Inertial-Visual Fusion For Camera Network Calibration

Hadi Aliakbarpour and Jorge Dias
Institute of Systems and Robotics, DEEC, University of Coimbra*
{hadi,jorge}@isr.uc.pt

***Abstract*—This paper proposes a novel technique to calibrate a network of cameras by fusion of inertial-visual data. There is a set of still cameras (*structure*) and one (or more) mobile agent(s) camera in the network. Each camera within the network is assumed to be rigidly coupled with an Inertial Sensor (IS). By fusion of inertial and visual data, it becomes possible to consider a virtual camera beside of each camera within the network, using the concept of *infinite homography*. The mentioned virtual camera is downward-looking , its optical axis is parallel to the gravity and has a horizontal image plane. Taking advantage of the defined virtual cameras, the transformations between cameras are estimated by knowing just the heights of two arbitrary points with respect to one camera within the structure network. The proposed approach is notably fast and it requires a minimum human interaction. Another novelty of this method is its applicability for dynamic moving cameras (robots) in order to calibrate the cameras and consequently localizing the robots, as long as that the two marked points are visible by them.**

***Index Terms*—Sensor fusion, inertial data, Inertial Sensor (IS), camera network, infinite homography, calibration, mobile robot and virtual camera.**

I. INTRODUCTION

Calibrating a camera network is demanding for many applications such as tracking, mobile robotics, 3D reconstruction, Human-Robot Interaction (HRI), human behavior understanding and surveillance. Beriault in [8] proposed a method for multi-camera network calibration for the sake of human gesture monitoring. Chen in [10] introduced a method to estimate epipole under a pure camera translation. Hu and Tan in [14] proposed an approach for depth recovery and affine reconstruction under pure camera translation. In [13] vanishing points are used for camera calibration in a vision system by He and Lei. Svoboda in [20] proposed a method for camera network calibration. Barreto and Daniilidis in [7] investigated the problem of multiple camera calibration and estimation of radial distortion.

The use of IS sensors to accompany compute vision applications is recently attracting attentions of the researchers. Nowadays, IS has become cheaper and more accessible. Thanks to the availability of MEMS cheapsets, there are many handy-phones (smart-phones) which are equipped with this sensor and camera as well. Dias in [11] investigated the cooperation between visual and inertial information. Lobo and Dias [16] proposed an efficient method to estimate the relative pose of a camera and an IS. Mirisola in[17] used a rotation-compensated imagery for the aim of trajectory of an airship by aiding inertial data. Ababsa in [1] proposed a localization method by fusing measurements from inertial and vision sensors. Integration of vision and inertial data for a roadway application is discussed in [19] by Randeniya. Calibration of a laser range finder and a stereo camera using IS is investigated in our former work [5]. In [18] Okatani et al. demonstrated that how the translation of camera between two images can be robustly estimated by using IS. Based on Okatani's work, Labrie and Hebert in [15] showed that how the camera 3D motion recovery can be improved by the using inertial data. In this last paper, the orientation obtained from inertial sensor was exploited in order to accelerate and improve the matching process between wide baseline images.

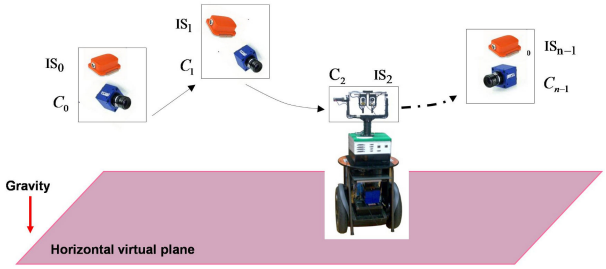


Figure 1. Multi IS-camera Setup: There is a set of still IS-camera couples (structure) and a mobile IS-camera couple (mobile agent) within the network

Our contribution in this paper is to use inertial-visual sensor fusion for the sake of the camera network calibration by using the concept of *infinite homography*. As previously mentioned in the state-of-the-art, the use of inertial sensors can improve the 3D recovery of camera position in two images [15], [18]. Further than just for camera motion recovery in two images, in this paper the inertial data is used in order to calibrate a camera network using just 2-points. There is a set of still cameras (structure) and one (or more) mobile agent camera in the network. Each camera in the network is rigidly coupled to an IS. Such an attached IS makes it possible to consider a virtual camera beside of each camera within the network. The mentioned (fusion-based) virtual camera is downward and has a horizontal image plane. In this method the only needs to calibrate the camera network is to just have the heights of two arbitrary points with respect to (w.r.t) one camera in the structure (still camera). In this approach the transformations among the cameras are estimated in a metric system. Similar to the most of the mentioned camera calibration approaches, our approach also needs an overlap between filed of views (FOV), but with the difference that in our approach observing even

*Hadi Ali Akbarpour is supported by the FCT (Portuguese Fundation for Science and Technology). This work is partially supported by the European Union within the FP7 Project *PROMETHEUS*, www.prometheus-FP7.eu. The authors would like to thank *Amilcar Ferreira, Hugo Faria* and *Kamrad Khoshhal* for their helps in the data collection phase.

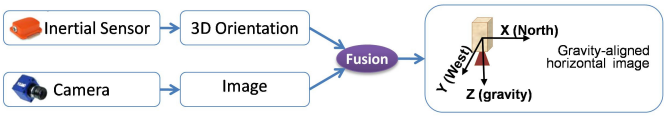


Figure 2. Fusion-based virtual camera.

a thin vertical object (such as a hanged string) is sufficient. Another novelty of this method is its applicability for dynamic moving cameras (robots) in order to calibrate the cameras and consequently localizing the robots, as long as that the two marked points are visible by them.

Apart of applicability of the proposed method to perform the calibration in a structure and mobile camera network, it also has some advantages even for being used for a just structure situations (no movement). For example, some scenes such as train stations most of the time are full of people and therefore there is a difficulty to move LED (used by most existing approaches) or other calibration object through the scene. Moreover in a very large space when the cameras are too far away from each other, it would be complicated to extrinsically calibrate them. The proposed approach is fast and its requirement for human interaction is minimum whereas in most other approaches a calibration object or spot light needs to be moved by a person for a while in the field. In this case (a network of just structure cameras) then just a single frame is sufficient and moreover the image frames between cameras do not need to be synchronized.

As another advantage, in the proposed approach a camera network can be constructed virtually by using just a single couple of IS-camera (can be even an smart-phone), mounted on a robot or held by a person, and placing it in different positions in the scene, which can be useful for some applications such as 3D reconstruction [2], [3]. Moreover, in applications which work based on having homography matrix between camera image and ground plane [2], [4], [6] our method can be useful, specially in some cases that there is a difficulty to find a suitable flat 3D plane in the environment, or if a 3D plane exists then there are not enough distinctive planar features to robustly estimate homography matrix (e.g. in natural scenes).

This article is arranged as following: The geometric models and reference frames are introduced in Sec. II. Definition of fusion-based virtual sensors and estimating the calibration parameters including translations and rotations are discussed in Sec. III. Section V is dedicated to the experiments implemented based on the proposed approach and eventually conclusion is described in Sec. VI.

II. Geometric models and reference frames

In a pinhole camera model, a 3D point $\mathbf{X} = [\ X \quad Y \quad Z \quad 1\ ]^T$ in the scene and its corresponding projection $\mathbf{x} = [\ x \quad y \quad 1\ ]^T$ (both $\mathbf{X}$ and $\mathbf{x}$ are expressed in normalized homogeneous form) are related via a 3×4 matrix P (called *Projection matrix*) through the following equation [12]:

$$\mathbf{x} = P\mathbf{X} \qquad (1) \qquad\qquad , \ P = K\,[R|\mathbf{t}] \qquad (2)$$

where K is the *camera calibration matrix,* R and $\mathbf{t}$ are the rotation matrix and translation vector between world and camera coordinate systems, respectively. The camera matrix K, which is also called *intrinsic parameter matrix,* is defined by [12]:

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

in which f_x and f_y represent the focal length of the camera (in terms of pixel scale) in the directions of x and y. The u_0 and v_0 are the elements of the principal point vector $\mathbf{p}$ [12]. In order to map points from one plane to another plane the concept of Homography [12] is used. Considering a 3D plane is observed by two cameras with $P = [I|0]$ and $P' = [R|\mathbf{t}]$ (concerning first camera center as world reference frame). Also assume that x_1 and x_2 are the image points of a 3D point $\mathbf{X}$ coincided on the 3D plane. Then $\mathbf{x_1}$ and $\mathbf{x_2}$ are called a pair of correspondence points and the relation between them can be expressed as $\mathbf{x_2} = H\,\mathbf{x_1}$ in which H is a 3×3 matrix called *planar homography* induced by the 3D plane [21] and is equal to (up to scale)

$$H = R + \frac{1}{d}\mathbf{t}\mathbf{n}^T \qquad (4)$$

in which R and $\mathbf{t}$ are rotation matrix and translation vector between the two cameras centers, $\mathbf{n}$ is Normal of the 3D plane and d is the orthogonal distance between of 3D plane from the camera center. Applying the camera calibration matrices K and K' and consequently having $P = K\,[I|0]$ and $P' = K'\,[R|\mathbf{t}]$ as camera projection matrices then corresponding equation will become [12]:

$$H = K'\,(R + \frac{1}{d}\mathbf{t}\mathbf{n}^T)\,K^{-1} \qquad (5)$$

For two image points $\mathbf{x_1}, \mathbf{x_2}$ from two different views corresponding to a single 3D point in the space the following relation is true:

$$\mathbf{x}_1^T\,F\,\mathbf{x}_2 = 0 \qquad (6)$$

in which F is called Fundamental matrix [21] and can be computed by having intrinsic and extrinsic parameters of two cameras:

$$F = K_1^{-T}\,T_x R K_2^{-1} \qquad (7)$$

where T_x is the skew-symmetric matrix of translation and R is the rotation between two camera.

A. Reference Frames Definition

Fig. 1 shows a setup with n couples of IS-camera and a horizontal (virtual) world plane. There is a set of still cameras (structure) and a mobile camera (a camera mounted on a mobile robot) in the network. In this setup each camera is rigidly fixed with an IS. Using the orientation given by IS it becomes possible to assume a virtual camera beside of each real camera within the network. Fig. 2 illustrates such a virtual camera. In order to calibrate the network, four different reference frames are involved (see Fig. 3). *Real camera reference frame*$\{C\}$: The local coordinate system of a camera C is expressed as $\{C\}$. *Earth reference frame* $\{E\}$: Which is an earth fixed reference frame having its X axis in the direction of *North*, Y in the direction of *West* and Z upward. *IS local reference frame* $\{IS\}$: This is the local reference frame

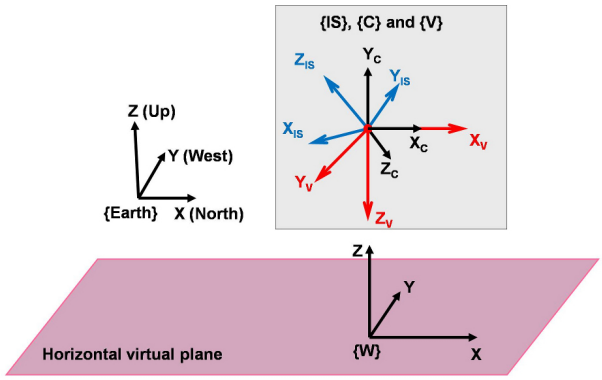


Figure 3. Involved reference frames in the proposed approach and a

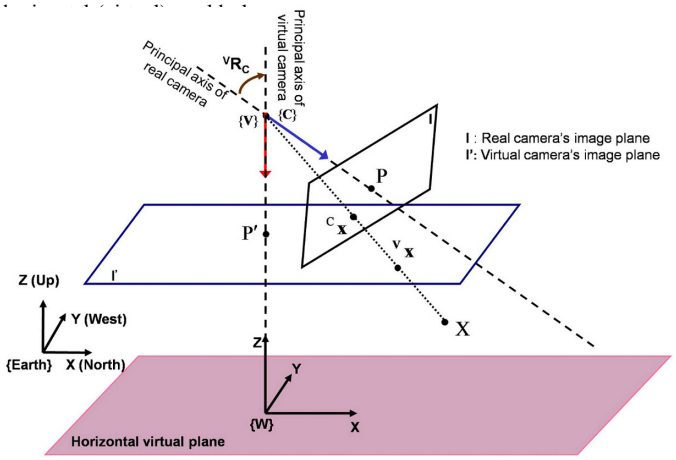


Figure 4. Fusion-based Virtual Camera Using *Infinite Homography*

of an IS which is defined w.r.t. to the earth reference frame $\{E\}$. *Virtual camera reference frame* $\{V\}$: As explained, for each real camera C, a virtual camera V, is considered by the aid of a rigidly coupled IS to that. $\{V\}$ indicates the reference frame of such a virtual camera. The centers of $\{C\}$ and $\{V\}$ coincide and therefore there is just a rotation among these two reference frames.

III. Fusion-based Virtual Camera

The idea of this section is to introduce a fusion-based virtual camera network by which the relative rotation among the virtual cameras are eliminated. A setup of structure (still) and mobile camera network is shown in Fig. 1. We start to explain the method for one camera and then it can be extended for all cameras. Fig. 4 shows the centers, optical axis, image plane and principal points of a real camera C and its corresponding virtual camera V. Here the image plane of real camera and virtual camera are named as I and $I^{'}$, respectively. Based on our assumption V is downward and has optical axis parallel to the gravity vector. Thus $I^{'}$ becomes a horizontal image plane at a distance f below the camera sensor, f being the focal length [17]. In this fashion, the intention is to produce the image plane of the virtual camera by having its corresponding real image plane and the 3D orientation data from IS. In fact, the idea is to register a 3D point such $\mathbf{X}$ onto $I^{'}$. This can be done in two steps. Firstly $\mathbf{X}$ is registered onto I as ${}^{c}\mathbf{x} = K[I|0]\mathbf{X}$. Then the 2D point ${}^{c}\mathbf{x}$ can be reprojected onto $I^{'}$ as follows:

$$ {}^{v}\mathbf{x} = {}^{v}H_{c}\,{}^{c}\mathbf{x} \tag{8} $$

in which ${}^{v}H_{c}$ is a 3×3 homography matrix between I and $I^{'}$. As described before, the real camera C and virtual camera V have their centers coincided to each other, so the transformation between these two cameras can be expressed just by a rotation matrix (see Fig. 4). In this case ${}^{v}H_{c}$ is called *infinite homography* since there is just a pure rotation between real camera and virtual camera centers [17]. Such an infinite homography can be computed using a limiting process on Eq. (5) by considering either $d\rightarrow\infty$ or $\mathbf{t}\rightarrow 0$:

$$ {}^{V}H_{C} = \lim_{d\rightarrow\infty} K({}^{V}R_{C} + \frac{1}{d}\mathbf{t}\mathbf{n}^{T})K^{-1} = K\,{}^{V}R_{C}\,K^{-1} \tag{9} $$

where ${}^{V}R_{C}$ is the rotation matrix between $\{C\}$ and $\{V\}$ [17]. The rotation matrix ${}^{V}R_{C}$ can be computed through three consecutive rotations (see the reference frames in Fig. 3) as follows:

$$ {}^{V}R_{C} = {}^{V}R_{E}\;{}^{E}R_{IS}\;{}^{IS}R_{C} \tag{10} $$

First one ${}^{IS}R_{C}$ is to transform from real camera reference $\{C\}$ to the IS local coordinate $\{IS\}$. The second one ${}^{E}R_{IS}$ transforms from the $\{IS\}$ to the earth fixed reference $\{E\}$ and the last one ${}^{V}R_{E}$ is to transform from $\{E\}$ to virtual camera reference frame $\{V\}$. Here we continue to explain how to compute these three consecutive rotation matrices. In order to estimate the rotation between camera and IS (${}^{IS}R_{C}$) , *Camera Inertial Calibration Toolbox* [16] is used which is a toolbox to calibrate a rigid IS-camera. Rotation from IS to earth ,${}^{E}R_{IS}$, is given by the IS sensor w.r.t $\{E\}$. Since the $\{E\}$ has the Z upward but the virtual camera is supposed to be downward-looking (with a downward Z) then the following rotation is applied to reach to the virtual camera reference frame:

$$ {}^{V}R_{E} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{11} $$

Having ${}^{V}R_{C}$ for each cameras and applying Eq. 9 for any camera within the network, we will have a set of parallel (and horizontal) virtual image planes (thought as the image planes of the fusion-based virtual camera) in such a way that there is no rotation among them. This means that by now we have solved the problem of relative rotation estimation for the cameras. In the next sub-section we proceed to explain a method to estimate the translations among the cameras within the network.

IV. Estimating Translation

In the previous section, it was explained how to reach to a network of virtual (fusion-based) cameras such a way that there is no relative rotation among them. It means that we have reduced the problem of calibration to just a *"pure translation"* case. This section is dedicated to propose a method to estimate the translation between the mentioned virtual cameras. Obviously since the center of each virtual camera is coincided to its corresponding real camera then the translations between virtual cameras set are equal to the real ones. As described before, the only requirement from the scene is to have the heights of two arbitrary 3D points such $\mathbf{X}_1 = [\;X_1 \quad Y_1 \quad Z_1\;]^{T}$ and $\mathbf{X}_2 = [\;X_2 \quad Y_2 \quad Z_2\;]^{T}$ (see Fig.

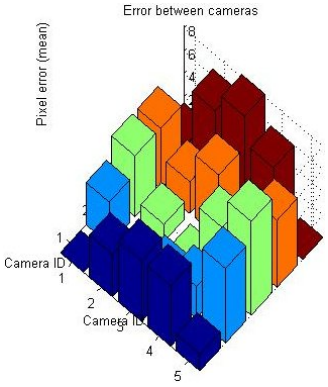


Figure 5. Values of the *mean* errors for the pixel correspondences for each camera pair calculated based on Eq. (7).

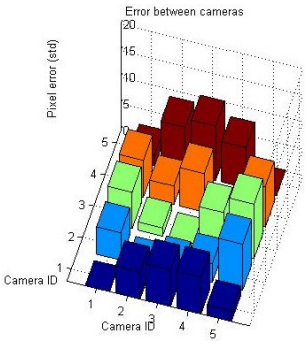


Figure 6. Values of the *std* errors for the pixel correspondences for each camera pair calculated based on Eq. (7).

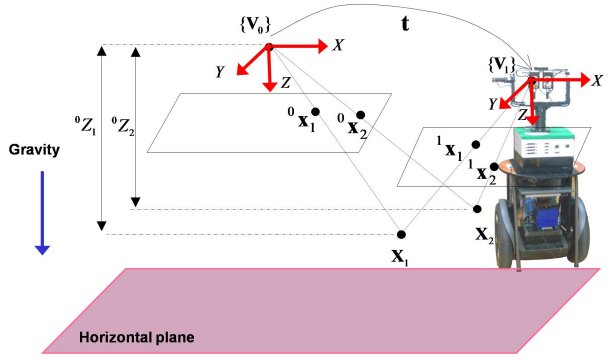
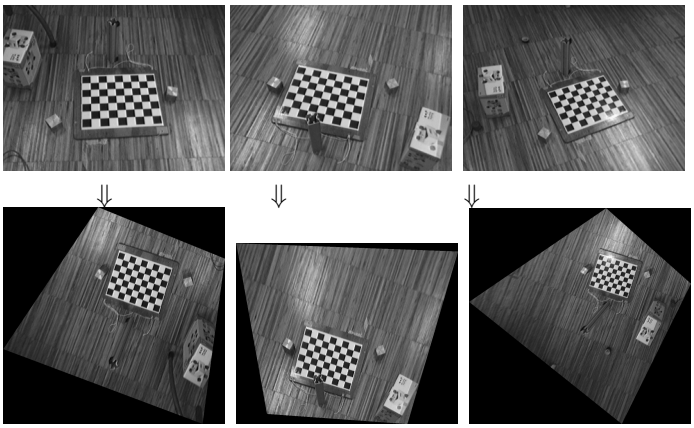


Figure 7. Translation between two virtual cameras (corresponding to two real cameras: first one from the structure camera and the other one here is on a mobile robot).

7) w.r.t one camera (namely V_0) within the network.

Suppose ${}^0\mathbf{X}_1 = [\ {}^0X_1 \quad {}^0Y_1 \quad {}^0Z_1\ ]^T$ and ${}^0\mathbf{X}_2 = [\ {}^0X_2 \quad {}^0Y_2 \quad {}^0Z_2\ ]^T$ are coordinates of the two 3D points $\mathbf{X}_1$ and $\mathbf{X}_2$ expressed in the first virtual camera center, respectively. Based on the assumption, the parameters ${}^0Z_1$ and ${}^0Z_2$ which indicate the heights of $\mathbf{X}_1$ and $\mathbf{X}_2$ in $\{V_0\}$ are known. Recalling that V_0 is downward and has its optical



(d) (e) (f)

Figure 8. Top: Real image planes (of three real cameras within the setup). Bottom: Virtual image planes (calculated from the real images shown in the top) . As can be seen all three virtual images in the bottom seem parallel to the floor and moreover there is no rotation among them. Notice that in this experiment (extrinsic calibration of camera network proposed by this paper) the checkerboard, which can be seen in the figures, is used just for validating the result and not as a calibration pattern.

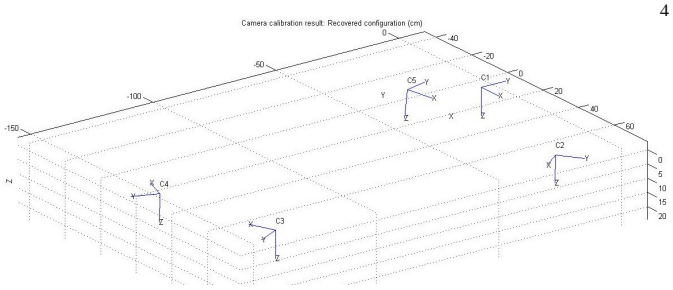


Figure 9. Result of camera calibration: Extrinsic parameters of cameras (expressed in the first camera reference frame).

axis parallel to the gravity. Therefore the term "*height*" here is also implies to the Z component of the 3D point. Then using projective property of a camera we can have all three components of ${}^0\mathbf{X}_1$ and ${}^0\mathbf{X}_2$ numerically obtained in a metric scale using the Eq. (12):

$$\begin{cases} {}^0\mathbf{X}_1 = {}^0Z_1\ (K_1^{-1}\ {}^0\mathbf{x}_1) \\ {}^0\mathbf{X}_2 = {}^0Z_2\ (K_1^{-1}\ {}^0\mathbf{x}_2) \end{cases} \tag{12}$$

where ${}^0\mathbf{x}_1$ and ${}^0\mathbf{x}_2$ are respectively the imaged points of $\mathbf{X}_1$ and $\mathbf{X}_2$ in the first virtual camera image plane. The same can be considered for the second virtual camera. Suppose ${}^1\mathbf{X}_1 = [\ {}^1X_1 \quad {}^1Y_1 \quad {}^1Z_1\ ]^T$ and ${}^1\mathbf{X}_2 = [\ {}^1X_2 \quad {}^1Y_2 \quad {}^1Z_2\ ]^T$ are respectively coordinates of the 3D points $\mathbf{X}_1$ and $\mathbf{X}_2$ expressed in the second virtual camera center ($\{V_1\}$). Then likewise using projective property of a camera we can have the following equation:

$$\begin{cases} {}^1\mathbf{X}_1 = {}^1Z_1\ (K_2^{-1}\ {}^1\mathbf{x}_1) \\ {}^1\mathbf{X}_2 = {}^1Z_2\ (K_2^{-1}\ {}^1\mathbf{x}_2) \end{cases} \tag{13}$$

In contrary to the Eq. (12), Eq. (13) can not be numerically obtained yet, since it has two unknown values for ${}^1Z_1$ and ${}^1Z_2$ (the heights of the 3D points w.r.t $\{V_1\}$). The terms $(K_2^{-1}\ {}^1\mathbf{x}_1)$ and $(K_2^{-1}\ {}^1\mathbf{x}_2)$ in Eq. (13) as well express the 3D position of the points ${}^1\mathbf{X}_1$ and ${}^1\mathbf{X}_2$ however up to scale factors ${}^1Z_1$ and ${}^1Z_2$. Here it is desirable to rewrite the Eq. (13) as the following:

$$\begin{cases} {}^1\mathbf{X}_1 = {}^1Z_1\ {}^1\hat{\mathbf{X}}_1 \\ {}^1\mathbf{X}_2 = {}^1Z_2\ {}^1\hat{\mathbf{X}}_2 \end{cases} \tag{14}$$

where ${}^1\hat{\mathbf{X}}_1 = (K_2^{-1}\ {}^1\mathbf{x}_1)$ and ${}^1\hat{\mathbf{X}}_2 = (K_2^{-1}\ {}^1\mathbf{x}_2)$. Then the Eq. (12) and Eq. (14) can be related through the translation vector between $\{V_0\}$ and $\{V_1\}$ as:

$$\begin{cases} {}^0\mathbf{X}_1 = {}^1\mathbf{X}_1 + \mathbf{t} & = {}^1Z_1\ {}^1\hat{\mathbf{X}}_1 + \mathbf{t} \\ {}^0\mathbf{X}_2 = {}^1\mathbf{X}_2 + \mathbf{t} & = {}^1Z_2\ {}^1\hat{\mathbf{X}}_2 + \mathbf{t} \end{cases} \tag{15}$$

where $\mathbf{t} = (\ t_1 \quad t_2 \quad t_3\ )^T$. In Eq. (15) there are five unknown parameters including ${}^1Z_1$, ${}^1Z_2$, t_1, t_2, t_3. Nevertheless there are also six linear equations which are adequate to obtain the unknowns. In order to estimate the five unknowns Eq. (15) can be arranged in the form of

$$A\,\mathbf{x} = B \tag{16}$$

where

$$A = \begin{bmatrix} {}^1\hat{\mathbf{X}}_1 & \mathbf{0}_{3\times1} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times1} & {}^1\hat{\mathbf{X}}_2 & \mathbf{I}_{3\times3} \end{bmatrix},$$

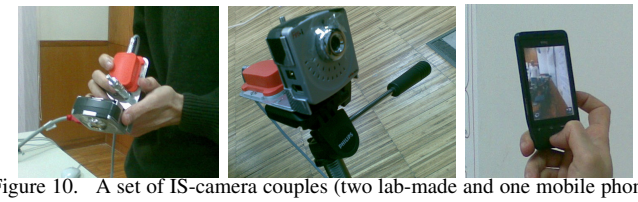


Figure 10. A set of IS-camera couples (two lab-made and one mobile phone)

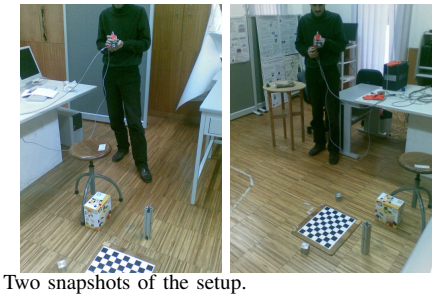


Figure 11. Two snapshots of the setup.

$$\mathbf{x} = \begin{bmatrix} {}^1Z_1 & {}^1Z_2 & t_1 & t_2 & t_3 \end{bmatrix}^T, \quad B = \begin{bmatrix} {}^0\mathbf{X}_1 \\ {}^0\mathbf{X}_2 \end{bmatrix}$$

Therefore $\mathbf{x}$ in Eq. (16) can be estimated using the least square approach as follows:

$$\mathbf{x} = (A^T A)^{-1} A^T B \tag{17}$$

and consequently the translation vector between the two virtual cameras' frames, $\{V_0\}$ and $\{V_1\}$, are estimated. Using the same mentioned method, the translation between all other virtual cameras and $\{V_0\}$ can be estimated.

V. Experiments

In this section the result of a performed experiment is discussed. The algorithm 1 describe the overall steps of the proposed calibration method. Fig. 10 shows a set of IS-camera couples (two lab-made and the other is a hand-held mobile phone equipped with inertial sensor). A MTi-Xsens containing gyroscopes, accelerometers and magnetometers is used as the IS. Firstly the intrinsic parameters of each camera is estimated using Bouguet Camera Calibration Toolbox[9] and then *Camera Inertial Calibration Toolbox* [16] is used for the sake of extrinsic calibration between the camera and IS. Fig. 12 illustrates the scene and Fig. 11 shows two snapshots of the scene. In Fig. 12 as can be seen one couple of IS-camera is used on a tripod, one couple is being carried in the hand of a person and the other one is on the top of a mobile robot.

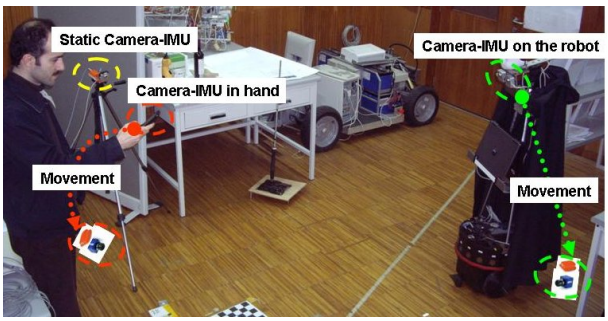


Figure 12. IS-camera network setup: One IS-camera couple is fixed on a tripod, one IS-camera in the hand of a moving person and the the other one on a mobile robot. The trajectories of the movable IS-camera are shown by dotted arrows and their next positions are shown by superimposing on the image.

Algorithm 1 Calibration procedure

Step 1- Obtaining the intrinsic parameters of the camera(s) (by using e.g. Bouguet's method [9]) .

Step 2- Calibrating the couple(s) of IS-camera (relative transformation among the camera and IS in each couple, by using Lobo's method [16]).

Step 3- Marking two 3D points in the scene and measuring just their height w.r.t one still camera.

Step 4- Capturing imagery and IS data for each couple.

Step 5- Performing camera network calibration based on the proposed approach.

Step 6- If there is any movement by the mobile robot(s) then repeating the steps 4 and 5 for the moved ones.

Step 7- End.

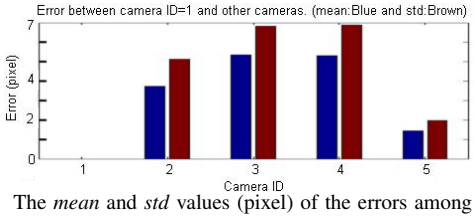


Figure 13. The *mean* and *std* values (pixel) of the errors among the camera ID=1 and the rest of cameras, calculated by Eq. (7).

The trajectories of the person and mobile robot are illustrated by dotted arrow lines. The new positions of moving IS-camera couples after movement are shown by superimposition on the picture. The static and two moving couples with their new two positions have made a network of five cameras. Note that at least one camera should be static (as a structure camera). A simple and thin string, which is visible by all cameras, is hanged in the scene. Two points of the string are marked. Then the relative heights between these two marked points and one still camera (indeed here the camera on the tripod) are measured manually. The relative heights can also be measured using some appropriate devices such as altimeters. Note that these two points do not need to necessarily lie on a vertical line, but since we did not have altimeter available, then we used two points from a vertically hanged string in order to minimize the measuring error.

For the aim of data collection, for each position a pair of image and inertial data is grabbed. As an example Fig. 8-top shows the image planes of three cameras (among five views). Fig. 8 bottom depicts image planes of the virtual cameras corresponding to real ones in the top, respectively. As can be seen in the figure, the virtual image planes are parallel to the horizon by using gravity data. Notice that in this experiment (calibration of camera network proposed by this paper) the checkerboard, which can be seen in the figures, is used just for validating the result and not as a calibration pattern. Then using the proposed approach, the rotations and translations among the cameras within the network are estimated. The recovered positions and orientations of the cameras are shown in Fig. 9.

Then some statistical operations, based on the properties of *Fundamental matrix* (see Eq. 7), are done in order to measure the errors in the result. Fig. 13 shows the mean and std of pixel errors between camera ID=1 and the rest of cameras.

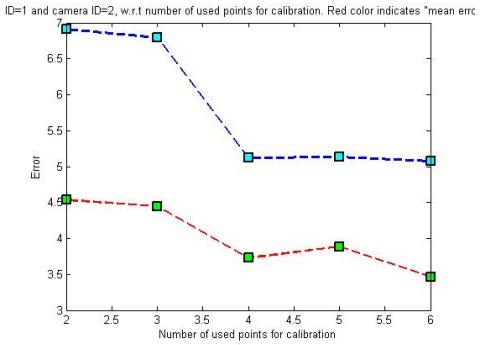


Figure 14. Errors among the camera ID=1 and the rest of the camera network w.r.t number of used points in the calibration process.

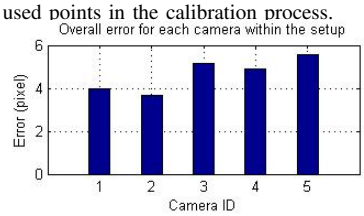


Figure 15. Overall *mean* error of each camera w.r.t the other cameras.

Note that the numbers are calculated by using *the average of the absolute value* of the errors (In Matlab it is: "mean(abs())") expressed by Eq. 7. Then Fig. 5 and Fig. 6 show the pixel errors between all cameras. The experiment is repeated also by using the heights of more than two points, in order to examine the improvement of the result w.r.t number of used points in the calibration process. Fig. 14 depicts the relation between number of used points for the calibration and value of errors. As can be seen with using just 2 points the difference in error value is fairly small. The overall errors for each camera (the sum error between each camera and all the rest) is shown in Fig. 15.

VI. Conclusion

A novel technique to calibrate a network of cameras based on inertial and visual sensor fusion has been proposed in this article. There is a set of still cameras (structure) and one (or more) mobile agent camera within the network and moreover each camera is rigidly coupled to an IS. Then a fusion-based virtual camera is defined for each IS-camera couple. In the proposed method, the only need to calibrate the camera network is to just have the heights of two arbitrary points with respect to (w.r.t) one camera. The experiments show that despite of just using the two points the errors are fairly small. Another novelty of this method is its applicability for dynamic moving cameras (robots) in order to calibrate the cameras and consequently localizing the robot, as long as that the two marked points are visible by them. As future work, the idea is to investigate the fusion of camera-calibration-based localization and robot's odometry in order to increase the accuracy of systems and also eliminating the need of always seeing the two points by the mobile robot. The intention is to continue our investigation in the direction of cloud robotics concept.

References

[1] F. Ababsa. Advanced 3d localization by fusing measurements from gps, inertial and vision sensors. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 871–875, 2009.

[2] Hadi Aliakbarpour and Jorge Dias. Imu-aided 3d reconstruction based on multiple virtual planes. In *DICTA'10 (the Australian Pattern Recognition and Computer Vision Society Conference), IEEE Computer Society Press, 1-3 December 2010, Sydney, Australia.*, 2010.

[3] Hadi Aliakbarpour and Jorge Dias. Multi-resolution virtual plane based 3d reconstruction using inertial-visual data fusion. In *International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP 2011), 5-7 March 2011, Algarve, Portugal.*, 2011.

[4] Hadi Aliakbarpour, J. F. Ferreira, K. Khoshhal, and Jorge Dias. A novel framework for data registration and data fusion in presence of multi-modal sensors. In *in Proceedings of DoCEIS2010- Emerging Trends in Technological Innovation, IFIP AICT 314-2010, Springer.*, volume 314/2010, pages 308–315, 2010.

[5] Hadi Aliakbarpour, Pedro Nunez, Jose Prado, Kamrad Khoshhal, and Jorge Dias. An efficient algorithm for extrinsic calibration between a 3d laser range finder and a stereo camera for surveillance. In *14th International Conference on Advanced Robotics (ICAR 2009)*, 2009.

[6] Dejan Arsic, B. Schuller, and Gerhard Rigoll. Multiple camera person tracking in multiple layers combining 2d and 3d information. 2008.

[7] Joao P. Barreto and Kostas Daniilidis. Wide area multiple camera calibration and estimation of radial distortion. In *Int. Work. on Omnidirectional Vision, Camera Networks and Non-Classical Cameras. Prague, May 2004.*, 2004.

[8] Silvain Beriault, Pierre Payeur, and Gilles Comeau. Flexible multi-camera network calibration for human gesture monitoring. In *ROSE 2007 - IEEE International Workshop on Robotic and Sensors Environments, Ottawa - Canada, 12-13 October 2007.*, 2007.

[9] Jean-Yves Bouguet. Camera calibration toolbox for matlab. In *www.vision.caltech.edu/bouguetj*, 2003.

[10] Zezhi Chen, Nick Pears, John McDermid, and Thomas Heseltine. Epipole estimation under pure camera translation. In *Proc. VIIth Digital Image Computing: Techniques and Applications, Sydney.*, 2003.

[11] Jorge Dias, Jorge Lobo, and Luis A. Almeida. Cooperation between visual and inertial information for 3d vision. In *Proceedings of the 10th Mediterranean Conference on Control and Automation - MED2002 Lisbon, Portugal, July 9-12, 2002.*, 2002.

[12] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. CAMBRIDGE UNIVERSITY PRESS, 2003.

[13] B.W. He and Y.F. Li. Camera calibration from vanishing points in a vision system. *Optics & Laser Technology, Elsevier.*, 40:555–561, 2007.

[14] Zhaozheng Hu and Zheng Tan. Depth recovery and affine reconstruction under camera pure translation. *Pattern Recognition, Elsevier*, 40:2826–2836, 2006.

[15] M. Labrie and P. Hebert. Efficient camera motion and 3d recovery using an inertial sensor. In *Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on*, pages 55 –62, May 2007.

[16] Jorge Lobo and Jorge Dias. Relative pose calibration between visual and inertial sensors. *International Journal of Robotics Research, Special Issue 2nd Workshop on Integration of Vision and Inertial Sensors*, 26:561–575, 2007.

[17] Luiz G. B. Mirisola, Jorge Dias, and A. Traca de Almeida. Trajectory recovery and 3d mapping from rotation-compensated imagery for an airship. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems San Diego, CA, USA, Oct 29 - Nov 2, 2007*, 2007.

[18] T. Okatani and K. Deguchi. Robust estimation of camera translation between two images using a camera with a 3d orientation sensor. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 275 – 278 vol.1, 2002.

[19] Duminda I. B. Randeniya, Sudeep Sarkar, and Manjriker Gunaratne. Vision-imu integration using a slow-frame-rate monocular vision system in an actual roadway setting. *IEEE Transactions on Intelligent Transportation Systems*, 11:256–266, 2010.

[20] T. Svoboda, D. Martienc, and T. Pajdla. A convenient multi-camera self-calibration for virtual environments , vol. 14(4), pp. 407–422, 2004. *PRESENCE: Teleoperators and Virtual Environments, Massachusetts Institute of Technology.*, 14:407–422, 2006.

[21] Jana Kosecka Yi Ma, Stefano Soatta and S. Shankar Sastry. *An invitation to 3D vision*. Springer, 2004.