A novel system for object pose estimation using fused vision and inertial data

Juan Li , Juan A. Besada, Ana M. Bernardos, Paula Tarrío, José R. Casar

Keywords:
Pose estimation
Error propagation
Augmented reality
Sensor fusion
Visual sensors
Inertial systems

A B S T R A C T

Six-degree-of-freedom (6-DoF) pose estimation is of fundamental importance to many applications, such as robotics, indoor tracking and Augmented Reality. Although a number of pose estimation solutions have been proposed, it remains a critical challenge to provide a low-cost, real-time, accurate and easy-to-deploy solution. Addressing this issue, this paper describes a multisensor system for accurate pose estimation that relies on low-cost technologies, in particular on a combination of webcams, inertial sensors and a printable colored fiducial. With the aid of inertial sensors, the system can estimate full pose both with monocular and stereo vision. The system error propagation is analyzed and validated by simulations and experimental tests. Our error analysis and experimental data demonstrate that the proposed system has great potential in practical applications, as it achieves high accuracy (in the order of centimeters for the position estimation and few degrees for the orientation estimation) using the mentioned low-cost sensors, while satisfying tight real-time requirements.

1. Introduction

The term 'pose' is usually employed to refer to the combined information on position and orientation of a moving target (i.e., an object or a human). Position is represented by the three-dimensional location of the object, while orientation may be expressed as a set of consecutive rotations. Determining the pose of a target in 3D space is an important task in many traditional application fields, such as robotics [1–3] (e.g., for robot guidance, object manipulation, etc.), indoor tracking and activity estimation, or interaction [4].

In particular, in recent years, an attractive application area requiring accurate pose estimation is indoor Augmented Reality (AR). AR has been widely explored in training, entertainment, education and tourism to facilitate a novel way for the users to interact with their surroundings [4–9]. Ideally, an AR system should be able to overlay the virtual information upon the real world with no error and no latency, thus it needs a perfectly estimated pose of the target relative to the real world. Despite the progress that has been made to date, current technologies for indoor deployments are not able to achieve these performance goals. Better said, they still offer limited performance in terms of accuracy, computational cost, usability, robustness, on-board power consumption and easiness of deployment. In this context, this paper describes a multisensor solution for accurate pose estimation using low-cost technologies. The designed system provides pose estimation in real time and may be easily adapted to different environments.

The enabling apparatus is simple: (a) one or more infrastructure vision sensors (commercial off-the-shelf cameras), which are fixed and calibrated beforehand, (b) a three-axis accelerometer in the object to be tracked (e.g., embedded accelerometers in mobile devices), (c) a printable colored marker to be stuck on the object and (d) a server. The pose calculation process is implemented on the server side, leaving computing power of the client side for applications. The proposed fiducial has a linear thin stripe-like geometry; it is thus different when compared against the conventional square fiducials that are used in ARToolKit [10] or ARTag [11]. Linear fiducials may better adapt to final services, because they are less invasive to the environment than the square fiducials, due to their smaller dimensions. The thinness also allows them to be attached to a small surface, for example, borders of mobile devices, hats or eyeglasses frames. Therefore, the proposed solution has the potential for indoor person tracking, robot tracking, mobile AR and interaction in smart spaces. With respect to pure vision-based approaches, the fusion of vision data with accelerometer measurements reduces the number of unknown pose parameters; therefore, robustness and computational efficiency are enhanced. Moreover, gravitational acceleration measurements are used to aid in the pose estimation, but no acceleration integration process is

performed. Therefore, our proposal generates zero-drift solutions and eliminates the requirement of having an initial state.

Within a bounded space, the system can work with a single active camera (monocular approach), or with two cameras (stereo vision approach), with the latter resulting in increased accuracy. To equip a room-like space with our pose estimation technology, more than two cameras may be needed to cover the whole space. The issues related to multi-camera management, such as object tracking and camera selection, will not be studied in this work. In our previous work [12], we proposed a six degree of freedom pose estimation system that fuses acceleration data and stereo vision. The system was evaluated by comparing to real measurements and a state-of-the-art marker-based system. Experimental results showed that the proposed stereo vision system provides high accuracy. This article introduces a new strategy to estimate 6-DoF pose by fusing data from the target object's accelerometer with input from one camera. Each component of the system is analyzed thoroughly. Besides, a complete pose estimation analytical error model for both the monocular and the stereo vision system is derived and validated by real tests. This paper provides more extensive experimental results and a thorough comparison to the state of the art to evaluate the system qualitatively and quantitatively. From our simulations and real tests of the system, it will be shown that the proposed pose estimation system has great potential in practical applications, as it achieves high accuracy (in the order of centimeters for the position estimation and few degrees for the orientation estimation) in real-time, using the mentioned low-cost sensors. Furthermore, the possible applications and the guidelines for the practical implementation of the system are addressed.

The rest of the article is organized as follows. Section 2 includes a review of previous work on object pose estimation systems. Section 3 states the mathematical formulation and explains the contributions of each sensing technology. Section 4 is dedicated to describe the pose estimation strategy, which fuses data from inertial and vision sensors. Section 5 models the errors of the system. Accuracy and computational load are assessed and the sensor error model is validated by experimental results in Section 6. The performance of the proposed approach applied to pointing applications is evaluated in Section 7. Finally, Section 8 concludes the paper and describes further lines of work.

2. Related work

Object pose estimation has been studied over the past several decades and a wide range of technologies have been explored [1–11,13–30]. Depending on the sensing technology, the available approaches may be classified into three main categories: sensor-based, vision-based and hybrid approaches. The existing literature on these categories is described below.

2.1. Sensor-based methods

Inertial sensors including accelerometers and gyroscopes have been widely used for robots [1], aircrafts and vehicles navigation [13]. The principle for determining position and orientation using these sensors is based on Newton's laws. Accelerometers measure the linear acceleration in the inertial reference frame, which is integrated to get the velocity and then integrated again to get the position. Gyroscopes measure the angular velocity and by integrating once, rotation angles can be calculated. Inertial Measurement Units (IMU) are maturely developed units for motion tracking which typically contain three orthogonal accelerometers and three orthogonal gyroscopes. They run at a high rate, therefore they are able to track fast and abrupt movements. Furthermore, they are not influenced by illumination and visual occlusion. On the downside, they suffer from a severe drift problem caused by accumulation of measurement errors, thus a periodic re-calibration is required. Several methods have been proposed to minimize the drift problem. For example, in [14], relative measurements were used instead of absolute measurements to reduce the drift error. It is worth mentioning that in inertial-based methods, the initial state is needed to calculate the absolute pose.

Magnetometers are used to get the heading angle by sensing the earth magnetic field [15]. In order to get a full pose estimation, they need to be combined with other technologies. The algorithm in [15] integrates inertial sensors with magnetometers and keeps the tracking results within about 2m of the true track throughout the entire in-building run. However, the measurements provided by magnetometers can be corrupted due to the presence of metallic objects in the surroundings, which is quite usual in indoor scenarios [16].

A different approach to positioning is the use of Radio-frequency (RF) technologies. They aim at locating moving objects (smartphones, robots, etc.) through diverse techniques(refer to e.g. [30] for a survey): WiFi, Bluetooth or ZigBee-based solutions usually rely on fingerprinting techniques (e.g. [31]) or channel modeling (e.g. [32]) to achieve a limited accuracy (3–4m in average). In addition, RF positioning systems do not generally support orientation estimation, therefore not providing a full pose estimation.

2.2. Vision-based methods

Visual sensing technologies try to interpret the environment through observations from cameras. Most of the available proposals estimate the spatial relationship between the camera and the object by finding the correspondence between 2D image points and 3D scene points. According to the tracked features, most of the methods can be grouped into marker-based, ready to decode a known external visual reference, and markerless methods, not needing any previously known symbol.

2.2.1. Marker-based methods

Marker-based methods recover the transformation between the fiducial (artificial) marker and the vision sensor by extracting the feature points previously defined in the marker. Several available libraries use planar fiducial for tracking, such as ARToolKit [10], ARTag [11], Studierstube [28], AprilTag [17] and OpenCV [29]. ARToolKit was developed in 1999 by Hirokazu Kato and has been widely used. Based on ARToolKit, ARTag was later developed to provide improved performance. The extended version of ARToolKit is ARToolKitPlus, which added more features over the ARToolKit. However, it is no longer developed and has a successor: Studierstube Tracker. It supports mobile phones as well as PCs and has low memory requirements. However, it is not open source. AprilTag has been recently developed for PCs and further improves accuracy and robustness. OpenCV is an open source cross-platform toolkit for image processing that supports PCs as well as mobile platforms. This library is still in development and has a large community of users. In [18], a chessboard pattern is tracked by OpenCV to implement mobile AR. Markers used in these libraries are black-white and have high contrast, so they are easily recognizable. On the downside, contrast-based detection is sensitive to lighting. Generally speaking, marker-based methods can provide high accuracy. However, the marker size and the distance as well as the viewing angle to the marker will affect the accuracy. These aforementioned markers need a big, flat surface to be placed. Therefore, they are unsuitable to be attached to a small object to be tracked, such as a mobile device. Instead, by using an on-body camera to track markers placed in known locations it is feasible to estimate the object's

pose. These approaches are widely used in simple scenarios because of their easy setup. However, their use can be complicated in a wide working area. For example, large quantities of markers need to be deployed and measured carefully. Markers can be intrusive to the environment (causing visual discomfort). In case of a mobile device, the on-board imaging processing is battery-draining.

In addition to paper markers, in commercial markets, retro-reflective elements are used in two of the most famous motion tracking systems: Opti-Track [19] and Vicon [20]. They use fixed high-speed infrared cameras to track markers and provide highly accurate results. However, they are expensive and not suitable for a low-cost and simple service. Light-emitting diodes (LEDs) are also used in several systems [2,9]. The system proposed in [9] is designed for a tablet-based AR service. It relies on tracking six LEDs mounted on the back of the tablet. Obviously, having six LEDs attached to the device makes the system complex in terms of real service-oriented feasibility. Moreover, the infrastructure cameras have to see the whole back of the tablet, which largely constrains the movement of the tablet, making it unnatural for the user. Another LED-based system is described in [2], in which four LEDs are placed on a robot to enable its tracking. Compared with paper colored fiducials, LEDs are easier to be detected and less sensitive to illumination changes, but they may be bulky and have to be powered either by wires or batteries.

2.2.2. Markerless methods

In order to get rid of artificial markers, researchers are making efforts to detect natural landmarks from image sequences, which is also referred as markerless methods. Many robust local descriptors including SIFT (Scale-Invariant Feature Transform) [33] and SURF (Speeded-Up Robust Features) [34] are stable under different viewpoints and lighting conditions and can be used to detect features existing in the scenes to build a markerless method. However, their computational requirements are stronger than those of methods relying on artificial markers. Although there are some proposals to adapt these methods to mobile platforms [8], the use of markerless methods is still a challenge for mobile devices with limited computational capabilities. Simultaneous localization and mapping (SLAM) systems [27,35] calculate pose from natural feature points. A survey on SLAM can be found in [36]. Generally speaking, markerless methods still have a large room for improvement in terms of accuracy, robustness and efficiency.

With the advances in imaging technologies, new types of vision sensors have been developed. RGB-Depth cameras with moderate prices, such as Microsoft Kinect devices [21], have attracted attention from researchers. These devices capture RGB images along with per-pixel depth information. In [3], a micro air vehicle is mounted with a Microsoft Kinect camera to track its pose. Similarly, authors in [22] have designed a system composed by an infrared dot-pattern projector and an infrared camera. The system recovers the transformation by finding correspondences between the reference pattern and the detected dot grids. A time-of-flight camera is a relatively new type of sensor that delivers 3-dimensional imaging at a high frame rate, simultaneously providing intensity data and range information for every pixel [23].

2.3. Hybrid sensor-vision methods

Each of the previous approaches has its strengths and limitations. An alternative solution aiming at taking advantage of the benefits of each of them, while softening their hindrances, is to combine both sensor types, as it is done in [5,24–26]. In our previous work [7], data from an ultrasound location system were fused with the magnetic sensor of a tablet device to provide pose information for AR services. The ultrasound location system obtained centimeter level of accuracy. However, orientation based on the magnetometer was noisy indoor due to the influence from metallic objects in the surroundings.

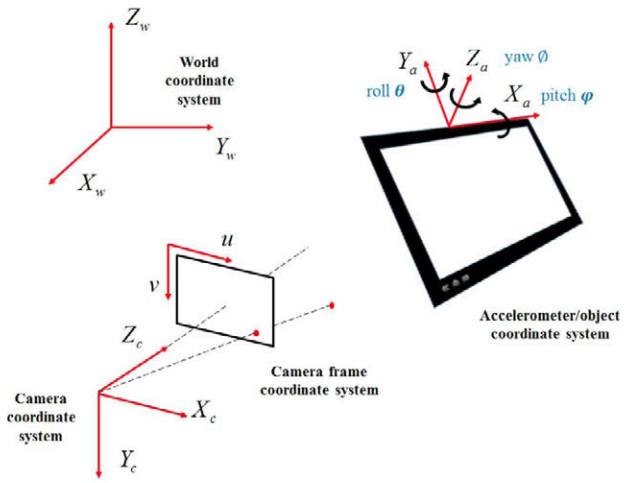


Fig. 1. Coordinate systems involved in the proposed system and orientation expressed in pitch-roll-yaw rotation angles.

Typically, visual and inertial measurements are combined using a filtering framework. Kalman filter and its derivatives (Unscented Kalman filter, Extended Kalman filter) are favorably selected to perform sensor fusion by integrating measurements from the vision-based system and from the sensor-based system [24,25].

Most of the hybrid systems choose gyroscopes for calculating the orientation through integration. Although the method is straightforward, gyroscopes suffer from drifts caused by zero rate offset. For example, within the tracking methods described in [5,26], gyroscope was adopted to measure the orientation, and in return, the vision-based system was focused on correcting the drift of the inertial system.

The method proposed in this paper works on a similar concept but rather than using gyroscope, we use gravitational acceleration, which avoids the common drift problem. The accelerometer contributes to two rotation angles and the calculation burden is largely reduced. To our knowledge, this is the first article to propose a linear fiducial based vision-inertial fusion approach which works on low-cost visual technologies delivering few centimeters error.

3. Pose estimation problem and sensor modeling

This section briefly introduces the necessary mathematical framework, stating definitions for the different involved coordinate systems, general notation and transformation equations. Then, the working principle and available data provided by each sensing technology are described.

3.1. Notation, coordinate systems and coordinate transformations

Our fusion strategy manages inputs from infrastructure (cameras) and mobile sensors, and therefore it is necessary to handle transformations between several coordinate systems. The necessary definitions and notation used for each coordinate system are provided below, together with the list of variables used throughout the paper. The coordinate transformation equations are also presented.

As schematically depicted in Fig. 1, four coordinate systems are involved:

- World coordinate system $\{w\}$: This is the global reference system used for describing the position and the orientation

Table 1
Summary of paper variables.

Variables	Meanings
R_{ij}	3×3 rotation matrix from $\{i\}$ to $\{j\}$
$\mathbf{t}_{ij}$	3D translation vector from $\{i\}$ to $\{j\}$
ψ	Rotation angle about x-axis (pitch)
θ	Rotation angle about y-axis (roll)
ϕ	Rotation angle about z-axis(yaw)
g	Magnitude of gravity
D	The distance between two reference points
$\mathbf{P}_i = [X_i, Y_i, Z_i]^T$	A point in $\{i\}$, $i = w, c, a$
$\mathbf{P}_f = [u, v]^T$	A point in $\{f\}$
$\mathbf{g}_w = [0, 0, -g]^T$	Gravitational acceleration in $\{w\}$
$\mathbf{g}_a = [g_{ax}, g_{ay}, g_{az}]^T$	Gravitational acceleration in $\{a\}$
$\mathbf{s} = [x, y, z, \psi, \theta, \phi]^T$	6-dimensional pose vector

(pose) of objects; its z-axis points towards the sky, being perpendicular to the ground and x-y axes are tangential to the ground.
- Camera coordinate system $\{c\}$: For each camera, it is a Cartesian reference system attached to the camera, whose origin is located in the camera optical center; its z-axis is along the optical axis and therefore x-y axes are parallel to the image plane.
- Camera frame coordinate system $\{f\}$: This 2D coordinate system is used to refer positions in the image plane in pixel units. The origin is the left-up corner of the image. The axes (to be called u-v) are parallel to the x-y axes of the camera coordinate system $\{c\}$.
- Accelerometer/object coordinate system $\{a\}$: Its orientation is aligned with the three accelerometer sensing axes. We assume the object coordinate system has its origin in the center of the colored marker, and the marker is parallel to the x-axis of the accelerometer. For example, for a mobile device, this means in practice that while the marker can be put in a user-defined position, it has to be aligned with the device's border.

Table 1 summarizes the notation and variables used in the following equations and in the rest of the paper.

Let us denote $\{i\}$ and $\{j\}$ as two arbitrary coordinate systems (any of them may have the values $\{w\}$, $\{c\}$, $\{a\}$ or $\{f\}$). A 3D vector $\mathbf{v}$ is expressed as $\mathbf{v}_i$ in $\{i\}$, but expressed as $\mathbf{v}_j$ in $\{j\}$. The relationship between $\mathbf{v}_i$ and $\mathbf{v}_j$ can be expressed using a rotation matrix as:

$$\mathbf{v}_j = R_{ij}\mathbf{v}_i \tag{1}$$

In general, the transformation of a 3D position from a given reference frame to another can be achieved by performing first a rotation between their reference frames and then a translation (related to the offset of the coordinate systems origins), which is mathematically expressed as:

$$\mathbf{P}_j = R_{ij}\mathbf{P}_i + \mathbf{t}_{ij} \tag{2}$$

Several representations can be used to express an object or camera orientation, for instance axis-angle, quaternions and Euler angles [37]. In this article, Euler angles, as depicted in Fig. 1, are adopted to allow solution for the roll and pitch angles from accelerometer measurements. Specifically, to obtain the rotation matrix from $\{w\}$ to $\{a\}$ (R_{wa}), three consecutive rotations might be performed in order: yaw–roll–pitch [38], resulting:

$$R_{wa} = \begin{bmatrix} \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\cos\theta \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\cos\theta \end{bmatrix} \tag{3}$$

Fig. 2. The colored and shape-based fiducial used in the proposed system. The two black crosses are depicted to indicate the position of the two reference points, but they do not exist in the actual marker. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2. Accelerometer as inclination sensor

Accelerometers sense both gravitational and dynamic (movement induced) accelerations. To measure inclination, gravitational accelerations need to be isolated, removing dynamic accelerations. To do so, a low-pass filter can be used. Some literature has done further research on this topic [39]. In the case of Android mobile devices, a "gravity sensor" is embedded since API Level 9 (Android 2.3) was released. Thus, we will assume that gravitational accelerations are isolated, focusing on the basic principle for converting the measurements of gravitational accelerations to inclination angles.

In world reference coordinates the gravity vector ($\mathbf{g}_w$) is perfectly aligned with the z-axis, pointing in negative direction. The 3-axis accelerometer provides the components of the gravitational acceleration expressed in the object reference frame ($\mathbf{g}_a = [g_{ax}, g_{ay}, g_{az}]^T$). Both gravity vectors are related through a rotation matrix, according to Eq. (1). For this case, the relation is:

$$\mathbf{g}_a = \begin{bmatrix} g_{ax} \\ g_{ay} \\ g_{az} \end{bmatrix} = R_{wa}\mathbf{g}_w = R_{wa}\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} = \begin{bmatrix} g\sin\theta \\ -g\sin\psi\cos\theta \\ -g\cos\psi\cos\theta \end{bmatrix} \tag{4}$$

From Eq. (4) pitch and roll angle can be deduced from the measurements of gravitational accelerations as follows:

$$\psi = \arctan\frac{g_{ay}}{g_{az}}; \quad \theta = \arcsin\frac{g_{ax}}{g} \tag{5}$$

3.3. Camera as position sensor

Apart from accelerometer-based measurements, our system exploits images of the fiducial to derive the object's position. A key aspect of the system is the design of the fiducial marker, made to facilitate the extraction of the reference positions that are afterwards used as inputs of the algorithms to calculate the device's position and orientation. In this section we will detail both the color and geometric features of the fiducial (two reference points) and the related fiducial detection algorithm based on image processing and computer vision procedures. An additional key aspect to be described in this section is the geometric relation between the reference points' 3D positions and their projections in the camera frame coordinate system.

3.3.1. Fiducial design and detection

The fiducial aims at serving as input to provide two reference points referred to the target object. It is designed to be: (a) easily recognizable within the environment, without causing confusion with other objects and resilient to illumination changes, (b) compatible and generalizable to different applications and (c) low-cost. Considering these features, we propose a thin colored printable marker, which embodies three colored rectangles, as depicted in Fig. 2. The central part (in magenta) shares two edges with the lateral parts (in yellow and cyan), whose centers are treated as the reference points in our system (indicated by two crosses in Fig. 2). The whole marker is just several millimeters in width and the length could vary to adapt to the object to be tracked. This linear feature makes it easy to be attached to object borders, e.g., the border of a mobile device. Thanks to its non-invasive characteristic with the environment, it can be considered for applications that are not compatible with obvious deployed markers in the environment. In our system, the marker

is placed in the border of the device in such a way that it is clearly visible by the cameras. Provided it is visible and well referenced to the device's geometry, it can be attached anywhere in the object.

The colors used in our system are magenta, yellow and cyan. However, they can be arbitrary combinations as long as they fulfill two principles. Firstly, the Hue range of selected colors should not overlap in Hue-Saturation-Value (HSV) color space. Secondly, they should be easily distinguishable from the typical colors in the surroundings (tracked object and environment).

The detection algorithm is designed taking into account both the color and the shape of the fiducial. The three colors are segmented by thresholding in HSV color space. Then, the obtained regions of interest are converted to binary images. Note that the three colored rectangles in the fiducial are successive. Each of the detected regions for each of the fiducial colors is morphologically dilated using a certain kernel. Then logic 'AND' operation is applied to the binary dilated images. The result of the operation are overlapped regions which are approximately centered in the reference points. Our experiments showed that a circular kernel with a radius of five pixels is adequate for this process, although the system accuracy and robustness is not extremely dependent on this parameter. In addition, the foreground is detected by frame differencing with a static background model [40], used to remove background (environment) areas with colors similar to those of the fiducial.

The centroids of these overlapped regions are considered as reference point candidates. Then, the algorithm calculates the percentage of 'magenta' (central segment color) pixels in the segment between each pair of candidates. This percentage is compared with an upper threshold of $T = 90\%$, which has been experimentally validated as a suitable trade-off between the detection rate and imperfect color perception. Apart from the fiducial, it is rare to find regions composed by three selected colors in the foreground (at least in our environment). Based on this idea, the algorithm simply chooses the longest candidate pair as the final result. In the stereo vision subsystem, further validation can be done by examining the distance between the final (reconstructed in 3D) reference points, given the known length of the fiducial. Algorithm 1 summarizes the previous processing.

Algorithm 1 Fiducial Detection

Input: Captured image I

Output: Reference points position in the image $(u^{(1)}, v^{(1)})$ and $(u^{(2)}, v^{(2)})$

1: Detect the foreground from the image I
2: Convert the image from RGB color space to HSV color space
3: Filter the image using the thresholds for each color and get 3 binary images, I_c(cyan), I_m(magenta) and I_y(yellow)
4: Mask previous detections with foreground detection
5: Morphologically dilate I_c, I_m and I_y separately using a disk kernel with a radius of 5 pixels
6: Do logic 'AND' operation and get $I_m \& I_c = I_{mc}$, $I_m \& I_y = I_{my}$
7: Find contours of I_{mc} and I_{my} and save contours' centroids as candidates
8: Check the pixels between each two distinct candidates. If magenta pixels/all checked pixels $> T$, save the pair as one pair candidate
9: Among all the pair candidates, choose the longest pair as the final result

3.3.2. Projection of fiducial reference points into the image plane

A final aspect to tackle in this section is how 3D points in the scene are projected into the image plane. In this paper, the pinhole camera projection model is adopted, which meets the collinear condition, i.e., the world point, the principal point and the projected point are collinear [41]. Then, the relationship between a point in world coordinates $\{w\}$ ($\mathbf{P}_w$) and its 2D position ($\mathbf{P}_f$) can be expressed as:

$$\lambda \begin{bmatrix} \mathbf{P}_f \\ 1 \end{bmatrix} = M \begin{bmatrix} \mathbf{P}_w \\ 1 \end{bmatrix} = K[R_{wc}\ \mathbf{t}_{wc}] \begin{bmatrix} \mathbf{P}_w \\ 1 \end{bmatrix} \tag{6}$$

where λ is a scale factor; M is a 3×4 projection matrix summarizing the whole projection process; K is composed of camera intrinsic parameters: focal length and principal point.

Camera calibration is a process to obtain intrinsic parameters (K) and extrinsic parameters (R_{wc} and $\mathbf{t}_{wc}$), or equivalently the M matrix for a given spatial scenario and camera deployment. In our deployment, the calibration was done offline, using Matlab® Calibration Toolbox, which implements the method proposed by Zhang [42], targeted to minimize the total re-projection error.

4. The pose estimation strategy: fusing data from inertial and vision sensors

In this section, we present two fusion algorithms combining previously described data. The difference between these two algorithms is the 3D reconstruction of the two fiducial reference points. We will first describe, in Section 4.1 the stereo vision object position extraction, while Section 4.2 will describe the monocular vision object position estimation. Once the 3D positions of those points in world coordinates are obtained by either method, the complete estimation of the 6-DoF pose is performed. The common procedure for this derivation is described in Section 4.3.

4.1. Stereo vision object positioning system

In the case that two cameras detect the fiducial simultaneously, 3D positions of each fiducial reference point can be obtained by triangulation, as depicted in Fig. 3a. The process described below is an adaptation of a linear least-squares method [43]. Other triangulation procedures [44] might be used to solve this part of the problem.

To convert this geometry to algebraic expressions, we let M_L and M_R denote the calibrated world to image plane projection matrices of the left and right cameras respectively, and λ_L and λ_R the respective multipliers. Let $[u_L, v_L]$ and $[u_R, v_R]$ be the 2D projections of one reference point $\mathbf{P}_w$ in the left and right image planes. Applying Eq. (6) to each camera, the following over-determined linear equations system can be obtained:

$$A\mathbf{P}_w = b \tag{7}$$

where A is a 4×3 matrix and b is a 4×1 matrix, described next:

$$A = \begin{bmatrix} M_L(1,1) - u_L M_L(3,1) & M_L(1,2) - u_L M_L(3,2) & M_L(1,3) - u_L M_L(3,3) \\ M_L(2,1) - v_L M_L(3,1) & M_L(2,2) - v_L M_L(3,2) & M_L(2,3) - v_L M_L(3,3) \\ M_R(1,1) - u_R M_R(3,1) & M_R(1,2) - u_R M_R(3,2) & M_R(1,3) - u_R M_R(3,3) \\ M_R(2,1) - v_R M_R(3,1) & M_R(2,2) - v_R M_R(3,2) & M_R(2,3) - v_R M_R(3,3) \end{bmatrix},$$

$$b = \begin{bmatrix} u_L M_L(3,4) - M_L(1,4) \\ v_L M_L(3,4) - M_L(2,4) \\ u_R M_R(3,4) - M_R(1,4) \\ v_R M_R(3,4) - M_R(2,4) \end{bmatrix}$$

To estimate the position of the fiducial reference point we may solve the equations using a least squares approach minimizing $\|A\mathbf{P}_w - b\|$. Then we will get the reference point coordinates in $\{w\}$ as:

$$\mathbf{P}_w = A^+ b \tag{8}$$

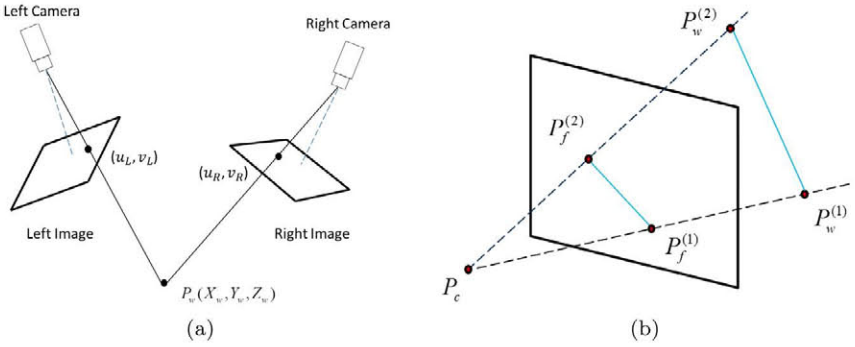


Fig. 3. (a) Stereo vision object positioning system geometry. $\mathbf{P}_w$ is the position of one fiducial reference point, to be obtained through triangulation. (b) Monocular vision object positioning system geometry. $\mathbf{P}_f^{(1)}$ and $\mathbf{P}_f^{(2)}$ are the projections of the two respective fiducial reference points $\mathbf{P}_w^{(1)}$ and $\mathbf{P}_w^{(2)}$ in the image plane.

where A^+ is the pseudo-inverse matrix of A.

The same process is done for the 3D reconstruction of the other fiducial reference point. This process provides the positions of both fiducial reference points expressed in world coordinates, to be called $\mathbf{P}_w^{(1)}$ and $\mathbf{P}_w^{(2)}$.

4.2. Monocular vision positioning system

In the case the fiducial is in the coverage of a single camera or it is partially occluded for some cameras, just being visible by one of them, a monocular positioning system may be used. The procedure relies on defining a set of constraints on fiducial reference points positions leading to a potential solution.

In our vision sensor modeling we referred to the collinear condition in the pinhole camera model, which also holds for the monocular vision system. Considering a 2D point in the image, there exists a collection of 3D points that are mapped onto the same point. These points lay on the ray connecting the camera projection center and the 2D point, as depicted in Fig. 3b for the two fiducial reference points.

From that figure, let $\mathbf{P}_f^{(1)}$ and $\mathbf{P}_f^{(2)}$ be the projection of $\mathbf{P}_w^{(1)}$ and $\mathbf{P}_w^{(2)}$ in $\{f\}$, respectively. Rearranging Eq. (6), all potential 3D points lying in the ray associated to a pixel point can be mathematically expressed as:

$$\left[X_w^{(i)}, Y_w^{(i)}, Z_w^{(i)}\right]^T = -R_{wc}^{-1}\mathbf{t}_{wc} + \lambda^{(i)} R_{wc}^{-1} K^{-1}\left[u^{(i)}, v^{(i)}, 1\right]^T, i = 1, 2 \quad (9)$$

Each of the reference points has an associated **collinearity constraint**: reference points must fulfill this equation for a given (unknown) value of $\lambda^{(i)}$. So, those collinearity constraints convert the problem of solving 6 unknown variables to a reduced problem of solving two unknown variables ($\lambda^{(1)}$ and $\lambda^{(2)}$ values in the parametric formulation of the projection lines). A simpler form of this equation is:

$$[X_w^{(i)}, Y_w^{(i)}, Z_w^{(i)}]^T = -R_{wc}^{-1}\mathbf{t}_{wc} + \lambda^{(i)}[X_t^{(i)}, Y_t^{(i)}, Z_t^{(i)}]^T, i = 1, 2 \quad (10)$$

where we define an auxiliary variable $[X_t^{(i)}, Y_t^{(i)}, Z_t^{(i)}]^T$ as follows: $[X_t^{(i)}, Y_t^{(i)}, Z_t^{(i)}]^T = R_{wc}^{-1}K^{-1}[u^{(i)}, v^{(i)}, 1]^T$

We can also define a **distance constraint** for the fiducial reference points. The distance between the two reference points in space is known (D), which allows defining the relation:

$$\|\mathbf{P}_w^{(1)} - \mathbf{P}_w^{(2)}\| = D \quad (11)$$

We just need one more constraint on reference points' positions, to be able to solve the problem. The additional constraint we are including is an **inclination constraint**, defined as follows. The positions of the reference points in the accelerometer/object coordinates are $\mathbf{P}_a^{(1)} = [-0.5D, 0, 0]^T$ and $\mathbf{P}_a^{(2)} = [0.5D, 0, 0]^T$ (the fiducial marker is centered at the origin of $\{a\}$ and aligned with its x-axis). The difference vector between those two points is expressed as $\Delta\mathbf{P}_a$ in $\{a\}$ and $\Delta\mathbf{P}_w$ in $\{w\}$. From Eq. (1), we get:

$$\Delta\mathbf{P}_w = \mathbf{P}_w^{(2)} - \mathbf{P}_w^{(1)} = R_{wa}^{-1}\Delta\mathbf{P}_a = R_{wa}^{-1}[D, 0, 0]^T \quad (12)$$

From Eq. (12) and Eq. (3) we get:

$$\Delta\mathbf{P}_w = D[\cos\theta\cos\phi,\ \cos\theta\sin\phi,\ -\sin\theta]^T \quad (13)$$

This relation imposes three constraints (one per Cartesian coordinate) on the fiducial reference points' relative positions in world coordinates, involving two Euler angles. The z-axis related **inclination constraint** is especially relevant for us:

$$\Delta Z_w = Z_w^{(2)} - Z_w^{(1)} = -D\sin\theta \quad (14)$$

Since the accelerometers provide an independent measurement of the inclination (roll) angle, as described in Eq. (5), we may write the inclination constraint as:

$$Z_w^{(2)} - Z_w^{(1)} = -Dg_{ax}/g \quad (15)$$

This idea allows a reformulation of the inclination constraint, building a bridge between the accelerometer measurements and the locations of the two reference points. Combining Eq. (15) and Eq. (10), we obtain the following relation between $\lambda^{(1)}$ and $\lambda^{(2)}$ parameters:

$$\lambda^{(1)} = \frac{Dg_{ax}}{Z_t^{(1)}g} + \frac{\lambda^{(2)}Z_t^{(2)}}{Z_t^{(1)}} \quad (16)$$

We may substitute $\lambda^{(1)}$ in Eq. (11) and rearrange it, obtaining a quadratic equation:

$$a\lambda^{(2)^2} + b\lambda^{(2)} + c = 0 \quad (17)$$

where

$$a = \left(\frac{Z_t^{(2)}}{Z_t^{(1)}}Y_t^{(1)} - Y_t^{(2)}\right)^2 + \left(\frac{Z_t^{(2)}}{Z_t^{(1)}}X_t^{(1)} - X_t^{(2)}\right)^2,$$

$$b = \frac{2Dg_{ax}}{Z_t^{(1)^2}g}\left[Z_t^{(2)}\left(X_t^{(1)^2} + Y_t^{(1)^2}\right) - Z_t^{(1)}\left(X_t^{(1)}X_t^{(2)} + Y_t^{(1)}Y_t^{(2)}\right)\right],$$

$$c = D^2\left[\left(\frac{g_{ax}}{gZ_t^{(1)}}\right)^2\left(X_t^{(1)^2} + Y_t^{(1)^2} + Z_t^{(1)^2}\right) - 1\right].$$

This equation yields two solutions, but only one (the positive) is physically feasible. Once we get the value of $\lambda^{(2)}$, $\lambda^{(1)}$ can be obtained using Eq. (16), and 3D positions of both reference points are easily calculated with Eq. (10).

4.3. Complete 6-DoF pose estimation

Both the monocular and stereo vision system provide the positions of both fiducial reference points expressed in world coordinates ($\mathbf{P}_w^{(1)}$, $\mathbf{P}_w^{(2)}$), and at the same time accelerometers are able to give pitch and roll estimation, as shown in Section 3.2 (Eq. (5)).

Due to the symmetrical design of the fiducial, the position of its central point is considered as the position of the object. To obtain the remaining Euler angle (yaw), we may exploit the remaining relations in Eq. (13). Dividing the x and y components of this equation we obtain:

$$\phi = \arctan((Y_w^{(2)} - Y_w^{(1)})/(X_w^{(2)} - X_w^{(1)})) \tag{18}$$

Summarizing, the 6-DoF object pose estimate results, from gravity measurements and reference points' 3D positions:

$$\mathbf{s} = \begin{bmatrix} x \\ y \\ z \\ \psi \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} (X_w^{(1)} + X_w^{(2)})/2 \\ (Y_w^{(1)} + Y_w^{(2)})/2 \\ (Z_w^{(1)} + Z_w^{(2)})/2 \\ \arctan(g_{ay}/g_{az}) \\ \arcsin(g_{az}/g) \\ \arctan((Y_w^{(2)} - Y_w^{(1)})/(X_w^{(2)} - X_w^{(1)})) \end{bmatrix} \tag{19}$$

5. System error modeling

In this section we will propose a procedure for the system error modeling, both for the monocular and for the stereo vision system. To develop the complete error model, Section 5.1 focuses on the accelerometer error modeling, while Section 5.2 describes a model of the reference point estimation error. Section 5.3 introduces a complete model of the pose estimation error based on the propagation of the previously described errors.

The modeling of the whole pose error distribution would be extremely complex due to the presence of different error sources in the input error terms and to the different weighting of those terms by the uncertainty propagation model. Therefore, in the following, we will just focus on the modeling of the first and second order statistics of this error. In other words, we are just assessing biases (error mean value), and error covariance matrices.

5.1. Accelerometer measurement error model

Accelerometers suffer from various error sources [45]. A simple model of the measurement of an accelerometer in the i-axis $\tilde{a}_i$, $(i = x, y, z)$ can be expressed:

$$\tilde{a}_i = a_i + S_i a_i + b_i + n_i \tag{20}$$

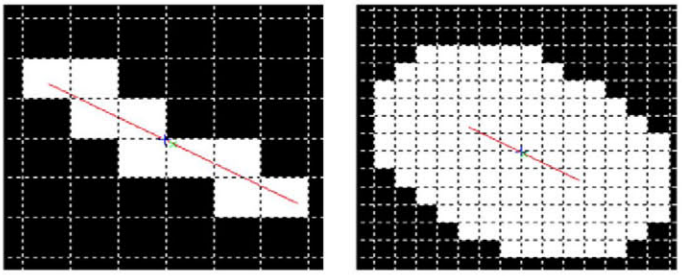
where S_i is a scale factor, a_i is the actual (true) acceleration along i-axis, b_i is the measurement bias and n_i is the random noise.

In our system, the accelerometers are just used to measure the gravity. With uncalibrated accelerometers, gravity sensing will be biased, and therefore we will have errors in the pose estimation, as will be described in Section 5.3. In order to calibrate the three-axis accelerometer, the Six-Position Static Test method [45] is used due to its simplicity and popularity. This method requires the accelerometer to be mounted on a leveled surface with each sensitive axis pointing alternately up and down. It can be easily done in real settings. Calculated bias and scale factors are used to correct the original measurements, so that in the rest of the system we use the calibrated (almost unbiased) gravity measurements.

Additionally, we estimated the noise standard deviations of accelerometer measurements from the samples used for calibration, resulting values equal to $\sigma_{ax} = 4.5 \times 10^{-3}$ m/s^2, $\sigma_{ay} = 4.3 \times 10^{-3}$ m/s^2, $\sigma_{az} = 4.6 \times 10^{-3}$ m/s^2 in our deployment.

5.2. Reference point position estimation error model

The errors in the estimation of reference points' coordinates (u, v) are due to image acquisition and processing, and to the algorithm used to estimate the reference points. Several factors affect the image acquisition and reference points extraction process, such as sensor spatial and intensity quantization, image noise or light flickering. A more detailed explanation of those effects can be found in [46,47]. In this section we will provide quantitative estimations of the estimation error due to the proposed image-processing algorithm.



(a) (b)

Fig. 4. (a) A line segment expressed in the image after quantization. (b) An example of line segment after dilation.

In our early experiments, it became evident that different sizes and orientations of the fiducial projection in the image resulted in different errors. From this basic idea, we performed a simulation to assess this error. In this simulation we assumed that the space quantification due to image resolution was the dominant error source, and discarded all other error sources. The ideal reference point is the center of the border segment between the different color rectangular areas in the fiducial. The measured reference point, extracted using the process in Section 3.3.1, is almost equivalent to the centroid of an area which can be obtained dilating the border segment projection in the image with the kernel described in step 5 of Algorithm 1, as can be seen in Fig. 4b. In Fig. 4a the border segment projected in image before dilation may be seen.

Depending on the border segment size and orientation, the error, defined as the difference between the estimated centroid and the ideal reference point projection, is different. In our real scenario experiments (with a camera resolution of 640 × 360 pixels) the shared edge covers between two pixels (fiducial far away from the camera, 4.5m) and eight pixels (fiducial very close to the camera, 30cm). Therefore, we model the projection of the border segment as a segment with length (L) from 2 to 8 pixels and inclination (θ, with respect to horizontal) from 0 to 180 degrees (note the image symmetry allows us to avoid modeling the angles between −180 and 0). The error is then measured as the difference between the centroid of the dilated region and the center of the line segment. In order to estimate the average error and the standard deviation of the error, we define a very fine grid (10 × 10 samples) for the position of the ideal center within a pixel. Then, we calculate the average value and the standard deviation of the centroid estimation from results for the different center positions in the grid, for a given border length and a given angle with respect to the horizontal. Algorithm 2 is an implementation of the previously described process.

Results are shown in Fig. 5. Due to problem symmetry, the mean error in both u-axis and v-axis (in Fig. 5a and Fig. 5b) is zero. The standard deviation of u-axis and v-axis, expressed in pixels, are shown in Fig. 5c and Fig. 5d. It is quite a complex function of border projection length and inclination.

From this result we developed an analytical model interpolating the simulated deviation. The basic idea behind this analytical model was realizing it had a distinct "periodical" pattern, due to the quantization effect. In a previous research for a somehow similar problem [48], a model for one dimensional (let us call it u) centroid estimation standard deviation was shown to be:

$$\sigma_u = \sqrt{(1 - 3 < L_u > + 3 < L_u >^2)/12} \tag{21}$$

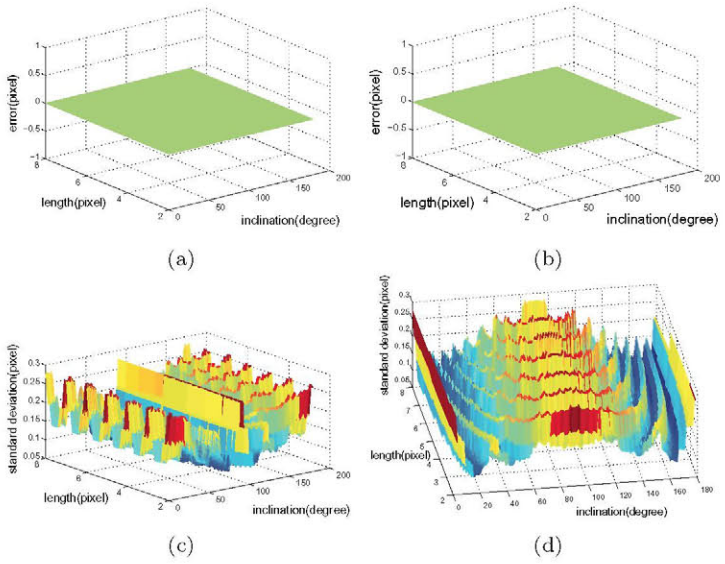


Fig. 5. Error modeling result. (a) Error in u-axis. (b) Error in v-axis. (c) Standard deviation in u-axis. (d) Standard deviation in v-axis.

Algorithm 2 Reference point position estimation error modeling

1: Create a binary image of 30×30 pixel with all the elements set to zero.
2: Run the loop, where L is the length of the line segment and θ is the angle between the line segment and the horizontal u axis.
3: **for** $L = 2; L \leq 8; L = L + 0.1$ **do**
4: **for** $\theta = 1/\pi; \theta \leq \pi; \theta = \theta + 1/\pi$ **do**
5: **for** $C_u = 14.55; C_u \leq 15.45; C_u = C_u + 0.1$ **do**
6: **for** $C_v = 14.55; C_v \leq 15.45; C_v = C_v + 0.1$ **do**
7: $(u^{(1)}, v^{(1)}) = (C_u - 0.5L\cos\theta, C_v - 0.5L\sin\theta)$;
8: $(u^{(2)}, v^{(2)}) = (C_u + 0.5L\cos\theta, C_v + 0.5L\sin\theta)$;
9: quantized_image = quantize the line segment
10: dilated_image = dilate the quantized_image
11: centroid_blob = find the centroid of the dilated_image
12: error = centroid_blob $-[C_u, C_v]$
13: **end for**
14: **end for**
15: **end for**
16: **end for**
17: Save 2D error vector for later analysis in an error matrix
18: Rearrange all the error we get from each loop.
19: Analyze the data statistic of the error varying from segment length and inclination, averaging results of all C_u and C_v values.

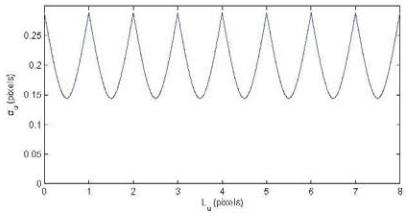


Fig. 6. Standard deviation (in pixels) for 1D quantification.

where L_u is the ideal length in pixels of the 1D image, and operator $< >$ denotes the fractional part of the real number L_u.

This periodical function is shown in Fig. 6, with maximum values equal to $1/\sqrt{12}$ pixels for integer L_u values, and minimum values equal to $1/(2\sqrt{12})$ pixels for values of L_u with fractional part equal to 0.5.

Any dilation on the basic blob has no effect on the centroid estimation error. In our 2D problem the extension of our blob along u-axis can be calculated through the projection of the segment border length (called L previously) along this axis. Additionally, for most inclinations, we are averaging the results in several rows, which results in a reduced error, and demands a more complex model. A good interpolation of the standard deviation of u in terms of L and θ may be obtained, finally, as:

$$\sigma_u = \begin{cases} \sqrt{(1 - 3 < L|\cos\theta| > + 3 < L|\cos\theta| >^2)/12} & if\ L|\cos\theta| < 1/2\sqrt{12} \\ (1/2 + 1/2|\cos\theta|)\sqrt{(1 - 3 < L|\cos\theta| > + 3 < L|\cos\theta| >^2)/12} & otherwise \end{cases} \tag{22}$$

For v-axis, the same relation appears, substituting $\cos\theta$ by $\sin\theta$. These detailed models might be exploited if we had very good knowledge of L and θ. Once the pose of the fiducial in 3D is calculated, these values, related with projection geometry, might be calculated approximately. But for a rough model of the error we might use a simplified model, applicable in worst case, assuming a standard deviation of the form:

$$\sigma_u = (1/2 + 1/2|\cos\theta|)/\sqrt{12}, \quad \sigma_v = (1/2 + 1/2|\sin\theta|)/\sqrt{12} \tag{23}$$

An even simpler model, in worst case, is assuming constant standard deviations:

$$\sigma_u = 1/\sqrt{12}, \quad \sigma_v = 1/\sqrt{12} \tag{24}$$

The three models in Eq. (22)–(24) might be used at different processing stages. The model in Eq. (22) needs too accurate control of the environment, and will just be difficult to apply in an operational environment. The model in Eq. (23) might be used to calculate covariances in real time, and the model in Eq. (24) is more adequate for worst-case analysis. As we are neglecting other sources of error, we will use Eq. (24) in the following.

An additional relevant conclusion was obtained from our simulation described in Algorithm 2: there is negligible cross covariance between u-axis and v-axis independently of L and θ. Finally, in the following sections we assume the error in the estimation of both

projected reference points are independent. In fact it is not completely true, but it is very complex to model this correlation effect, and we will show in the Section 6 this lack of information is not too important for the overall system accuracy assessment.

5.3. Pose estimation error modeling

In our proposed pose estimation algorithms (stereo and monocular), the output of our system is the estimated pose vector $\mathbf{s} = [x, y, z, \psi, \theta, \phi]^T$. Depending on the available cameras we have different inputs:

(a) For the stereo vision system we have two pairs of reference points projections, coming from each of the cameras: $[u_L^{(1)}, v_L^{(1)}]$ and $[u_L^{(2)}, v_L^{(2)}]$ from the "left" camera, and $[u_R^{(1)}, v_R^{(1)}]$ and $[u_R^{(2)}, v_R^{(2)}]$ from the "right" camera. Summarizing, the measurement vector in this case results: $\mathbf{X}_{in} = [g_{ax}, g_{ay}, g_{az}, u_L^{(1)}, v_L^{(1)}, u_L^{(2)}, v_L^{(2)}, u_R^{(1)}, v_R^{(1)}, u_R^{(2)}, v_R^{(2)}]^T$

(b) For the monocular vision system we have one pair of reference points projections: $[u^{(1)}, v^{(1)}]$ and $[u^{(2)}, v^{(2)}]$. So we may define the measurement vector: $\mathbf{X}_{in} = [g_{ax}, g_{ay}, g_{az}, u^{(1)}, v^{(1)}, u^{(2)}, v^{(2)}]^T$

In both cases, we may summarize the estimation algorithm as a function relating the available measurement vector and the pose estimator:

$$\mathbf{s} = f(\mathbf{X}_{in}) \tag{25}$$

Of course, the function is different for each of the proposed pose estimation approaches. To analyze estimation error we first divide the error sources in two kinds. The first kind is related to the propagation of the errors in the available inputs to the estimator. The second kind is the systematic errors (e.g., imperfect camera calibration), which lead to potential biases in the estimation.

To analyze the propagation of input errors, we relate the errors in the measurements and the errors in the estimation through a first order Taylor approximation of the ideal pose as follows:

$$\mathbf{s} - \Delta\mathbf{s} = f(\mathbf{X}_{in} - \Delta\mathbf{X}_{in}) = f(\mathbf{X}_{in}) - F_X \Delta\mathbf{X}_{in} + \ldots \tag{26}$$

where $\mathbf{s}$ is the estimated pose (with errors), $\Delta\mathbf{s}$ is the estimated pose error, $\mathbf{X}_{in}$ is the measurement vector (with errors), $\Delta\mathbf{X}_{in}$ is the measurement error vector and therefore the term $(\mathbf{X}_{in} - \Delta\mathbf{X}_{in})$ would be an "ideal" measurement without errors and F_X is the Jacobian matrix of function $f(.)$ evaluated at $\mathbf{X}_{in}$. F_X can be calculated analytically from previous derivations or approximated numerically.

Neglecting higher order terms we could therefore relate measurement and pose estimation error as follows:

$$\Delta\mathbf{s} \approx F_X \Delta\mathbf{X}_{in} \tag{27}$$

So the estimated pose error is the result of the propagation of the input error (uncertainty) through a system specific uncertainty propagation function F_X. It should be emphasized that F_X is different for monocular and stereo vision system (even its size is different, it is a 6×7 matrix for the monocular vision system and 6×11 for the stereo vision system).

From Eq. (27), using the expectation operation over errors, we may define the following relation between pose estimation bias ($\mathbf{b}_s$ in the following), and input measurement vector bias (denoted $\mathbf{b}_X$):

$$\mathbf{b}_s \approx F_X \mathbf{b}_X \tag{28}$$

Also, we may define the following approximate relation between measurement error covariance (denoted C_X) and the resulting pose estimation error covariance (C_s):

$$C_s \approx F_X C_X F_X^T \tag{29}$$

Table 2
Errors in position and orientation estimation from different distances and different orientations.

Distance (cm)	Position error (mm)			Orientation error (degree)		
	x axis	y axis	z axis	x axis	y axis	z axis
50	3.2	6.5	2.4	0.6	1.5	1.8
150	9.1	9.1	3.1	0.8	0.4	1.6
250	11.5	39.2	7.4	1.3	1.1	3.7
350	15.3	87.1	6.1	1.5	1.0	3.0

Regarding the covariance matrix, applying Eq. (29) to the stereo vision system results in:

$$C_s = F_X diag(\sigma_{g_{ax}}^2, \sigma_{g_{ay}}^2, \sigma_{g_{az}}^2, \sigma_u^2, \sigma_v^2, \sigma_u^2, \sigma_v^2, \sigma_u^2, \sigma_v^2, \sigma_u^2, \sigma_v^2) F_X^T \tag{30}$$

where $diag(.)$ represents a diagonal matrix (in this case, the size 11×11) whose diagonal elements are listed as parameters, and all standard deviations were introduced in Section 5.1 and Section 5.2. For σ_u and σ_v we will use in general the value predicted in Eq. (24).

Again, applying Eq. (29) to the monocular vision system results in:

$$C_s = F_X diag(\sigma_{g_{ax}}^2, \sigma_{g_{ay}}^2, \sigma_{g_{az}}^2, \sigma_u^2, \sigma_v^2, \sigma_u^2, \sigma_v^2) F_X^T \tag{31}$$

In Section 6 we will use the above just defined models to predict the system performance in a realistic scenario, and we will also use real error measurements to validate this model.

6. Experiments

In this section we will experimentally assess our system accuracy, validate the system error model and calculate the computational load. Section 6.1 includes two methodologies to assess the system accuracy. Section 6.2 validates the error model both for the stereo vision and monocular vision approaches. Additionally, the computational load of both solutions in a currently commercial workstation (HP Z420) is featured in Section 6.3.

6.1. Accuracy assessment

Two methodologies are proposed to evaluate the system accuracy. The first experiment is to benchmark to real measurements in terms of position and orientation estimation. The second experiment compares the proposed stereo vision system to a marker-based system in terms of projection errors of 25 test points.

6.1.1. Accuracy: benchmark to real measurements

In our previous work [12], we evaluated a preliminary version of the proposed stereo vision system in 16 poses, changing the distance between the tablet and the cameras (50, 150, 250 and 350 cm) along the depth direction and the inclination of the tablet (0°, 45°, 90° and 180°). The real position and orientation were measured by a Laser Distance Meter and a goniometer. The results are summarized in Table 2 by averaging errors for all the inclinations in each distance.

In this paper, we have implemented another experiment with different camera setup and assessed the accuracy of both the stereo vision system and the monocular vision system. We have used two Logitech HD Pro webcam C920 cameras. They are deployed in the ceiling of our laboratory to have a bird view of the scene, as depicted in Fig. 7a. Examples of captured images from the left and right camera are shown in Fig. 7b and Fig. 7c. The resolution of the image is set to 640×360 pixels and the cameras run at 20 frames/s. The fiducial, whose size is 19×1 cm, is tagged on the upper border of a Nexus 10 tablet, which is held in landscape mode. The image processing and pose estimation process

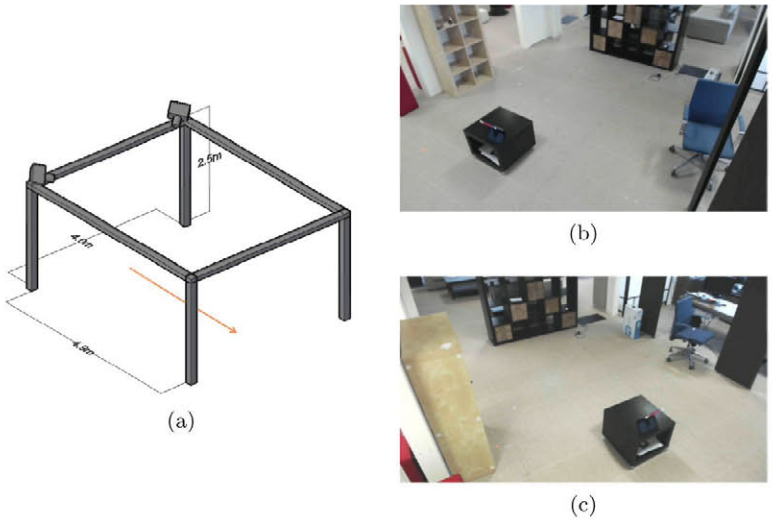


Fig. 7. (a) Deployment of cameras in our lab. (b) A captured image from the left camera. (c) A captured image from the right camera.

Table 3
RMS of estimated position and rotation error in stereo vision system and monocular vision system.

Distance (cm)	Stereo vision system		Monocular vision system	
	Position error (mm)	Orientation error (degree)	Position error (mm)	Orientation error (degree)
113	8.8	0.4	48.3	1.4
163	2.6	1.3	74.3	0.7
213	13.2	0.7	79.4	2.6
263	15.2	1.7	112.2	1.7
313	17.1	3.1	212.6	1.6
363	19.0	2.7	136.1	2.3
413	28.9	4.2	275.4	2.6

are performed in real time in an HP Z420 workstation. Cameras and accelerometer are calibrated offline in advance, as described in Section 3.3.2 and Section 5.1.

On this infrastructure, we have performed static accuracy measurements in seven positions, centered between both cameras, and at increasing distance from the wall next to both cameras, as indicated by the orange line in Fig. 7a. The distance to the wall varies from 113 cm to 413 cm and tests have been performed every 50cm. The tablet is placed in a static stand at a predefined pitch angle (50°), measured by a goniometer with 0.1° resolution. The distance has been measured using a Laser Distance Meter with high accuracy. At each position, 50 images from left and right cameras have been captured and corresponding accelerometer measurements have been transmitted to the central workstation, which processes the images and locates the reference points in the captured images. Then, we have compared the estimated position and rotation accuracy with the ground truth and the results of the root-mean-square (RMS) errors are shown in Table 3.

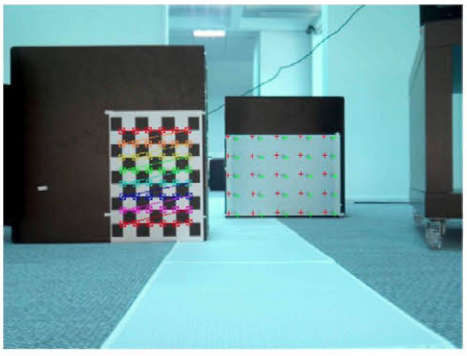


Fig. 8. A sample image showing the results of the two pose estimation systems. The green and red crosses correspond to the projections of the 25 test points using the results from the proposed stereo vision system and the reference system, respectively. The 6 × 8, 30 × 30 mm chessboard pattern is used by the reference system to estimate the device's 3D pose.

6.1.2. Projection error: benchmark to OpenCV

Additionally, we have designed an experiment to compare our proposed stereo vision system to the accurate marker-based camera pose method by OpenCV, an open source, widely-used and cross-platform computer vision library. OpenCV finds the position of internal corners of a chessboard using the function *findChessboardConers()* and then finds an object pose from 3D-2D point correspondences using the function *solvePnP()*. Based on the estimated device pose from the proposed system and OpenCV, 25 test points with known world coordinates are projected back to the image captured by the device camera. Then, the comparison of two systems is done by calculating the mean projection error of the 25 test points, understood as the image distance between the projected points and the real points in the device camera image.

As we mentioned before, the markers used in OpenCV need a big and flat surface to be placed. Therefore, they are not suitable to track small objects, such as a mobile device. The reference system is accomplished by tracking a marker deployed in the environment from the internal device camera. To set the benchmark up, a chessboard composed of 6 × 8 grids of 30mm × 30mm (overall size 180mm × 240mm) has been used as an external marker and detected by the internal device camera. 25 test points have been provided by a 4 × 4 grid with a total dimension of 360 mm × 280 mm. An example of the experiment is shown in Fig. 8. The experiment has been carried out with different measurement

Table 4
Errors of the two considered pose estimation systems.

Measurement distance (cm)	Observation distance (cm)	Proposed system error (pixel)	Marker-based reference system error (pixel)
100	50	7.4 ± 2.7	19.7 ± 7.0
100	150	4.0 ± 0.6	8.9 ± 0.6
100	250	15.2 ± 0.3	12.4 ± 0.4
100	350	9.7 ± 0.4	5.1 ± 0.2
200	50	18.3 ± 5.4	17.4 ± 31.0
200	150	12.9 ± 1.5	8.9 ± 0.6
200	250	12.0 ± 0.3	12.4 ± 0.4
200	350	11.6 ± 0.3	5.1 ± 0.5
300	50	10.0 ± 4.8	N/A
300	150	17.6 ± 6.5	N/A
300	250	15.3 ± 1.0	N/A
300	350	4.4 ± 0.6	N/A
Average		11.5 ± 2.0	10.3 ± 5.1

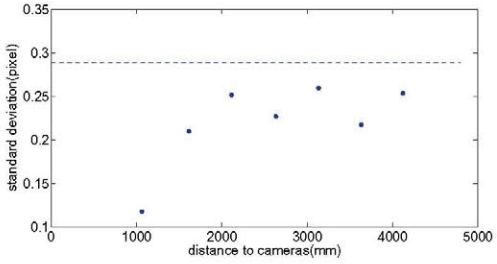


Fig. 9. Standard deviation of pixel measurements from different distances to the cameras.

distances (distance between the camera and the marker) and observation distances (distance between the test points and the tablet camera). The results are shown in Table 4. More details about this experiment can be found in [12].

Both systems have an average error around 10 pixels. The proposed system has lower variance than the reference system, thus gaining in stability. An important feature is that, when the tablet is 3m away from the cameras and the chessboard, the reference system stops working, whereas the proposed system still works providing consistent accuracy (11.8 pixels in average).

6.2. System error model validation

The standard deviations of the reference point position estimation results have been calculated and compared with the error model statistics as described in Section 5.2. In Fig. 9, the horizontal line corresponds to the standard deviation predicted by our input error models (Eq. (24) for reference point measurements). We can see that the standard deviations of the error are similar to those in the model, which tends to overestimate them. This is expected from the error model discussion and derivation in Section 5.2.

The system then calculates the pose vector based on the estimated reference points and sampled accelerometer measurements. In each position, the standard deviation of pose is estimated from 50 measurements, and the results are depicted in Fig. 10a, Fig. 10b for the stereo vision system and Fig. 11a, Fig. 11b for the monocular vision system (using the left camera in our lab deployment). Meanwhile, corresponding standard deviations predicted by the error model are presented in Fig. 11c, Fig. 11d for the stereo vision system and Fig. 11c, Fig. 11d for the monocular vision system. We can find that the curve of experimental results is consistent with the standard deviations predicted by the error model, both for the stereo vision system and the monocular vision system. If we compare Fig. 10 and Fig. 11, we can see that the stereo vision system is much more stable (expected errors in few mm instead of few cm level).

Table 5
Execution time of the system.

Function	Mean (ms)	Min (ms)	Max (ms)
Vision task	96.34	21	223
Pose computation	0.12	0.01	2
Total time	96.46	21.03	230

This model is important for system stability assessment, as it analyzes the propagation of the uncertainty of the accelerometer and the reference point detection to the uncertainty of pose estimation. It can also be used for post-processing, for example, to calculate the covariance of noises in a Kalman filter. However, it is not enough to estimate the bias due to some systematic factors, such as imperfect camera calibration.

6.3. Computational load assessment

To assess the computational load, we carried out an online experiment in which we used the proposed system to continuously calculate the pose of a moving tablet during ten minutes, using a prototype implementation of the described algorithms in a mid-range HP Z420 workstation (with a quad-core Intel® Xeon® CPU E5-1620 @ 3.60GHz and NVIDIA Quadro 4000 GPU). We collected the duration of the vision task and of the pose computation separately, as listed in Table 5. On average, the prototype system is able to update the pose estimation with a rate of 10 times per second, which fulfills the real-time requirement of most applications.

7. Pointing applications

The proposed pose estimation system has potential in several application fields such as indoor AR, person tracking, robot localization and pointing-related applications. In this section, we take pointing as an extended application example and try to model the accuracy of our proposal for it. Let us assume the system is applied to estimate the pose of a pointing device, which could be a pen or a mobile device. We are interested in the stability of the system. In other words, how the uncertainty of the estimated pose from Section 6 will affect the estimation of the projected point (a point/target located in the wall, towards which the device is pointing at). The performance depends on the geometry relationship among the cameras, the device and the target. Here we make some assumptions. The first one is that the pointing direction starts from the center of the fiducial and is aligned with the negative z-axis of the accelerometer/object coordinate system. A unit vector can be mathematically expressed as $\mathbf{v}_a = [0, 0, -1]^T$. Secondly, we assume that the targets (projected points) are located in the wall, where

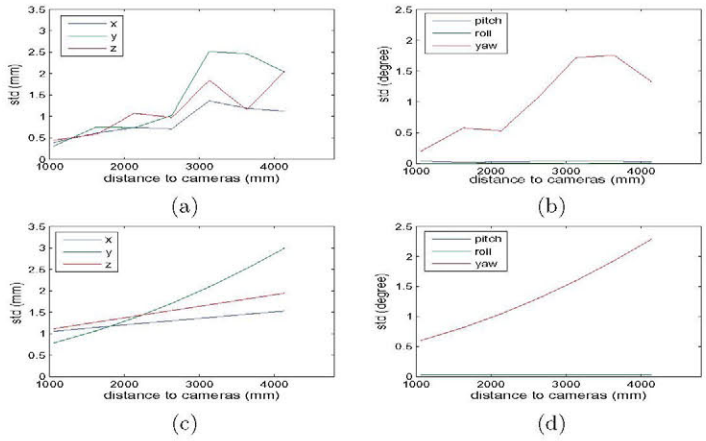


Fig. 10. Comparison of position and Euler angles between experimental results and predicted results in the stereo vision system. (a) Experimental position standard deviation. (b) Experimental rotation standard deviation. (c) Predicted position standard deviation. (d) Predicted rotation standard deviation.

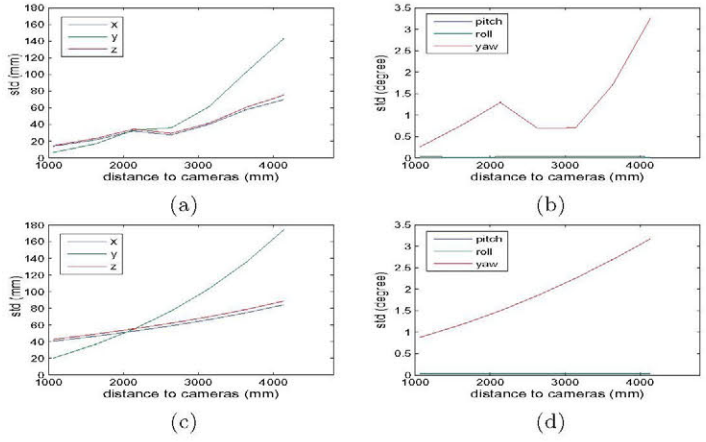


Fig. 11. Comparison of position and Euler angles between experimental results and predicted results in the monocular vision system. (a) Experimental position standard deviation. (b) Experimental rotation standard deviation. (c) Predicted position standard deviation. (d) Predicted rotation standard deviation.

the cameras are deployed. In our case, they have $Y_w = 0$. All this is done on the experimental setup from Section 6.1.1.

Let us denote $P_w = [X_w, 0, Z_w]^T$ as the position of the projected point in the world coordinate system. With the knowledge of the estimated pose vector $\mathbf{s} = [x, y, z, \psi, \theta, \phi]$, we can obtain the vector from the center of the fiducial to the projected point as:

$$\mathbf{v}_w = \mathbf{P}_w - [x, y, z]^T \tag{32}$$

The relationship between $\mathbf{v}_a$ and $\mathbf{v}_w$ can be expressed as:

$$\mathbf{v}_w = \lambda R_{wa}^{-1}\mathbf{v}_a = \lambda \begin{bmatrix} -cos\psi sin\theta cos\phi - sin\psi sin\phi \\ -cos\psi sin\theta sin\phi + sin\psi cos\phi \\ -cos\psi cos\theta \end{bmatrix} \tag{33}$$

where λ is a scale factor.

Therefore, we can get the position of the projected point as:

$$\mathbf{P}_w = \left[x + \frac{cos\psi sin\theta cos\phi + sin\psi sin\phi}{sin\psi cos\phi - cos\psi sin\theta sin\phi} y, \quad 0, \right.$$
$$\left. z + \frac{cos\psi cos\theta}{sin\psi cos\phi - cos\psi sin\theta sin\phi} y \right]^T \tag{34}$$

We may summarize the relationship between the pose vector $\mathbf{s}$ and the projected point as a function

$$\mathbf{P}_w = f_p(\mathbf{s}) \tag{35}$$

The standard deviation of the projected point in both stereo vision system and monocular vision system in x-axis and z-axis is depicted in Fig. 12. It was calculated using the error propagation procedure described in Section 5.3 and the test data from the 7 position static test in Section 6.1.1. We find that the small error in pose estimation is magnified in the projected point, as expected. In the stereo vision system, the error is still kept low (less than 10cm about 4m away). This level of accuracy is acceptable for pointing applications as long as the objects are placed with a larger separation than the error. In the monocular vision system, the error is bigger. However, it can still be used for applications without high accuracy requirements, with lower cost and simpler infrastructure deployment. An example of the proposed system applied to control the lamps by pointing is shown in Fig. 13.

8. Conclusions and future work

In this article, we have proposed a hybrid pose estimation approach based on a colored fiducial, mobile accelerometers, and multi-sensor data fusion techniques. Two different fusion approaches for the pose estimation have been proposed, one based on stereo vision and the other one based on monocular vision. The experimental results show that the proposed system has achieved an accuracy in the order of centimeters for the position estimation and few degrees for the orientation estimation, providing measurements in real-time. We have also proposed error models for both methods, and validated them experimentally. As previously underlined, the system is built on inexpensive cameras (webcams), low-cost accelerometers which are typically embedded in tablets,

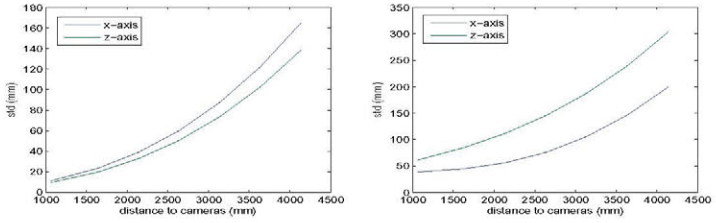


Fig. 12. (a) Projected point standard deviation in the stereo vision system. (b) Projected point standard deviation in the monocular vision system.

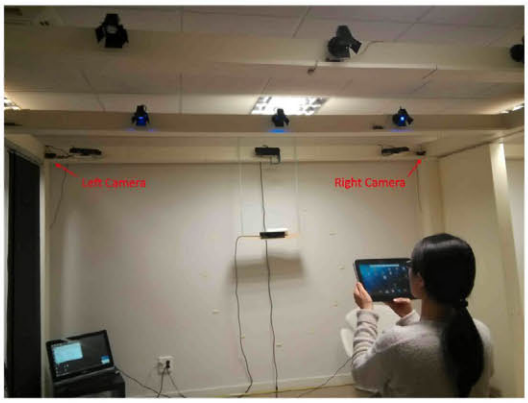


Fig. 13. An pointing application example.

smartphones or wearable devices, and a colored paper marker, being an attractive option in terms of cost. Specially, it does not need a dense camera deployment, as pose estimation based on a single camera (with reduced performance) is still possible. Additionally, the system is easy to deploy, requiring a minimum of setup and configuration to get running. The calibration processes (both for the cameras and the device's accelerometer) only have to be done once during the system installation, which can be quickly accomplished. The calibration process would take approximately 5 $\sim$ 10 minutes for each camera with the current calibration tool and two minutes for the accelerometers. All in all, we can claim that the system can provide accurate, real-time and stable results, thus being suitable for most applications in smart spaces.

The proposed pose estimation system has potential in several application fields such as indoor AR (e.g., for museums, retail applications or gaming), person and object tracking (e.g., for warehouse analytics, indoor drone tracking or activity assessment), robot localization (e.g., for industrial applications or autonomous robot navigation) and pointing-related applications (e.g., for interactions in smart spaces or body-controlled user interfaces). Some of those applications would need the system to be enhanced according to the lines to be commented in the next paragraph to be actually applicable.

The two main limitations of the current version of the proposed system are: (a) it is able to track only one user; (b) current implementation of the system has very limited coverage, as it is just based on two cameras. In future work we will extend the proposed system to address those limitations. To address single user limitation, we will explore and maybe combine several potential approaches. For example, devices may be tracked in the 3D space and distinguished by their trajectories. Also, the users' body information (e.g., color histogram) or different marker designs can also be applied to distinguish and identify each user. On the other hand, to address the coverage constraint, an extension of the system to more than two cameras is needed. Related issues, such as multi-camera management to maintain the tracking continuity and potential fusion of redundant information will be studied. Finally, a user-in-motion accuracy assessment of the proposed system will be also performed to validate the system even more thoroughly.

Acknowledgement

This work has been supported by the Spanish Ministry of Economy and Competitiveness under grant TEC2014-55146-R and by the Technical University of Madrid under grant RP150955017. Juan Li acknowledges the China Scholarship Council for her scholarship.

References

[1] N. Kyriakoulis, A. Gasteratos, Color-based monocular visuoinertial 3-d pose estimation of a volant robot, IEEE Trans. Instrum. Meas., 59 (10) (2010) 2706–2715.
[2] M. Faessler, E. Mueggler, K. Schwabe, D. Scaramuzza, A monocular pose estimation system based on infrared leds, in: IEEE International Conference on Robotics and Automation (ICRA), 2014, IEEE, 2014, pp. 907–913.
[3] A.S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, N. Roy, Visual odometry and mapping for autonomous flight using an rgb-d camera, in: International Symposium on Robotics Research (ISRR), 2011, pp. 1–16.
[4] F. Zhou, H.B.-L. Duh, M. Billinghurst, Trends in augmented reality tracking, interaction and display: A review of ten years of ismar, in: Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, IEEE Computer Society, 2008, pp. 193–202.
[5] S. You, U. Neumann, R. Azuma, Hybrid inertial and vision tracking for augmented reality registration, in: Virtual Reality, 1999. Proceedings., IEEE, IEEE, 1999, pp. 260–267.
[6] P. Föckler, T. Zeidler, B. Brombach, E. Bruns, O. Bimber, Phoneguide: museum guidance supported by on-device object recognition on mobile phones, in: Proceedings of the 4th international conference on Mobile and Ubiquitous Multimedia, ACM, 2005, pp. 3–10.
[7] D. Gómez, P. Tarrío, J. Li, A.M. Bernardos, J.R. Casar, Indoor augmented reality based on ultrasound localization systems, in: Highlights on Practical Applications of Agents and Multi-Agent Systems, Springer, 2013, pp. 202–212.
[8] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, D. Schmalstieg, Real-time detection and tracking for augmented reality on mobile phones, IEEE Transactions on Visualization and Computer Graphics, 16 (3) (2010) 355–368.
[9] G. Klein, T. Drummond, Sensor fusion and occlusion refinement for tablet-based ar, in: Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on, IEEE, 2004, pp. 38–47.
[10] ARToolKit, URL: (http://www.hitl.washington.edu/artoolkit/).
[11] M. Fiala, Artag, a fiducial marker system using digital techniques, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 2, IEEE, 2005, pp. 590–596.
[12] J. Li, A.M. Bernardos, P. Tarrío, J.R. Casar, A combined vision-inertial fusion approach for 6-dof object pose estimation, in: Seventh International Conference on Machine Vision (ICMV 2014), International Society for Optics and Photonics, 2015, 944518–944518.
[13] S.Z. Jamal, Tightly coupled gps/ins airborne navigation system, Aerosp. Elec. Syst. Magaz., IEEE 27 (4) (2012) 39–42.
[14] J.P. Rolland, L. Davis, Y. Baillot, A survey of tracking technology for virtual environments, Fundam. Wearable Comput. Augmented Reality 1 (2001) 67–112.
[15] J. Bird, D. Arden, Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors [emerging opportunities for localization and tracking], IEEE Wireless Commun., 18 (2) (2011) 28–35.
[16] H. Martin, J.A. Besada, A.M. Bernardos, E. Metola, J.R. Casar, Simplified pedestrian tracking filters with positioning and foot-mounted inertial sensors, Int. J. Distributed Sensor Netw. 2014 (2014).
[17] E. Olson, Apriltag: A robust and flexible visual fiducial system, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, 2011, pp. 3400–3407.
[18] D. Prochazka, T. Koubek, (2011) Augmented reality implementation methods in mainstream applications, arXiv preprint arXiv:1106.5569
[19] OptiTrack, URL: (https://www.optitrack.com/).
[20] Vicon motion capture system, URL: (http://www.vicon.com/).

[21] Kinect, URL: (https://www.microsoft.com/en-us/kinectforwindows/).
[22] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohi, J. Shotton, S. Hodges, A. Fitzgibbon, Kinectfusion: Real-time dense surface mapping and tracking, in: 10th IEEE International Symposium on Mixed and augmented reality (ISMAR), IEEE, 2011, pp. 127–136.
[23] S. Foix, G. Alenya, C. Torras, Lock-in time-of-flight (tof) cameras: a survey, IEEE Sensors J. 11 (9) (2011) 1917–1926.
[24] J. Kelly, G.S. Sukhatme, Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration, The Int. J. Robot. Res. 30 (1) (2011) 56–79.
[25] G. Ligorio, A.M. Sabatini, Extended kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: comparative analysis and performance evaluation, Sensors 13 (2) (2013) 1919–1941.
[26] K. Satoh, S. Uchiyama, H. Yamamoto, A head tracking method using bird's-eye view camera and gyroscope, in: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality, IEEE Computer Society, 2004, pp. 202–211.
[27] P. Henry, M. Krainin, E. Herbst, X. Ren, D. Fox, Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments, Int. J. Robot. Res. 31 (5) (2012) 647–663.
[28] D. Schmalstieg, D. Wagner, Mobile phones as a platform for augmented reality, Connections 1 (2009) 3.
[29] OpenCV, URL: (https://www.opencv.org).
[30] H. Liu, H. Darabi, P. Banerjee, J. Liu, Survey of wireless indoor positioning techniques and systems, IEEE Trans. Syst., Man, Cybern., Part C Appl. Rev., 37 (6) (2007) 1067–1080.
[31] W. Xiao, W. Ni, Y.K. Toh, Integrated wi-fi fingerprinting and inertial sensing for indoor positioning, in: International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2011, pp. 1–6.
[32] P. Tarrío, A.M. Bernardos, J.R. Casar, Weighted least squares techniques for improved received signal strength based localization, Sensors 11 (9) (2011) 8569–8592.
[33] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int.J. Comput. Vis. 60 (2) (2004) 91–110.
[34] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, in: Computer vision–ECCV 2006, Springer, 2006, pp. 404–417.
[35] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, W. Burgard, An evaluation of the rgb-d slam system, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2012, pp. 1691–1696.
[36] J. Fuentes-Pacheco, J. Ruiz-Ascencio, J.M. Rendón-Mancha, Visual simultaneous localization and mapping: a survey, Artif. Intell. Rev. 43 (1) (2015) 55–81.
[37] J. Diebel, Representing attitude: Euler angles, unit quaternions, and rotation vectors, Matrix 58 (2006) 15–16.
[38] M. Pedley, Tilt sensing using a three-axis accelerometer, Freescale Semiconductor Appl. Note (2013).
[39] V.T. van Hees, L. Gorzelniak, E.C.D. Leon, M. Eder, M. Pias, S. Taherian, U. Ekelund, F. Renström, P.W. Franks, A. Horsch, et al., Separating movement and gravity components in an acceleration signal and implications for the assessment of human daily physical activity, PLoS One 8 (4) (2013) e61691.
[40] M. Piccardi, Background subtraction techniques: a review, in: IEEE international conference on Systems, Man and Cybernetics, 4, IEEE, 2004, pp. 3099–3104.
[41] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge university press, Cambridge, 2003.
[42] Z. Zhang, Flexible camera calibration by viewing a plane from unknown orientations, in: The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999., 1, IEEE, 1999, pp. 666–673.
[43] J. Nocedal, S.J. Wright, Least-Squares Problems, Numerical optimization (2006) 245–269.
[44] R.I. Hartley, P. Sturm, Triangulation, Comput.Vis. Image Understand. 68 (2) (1997) 146–157.
[45] P. Aggarwal, Z. Syed, X. Niu, N. El-Sheimy, A standard testing and calibration procedure for low cost mems inertial sensors and units, J. Navigation 61 (02) (2008) 323–336.
[46] G. Di Leo, C. Liguori, A. Paolillo, Covariance propagation for the uncertainty estimation in stereo vision, IEEE Trans. Instrum. Meas. 60 (5) (2011) 1664–1673.
[47] M. De Santo, C. Liguori, A. Paolillo, A. Pietrosanto, Standard uncertainty evaluation in image-based measurements, Measurement 36 (3) (2004) 347–358.
[48] J. Garcia Herrero, J. Besada Portas, F. Jimenez Rodriguez, J. Corredera, Surface movement radar data processing methods for airport surveillance, IEEE Trans. Aerosp. Elec. Syst., 37 (2) (2001) 563–585.