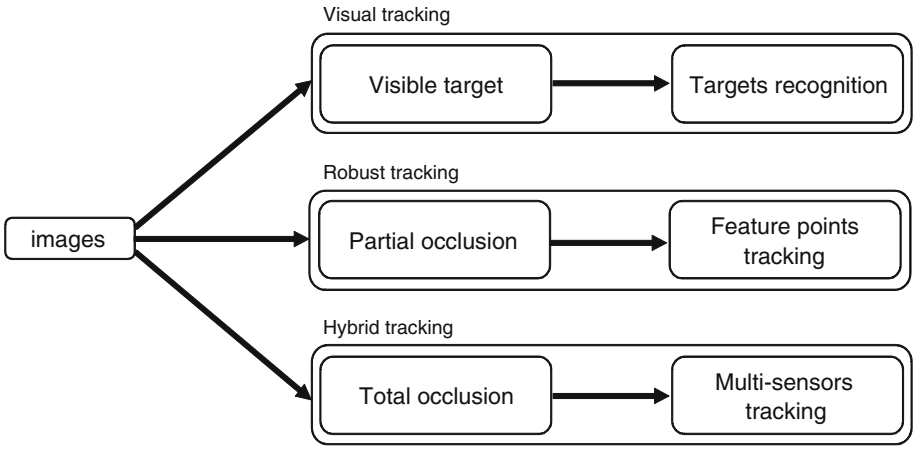


Fig. 1 Synoptic diagram for tracking and occlusion handling

system for AR real-time tracking on mobile phone. The recognition process allows books and CDs cover tracking. The system operates in real-time, and the matching part is performed on a base of 20,000 images stored in a server. Klein and Murray [13] described an implementation of SLAM method on a 3G iPhone. The authors demonstrated that the SLAM approach can operate on mobile phones; however, the accuracy and the execution time of the system are limited compared to PC. Wagner et al. [28] presented three techniques for object tracking in real-time mobile application. The authors combined several existing approaches in literature to implement an AR tracking system. SIFT [14] and Ferns [23] methods were employed with a tracker-based model matching. The authors combined both approaches to overcome drawbacks of each technique.

Smartphones have known large progress in application development; however, there is a gap between this technological progress and the hardware performance of these devices (memory, processor, power supply, etc.). Consequently, the mobile phone is not yet adapted to real-time applications, which demand high material resources. The technical development in nanotechnology would allow in a near future to improving performances of phones CPUs while increasing their capacity of storage and addressing [20].

The literature review enabled to have an overview of the main issues and solutions related to our research area. In this study, we are particularly interested to works connected to multi-sensors tracking. In this field, the work of You and Neumann [30] and Foxlin and Naimark [10] is closest to our problematic. These authors directed their research toward hybrid systems, data fusion and robust tracking architectures for real-time AR registration. However, some residual issues unsolved by the existing solutions led us to propose an original multi-modal tracking and occlusion handling approach for AR applications.

The remainder of the paper is organized as follows: in Sect. 2, we detail the pose estimation procedure. Section 3 describes the vision-based tracking module. The hybrid tracking modality is detailed in Sect. 4. In Sect. 5, we present the experimental results. Section 6 presents a discussion and we conclude with Sect. 7.

2 Camera pose estimation

2.1 Problem formulation

To overlay virtual objects on the real scene, the camera pose must be computed. The pose estimation is based on the extraction of geometric primitives that enables the matching of 2D image points with the 3D object points. The 3 modules of the proposed solution presented in Fig. 1 aim to estimate the camera pose in order to project synthetic graphics on real images.

The pose estimation is formulated as a function minimization which relates 3D points to the 2D points by the following equation (Fig. 2):

$$F(p, P, I_x, R, T) = 0 \tag{1}$$

where $P_i = (X, Y, Z)$ is a 3D point in object reference frame and $p_i = (u, v)^T$ its projection in the image. Using the perspective model of the camera, we have:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = I_x \begin{pmatrix} R & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2}$$

where s represents the scale factor, (R, T) are the rotation and the translation of the object reference frame according to the camera reference frame and I_x is the intrinsic matrix of the camera [19].

To estimate the camera pose, first, it is necessary to determine the 2D–3D matching points. Then, we have to find the perspective transformation which defines the 2D–3D correspondences. The pose estimation requires a calibration procedure to retrieve the camera intrinsic parameters.

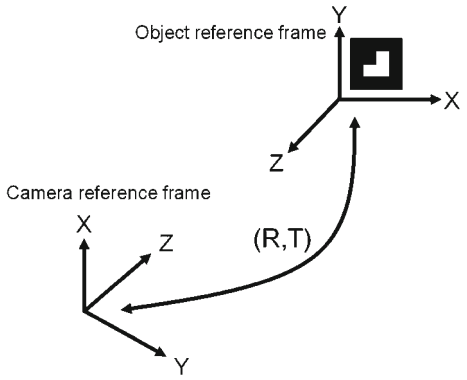


Fig. 2 Reference frames used for pose estimation

2.2 Proposed solution

The solution we propose consists in developing a hybrid pose estimation method which combines the extended Kalman filter (EKF) and an analytical algorithm. Indeed, the EKF algorithm converges to an optimum for any set of observed points; however, in order to ensure this convergence into the correct pose in a minimum time, a good pose parameters initialization is required. The EKF algorithm can be initialized using initial rotation guess R_0 and translation T_0. Therefore, an analytical pose estimator is used to compute the correct initial parameters to allow the convergence of the EKF toward an optimal solution. Our contribution in this part is to develop a new pose estimator based on EKF and initialized by an other pose estimator in order to improve the accuracy of the camera localization.

2.2.1 Parameters initialization

To compute the first guess of our pose parameters (R_0, T_0), we use the algorithm of Zhang [31]. This algorithm is adapted to planar square targets and requires the knowledge of:

1. Intrinsic parameters of the camera.
2. Coordinates of the 4 corners of the fiducial in the image and their 3D matchings.

This technique requires 2D/3D matching points, and the relationship between a 3D point P and its image projection p is given by Eq. 2.

Let's denote the i^{th} column of the rotation matrix R by r_i. From Eq. 2 and considering planar objects ($Z = 0$), one has [31]:

$$s\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = I_x \begin{pmatrix} r_1 & r_2 & r_3 & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = I_x \begin{pmatrix} r_1 & r_2 & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (3)$$

Given an image of the model plane, a homography can be estimated. One denotes it by $H = (h_1\ h_2\ h_3)$ which is identified to $H = I_x(r_1\ r_2\ T)$. Once I_x is known, the pose parameters for each image are readily computed as follows [31]:

$$\begin{aligned} r_1 &= \lambda I_x^{-1} h_1 \\ r_2 &= \lambda I_x^{-1} h_2 \\ r_3 &= r_1 \times r_3 \\ T &= \lambda I_x^{-1} h_3 \end{aligned} \quad (4)$$

where $\lambda = 1/\left\|A^{-1}h_1\right\| = 1/\left\|A^{-1}h_2\right\|$.

2.2.2 Fitting parameters

To fit the pose parameters estimated by the previous analytical method, we use a second iterative method based on the EKF [29].

First, we need to define the state vector of the EKF. Since our goal is to estimate the camera pose, we use the rotation angles (ψ, θ, φ) and the translation components (t_X, t_Y, t_Z) to represent the system state. The measurement input is provided by the camera. We have to estimate 6 variables of the state vector, and so we use the following 8×1 measurement vector:

$$z = \left(u_1\ u_2\ u_3\ u_4\ v_1\ v_2\ v_3\ v_4\right)^t \quad (5)$$

where (u_i, v_i) are feature points in the image.

Time update The time update produces estimates of the state vector $\hat{x}$ and the error covariance matrix P. The equations of projection are given by the following a priori state estimate and covariance error:

$$\hat{x}_{k+1}^- = \hat{x}_k \quad (6)$$

$$P_{k+1}^- = A_k P_k A_k^t + Q_k \quad (7)$$

where Q represents the covariance matrix of the process noise and A is the transition matrix represented by:

$$A = I_6 \quad (8)$$

Measurement update The measurement update model relates the state vector to the measurement vector. The measurement vector is represented by the 2D features points. Based on the knowledge of the feature points position in the camera reference frame, we use the perspective projection model as follows:

$$u_i = f\,(M, P_i) \quad (9)$$

$$v_i = g\,(M, P_i) \quad (10)$$

The measurement function is given by:

$$z_{k+1} = h\left(\hat{x}_k\right) + v_k \quad (11)$$

where ν_k represents the measurement noise and h is given by:

$$h\left(\hat{x}_k\right) = M \times P_i \times x_k \tag{12}$$

and x_k is the state vector defined before.

To perform the measurement update, first we compute the Kalman gain:

$$K_k = P_k^- H_k^T \left(H_k P_k^{-1} H_k^T + V_k R_k V_k^T\right) \tag{13}$$

where:

$$\begin{cases} H_{ij} = \frac{\partial h_i}{\partial x_j}\left(\hat{x}_k, 0\right) \\ V_{ij} = \frac{\partial h_i}{\partial v_j}\left(\hat{x}_k, 0\right) \end{cases} \tag{14}$$

The estimation is updated with measurement z_k:

$$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - h\left(\hat{x}_k^-, 0\right)\right) \tag{15}$$

In practice, generally, one does not know the values of the noise at each time step. Therefore, in Eqs. 14 and 15, one can approximate the measurement vector without noise ($\nu_k = 0$) [29].

Finally, we update the error covariance:

$$P_k = \left(I - K_k H_k\right) P_k^- \tag{16}$$

where:

$$H = \begin{pmatrix} \frac{\partial h_1}{\partial \psi} & \frac{\partial h_1}{\partial \theta} & \frac{\partial h_1}{\partial \varphi} & \frac{\partial h_1}{\partial t_x} & \frac{\partial h_1}{\partial t_y} & \frac{\partial h_1}{\partial t_z} \\ \frac{\partial h_2}{\partial \psi} & \frac{\partial h_2}{\partial \theta} & \frac{\partial h_2}{\partial \varphi} & \frac{\partial h_2}{\partial t_x} & \frac{\partial h_2}{\partial t_y} & \frac{\partial h_2}{\partial t_z} \end{pmatrix} \tag{17}$$

The state vector and the error covariance matrix are updated using the measurement input from the camera. Once this step is performed, this data will be the input of the time update step. These two steps are carried out recursively to estimate the rotation angles and the translation vector of the camera coordinate frame according to the workspace coordinate frame.

2.2.3 Hybrid extended Kalman filter algorithm

This method is the combination of the two algorithms presented before: the EKF and the analytical algorithm. Indeed, we already mentioned that the EKF problem is the parameter first guess, so we use the analytical algorithm to initialize the pose values to accurately estimate the EKF states (Fig. 3).

3 Vision-based tracking modality

We begin by presenting the first module of our architecture, which enables visible target tracking.

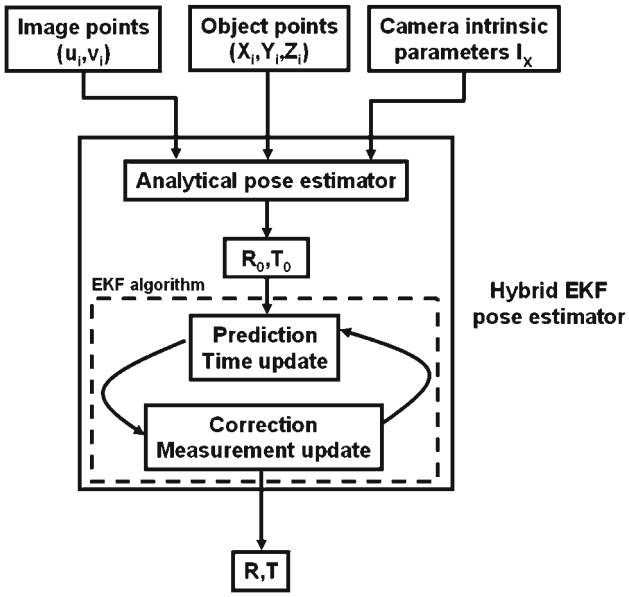


Fig. 3 Hybrid Kalman filter pose estimator diagram

3.1 Fiducial recognition approach

Our system detects and identifies objects according to a specific code associated with the marker. To extract the object of interest from the scene, images are preprocessed to reduce the detection error rate. Many operations are applied to process the image and detect the object shape. The proposed system has the advantages of being fast and flexible compared to ARToolKit or the Intersense system. Indeed, our method extracts in real-time, the object of interest from the image by computing the binary code located inside the target. The used code is composed of 16 bits which allows to reduce the computing time compared to CyberCode which uses 33 bits. Finally, in our system, only one marker is sufficient to estimate the camera pose contrary to the system of Intersense which requires several visual landmarks to calculate the same pose (4 targets to determine the pose).

Our object detection and identification is composed of the following steps (Fig. 4):

- Detect contours in image.
- Smooth the image to reduce the noise and eliminate pixel variations in the contour segments.
- Dilate the smoothed image to remove potential holes between edge segments.
- Approximate contour with accuracy proportional to the contour perimeter.
- Find the number of object vertices.
- Identify object boundaries as 4 intersecting lines by testing collinearity of vertices.
- Find the minimum angle between joint edges.

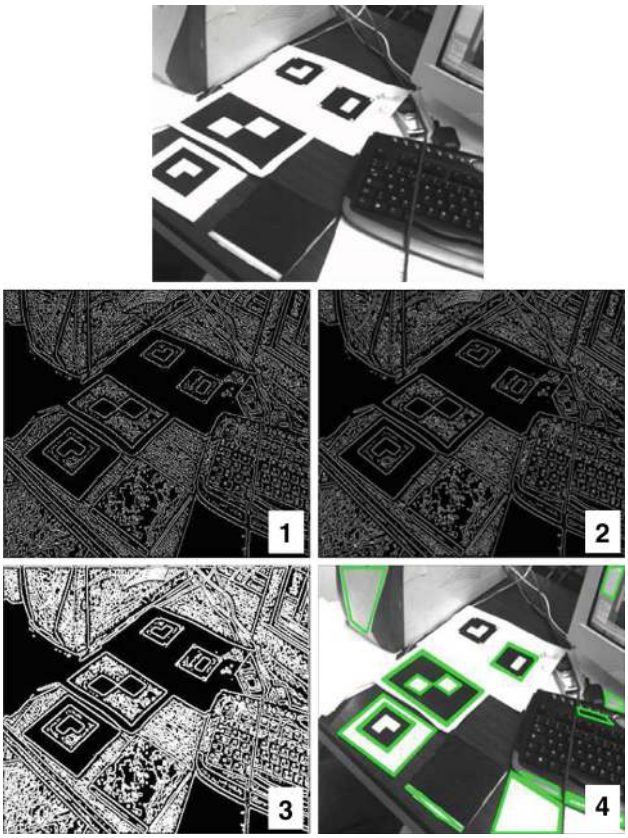


Fig. 4 Fiducial detection process. *1* Contours detection, *2* Image smoothing, *3* Image dilatation, *4* Polygonal approximation

Finally, only objects with 4 vertices and right angles are retrieved and considered as square shapes. Once a square object is detected, the next step is to identify this object and match it with a defined template. Our goal is to design fiducials which can be robustly extracted in real-time from the scene. Therefore, we use two kinds of square fiducials with different patterns inside (Fig. 5).

The internal code of the fiducial is computed by spatial sampling of the 3D fiducial model. Then, we project the sample points on the 2D image using the homography computed from the 4 vertices of the detected fiducial. We compute the fiducial code from the sampling grid, and this code is composed of 16 bits and represents the fiducial sample color (Fig. 6). However, only 4 bits are useful to compute the effective target code. Finally, the fiducial code can have 4 values following the 4 possible orientations, this reduces by 4 the number of target class.

The target system must respect a strong constraint, which is to allow the detection of target orientation. Each target rotated by 90° has a distinct code in the identification phase. Thus, targets have 4 codes following their orientations, and consequently, the number of target classes is divided by 4 which reduces the number of possible codes (Fig. 7). Moreover, targets should not have a central symmetry because we could not distinguish the target orientation.

Fig. 5 Models of fiducials

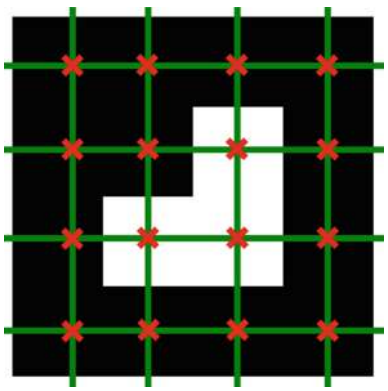


Fig. 6 Fiducial sampling

3.2 Robust points tracking approach

The second module of our architecture allows feature point tracking. The aim of this part is to manage partial occlusions. We use the RANSAC algorithm [9] to track 2D points in image sequences. For many applications, simple geometric or photometric templates can be sufficient. Projective geometry is a mathematical tool well suited to model the environment and the camera acquisition process. However, when we are confronted to real images (robotic applications), the modeling becomes inaccurate and the use of robust algorithms is required. In addition, outliers can appear in case of lighting changes or in presence of occlusions. To overcome all these problems, we propose a robust method based on feature points tracking using RANSAC algorithm.

3.2.1 Feature point detection and matching

For feature points detection, we used the Harris detector [11]. To match the detected feature points in two successive images, a correlation method is employed. This method finds similarity areas in two successive frames using a correlation measure on a window around the point to match (Fig. 8b).

The 3D–2D matching is realized from the camera pose estimation. The 3D points of the object model are matched

Fig. 7 Codes corresponding to different target orientations

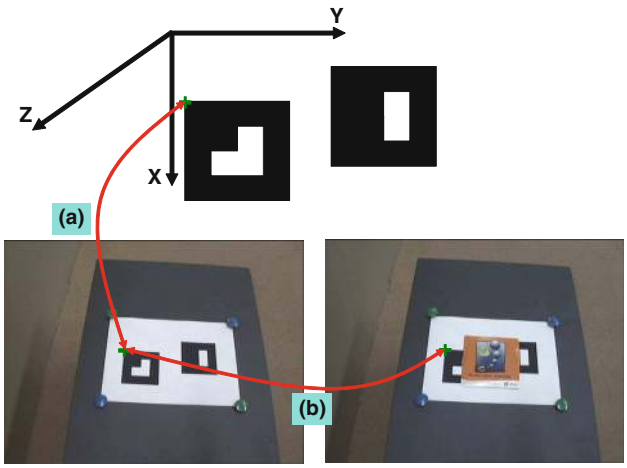


Fig. 8 Points matching. *a* 3D–2D matching, *b* 2D–2D matching

to their 2D projections, using the camera perspective matrix (Fig. 8a).

3.2.2 Robust matching

The robust matching method based on RANSAC algorithm uses the 2D–2D matching points in two successive images. The setting up of strong matching of these points of interest is made by an estimate of the RANSAC homography. This homography represents a geometric constraint between two images. The determination of this matrix allows to find the transformation that connects primitive from one image to another. Points likely to be a good match are validated, while the false candidates are rejected and eliminated by RANSAC.

3.2.3 Our method

The object of interest is represented by our target models illustrated in Fig. 5. The feature points tracking is performed by estimating the homography matrix which binds homologous 2D points in two successive images. These 2D points are related by the following homography:

$$\begin{pmatrix} \lambda u_2 \\ \lambda v_2 \\ \lambda \end{pmatrix} = \underbrace{\begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix}}_{G} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \tag{18}$$

where (u_1, v_1) and (u_2, v_2) are two homologous points in two successive images and λ is an arbitrary scale factor.

The principle of the occlusion handling is explained in the diagram of Fig. 9. We use two different types of fiducial models with two distinct codes. Initially, both fiducial models must be visible to the camera to identify and extract their 4 feature points. If these points are visible, then, they are tracked with the identification algorithm presented in Sect. 3.1. If one of the 4 target points is occluded, the target is not identified and the robust tracking algorithm is launched. This algorithm based on RANSAC algorithm allows the tracking of visible feature points by computing a rigid transformation between two successive images acquired by the camera. Finally, if both fiducials are occluded, the robust algorithm fails and cannot track feature points anymore.

The robust method enables tracking when targets are partially occluded. However, the main problem of this method is the number of visible points in case of partial occlusion. Indeed, the constraint is to have a number of points ≥ 4 to be able to estimate the camera pose. Nevertheless, in a real environment, the number of visible points can change constantly and the constraint we mentioned before is not always respected. To solve this problem, we use another motion sensor which is the inertial measurement unit (IMU) in order to locate the vision system when the visual markers are less than 4 points.

4 Hybrid tracking modality

4.1 System description and calibration

Our hybrid tracking system is composed of a camera and an IMU. Each sensor substitutes the other to compute the camera pose. The orientation is determined separately by the two sensors. For the translation, the camera is used to correct the IMU drift. Our approach of data substitution switches the system to the adapted tracking module according to target visibility.

The multi-sensors system contains an IMU: MTi from Xsens which provides the following measurements: accelerations, angular velocities, magnetic fields and angular rotations, and a Sony camera model XC-555P with a 6 mm focal

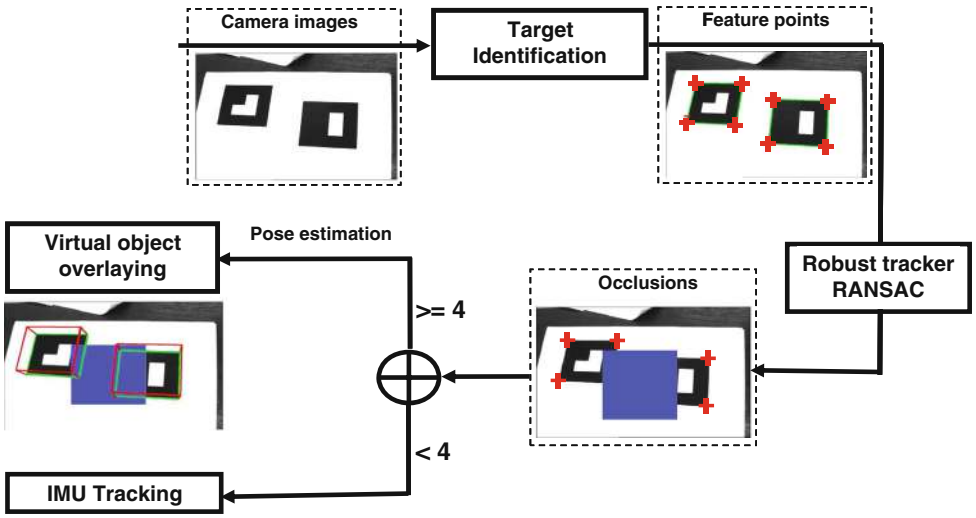


Fig. 9 Synoptic Diagram for occlusion handling

length, allowing real-time video capture. The data of the two sensors are provided at a sampling time and expressed in their respective reference frame.

We describe, now, the rotation calibration between the IMU and the camera. This procedure determines the rotation between the IMU reference frame R^I and the camera reference frame R^C [16]. The rotation between these reference frames is given by:

$$R_{CI} = R_{CW} R_{WI} \tag{19}$$

with:

- R_{CI}: rotation of the IMU reference frame R^I according to the camera reference frame R^C.
- R_{CW}: rotation of the world reference frame R^W according to the camera reference frame R^C.
- R_{WI}: rotation of the IMU reference frame R^I according to the world reference frame R^W.

The rotation calibration between the camera reference frame and the IMU reference frame is carried out in two steps: first, the calibration of the IMU global reference frame to compute R_{WI} rotations and then, a camera pose estimation to determine R_{CW}.

The IMU computes the rotations of its local reference frame R^I according to the global reference frame R^G (defined according to the north magnetic). We define a new global reference frame using the reset functionality of the IMU software to define a new global reference frame superimposed to the world frame.

The rotation of the camera reference frame is derived from the pose estimator algorithm. Indeed, the pose estimation determines the rotation R_{CW}. Finally, the determination of the rotation between the IMU and the camera R_{CI} is computed by Eq. 19 using the two rotations: R_{CW} and R_{WI}.

4.2 Visual-inertial data substitution

We present now the hybrid algorithm used for occlusion handling and the IMU and the camera data substitution. For the rotation data, the camera estimates the orientation when targets are visible and the IMU determines the orientation if targets are occluded. The data substitution is carried out for translation, and the two sensors compute the translation at the same time because the IMU requires measurements of the camera to correct its drift. The hybrid system is composed of 3 modules (Fig. 10):

- Module of camera pose estimation.
- Module of orientation estimation from the IMU.
- Module of position estimation from the IMU accelerations.

The temporal diagram of data substitution from the camera and the IMU is represented in Fig. 11. If the target is visible and identified, the vision-based module enables tracking and estimates the camera pose. However, when the target is not detected due to occlusion or motion blur, the pose in this case is estimated by the IMU. This diagram shows also the collaboration between the two sensors. The data substitution between the IMU and the camera allows to correct the drift of IMU positions and maintains camera tracking in presence of occlusions.

4.3 Kalman filter implementation

To use the Kalman filter, the motion must be modelled to perform the prediction and the correction of Kalman filter states and covariances. In our study, the Kalman filter estimates the camera translation from the IMU accelerations. The state vector is defined as following:

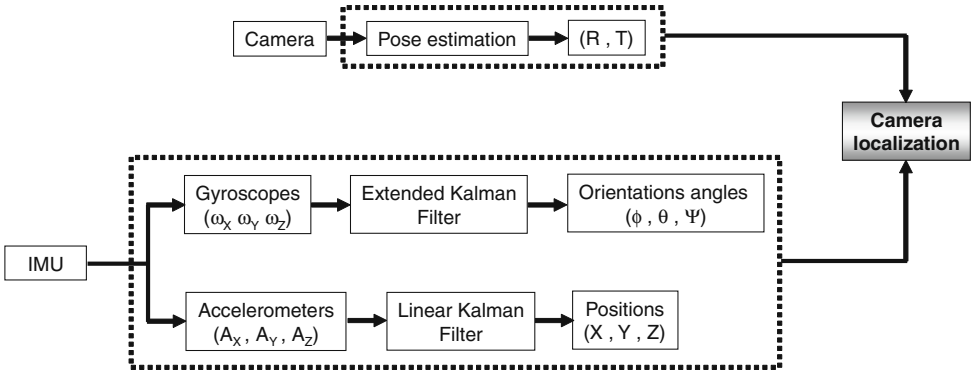


Fig. 10 Data substitution in the hybrid tracker

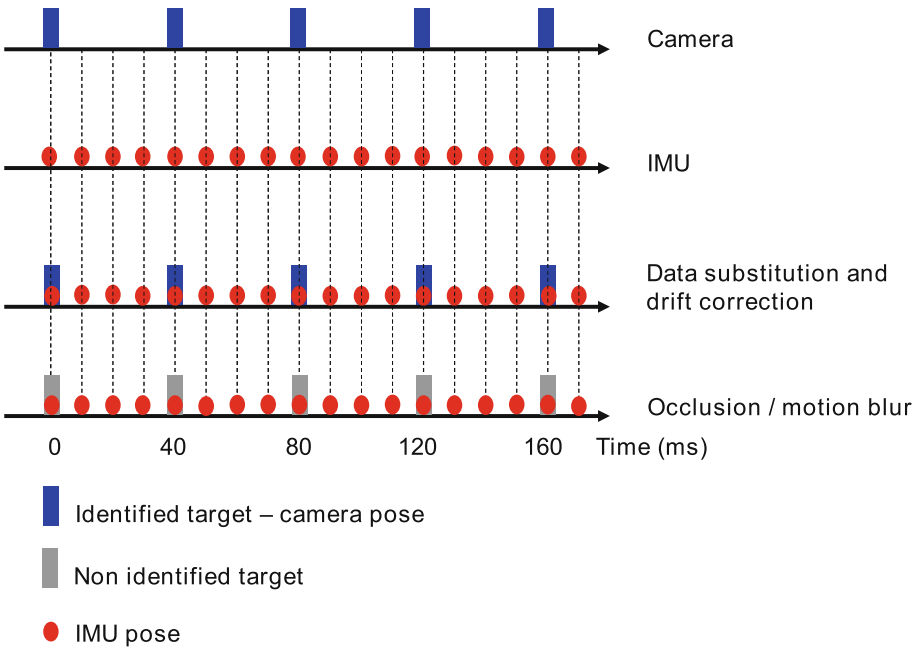


Fig. 11 Temporal diagram of data substitution

$$\hat{x} = \left(X \;\; Y \;\; Z \;\; V_X \;\; V_Y \;\; V_Z \;\; A_X \;\; A_Y \;\; A_Z \right)^T \tag{20}$$

The state model of the system is given by:

$$\begin{pmatrix} X_k \\ V_k \\ A_k \end{pmatrix} = \begin{pmatrix} X_{k-1} + \Delta T V_{k-1} + \frac{(\Delta T)^2}{2} A_{k-1} \\ V_{k-1} + \Delta T A_{k-1} \\ A_{k-1} \end{pmatrix} + \begin{pmatrix} W_{k-1}^1 \\ W_{k-1}^2 \\ W_{k-1}^3 \end{pmatrix} \tag{21}$$

where W_{k-1}^i represents the measurement noise and X_k, V_k, A_k are 3×1 vectors representing respectively, position, velocity and acceleration.

Now, we use the Kalman filter to model the camera. Measurements are provided by the IMU, and the filter uses acceleration measurements and computes a double integration in order to estimate positions. When the sensor data are available, the filter predicts and corrects the states. The prediction step uses the state estimated at the previous moment to produce an estimate in a current state. In the correction or update step, observations at the current time are used to correct the predicted state in order to improve the estimate accuracy.

4.3.1 Prediction

The state prediction and the error covariance are defined by the following projection equations:

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned} \tag{22}$$

The translation model is represented by:

$$\begin{pmatrix} X_k \\ V_k \\ A_k \end{pmatrix} = \begin{pmatrix} 1 & \Delta T & \frac{(\Delta T)^2}{2} \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_{k-1} \\ V_{k-1} \\ A_{k-1} \end{pmatrix} \tag{23}$$

where ΔT is the sampling time.

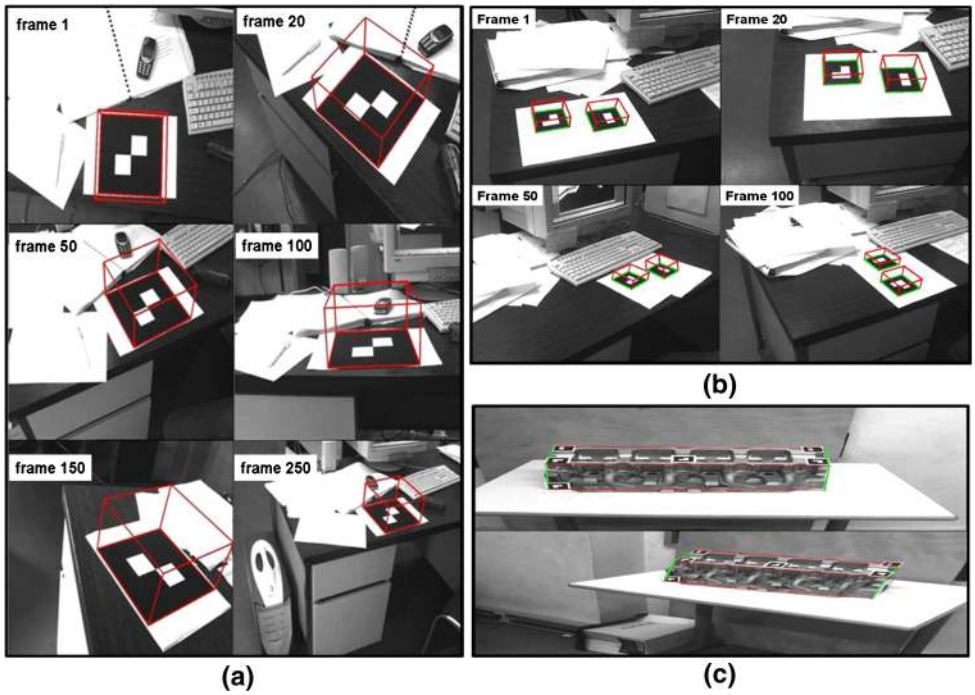


Fig. 12 Virtual object overlay in a tracking sequence

4.3.2 Correction

In this step, the filter updates the system states with accelerations data of the IMU, first of all, the Kalman gain is computed by:

$$K_k = P_k^- H^T \left(H P_k^- H^T + R \right)^{-1} \tag{24}$$

where H is the measurement matrix and R is the measurement noise covariance.

The states are, thereafter, updated with accelerations measurement by the following equation:

$$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H \hat{x}_k^- \right) \tag{25}$$

with z_k, the measurement vector represented by the camera accelerations A_C.

Finally, the covariance error is updated by:

$$P_k = (I - K_k H)\, P_k^- \tag{26}$$

where

$$H = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}^T \tag{27}$$

Our Kalman filter computes the IMU position according to a reference point. However, we are interested in estimating the camera position according to the world reference frame. Indeed, the IMU device is connected to the camera. When the hybrid sensor makes a translation motion, the acceleration of both sensors is the same and given by:

$$A_C = R_{CI} A_I \tag{28}$$

5 Experiments

5.1 Visual tracking

For the visual tracking module, we used a single camera with 6 mm focal length. First, we test the algorithm of fiducial identification and pose estimation. We printed on a standard laser printer 80×80 mm black and white fiducials. The identification algorithm detects squares in image then computes the target codes, and if there is a matching with the template code, the target is identified and can be tracked in the image.

Since the pose parameters were determined, we have projected a virtual cube on the detected real target in the image to evaluate the visual rendering stability. In this experiments, the camera is moved freely around fiducials. The identification algorithm detects and tracks targets in frames, and the hybrid EKF estimates position and orientation of the camera. Figure 12a, b show virtual objects overlaying upon real targets, different type of targets are defined to test the effectiveness of the marker identification algorithm. In Fig. 12c, the planar marker is used to compute the camera pose parameters which are used afterward to project a virtual wire model of a cylinder head.

The second part of the visual tracking experiments is the evaluation of different localization methods. A comparison between these methods is performed in order to determine the performances and the weakness of each one. We compared our hybrid EKF method to the 3 other algorithms which are the analytical algorithm of Zhang [31], the hybrid Orthogonal Iteration (OI) [7,15] and the EKF [19]. The comparison

between these algorithms is carried out according to the following criteria:

- Execution time.
- Reconstruction error.
- Generalization error.

The experimental tests were realized using the following hardware configuration:

- Pentium III 1.1 GHz.
- Matrox Meteor II framegrabber.
- Sony XC-555P camera.

5.1.1 Execution time

The first analysis, which is the execution time of algorithms, shows that the analytical algorithm is the fastest method with 19.40 µs for one pose estimation, the hybrid EKF makes 112.27 µs to estimate the same pose, then we have 153.25 µs for the hybrid OI and finally, 13,530.30 µs are necessary for the EKF to determine the pose parameters. So, in terms of computation time, the analytical algorithm presents the best performance than the other methods unlike the EKF which is very slow and seems to be unadapted to real-time applications.

5.1.2 Reconstruction error

In this experimentation, the camera is moved around the target, and the 4 algorithms estimate the pose parameters and we evaluate the reconstruction error within the image. The 4 algorithms computed 1,400 poses, and the error is estimated by re-projecting the object model on the image. For each pose computation, we re-project the target model on the image and we measure the difference between the real target corners and the projected corners. We notice that the hybrid EKF is the most stable and accurate method compared to the other algorithms. From Fig. 13, when the distance between fiducials belongs to [0.10, 0.45] m, the analytical method and the hybrid EKF present the lowest reconstruction error, the two algorithms are accurate and stable in this interval. However, when this distance becomes greater than 0.45 m, the hybrid OI is more accurate than the other methods. The reconstruction error is important in the EKF, because the algorithm did not converge to the optimal solution due to bad parameter initialization.

5.1.3 Generalization error

To determine the generalization error, we printed 4 square targets with 5 cm side on a paper. One of the targets is used to compute pose parameters, and the 3 others are used for

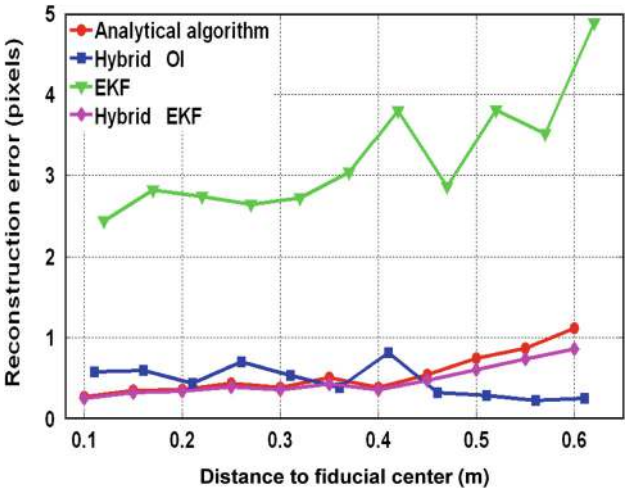


Fig. 13 Reconstruction error according to fiducial center

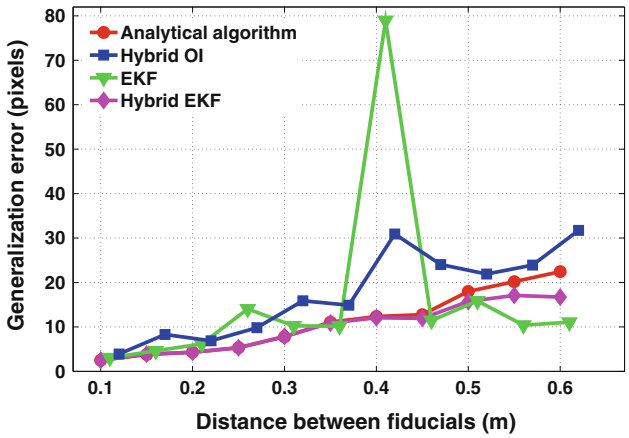


Fig. 14 Generalization error according to distance between fiducials

generalization error. This generalization error is computed by re-projecting the models of objects which were not used in the pose estimation procedure and we project them on the image. The obtained results on generalization error are represented in Fig. 14. The hybrid EKF presents the best performance in terms of generalization error comparing to other algorithms.

5.1.4 Discussion

Comparative studies of pose estimation were presented in the literature [1,7,24]. An analysis of re-projection error enabled to find comparative elements between our method and the pose estimation method proposed by Ansar and Daniilidis [1]. The authors estimated the re-projection error to <0.5 pixels for a distance representing 12 times the size of target side. In our study, the lowest re-projection error is <0.5 pixels for a distance of <0.6 m, which represents 12 times

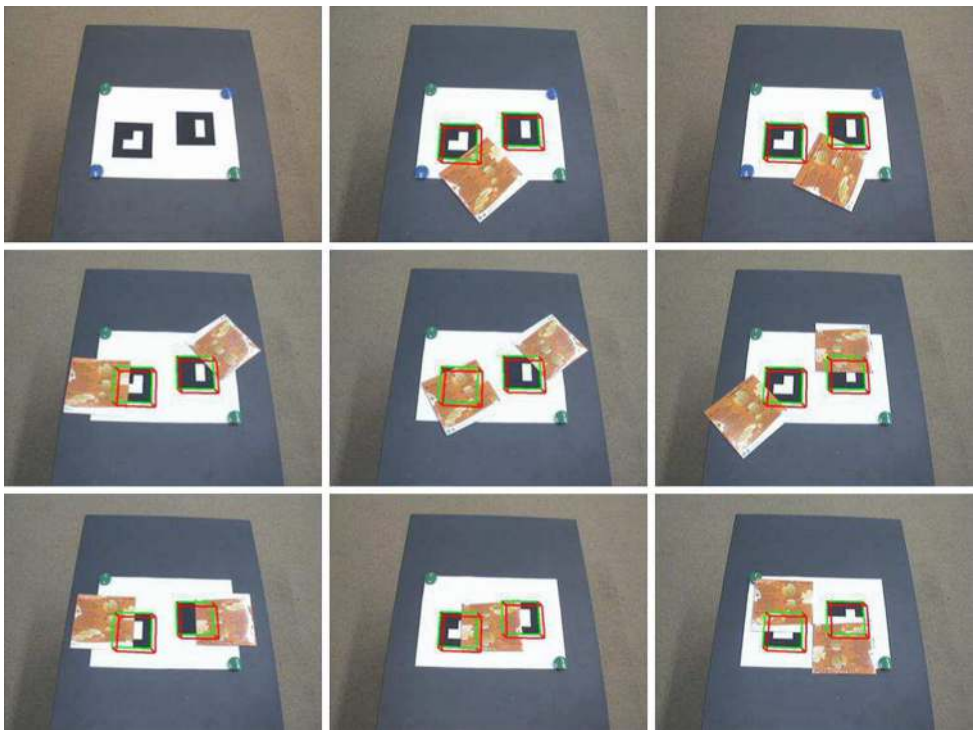


Fig. 15 Partial occlusion of targets

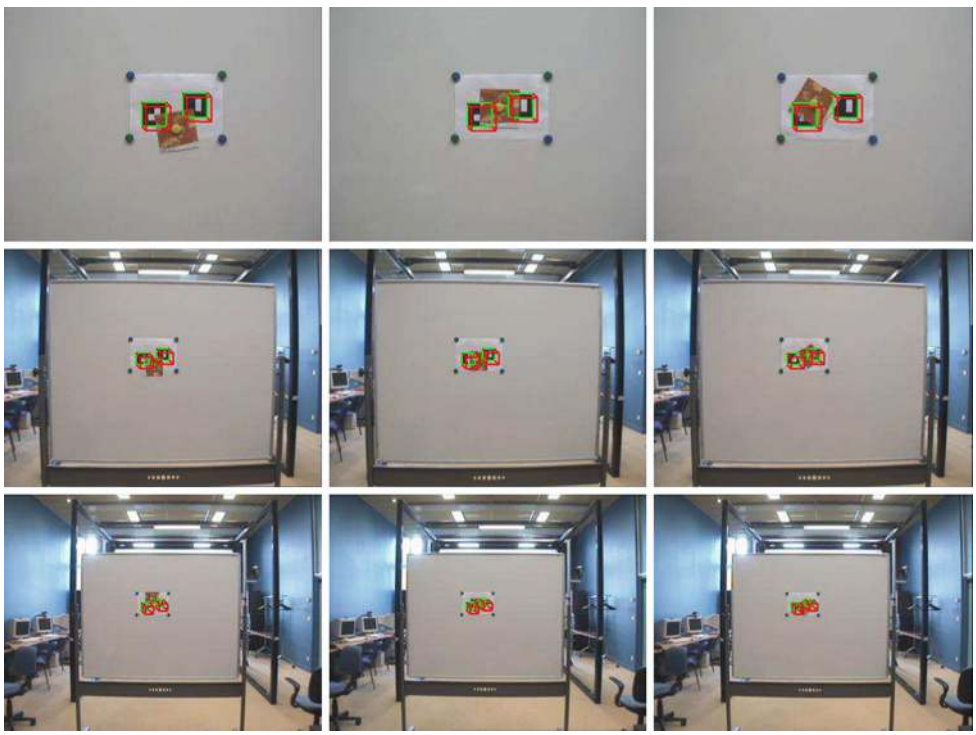


Fig. 16 Partial occlusion with change in scale

the size of target side. Moreover, our method presents a better precision in the pose parameters estimation compared to linear N-points method of Ansar and Daniilidis [1]. In the work of Didier [7], the OI method presented the best compromise in terms of computing time, generalization error and real distance estimation compared to the least squares and the analytical algorithms. Whereas, in our evaluations, the hybrid Kalman filter performs the best performances during the experimental tests compared to other methods including the hybrid OI presented in [7].

We can conclude that the hybrid techniques are well adapted to estimate the camera pose. This type of methods contributes to the improvement of the execution time and the registration accuracy.

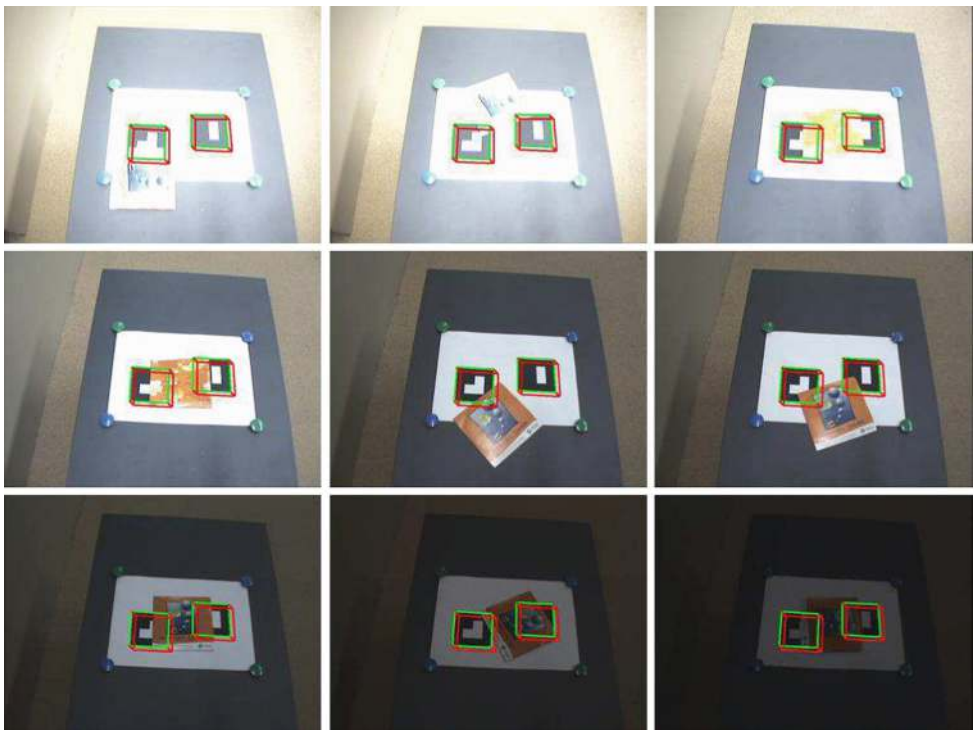


Fig. 17 Partial occlusion with change in illumination

Fig. 18 Partial occlusion with change in orientation

5.2 Robust tracking results

We present now the experiments we realized to evaluate the robust tracker. The test consists in using a set of images taken from various environment conditions. As we see in Fig. 15, the targets are tracked even if they are not identified (visible points less than 4 for one target). The robust tracker handles occlusions, maintains the virtual cube overlaying and it is robust to change in scale (Fig. 16), illumination (Fig. 17) and orientation (Fig. 18).

Table 1 shows the execution time and reconstruction error of the robust algorithm according to the number of visible points. We notice that the accuracy of the pose estimation depends on the used number of points. A better pose estimation is obtained when the number of visible points is significant; however, the time execution increases also in this case.

Table 1 Execution time and reconstruction errors of the robust method

Visible points	Exec. time (ms)	Recons. err. (pixel)
4	35.37	0.75
5	36.11	0.41
6	36.91	0.34
7	37.57	0.30

5.3 Hybrid tracking results

5.3.1 Visible target

In this part, we present the obtained results from the hybrid tracker. In this test, the target is totally visible by the camera, so the pose can be estimated either by the camera or the IMU. The camera and the IMU frequencies are, respectively, 25 and 100 Hz. The goal of this experiment is to check the registration precision by projecting a virtual cube using the IMU data.

In Fig. 19, the virtual cube is superimposed on the target using the IMU pose parameters. The results show that the estimated pose using the IMU is accurate. Indeed, the virtual object is correctly aligned on real targets for different camera pose.

We observe in Fig. 20a that the two curves are practically superimposed and the IMU does not present a drift. Indeed, the camera corrects the IMU drift for each acceleration data used in the Kalman filter because the sensors have the same frequency. On the other side (Fig. 20b), we notice that the two curves are shifted, and this drift is due to the camera sampling time which is 4 times greater than the IMU data sampling.

The Fig. 21 illustrates the IMU drift. The accumulation of constants of integration is the main cause of the drift. Indeed, the IMU positions are computed from double integration of

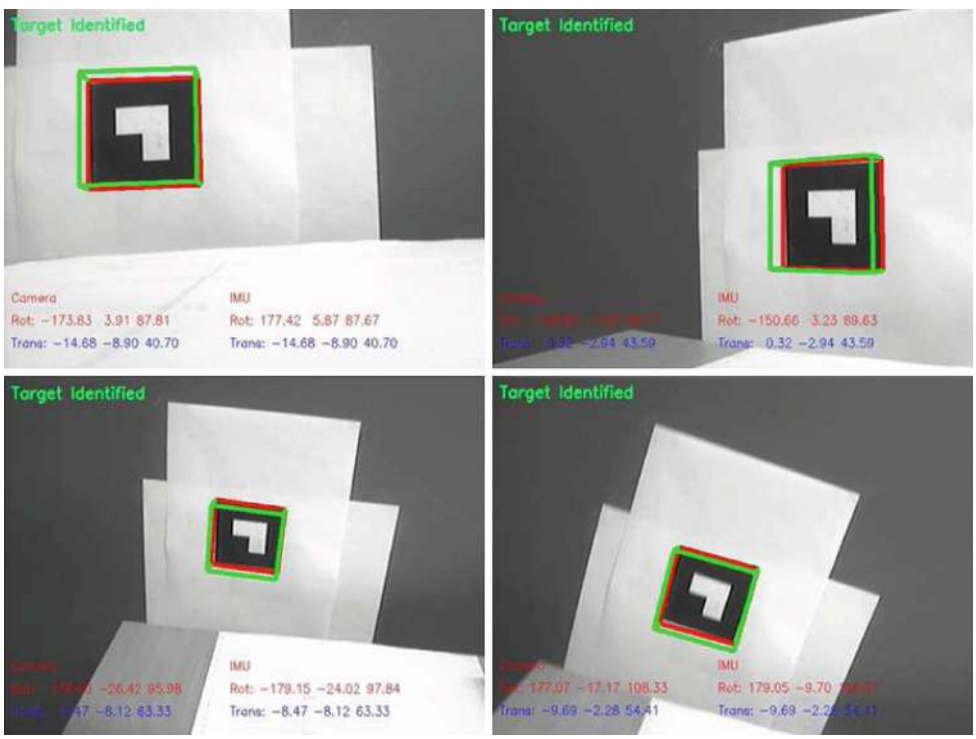


Fig. 19 Tracking results with the hybrid system when target is visible

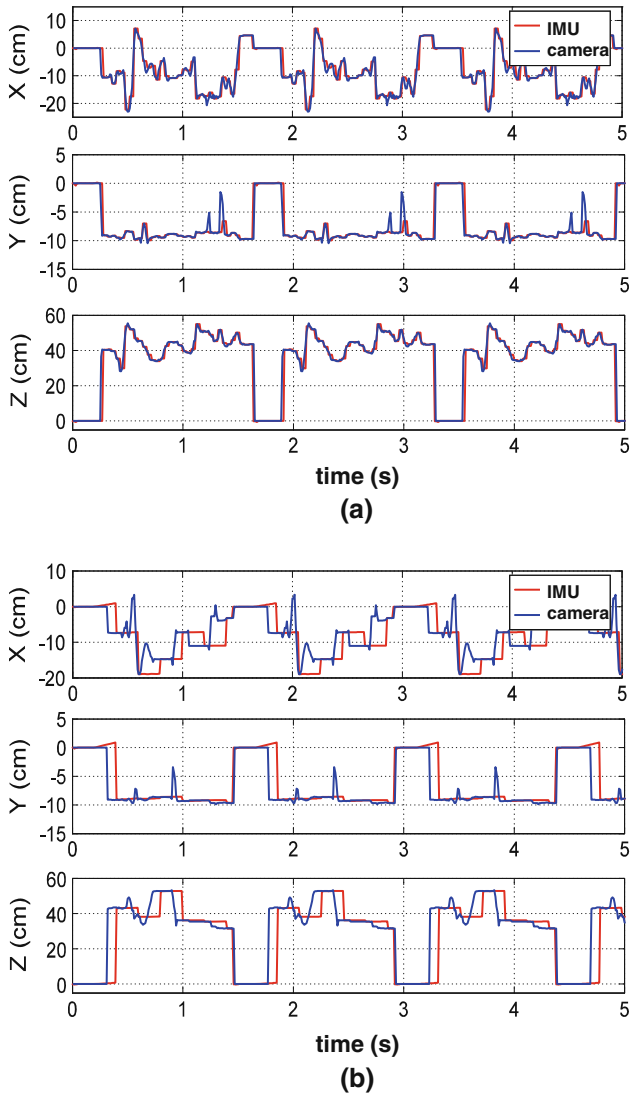


Fig. 20 Tracking with the hybrid system. **a** IMU frequency: 25 Hz, **b** IMU frequency: 100 Hz

accelerations. This drift error is represented by the difference between the IMU positions and the camera positions.

5.3.2 Occluded target

In this experiment, we tested our hybrid system in presence of occlusions. We occluded partially and totally the target to check the IMU capability to estimating the pose. Results illustrated in Fig. 22 show that the IMU computes correctly the position and the orientation and allows the overlying of the virtual cube on targets, whereas the camera is not able to determine the pose because the object of interest is not identified. In this experiment, the first tracking module based on the vision localization system fails. This registration module is launched if the target is totally visible to the camera; however, if one feature point of the target is not visible, then the identification algorithm stops tracking the target. The hybrid module replaces the camera tracking, the IMU estimates the pose of its local reference frame according to the target reference frame, and since the hybrid system is calibrated, then the IMU pose is projected into the camera reference frame.

The hybrid sensor enables target tracking when markers are totally occluded by another object in the scene. The advantage of this module is to keep tracking and to allow localization of the system in worst conditions of visibility.

Figure 22 shows a complete scenario of occlusion management of the hybrid tracker. Indeed, when the target is visible, the visual-based tracker enables identification and real-time tracking. However, when the target is occluded and the number of feature points is not sufficient to estimate the pose transformation, in this case the IMU replaces the camera and estimates the pose.

The hybrid tracking manages target occlusions when they are not identified by the vision system. The condition of total or partial visibility of the target is not required for this registration module. The multi-sensors system provides an interesting solution for the occlusion management; however, drifts in translation allow a short duration use of the IMU, especially, when the hybrid system is moving.

5.3.3 Motion blur

In order to test the robustness of our system in case of image blur generated by abrupt motions of the hybrid sensor, we carried out an experiment which consists in moving the tracking device very quickly around the target. We notice in Fig. 23 that the target is tracked although it is not detected by the camera due to bad image quality. However, the IMU replaces the camera and enables the pose estimation and the overlaying of the virtual cube. This experiment shows another advantage of the use of the IMU. In case of motion blur, the vision system is not very robust to noninstantaneous image acquisitions. If the motion frequency of the hybrid sensor is higher than the camera frequency, this phenomenon appears. The IMU does not require the target visibility to locate the sensor reference frame in 3D space. However, if the motion sensor is carried out in a frequency higher than that of the IMU, a latency is generated in this case, and this will cause a positioning shift of the virtual overlaying.

5.4 Overall system test

In order to test the overall system of occlusion management, we integrated the visual and the robust trackers presented previously with the hybrid system developed in this section (Fig. 24). Indeed, when the target is partially occluded, the RANSAC algorithm allows to track the object feature points, and then, the pose is estimated and the virtual cubes are

Fig. 21 IMU drift

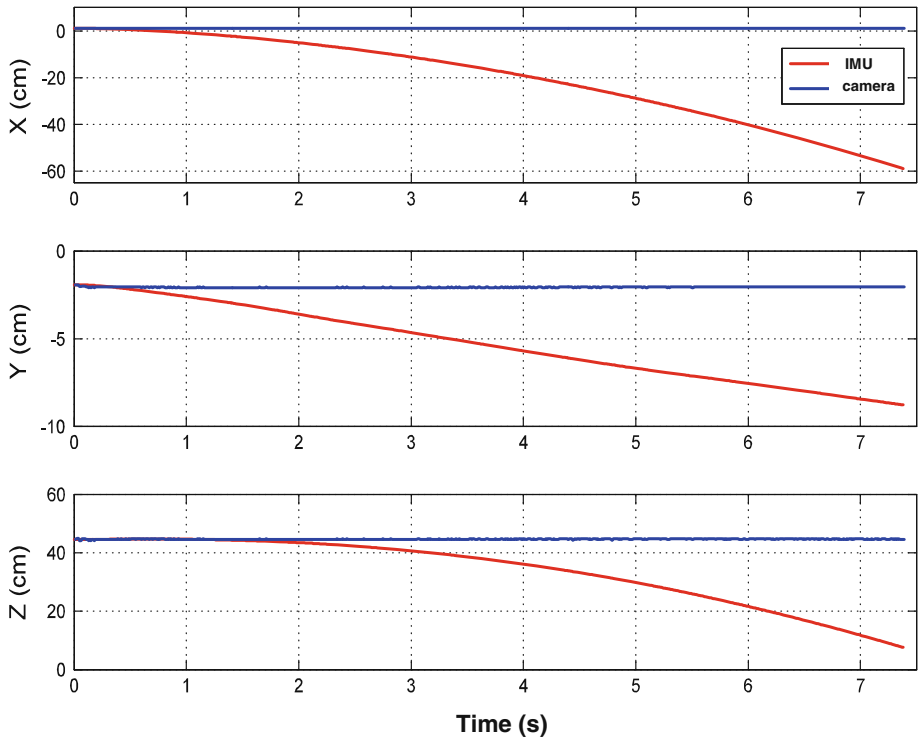
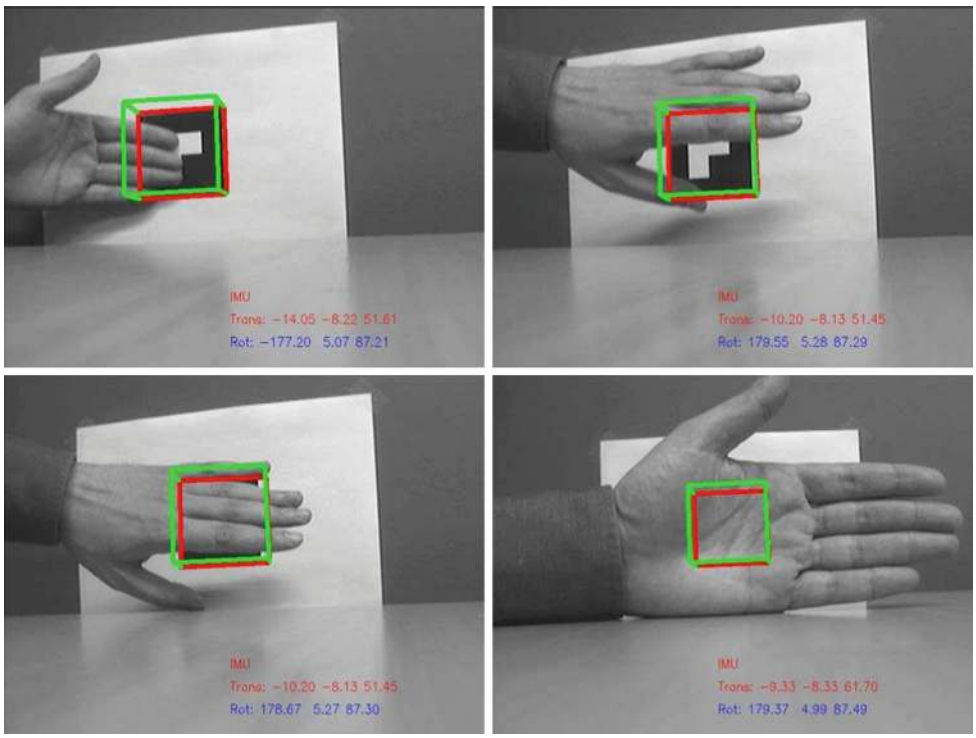


Fig. 22 Tracking results with the hybrid system when target is occluded



overlaid upon targets. However, if both targets are totally occluded, the robust algorithm does not manage anymore the tracking and, consequently, the hybrid module starts and enables the camera localization.

6 Discussion

In this paper, we contribute to handle some limitations related to AR systems working in real-time within real environments.

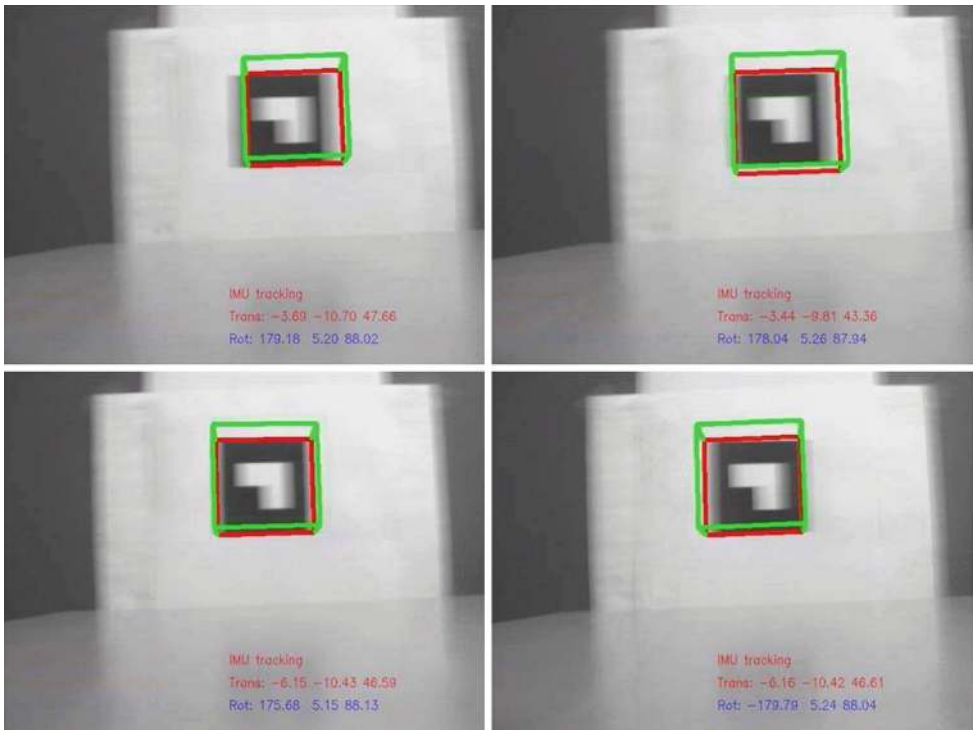


Fig. 23 Tracking results with the hybrid system in presence of motion blur

We proposed and developed a detection and identification system for coded targets. This system is based on a spatial sampling of rectangular areas within the image. The visual tracker enables real-time 2D localization with high accuracy. Indeed, our system uses a code of 16 bits, while other system such as CyberCode [25] computes 33-bits code and also imposes some constraints to object shapes. Besides, only a single target is required for the identification process, contrarily to Intersense system [21] which needs multiple targets to compute its circular codes.

For target tracking in images, we developed a hybrid algorithm of pose estimation to improve the accuracy and speed of localization by combining an analytical method and an iterative technique based on the Kalman filter. We proved, experimentally, that our pose estimator enables better localization compared to the existing techniques (Analytical algorithm, Hybrid OI, EKF).

Moreover, the occlusion handling represents an important contribution in our work. The robust tracker and the vision-inertial tracker enable target tracking in case of occlusions and environment changes. Similar works are presented in the state of the art; however, compared to existing works, our tracker estimates both translation and rotation of the camera using the IMU data. Besides, the calibration procedure of the hybrid system requires, only, a single camera and IMU pose to determine the transformation between reference frames.

Experiments in a real life conditions demonstrated that our hybrid tracking device brings significant improvements to vision-based tracking systems. This system can run in indoor and outdoor environment and meet real-time and accuracy requirements.

7 Conclusion

This work enabled to overcome some limitations related to tracking systems in augmented reality. We contributed in this paper to solve the problem of markers registration by establishing a multi-modal architecture of tracking and occlusion handling. This architecture consists of a registration module of coded targets based on a hybrid algorithm of pose estimation. We extended our space-time localization method by a module of feature points tracking and occlusion management. This step is based on the study of the tracking properties (feature points detection, 2D–2D and 2D–3D matching) and on the adaptation and the management of various experimental conditions. Finally, we included into our system, a multi-sensors tracking part. This module is composed of a hybrid device which overcomes total target occlusion. The fusion of kinematic data from the IMU and the camera images opens new prospects to improve the robustness of vision-based systems in augmented reality applications. However, such multi-sensor systems present some limitations which are difficult to overcome. In this context, it is important to identify the intrinsic parameters of the sensors in order to characterize more finely the duality and the collaboration of the heterogeneous data resulting from the sensors.

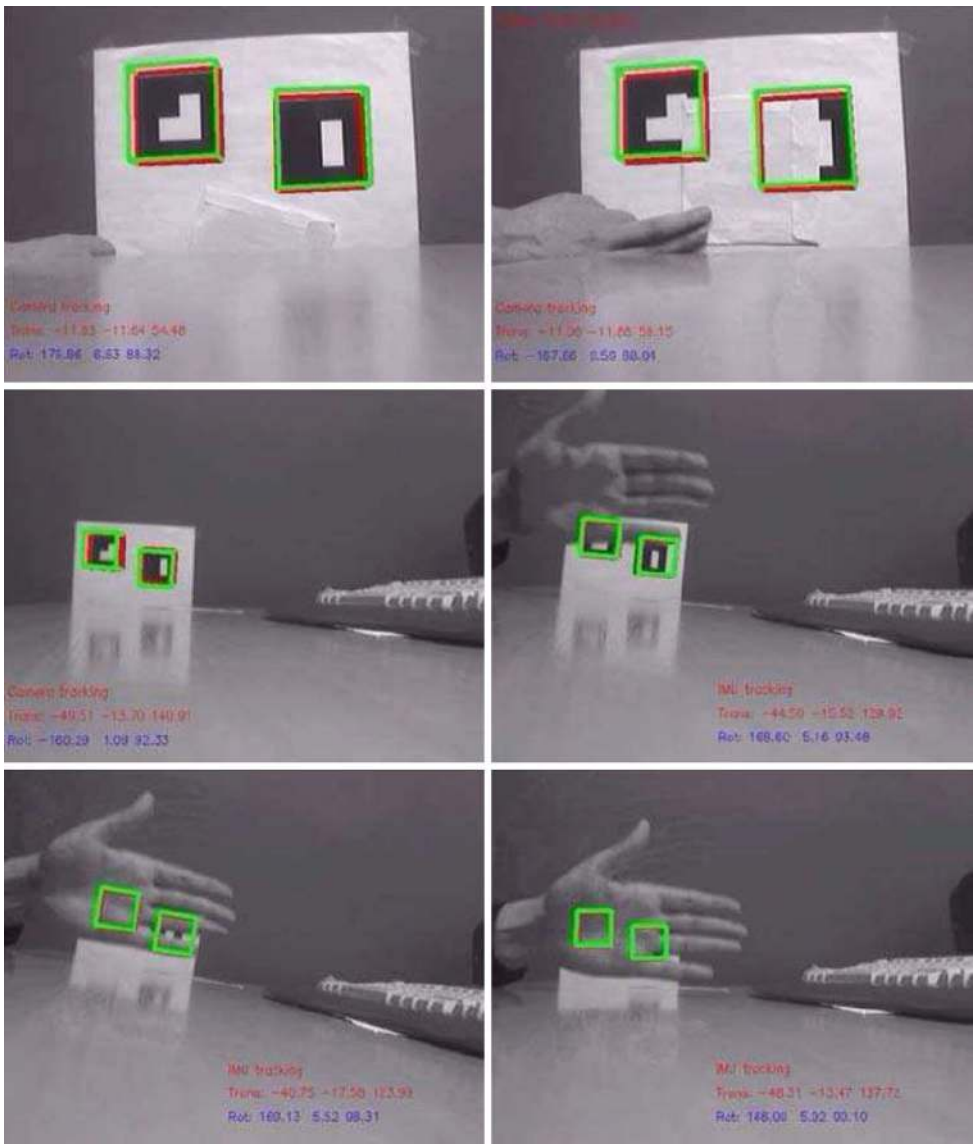


Fig. 24 Overall tracking system results

In near future, we plan to use the AR tracking system in outdoor environments. In the application, we associate the vision-inertial tracker with a GPS in order to localize robustly and accurately the user in the environment where the data acquisition conditions cannot be controlled. Moreover, the AR hybrid system can be used as a wearable self-tracker to perform robust localization. A motivating application would be the use of the hybrid sensor with a mixed reality helmet. Our system enables real-time registration of virtual object overlaid on the user's visor. The tracking should be stable and robust to motion blur and illumination changes.

References

1. Ansar, A., Daniilidis, K.: Linear pose estimation from points or lines. IEEE Trans. Pattern Anal. Mach. Intell. **25**(5), 578–589 (2003)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf), vol. 110, pp. 346–359. Elsevier Science Inc., New York, NY, USA (2008)
3. Bleser, G., Stricker, D.: Advanced tracking through efficient image processing and visual-inertial sensor fusion. In: IEEE Virtual Reality (VR'08), Reno, Nevada, USA, pp. 137–144, March 2008
4. Chen, D.M., Tsai, S.S., Vedantham, R., Grzeszczuk, R., Girod, B.: Streaming mobile augmented reality on mobile phones. In: ISMAR '09: Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality, Washington, DC, USA, pp. 181–182 (2009)
5. Cho, Y., Neumann, U.: Multi-ring color fiducial systems for scalable fiducial tracking augmented reality. In: VRAIS'98: Proceedings of the Virtual Reality Annual International Symposium, Atlanta, GA, USA, p. 212 (1998)
6. Comport, A.I., Marchand, É., Chaumette, F.: A real-time tracker for markerless augmented reality. In: ISMAR'03, Tokyo, Japan, pp. 36–45, October 2003
7. Didier, J.-Y., Ababsa, F., Mallem, M.: Hybrid camera pose estimation combining square fiducials localization technique and orthog-

onal iteration algorithm. Int. J. Image Graph. (IJIG) **8**(1), 169–188 (2008)
8. Fiala, M.: Artag, a fiducial marker system using digital techniques. In: CVPR'05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, vol. 2, pp. 590–596 (2005)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (June 1981)
10. Foxlin, E., Naimark, L.: Vis-tracker: a wearable vision-inertial self-tracker. In: VR'03: Proceedings of the IEEE Virtual Reality 2003, Los Angeles, California, USA, pp. 199–206, March 2003
11. Harris, C., Stephens, M.: Combined corner and edge detector. In: Proceedings of the Alvey Conference, pp. 147–151 (1988)
12. Kato, H., Billinghurst, M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: IWAR'99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, San Francisco, CA, USA, pp. 85–92 (1999)
13. Klein, G., Murray, D.: Parallel tracking and mapping on a camera phone. In: ISMAR '09: Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality, Washington, DC, USA, pp. 83–86. IEEE Computer Society (2009)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
15. Lu, C.P., Hager, G.D., Mjolsness, E.: Fast and globally convergent pose estimation from video images. IEEE Trans. Pattern Anal. Mach. Intell. **22**(6), 610–622 (2000)
16. Maidi, M., Ababsa, F., Mallem, M.: Vision-inertial system calibration for tracking in augmented reality. In: 2nd International Conference on Informatics in Control, Automation and Robotics (ICINCO'05), Barcelona, Spain, pp. 156–162 (2005)
17. Maidi, M., Ababsa, F., Mallem, M.: Robust augmented reality tracking based visual pose estimation. In: 3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO'06), Setbal, Portugal, pp. 346–35 (2006a)
18. Maidi, M., Ababsa, F., Mallem, M.: Robust fiducials tracking in augmented reality. In: The 13th International Conference on Systems, Signals and Image Processing (IWSSIP 2006), Budapest, Hungary, pp. 423–42 (2006b)
19. Maidi, M., Didier, J.Y., Ababsa, F., Mallem, M.: A performance study for camera pose estimation using visual marker based tracking. Mach. Vis. Appl. Int. J. **21**(3), 365–376 (2010)
20. Maidi, M., Preda, M., Le, V.-H:. Markerless tracking for mobile augmented reality. In: IEEE International Conference on Signal and Image Processing Applications (ICSIPA2011), Kuala Lumpur, Malaysia, pp. 301–306. IEEE Signal Processing Society, November 2011
21. Naimark, L., Foxlin, E.: Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR'02), Darmstadt, Germany, pp. 27–36 (2002)
22. Naimark, L., Foxlin, E.: Encoded led system for optical trackers. In: ACM and IEEE International Symposium on Mixed and Augmented Reality (ISMAR'05), Vienna, Austria, October 2005
23. Ozuysal, M., Fua, P., Lepetit V.: Fast keypoint recognition in ten lines of code. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR 07), pp. 1–8 (2007)
24. Quan, L., Lan, Z.D.: Linear n-point camera pose determination. IEEE Trans. Pattern Anal. Mach. Intell. **21**(8), 774–780 (1999)
25. Rekimoto, J., Ayatsuka, Y.: Cybercode: designing augmented reality environments with visual tags. In: DARE'00: Proceedings of DARE 2000 on Designing Augmented Reality Environments, Elsinore, Denmark, pp. 1–10 (2000)
26. Stricker, D., Klinker, G., Reiners, D.: A fast and robust line-based optical tracker for augmented reality applications. In: Proceedings of First International Workshop on Augmented Reality (IWAR'98), San Francisco, USA, pp. 129-145 (1998)
27. Takacs, G., Chandrasekhar, V., Gelfand, N., Xiong, Y., Chen, W.-C., Bismpigiannis, T., Grzeszczuk, R., Pulli, K., Girod, B.: Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In: MIR '08: Proceeding of the 1st ACM International Conference on Multimedia Information Retrieval, New York, NY, USA, pp. 427–434. ACM (2008)
28. Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., Schmalstieg, D.: Real-time detection and tracking for augmented reality on mobile phones. IEEE Trans. Vis. Comput. Graph. **16**(3), 355–368 (2010)
29. Welch, G., Bishop, G.: An introduction to the Kalman filter. Technical report N. TR 95-041, Department of Computer Science, University of North Carolina, USA (2004)
30. You, S., Neumann, U.: Fusion of vision and gyro tracking for robust augmented reality registration. In: VR'01, Yokohama, Japan, pp. 71–78, March 2001
31. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)