A New Solution to the Relative Orientation Problem Using Only 3 Points and the Vertical Direction

Mahzad Kalantari, Amir Hashemi, Franck Jung, Jean-Pierre Guédon

J Math Imaging Vis (2011) 39: 259–268
DOI 10.1007/s10851-010-0234-2

A New Solution to the Relative Orientation Problem Using Only 3 Points and the Vertical Direction

Mahzad Kalantari · Amir Hashemi · Franck Jung · Jean-Pierre Guedon

Published online: 9 November 2010

Abstract This paper presents a new method to solve the relative pose between two images, using three pairs of homologous points and the knowledge of the vertical direction. The vertical direction can be determined in two ways: The first requires direct physical measurements such as the ones provided by an IMU (inertial measurement unit). The other uses the automatic extraction of the vanishing point corresponding to the vertical direction in an image. This knowledge of the vertical direction solves two unknowns among the three parameters of the relative rotation, so that only three homologous couples of points are requested to position a couple of images. Rewriting the coplanarity equations thus leads to a much simpler solution. The remaining unknowns resolution is performed by "hiding a variable" approach. The elements necessary to build a specific algebraic solver are given in this paper, allowing for a real-time implementation. The results on real and synthetic data show the efficiency of this method.

Keywords Relative orientation · Vertical direction · Minimal solution

M. Kalantari (✉)
Ecole Nationale des Sciences Géographiques, IGN France, Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), UMR CNRS 6597, Polytech'Nantes, Nantes, France
e-mail: mahzad.kalantari@ensg.eu

A. Hashemi
Department of Mathematical Sciences, Isfahan University of Technology, Isfahan 84156-83111, Iran
e-mail: amir.hashemi@cc.iut.ac.ir

F. Jung
Commissariat général au développement durable, Direction de la recherche et de l'innovation, Ministère de l'écologie, de l'énergie, du développement durable et de la mer, en charge des technologies vertes et des négociations sur le climat, La Défense, France
e-mail: franck.jung@developpement-durable.gouv.fr

J.-P. Guedon
Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), UMR CNRS 6597, Polytech'Nantes, Nantes, France
e-mail: jean-pierre.guedon@polytech.univ-nantes.fr

1 Introduction

This paper presents an efficient solution to solve the relative orientation problem in calibration setting. In such a situation, the intrinsic parameters of the camera, e.g. the focal length, the camera distortion are assumed to be a priori known. In this case, the relative orientation linking two views is modeled by 5 unknowns: the rotation matrix (3 unknowns) and the translation (2 unknowns up to a scale). Its resolution using only five points, in a direct and fast way, has been considered as a major research subject since the eighties [16] up to now [3, 10, 11, 15, 21, 23].

In this paper we use the knowledge of the vertical direction to solve the relative orientation problem for two following reasons:

- The use of MEMS[1]-IMU (inertial measurement unit) in electronic personal devices such as smart phones and digital cameras is continuously increasing, and the prices of IMUs is decreasing as well. The camera-IMU sensors fusion is not the topics of this paper, as many authors have shown the advantage of coupling them [12]. In MEMS-IMU the accuracy of heading (rotation around the vertical

[1] Microelectromechanical systems.

axis Z) is worse than for pitch (rotation around X axis) and roll (rotation around Y axis), due to the strength of the gravity field, which has no effect on a rotation around the vertical axis. Thus the new method presented in this paper takes a considerable benefit from a combination of data from MEMS-IMU and from use of 3 homologous points, that strengthen the very weakness of IMU data.

- Today very performant algorithms based on image analysis are available, that allow to calculate the vertical direction with a high accuracy. If we have a calibrated image we can also determine the vertical direction using automatic vanishing points extraction. A lot of algorithms [2, 14, 19], on such topics exist in the literature. These algorithms are very useful in urban and man-made environments [1, 9, 17, 24].

The use of the vertical direction so as to reduce the disparity between two frames, to simplify 3D vision, has already been considered by [25]. But most papers consider a fixed stereoscopic baseline, and here we are supposed to have no information on it. Furthermore, most papers [25] try to solve the problem using iterative methods or non minimal settings (e.g. more than three points).

The main contribution of this paper is to provide an efficient algorithm to estimate the relative orientation using the vertical direction as an external information in the minimal case, using 3 points. Once the vertical direction is defined, we inject this information in relative orientation, based on the so-called coplanarity equation. The knowledge of the vertical direction removes 2 degrees of freedom to the problem of the relative orientation. Therefore it will be enough to have only 3 homologous couples of points to solve for the 3 other unknowns: two parameters of the baseline, as it is up to a scale, and the angle of rotation around the vertical axis. These coplanarity constants can be written as a system of polynomial equations. The possibility to build a solution with only 3 points is an obvious advantage in terms of computation time, in particular when sorting the undesirable solutions by classic robust estimators such as Ransac (RANdom SAmple Consensus) [6]. In Section 5, we show that the new 3-point method provides better accuracy and robustness to noise on relative orientation estimation.

For the resolution of the relative orientation equation, we use the "hiding a variable" method. This approach consists in hiding a variable (that is, in considering one of the variable as a parameter). This yields a new system which is linear for our example. Then, we search the values of this hidden variable for which the initial system has a solution. Finally, replacing any value of the hidden variable in the new system, we can find a solution of the initial system (see [5] for more details).

The paper is organized as follows. In Sect. 2, we present the geometric framework of our system. Section 3 rewrites the coplanarity constraint using the vertical direction knowledge. The resolution of our polynomial system is described in Sect. 4. The assessment of the algorithm in noisy conditions is studied in Sect. 5.1, where the 3-points algorithm is compared to the well known 5-points algorithm. In Sect. 5.2, a comparison with real image database is performed.

2 Coordinate Systems and Geometry Framework

The classical coordinate system of camera (cf. Fig. 1) used in computer vision has been chosen [7]. In this camera system $(X_{cam}, Y_{cam}, Z_{cam})$, the focal plane is at $Z_{cam} = F$, where F is the focal length. Given the calibration matrix K (a 3×3 matrix that includes the information of focal length, skew of the camera, etc.), the view is normalized by transforming all points by the inverse of K, $\hat{m} = K^{-1}m$, in which m is a 2-coordinates point in the image. Thus the new calibration matrix of the view becomes the identity matrix. M is the object point. In the rest of this paper, we suppose that all image 2D-coordinates of the point are normalized. For a stereo system in relative orientation, the center of the world space coordinate system is the optical center C of the left image, with the same directions of axes. The world coordinate system is denoted by (X_w, Y_w, Z_w). In this system the Y_w axis is along the physical vertical of the world space.

3 Using the Vertical Direction Knowledge for Relative Orientation

3.1 Use the IMU Information

If we have of an IMU coupled with the camera, we need only to know the rotation angle (α) around X and Z axes (γ), based on our coordinates system. So the rotation matrix equals to:

$$R_{ver} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}. \quad (1)$$

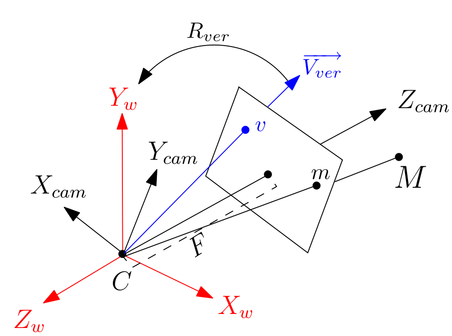


Fig. 1 Coordinate systems and geometry overview. The vector V_{ver} is the vector of vertical vanishing point and pierces the image plane in v. R_{ver} is defined in Sect. 3.2

3.2 Use the Information Given by Vertical Vanishing Point

If we only have a calibrated image of a man-made environment, we can get the vertical direction using the automatic extraction of the vertical vanishing point. Let us suppose that $\overrightarrow{V}_{ver}$ be the vector joining C to the vanishing point in the image plane expressed in the camera system, and $\overrightarrow{Y}_w$ $(0, 1, 0)$ be the Y axis of the world system, see Fig. 1. We perform the rotation that transforms $\overrightarrow{V}_{ver}$ into $\overrightarrow{Y}_w$. Thus, we determine the rotation axis $\overrightarrow{\omega}$ and the rotation angle θ in the following way: $\overrightarrow{\omega} = \overrightarrow{V}_{ver} \otimes \overrightarrow{Y}_w$, after simplification and normalisation $\overrightarrow{\omega} = [\frac{V_z}{d}, 0, \frac{-V_x}{d}]$, where $d = \sqrt{V_z^2 + V_x^2}$, $\theta = \arccos(\overrightarrow{V}_{ver} \cdot \overrightarrow{Y}_w)$, so after simplification, $\theta = \arccos(V_y)$. Using Olinde-Rodrigues formula we get the following rotation matrix:

$$R_{ver} = I\cos\theta + \sin\theta\,[\omega]_\times + (1-\cos\theta)\omega^t\omega. \tag{2}$$

The rotation (R_{ver}) given by equation 1 or 2 is then applied to all 2D points obtained in each image, $\hat{m}$ is replaced by $R_{ver}\hat{m}$.

3.3 Rewriting the Coplanarity Constraint

We recall that for a pair of homologous points $\hat{m}^1$ and $\hat{m}^2$ of a perfect pinhole camera, the constraint on these 2 points is expressed by the equation of coplanarity:

$$\begin{bmatrix}\hat{m}_x^2 & \hat{m}_y^2 & 1\end{bmatrix} E \begin{bmatrix}\hat{m}_x^1 \\ \hat{m}_y^2 \\ 1\end{bmatrix} = 0, \tag{3}$$

where E is a 3×3 rank-2 so-called "essential" matrix [7]. We can also express this constraint by the equation 4.

$$\begin{bmatrix}\hat{m}_x^2 & \hat{m}_y^2 & 1\end{bmatrix} \begin{bmatrix}0 & T_z & -T_y \\ -T_z & 0 & T_x \\ T_y & -T_x & 0\end{bmatrix} R \begin{bmatrix}\hat{m}_x^1 \\ \hat{m}_y^2 \\ 1\end{bmatrix} = 0. \tag{4}$$

However, if we apply the rotation (R_{ver}) obtained in (2) to all homologous points, before we take into account this constraint (4), the rotation R is expressed in a simpler way, as it remains only one parameter of rotation to estimate, the angle ϕ around the Y axis (vertical axis). Thus:

$$R_\phi = \begin{bmatrix}\cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi\end{bmatrix}. \tag{5}$$

Using $t = \tan\frac{\phi}{2}$, we replace $\cos\phi$ by $(1-t^2)/(1+t^2)$ and $\sin\phi$ by $2t/(1+t^2)$. The new coplanarity equation is rewritten as:

$$\begin{aligned}&(-2\hat{m}_x^2 T_y t + \hat{m}_y^2(T_z(1-t^2) + 2T_x t) \\ &\quad - \hat{m}_z^1 T_y(1-t^2)\hat{m}_x^1 + (\hat{m}_x^2(1+t^2))T_z \\ &\quad + \hat{m}_z^2(1+t^2)T_x)\hat{m}_y^1 + (\hat{m}_x^2 T_y(1-t^2) \\ &\quad + \hat{m}_y^2(2T_z t - T_x(1-t^2)) - 2\hat{m}_z^2 T_y t)\hat{m}_z^1 = 0.\end{aligned} \tag{6}$$

3 pairs of homologous points allows for instancing (6) as $\{f_2, f_3, f_4\}$ with remaining unknowns T_x, T_y, T_z and t. The corresponding base has only two degrees of freedom, since no scale modeling has been performed yet. Therefore it is necessary either to fix a component of the base to 1, or to add the constraint of normality. We have chosen this last one: $f_1 \equiv T_x^2 + T_y^2 + T_z^2 - 1 = 0$. We have therefore a system of 4 polynomial equations $\{f_1, f_2, f_3, f_4\}$ of degree ≤ 3. This general model is simplified in the next subsection to get more convenient equations.

3.4 Simplification of the Polynomial System

Let $f_1, \ldots, f_4 \in \mathbb{C}[T_x, T_y, T_z, t]$ be the system of polynomials as defined in Sect. 3.3. Let (T_x, T_y, T_z, t) be a solution of the system $f_1 = \cdots = f_4 = 0$. Two cases are possible:

- If $T_x \neq 0$, then $(T_y/T_x, T_z/T_x, t)$ is a solution of the system $\{f_2 = f_3 = f_4 = 0\}|_{T_x=1} \subset \mathbb{C}[T_y, T_z, t]$, because f_2, f_3, f_4 are homogeneous polynomials in T_x, T_y and T_z. Also, from any solution (T_y, T_z, t) of this system, we can obtain two solution $(1, T_y/\sqrt{1+T_y^2+T_z^2}, T_z/\sqrt{1+T_y^2+T_z^2}, t)$ and $(-1, -T_y/\sqrt{1+T_y^2+T_z^2}, -T_z/\sqrt{1+T_y^2+T_z^2}, t)$ of the initial system $f_1 = \cdots = f_4 = 0$.
- If $T_x = 0$, the following cases are possible:
  - If $T_y \neq 0$, then $(0, 1, T_z/T_y, t)$ is a solution of the system $\{f_2 = f_3 = f_4 = 0\}|_{T_x=0, T_y=1} \subset \mathbb{C}[T_z, t]$. Reciprocally, from any solution (T_z, t) of this system, we can find two solutions $(0, 1/\sqrt{1+T_z^2}, T_z/\sqrt{1+T_z^2}, t)$ and $(0, -T_y/\sqrt{1+T_z^2}, -T_z/\sqrt{1+T_z^2}, t)$ of the initial system $f_1 = \cdots = f_4 = 0$.
  - If $T_y = 0$, then $(0, 0, T_z, t)$ is a solution of $\{f_2 = f_3 = f_4 = 0\}|_{T_x=0, T_y=0} \subset \mathbb{C}[T_z, t]$. Conversely, from any solution (T_z, t) of this system, we can obtain two solutions $(0, 0, \pm 1, t)$ of the initial system $f_1 = \cdots = f_4 = 0$ (note that from the constraint of normality, $T_z \neq 0$).

We have tested the above cases for generic values of input points, and we have observed that only the case $T_x \neq 0$ has solutions. Therefore, in the next section, we will solve only the system $\{f_2 = f_3 = f_4 = 0\}|_{T_x=1} \subset \mathbb{C}[T_y, T_z, t]$. Let us denote this new polynomials by $G = \{g_1, g_2, g_3\}$.

4 Resolution of the Relative Orientation Equation

In this section, we solve the system $g_1 = g_2 = g_3 = 0$ by hiding a variable [5]. Then, using the solutions of the system, we compute the final relative orientation between the images.

4.1 Solving by Hiding a Variable

This method may be applied only for the system where the number of equations is equal to the number of unknowns. To solve the system $g_1 = g_2 = g_3 = 0$ in the variables Ty, Tz and t, we "hide" the last variable t. The new overdetermined system $g_1 = g_2 = g_3 = 0$ is a linear system in T_y and T_z where t is considered as a parameter. This leads to a matrix $M(t)_{3\times 3}$ with polynomial entries of degree 2 in t, and solving the initial system is equivalent to solve the linear system

$$M(t)\begin{bmatrix} 1 \\ T_y \\ Tz \end{bmatrix}^t = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^t. \quad (7)$$

Now, we search the values of t for which the above system has a solution. This implies that the solutions of $\det(M(t)) = 0$ yields the corresponding values of t, which roots give the solutions of the relative pose problem. Then, replacing any root of this equation in (7), we can compute the corresponding values of T_y and T_z. In the following, we illustrate this method with a real example. Let $\{(X_i^c, Y_i^c, Z_i^c)\}$ for $i \in \{1, 2\}$ and $c \in \{m, n, p\}$ be the coordinates of the 3 pairs of homologous points. The following is a real example of coordinates of 3 pair of points:

$$\begin{array}{lll}
X_1^m = -922619 & Y_1^m = -787701 & Z_1^m = 2476100, \\
X_1^n = 1214650 & Y_1^n = -1335824 & Z_1^n = 1530804, \\
X_1^p = 2006952 & Y_1^p = 129983 & Z_1^p = 3082258, \\
X_2^m = 16672 & Y_2^m = -838755 & Z_2^m = 2489002, \\
X_2^n = 1788337 & Y_2^n = -1321237 & Z_2^n = 1521395, \\
X_2^p = 2987423 & Y_2^p = 89776 & Z_2^p = 3076520.
\end{array}$$

The set G includes the following three polynomials:

$$\begin{aligned}
g_1 = {} & -122952718964647 T_y t - 13132551072 T_z(1-t^2) \\
& - 26265102144t - 2337682075438 T_y(1-t^2) \\
& - (773851299345(1+t^2))T_z - 116251891098 \\
& - 4037430619902t^2 - 3921178728804 T_z t \\
g_2 = {} & -9002322177260 T_y t - 2388903484688 T_z(1-t^2) \\
& - 4777806969376t - 889630996198 T_y(1-t^2) \\
& + (1604840522050(1+t^2))T_z + 9761069932 \\
& - 4054870839028t^2 - 4064631908960 T_z t \\
g_3 = {} & -30956485893712 T_y t + 388314203809 T_z(1-t^2) \\
& + 776628407618t - 3033580474094 T_y(1-t^2) \\
& - (180176122752(1+t^2))T_z - 123182504952 \\
& + 676608093368t^2 + 799790598320 T_z t.
\end{aligned}$$

The following algorithm (written in MAPLE) computes the coefficient matrix $M(t)$.

Algorithm 1 SYLVESTER

```
Input: A list G of polynomials
Output: The coefficient matrix M(t) of G
T := plex(T_y, T_z)
E := [T_y, T_z, 1]
Syl := Matrix(3, 3)
for r from 1 to 3 do
  w := collect(G[r], [Ty, Tz])
  for s from 1 to 3 do
    if LeadingMonomial(w, T) = E[s] then
      Syl[r, s] := LeadingCoefficient(w, T)
      w := w - Syl[r, s]E[s]
    end if
  end for
end for
Return Syl
```

In the following, we explain the functions used in the above algorithm. The function `collect`, collects all coefficients of the first argument with the same power of the second argument. Let $R = \mathbb{Q}[x_1, \ldots, x_n]$ be a polynomial ring where $\mathbb{Q}$ is the field of rationals. We recall here the definition of the *lexicographic ordering* (plex), denoted by $\prec$. For this we denote by $\deg_i(m)$ the degree in x_i of a monomial $m \in R$. If m and m' are monomials, then $m \prec m'$ if and only if the first non zero entry in the sequence $(\deg_1(m') - \deg_1(m), \ldots, \deg_n(m') - \deg_n(m))$ is positive. The *leading monomial* of a polynomial $f \in R$ is the greatest monomial (with respect to $\prec$) appearing in f. The coefficient of the leading monomial of f is called the *leading coefficient* of f (see [4] for more details).

Using the above algorithm, we have computed $\det(M(t))$ which is equal to

$$\begin{aligned}
& 4(1+t^2)(154250683864538173123428616949105594 99t^4 \\
& - 1532372929550339616040441670035833018 33t^3 \\
& - 4322571339694270991862493086096290920 2t^2 \\
& - 2221712029788092687039974927068304611t \\
& + 466372661808434607479806415557855 69).
\end{aligned}$$

The real roots of this polynomial (of degree 6) yields the corresponding values of t

$$-0.1965528582, \quad -0.09512634137,$$
$$0.01583776781, \quad 10.21014377.$$

Now, let us consider the value $t = -0.1965528582$. Then, using the following MAPLE commands

```
with(LinearAlgebra):
t:=-0.1965528582;
LinearSolve(Syl(G));
```

we obtain the corresponding values

$$-0.450609301917608285 \quad \text{and} \quad 4.92717236468391206$$

for T_y and T_z. The CPU time of computing $M(t)$ and its determinant is 0.004 sec. where the timing was conducted on a personal computer with 3.2 GHz, 2×Intel(R)-Xeon(TM) Quad core, 24 GB RAM and 64 bits under the Linux operating system.

It is worth noting that we can compute symbolically $\det(M(t))$ which is a polynomial in t and X_i^c, Y_i^c, Z_i^c for $i \in \{1, 2\}$ and $c \in \{m, n, p\}$. If we consider it as a polynomial in t, it has degree 6, and using the MAPLE command `factor`, we can see that it is divisible by $1 + t^2$. Therefore, the number of real solutions corresponding to the system $g_1 = g_2 = g_3 = 0$ is equal to 4.

4.2 Computation of the Final Relative Orientation

After the resolution of the polynomial system, and the obtention of the parameters T_x, T_y, T_z and t, it is possible to compute the final relative orientation between the images. If we suppose that R_{ver1} is the rotation matrix defined in Sect. 3.2 for the image 1, and R_{ver2} the same for the image 2, and R_ϕ the rotation matrix defined by t (5), the final relative orientation between the images 1 and 2 is:

$$\begin{aligned} R_{final} &= R_{ver2}^t R_\phi R_{ver1}, \\ \overrightarrow{T}_{final} &= R_{ver2}^t \overrightarrow{T}, \quad \text{where } \overrightarrow{T} = [T_x, T_y, T_z]^t. \end{aligned} \tag{8}$$

5 Experiments

The accuracy of the relative orientation resolution, using a vertical vanishing point and 3 tie points, is based on three factors:

1. the accuracy of the polynomial resolution of the translation parameters (T_x, T_y, T_z), and of the rotation around the Y axis,
2. the geometric accuracy for the estimation of the vertical direction,
3. the accuracy of the algorithm on tie points in presence of noise.

In order to evaluate the different impacts, we have in a first time worked on synthetic data in Sect. 5.1, then we have used real data in Sect. 5.2.

5.1 Performance Under Noise

In this section, the performance of the 3 points method in noisy conditions has been studied and compared to the 5 points algorithm [21] using the software provided in [20]. The employed experimental setup is similar to [15]. The distance to the scene volume is used as the unit of measure, the baseline length being 0.3. The standard deviation of the noise is expressed in pixels of a 352×288 image as $\sigma = 1.0$. The field of view is equal to 45 degrees. The depth varies between 0 to 2 units. Two different translation values have been considered, one in X (sideway motion) and one in Z (forward motion). The experiments involve 2500 random samples trials of point correspondences. For each trial, we determine the angle between estimated baseline and true baseline vector. This angle is called here *translational error*, and expressed in degrees. For the error estimation on the rotation matrix, the angle of $(R_{true}^T R_{estimate})$ is calculated, and the mean value for the 2500 random trials for each noise level is displayed. From Figs. 2, 3, 4 and 5, we see that the 3-points algorithm is more robust to error caused by noise in sideway and forward motion for estimation of rotation and translation.

Now let us compare 3-points and 5-points algorithms on a planar scene. In this configuration all the points of the scene in the world have the same Z (here equal to 2). The results for the estimation of the rotation (Figs. 6 and 8) show that the two algorithms provide a good determination of the rotation, but the 3-points gives much better results than the 5-points one for the base determination in sideway motion (Fig. 7) and in forward motion (Fig. 9). This weakness of the 5-points algorithm in planar scene has been discussed in [18].

In the following, we discuss the impact of the accuracy of the vertical direction on the estimation of relative orientation. We have introduced an error of 0 to 0.5° on the angular accuracy of the vertical direction. Today for example, a low-cost inertial sensor such as Xsens-MTi [8] gives a precision around 0.5° on the rotation angle around the X and Z axes (the vertical direction being Y axis). Of course, some high accuracy IMU are available, they may reach an accuracy better than 0.01° on the orientation angles if properly coupled with other sensors (e.g. GPS). Using an automatic vanishing point detection, especially in an urban scene, we get a very precise vertical direction (better than 0.001°), as it will be shown later. We have checked the impact of this accuracy on the determination of the rotation and the base orientation. (Figs. 10 and 11).

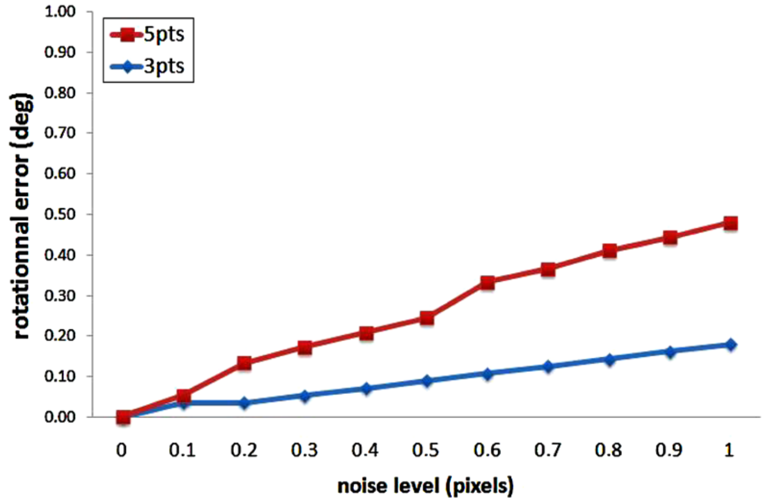


Fig. 2 Error on the rotation (in degrees, sideway motion)

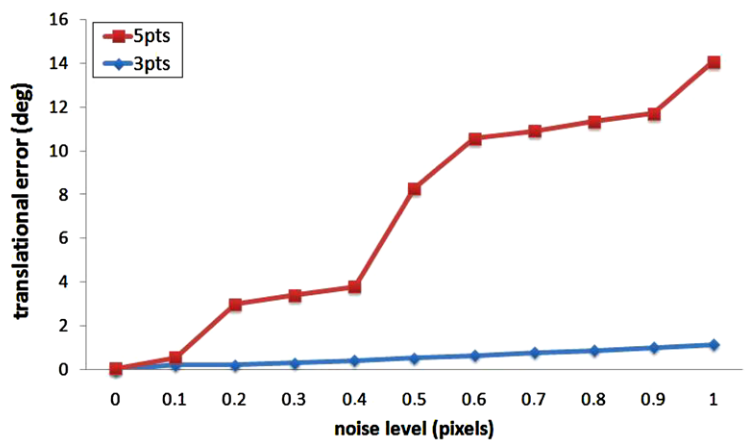


Fig. 3 Error on the baseline orientation (in degrees, sideway motion)

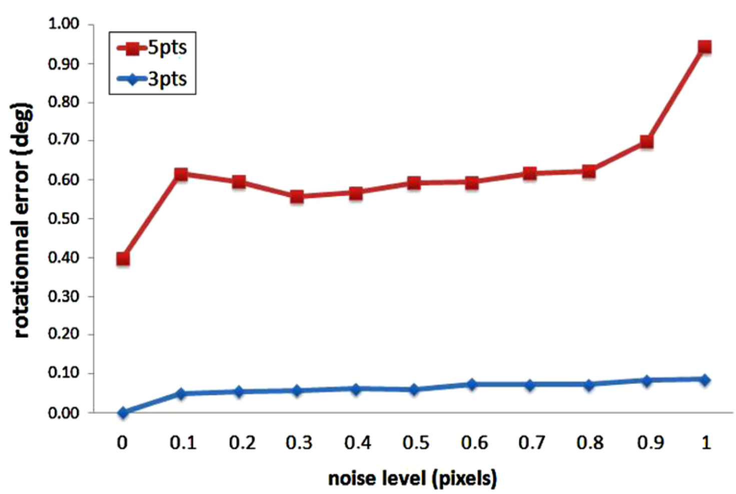


Fig. 4 Error on the rotation (in degrees, forward motion)

5.2 Real Example

So as to provide a numerical example on real images, we have chosen to work on the 9-images sequence “entry-P10” of the online database [22]. In this database we know all the intrinsic and external parameters. First, we extracted the vanishing points on each image. We used the algorithm of [9] because beyond its high speed, it allows an error propagation on the vanishing points according to the error on the segments detection. We express this error in an angu-

Fig. 5 Error on the baseline orientation (in degrees, forward motion)

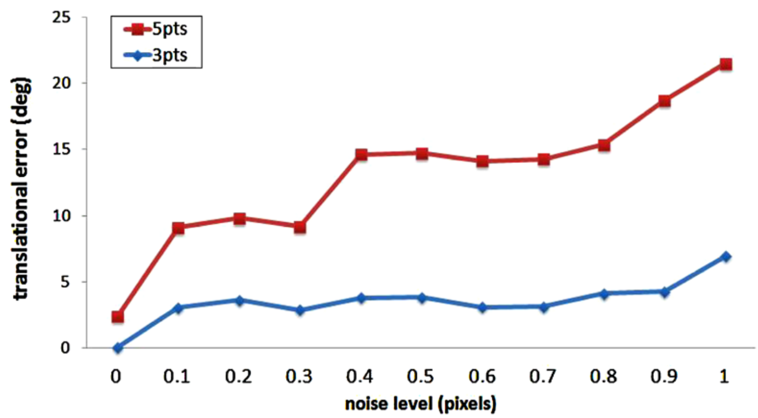


Fig. 6 Error on the rotation (in degrees) in planar configuration (sideway motion)

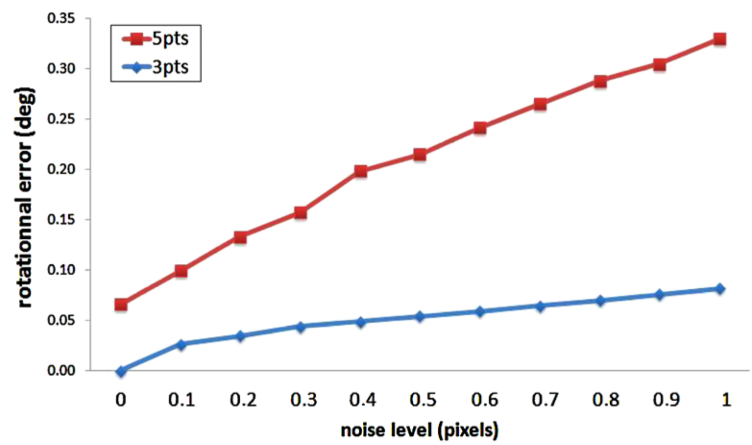


Fig. 7 Error on the base orientation (in degrees) in planar configuration (sideway motion)

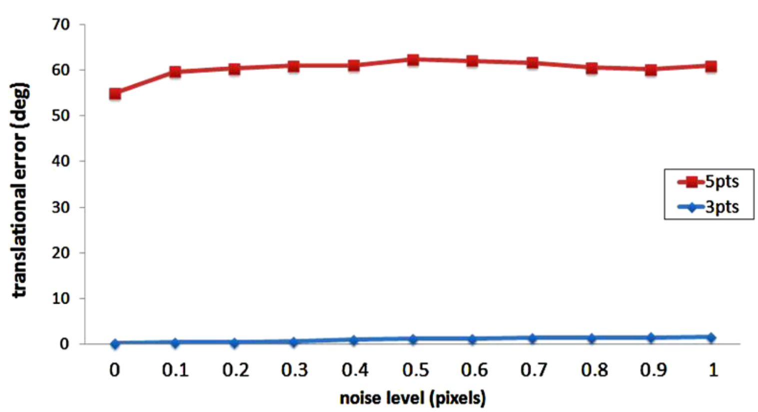
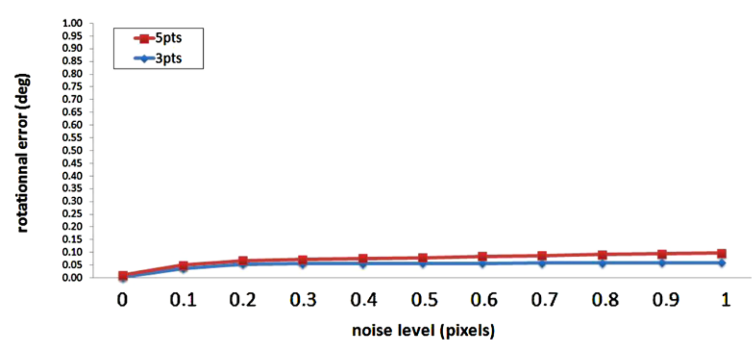


Fig. 8 Error on the rotation (in degrees) in planar configuration (forward motion)



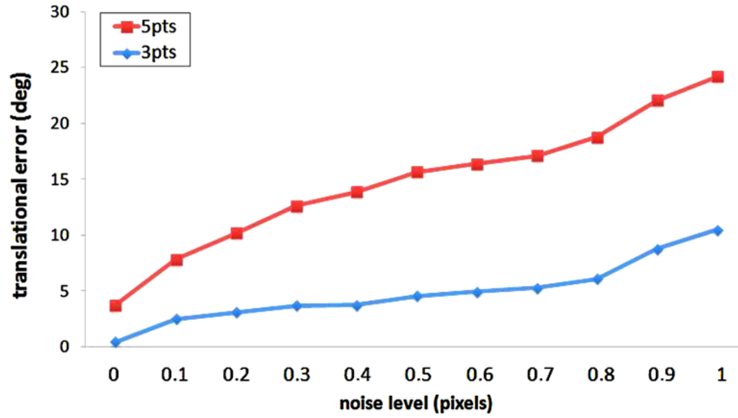


Fig. 9 Error on the base orientation (in degrees) in planar configuration (forward motion)

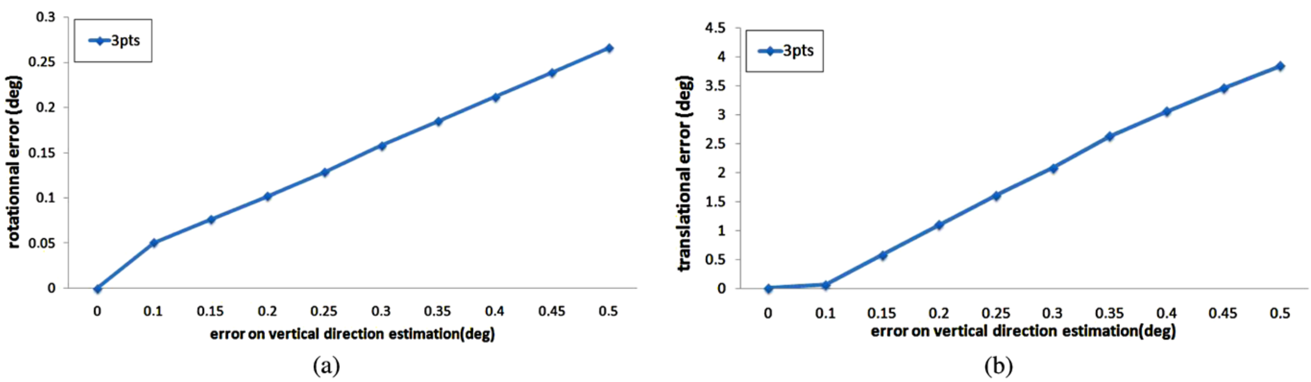


Fig. 10 Impact of the geometric accuracy of the vertical direction on the estimation of (**a**) the rotation (in degrees), and (**b**) the base orientation (in degrees) in sideway motion

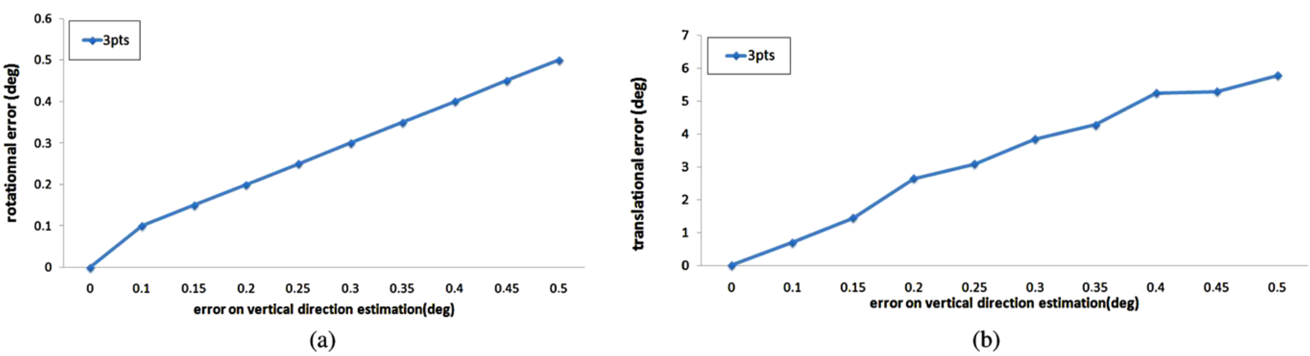


Fig. 11 Impact of the geometric accuracy of the vertical direction on the estimation of (**a**) the rotation (in degrees), and (**b**) the base orientation (in degrees) in forward motion

lar manner. The results of the angular errors are shown in Table 1. As one can see it, the determination of the vertical vanishing point is very precise and according to Figs. 10 and 11 it induces an error close to zero. Then, we have computed the relative orientation for 3 successive images (each time, 2 following couples of images). The points of interest are extracted using SIFT [13] algorithm. The results are presented in Fig. 12. The mean value of angular errors on the rotation amounts to 0.82 degree. For the estimation of the translation, this error amounts to 1.33 degree. These results show clearly the efficiency and robustness of the method.

5.3 Time Performance

The resolution of the polynomial system and detection of vanishing point was written in C++. With a 1.60 GHz PC the time of each resolution is about 2 µs, allowing real-time application. We may note that the selection process using

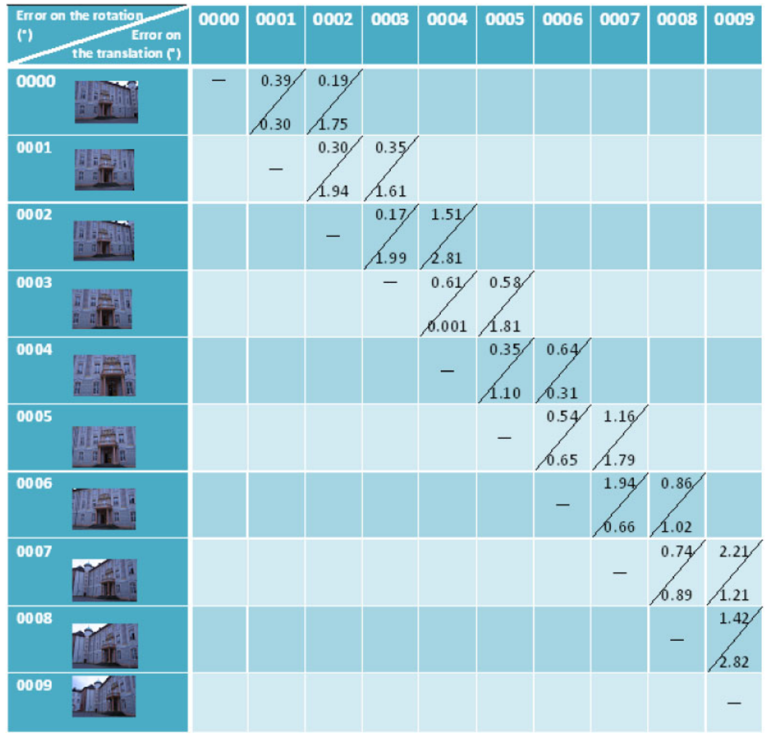


Fig. 12 Result on "entry-P10" sequence. Each cell contains the error on rotation in degrees (*upper left*) and error on the translation in degrees (*bottom right*)

Table 1 Results. Vertical direction detection using the vertical vanishing point

Image	Angular error on vertical direction in degree
0000	0.002569
0001	0.0066
0002	0.001584
0003	0.001443
0004	0.000899
0005	0.00115
0006	0.001445
0007	0.005018
0008	0.002424
0009	0.002223

RanSac [6] among the SIFT points is running considerably faster on 3-point than on 5-point algorithm.

6 Summary and Conclusions

Today, more and more low-cost personal devices include MEMS-IMU in complement to cameras, these devices allow to provide very easily the direction of the vertical in the image. Furthermore, image based automatic extraction of the vertical vanishing point offers a very high accuracy alternative, if needed. So, here, we have demonstrated the advantage of using the vertical direction, and an efficient algorithm for solving the relative orientation problem with this information has been presented. In addition to a considerable acceleration, compared with the classical 5 points solution, our algorithm provides a noticeable accuracy improvement for the baseline estimation. Another interesting feature improvement has been demonstrated: the planar scenes raise no more problem in baseline estimation. This advantageous result is due to an appropriate problem formulation using in a explicit way the significant parameters of the relative orientation (parameters of the rotation and the translation).

Acknowledgements The authors would like to thank the referees for their helpful reports, in particular for pointing out the method of solving by hiding a variable. This work (for the second author) was supported in part by the CEAMA, Isfahan University of Technology, Isfahan 84156, Iran.

References

1. Antone, M., Teller, S.: Automatic Recovery of Relative Camera Rotations for Urban Scenes, vol. 02, pp. 282–289. IEEE Computer Society, Los Alamitos (2000)
2. Barnard, S.T.: Interpreting perspective images. Artif. Intell. **21**, 435–462 (1983)
3. Batra, D., Nabbe, B., Hebert, M.: An Alternative Formulation for Five Point Relative Pose Problem, pp. 21–21 (2007)
4. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms, 3rd edn. Undergraduate Texts in Mathematics. Springer, New York (2007)
5. Elkadi, M., Mourrain, B.: Symbolic-numeric methods for solving polynomial equations and applications. In: Dickenstein, A.,

Emiris, I.Z. (eds.) Solving Polynomial Equations: Foundations, Algorithms, and Applications, pp. 124–168 (2005)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
7. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
8. http://www.xsens.com/
9. Kalantari, M., Jung, F., Paparoditis, N., Guédon, J.P.: Robust and automatic vanishing points detection with their uncertainties from a single uncalibrated image, by planes extraction on the unit sphere. In: IAPRS, Beijing, China, vol. 37 (Part 3A), pp. 203–208 (2008)
10. Kukelova, Z., Bujnak, M., Pajdla, T.: Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems (2008)
11. Li, H.D., Hartley, R.I.: Five-Point Motion Estimation Made Easy, pp. I: 630–633 (2006)
12. Lobo, J., Dias, J.: Vision and inertial sensor cooperation using gravity as a vertical reference. IEEE Trans. Pattern Anal. Mach. Intell. **25**(12), 1597–1608 (2003)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
14. Lutton, E., Maitre, H., Lopez-Krahe, J.: Contribution to the determination of vanishing points using Hough transform. IEEE Trans. Pattern Anal. Mach. Intell. **16**(4), 430–438 (1994)
15. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**(6), 756–777 (2004)
16. Philip, J.: A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. Photogramm. Rec. **15**(88), 589–599 (1996)
17. Schaffalitzky, F., Zisserman, A.: Planar grouping for automatic detection of vanishing lines and points. Image Vis. Comput. **18**, 647–658 (2000)
18. Segvic, M., Schweighofer, G., Pinz, A.: Performance evaluation of the five-point relative pose with emphasis on planar scenes. In: Performance Evaluation for Computer Vision, Workshop of the Austrian Association for Pattern Recognition, Austria, pp. 33–40 (2007)
19. Shufelt, J.A.: Performance evaluation and analysis of vanishing point detection techniques. IEEE Trans. PAMI **21**(3), 282–288 (1999)
20. Stewénius, H.: Matlab code for solving the fivepoint problem. http://vis.uky.edu/~stewe/FIVEPOINT/
21. Stewénius, H., Engels, C., Nistér, D.: Recent developments on direct relative orientation. ISPRS J. Photogramm. Remote Sens. **60**(4), 284–294 (2006)
22. Strecha, C., von Hansen, W., Van Gool, L.J., Fua, P., Thoennessen, U.: On Benchmarking Camera Calibration and Multi-view Stereo for High Resolution Imagery, pp. 1–8 (2008)
23. Triggs, B.: Routines for relative pose of two calibrated cameras from 5 points. Technical report, INRIA (2000)
24. van den Heuvel, F.A.: Vanishing point detection for architectural photogrammetry. Int. Arch. Photogramm. Remote Sens. **32**(5), 652–659 (1998)
25. Vieville, T., Clergue, E., Facao, P.E.D.: Computation of ego motion using the vertical cue. Mach. Vis. Appl. **8**(1), 41–52 (1995)

Mahzad Kalantari is born in 1982 in Ispahan (Iran). She got her engineer's diploma in Geomatics from the University of Ispahan, then her Master degree in Geographical Information from the École Nationale des Sciences Géographiques—University of East Paris, in 2005. The topics of her PhD in Informatics, defended in 2009, was about photogrammetry and computer vision. She follows currently a post-doctoral specialization in electronics.

Amir Hashemi is born in 1975. He got his license in Mathematics at the University of technology of Sharif (Iran), his master in Mathematics at the University of technology of AmirKabir (Iran), and his doctorate in Informatics at the University of Paris 6 (France). His research activities are dedicated to the improvement of the efficiency of the calculation of the Gröbner bases and the resolution of the polynomial systems, and their application. He is currently assistant professor at the University of technology of Isfahan, Iran.

Franck Jung is born in 1970 in Strasbourg. Graduated from the Paris Polytechnic School, he got a doctorate in applied mathematics on training applied to image processing. After having directed the superior school of the geometers and topographers (ESGT) in Le Mans, as assistant director he is currently in charge of the scientific and technical animation in the French Ministry of Ecology. His main topics are about the problematic of training, of statistical inference and the techniques of optimization.

Jean-Pierre Guedon is born in Grosbreuil (France) in 1962. After studies in mathematics and Informatics at the University of Nantes, he got a Master degree in automatic and computer and a doctorate of the Ecole Centrale de Nantes about tomography. After a postdoc at the FDA in USA, he became professor at the Polytechnic School of the University of Nantes. His main field of activity is about the Mojette transform (exact discrete Radon transform) that he developed for fifteen years. He proposed the Mojette game in 2010.