
Orientation relative à partir de 3 points homologues et de la direction verticale

Une approche directe

**Mahzad Kalantari^{1,3}, Amir Hashemi², Franck Jung⁴,
JeanPierre Guedon³**

*1. Université Paris-Est, Laboratoire de Géomatique Appliquée, équipe GTMC
École nationale des sciences géographiques
6 et 8 Avenue Blaise Pascal, Cité Descartes – Champs-sur-Marne
77455 Marne la Vallée cedex 2, France*

mahzad.kalantari@ensg.eu

*2. Department of Mathematical Sciences
Isfahan University Technology
Isfahan, 84156-83111, Iran*

Amir.Hashemi@cc.iut.ac.ir

*3. Institut de recherche en communications et cybernétique de Nantes (IRCCyN)
UMR CNRS 6597, École Polytechnique de l'Université de Nantes, France
jean-pierre.guedon@polytech.univ-nantes.fr*

*4. 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, France
franck.jung@equipement.gouv.fr*

RÉSUMÉ. Cet article propose une nouvelle méthode pour trouver l'orientation relative entre deux images, en utilisant trois points homologues et l'identification de la direction verticale dans les images. La direction verticale peut être déterminée de deux façons : 1- avec une mesure physique directe comme par exemple une centrale inertielle (IMU), 2- l'utilisation du point de fuite correspondant aux verticales. Cette connaissance de la direction verticale résout deux inconnues parmi les trois paramètres de la rotation relative, de sorte que seulement trois points homologues sont nécessaires dans les deux images. En réécrivant les équations de coplanarité, on parvient à une solution beaucoup plus simple. La résolution des inconnues restantes est exécutée par une méthode algébrique qui utilise les bases de Gröbner. Les éléments nécessaires pour construire une solution algébrique spécifique sont donnés dans cet article, en tenant compte d'une mise en œuvre temps réel. Les résultats sur des données synthétiques puis sur des données réelles montrent l'efficacité de cette méthode.

ABSTRACT. This paper proposes to use the knowledge of the vertical direction to estimate the relative orientation of images. The presented algorithm can use information about the vertical direction, either computed by image based techniques (information taken from the vertical vanishing point), or obtained by a direct physical measurement. This knowledge solves 2 unknowns among the 3 parameters of the relative rotation, so that only 3 homologous points are requested to position a couple of images. The coplanarity constraint equations are re-written to lead to a simpler version. The remaining unknowns resolution is performed by an algebraic method using Gröbner bases. 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.

MOTS-CLÉS : orientation relative, direction verticale, bases de Gröbner.

KEYWORDS: relative orientation, vertical direction, Gröbner basis.

DOI:10.3166/TS.27.325-348 © 2010 Lavoisier, Paris

Extended abstract

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 modelled 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 up to now. Here we use the knowledge of the vertical direction to solve the relative orientation problem for the two following reasons:

- The use of MEMS-IMU in electronic personal devices such as smart phones and digital cameras is continuously increasing, and the prices of IMUs is correspondingly decreasing. The camera-IMU sensors fusion is not the topics of this paper, as many authors have shown the advantage of coupling them. In MEMS-IMU the accuracy of heading (rotation around the vertical 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, which strengthen the very weakness of IMU data.

- Today very efficient 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 on such topics exist in the literature. These algorithms are very useful in urban and man-made environments.

The use of the vertical direction so as to reduce the disparity between two frames, to simplify 3D vision, has already been considered. But most papers consider a fixed stereoscopic baseline, and here we are supposed to have no information on it. Furthermore, most papers 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 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. Hence, we solve these equations using the Gröbner bases in a direct way.

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). Here 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 Gröbner bases technique. Most of mathematical softwares allow to compute Gröbner bases. But, in the framework of our research work, we are looking for operational applications, running as fast as possible. Thus, we need to design a special algorithm for computing a Gröbner basis of our system, more efficient than classical existing ones. The commonly-used algorithms to compute the Gröbner bases are based on Buchberger's algorithm, which are inefficient in most cases. The Gröbner basis method for solving systems of polynomial equations (based on matrices) was recently used to solve many minimal problems in computer vision. This method was first used for such systems by Stewenius. Later many other new minimal problems were solved using similar methods. Kukelova et al. in have observed that the action matrix can be constructed without computing a complete Gröbner basis. Recently they have presented the automatic generator of such Gröbner bases solvers for minimal problems in computer vision.

Our approach is based on constructing the Macaulay matrix, and is followed by a Gaussian elimination procedure. This approach was first used for solving polynomial system by Macaulay when he generalized Sylvester's matrix to multivariate polynomials. The idea is to build a matrix whose lines contain the multiples of the polynomials in the original system, the columns representing a basis of monômials up to a given degree. It was observed by Lazard that for a high enough degree, ordering the columns according to a monômial ordering and performing a row reduction without column pivoting on the matrix gives directly a Gröbner basis for the input system. It seems that this solver was not previously used in computer vision community. Moreover, our experiments show that our proposed solver is correct, numerically stable, robust and reasonably fast.

Our approach consists in finding the matrix representation of the useful multiples of the input polynomials. Moreover, this construction does not depend on the value of input points. Therefore, for any given value of input points, we can calculate directly this matrix, and then the Gröbner basis of the corresponding ideal is obtained via only a single Gaussian elimination (this matrix for our problem is of size $15 \times$

21). Once we have the Gröbner basis, we can compute the solutions. However, the advantage of our method is that it avoids any useless further computation.

Another efficient way for solving a polynomial system, by means of its matrix, is resultants. In general, a resultant is a single polynomial derived from a system of polynomial equations that encapsulates the solution of the system. The polynomial system that we concentrate on is a system such that the variables can be partitioned into two groups, so that every polynomial is homogeneous in each group. Such a system is called multihomogeneous, and its resultant is called multihomogeneous resultant. In this paper, we apply the method of Z. Emiris and A. Mantzaflais to compute the multihomogeneous resultant of our system. So, we have compared the efficiency of Gröbner bases and resultants techniques.

First, we present the geometric framework of our system. Then we rewrite the coplanarity constraint using the vertical direction knowledge, and the resolution of a polynomial system with the help of Gröbner bases and resultant. At the end, we present the assessment of the algorithm in noisy conditions, and the 3-points algorithm is compared to the well known 5-points algorithm, and finally a comparison with real image database is performed.

1. Introduction

La présente étude propose une solution effective au problème de l'orientation relative dans le cadre de systèmes étalonnés. Dans une telle situation, les paramètres intrinsèques de l'appareil-photo, par exemple la distance focale et sa distorsion, sont supposés connus *a priori*. Dans ce cas l'orientation relative qui lie deux vues est modélisée par 5 inconnues : la matrice de rotation (3 inconnues) et la translation (2 inconnues avec un facteur d'échelle indéterminé). Dans cet article nous utilisons la connaissance de la direction verticale pour résoudre le problème de l'orientation relative pour deux raisons :

1. L'usage de plus en plus fréquent de dispositifs mixtes micro-mécaniques et inertiels (MEMS-IMU) dans appareils personnels électroniques tels que téléphones intelligents, appareils-photo numériques et centrales inertiels de faible prix. La fusion des capteurs (camera-IMU) n'est pas le but de cet article, puisque beaucoup d'auteurs ont déjà montré l'intérêt de les associer (Lobo *et al.*, 2003). Dans les MEMS-IMU la précision en cap (rotation autour de l'axe vertical Z) est moins bonne que le tangage (rotation autour de l'axe X) et le roulis (rotation autour de l'axe Y), lié au fait que le champ de la gravité n'a aucun effet sur une rotation autour de l'axe vertical. La nouvelle méthode, présentée ici, tire tout son intérêt d'une combinaison de données inertiels et de l'usage de 3 couples de points homologues qui confortent précisément le point faible de données inertiels.

2. Aujourd'hui des algorithmes performants basés sur l'analyse d'image sont disponibles, et ils permettent de calculer la direction verticale avec une grande précision. Si nous ne disposons que d'un ensemble d'images calibrées, sans autres mesures externes, nous pouvons déterminer aussi la direction verticale par extraction automatique des points de fuite dans ces images.

2. État de l'art

Le calcul des paramètres extrinsèques (orientation et position) de la caméra à partir de plusieurs vues à l'aide de points homologues est la base de beaucoup d'applications, aussi bien du domaine de la photogrammétrie, que de celui de la vision par ordinateur. Quand il s'agit de définir les paramètres extrinsèques d'un couple d'images, on parle en général d'orientation relative car la position et l'orientation de la deuxième image sont calculées par rapport à la première image. L'orientation relative est connue depuis un siècle environ par la communauté des photogramètres.

L'inconvénient majeur de l'approche photogrammétrique, qui est basée sur la condition de colinéarité, est dû à la non-linéarité du problème, et qui dit problème non linéaire dit nécessité de disposer de valeurs approchées autour desquelles on opère une linéarisation de la fonction analytique que l'on cherche à minimiser. Dans le cas de la photogrammétrie aérienne (Wolf *et al.*, 2000), (Kasser *et al.*, 2001), les valeurs initiales sont très simples à estimer en raison de la régularité d'échantillonnage spatial et des axes quasi verticaux des clichés. En plus, l'avion ne subit que de faibles variations de rotation, et un ensemble de points d'appui peut toujours être obtenu.

Pour les applications terrestres la détermination de ces valeurs initiales devient plus hasardeuse, c'est pour cela que la communauté de vision par ordinateur depuis plusieurs années s'est intéressée au calcul des paramètres extrinsèques de la caméra à l'aide d'équations linéaires. Le modèle employé est alors basé sur la condition de coplanarité. Longuet-Higgins publia la première fois en 1981 la notion de matrice essentielle (E) (Longuet Higgins, 1981).

En utilisant la matrice essentielle (E) la condition de coplanarité devient linéaire. Résoudre l'orientation relative revient donc à estimer les 9 éléments de la matrice essentielle. En fait 8 points suffisent pour résoudre le système, car l'estimation se fait à une échelle près. La matrice essentielle suppose que la calibration de la caméra est disponible. Cette résolution, bien qu'efficace en termes de calcul, ajoute 3 degrés de liberté à l'orientation relative. Un autre inconvénient majeur est qu'elle ne peut fonctionner que si tous les points homologues sont coplanaires dans l'espace objet, ce qui peut très bien arriver par exemple sur des façades. Il y a donc eu beaucoup d'efforts au sein de la communauté de vision par ordinateur pour réduire le degré de liberté de la matrice essentielle et diminuer au strict minimum le nombre de points homologues requis, d'ailleurs les différentes méthodes de résolution de la matrice essentielle ont pour nom ce nombre de points homologues minimum. On peut ainsi citer les méthodes des 8 points (Longuet Higgins, 1981), (Hartley *et al.*, 2004), (Ma *et al.*, 2003), 7 points (Sturm, 1869), 6 points (Philip, 1996) et plus récemment 5 points.

Il a été pour la première fois démontré par Kruppa (1913) que la résolution directe de l'orientation relative à partir de 5 points contenait en général 11 solutions au plus. La méthode décrite par celui-ci consistait à trouver toutes les intersections de deux courbes de degré 6. Malheureusement, en son temps, sa méthode n'a pu donner lieu à une implémentation numérique. Plus récemment dans (Demazure, 1988), (Faugeras *et al.*, 1990), (Faugeras *et al.*, 1993), (Maybank, 1992), (Heyden *et al.*, March 1999) il a été démontré que le nombre de solutions est en général égal à 10, incluant les solutions complexes. Triggs (Triggs, 2000) a donné une version

détaillée pour une implémentation numérique. Philip (Philip, 1996) a présenté en 1996 une solution utilisant un polynôme de degré 13, et donné par ailleurs une méthode numérique pour résoudre son système. Les racines de son polynôme fournissent directement l'orientation relative. L'idée de Philip a été reprise en 2004 par Nistér (2004) qui a raffiné cet algorithme. Depuis, nombre de publications ont visé des améliorations à la méthode de Nister, notamment Stewenius (Stewenius *et al.*, 2006) qui a donné une résolution polynomiale à l'aide des bases de Gröbner. D'autres articles ont proposé quelques modifications de la méthode de Nister en vue d'une amélioration numérique, ou bien d'une simplification d'implémentation (Li *et al.*, 2006), (Segvic *et al.*, 2007), (Batra *et al.*, 2007), (Kukelova *et al.*, 2008). Comme on le constate aisément, la plupart des efforts lors de ces dernières années ont porté sur la diminution du nombre de points homologues pour arriver au minimum absolu c'est-à-dire 5. Dans cet article nous proposons d'employer l'identification de la direction verticale dans l'image, soit à partir de capteurs externes soit à partir du point de fuite des verticales. Dans ce cas 3 points homologues suffisent pour résoudre le problème de l'orientation relative.

L'utilisation de la direction verticale pour réduire la disparité entre deux systèmes, et pour simplifier la vision en 3D, a été déjà publiée dans (Vieville *et al.*, 1995). Mais la plupart utilisent une ligne de base stéréoscopique fixe, et ici nous considérons que nous n'avons pas de connaissance de celle-ci. En outre, la plupart des publications (Vieville *et al.*, 1995) essaient de résoudre le problème par des méthodes itératives ou non minimales (par exemple plus de trois points).

3. Notre approche du problème de l'orientation relative

La connaissance de la direction verticale enlève 2 degrés de liberté au problème de l'orientation relative. Par conséquent seuls 3 couples de points homologues suffisent pour résoudre les 3 autres inconnues, ce qui simplifie considérablement les

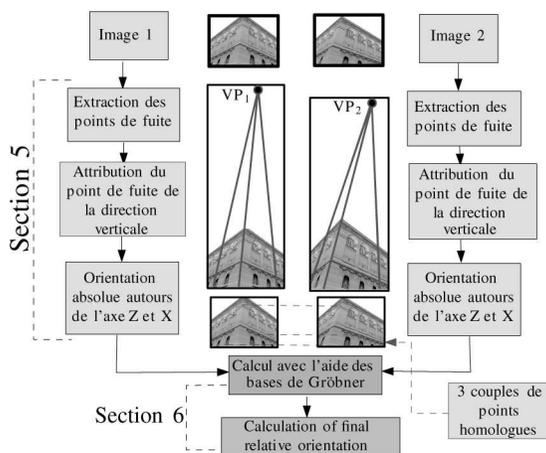


Figure 1. Vue d'ensemble de l'algorithme présenté dans cet article

équations d'orientation relative, basées sur la condition de coplanarité. Nous résolvons ces équations avec les bases de Gröbner. Les différentes étapes de l'algorithme et les sections correspondantes sont illustrées dans la figure 1. Le présent article est structuré comme suit : la section 5 présente la simplification de la contrainte de coplanarité lorsqu'on utilise la connaissance de la direction verticale. La résolution du système polynomial avec l'aide des bases de Gröbner est décrite dans la section 6. Les performances de l'algorithme sur des images de synthèse puis de vraies images sont données dans la section 7.

4. Systèmes de coordonnées et éléments de la géométrie d'ensemble

C'est le système classique de coordonnées image (cf. figure 2) utilisé en vision par ordinateur qui a été choisi (Hartley *et al.*, 2004). Dans le système de la camera $(X_{cam}, Y_{cam}, Z_{cam})$, le plan focal est $Z_{cam} = F$, F étant la distance focale. Étant donnée la matrice K de calibration, la vue est normalisée en transformant tous les points par l'inverse de K , $\hat{m} = K^{-1}m$, dans lequel m est un point dans le plan image. Donc la nouvelle matrice de calibration devient la matrice identité, M étant un point dans l'espace objet. Dans le reste de cette étude nous supposons que toutes les coordonnées des points dans le plan image sont normalisées. Pour un système stéréoscopique, en orientation relative, le centre du système de coordonnées de l'espace objet est le centre optique C de l'image gauche, avec les mêmes directions d'axes. Le système de coordonnées objet est dénoté par (X_w, Y_w, Z_w) . Dans ce système l'axe Y_w est le long de la verticale physique de l'espace objet.

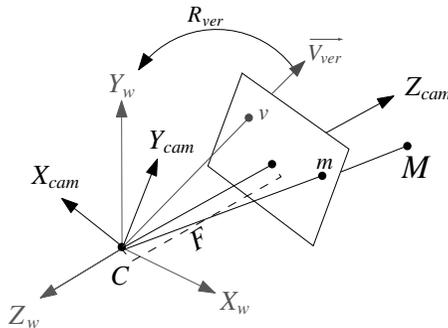


Figure 2. Système de coordonnées employé. Le vecteur V_{ver} est le vecteur du point de fuite vertical et il coupe le plan image dans v . R_{ver} est définie dans la section 5.2

5. Emploi de la direction verticale pour l'orientation relative

Cette section concerne l'identification de la verticale selon une approche image. Si cette information est obtenue directement par une mesure physique, l'algorithme décrit dans cette section ne sera pas nécessaire. Dans la géométrie conique, caractéristique de la vision humaine ou de la photographie, les lignes parallèles de

l'espace de l'objet forment dans l'image un faisceau de droites, concourantes sur les points de fuite. Par conséquent à chaque point de fuite, une direction est associée. Depuis les années 1980, de nombreux auteurs s'intéressent à l'extraction des points de fuite. Beaucoup d'algorithmes sur de tels sujets existent dans la littérature (Barnard, 1983), (Lutton *et al.*, 1994), (Shufelt, 1999), (van den Heuvel, 1998), (Antone *et al.*, 2000), (Schaffalitzky *et al.*, 2000). Dans cet article nous avons choisi l'approche de (Kalantari *et al.*, 2008) pour sa simplicité, sa robustesse et sa vitesse de calcul.

5.1. Identification automatique de la direction verticale

Une fois que les différents points de fuite et leurs directions associées sont extraits, nous procédons à une identification automatique du point de fuite qui correspond à la direction verticale, le seul qui nous intéresse. Supposons que nous ayons n vecteurs unité des directions de l'espace liant le centre optique aux points de fuite sur le plan image : $\{\vec{V}P_1, \vec{V}P_2, \vec{V}P_3, \dots, \vec{V}P_n\}$. Ces vecteurs sont exprimés dans le système de la caméra. Chaque vecteur est exprimé de la manière suivante : $\vec{V}P = [V_x \ V_y \ V_z]^t$. Nous choisissons parmi les différents points de fuite extraits, celui qui remplit les conditions suivantes :

1) le rapport $\frac{V_x}{V_y}$ est minimum,

2) cette direction est perpendiculaire à celle des autres points de fuite, considérant que la plupart d'entre elles sont horizontales (seul le point de fuite vertical est unique). La prise en compte de la calibration est indispensable ici, et nous écrivons : $90^\circ - \epsilon < \arcsin(|\vec{V}P_{vertical} \otimes \vec{V}P_i|) < 90^\circ + \epsilon$. Par expérience nous avons choisi la valeur de 1° pour ϵ .

Ces conditions ont été testées sur 250 images de tous types. L'identification automatique du point de fuite vertical a été efficace à 100 %. Quelques résultats sont présentés à la figure 3. La méthode utilisée pour sélectionner le point de fuite vertical est citée ici juste comme une possibilité, beaucoup d'autres peuvent être choisies, par exemple des données MEMS externes peuvent être préférées si disponibles.



Figure 3. Quelques résultats sur l'extraction et l'identification automatique des lignes et directions verticales.

5.2. Orientation absolue autour des axes Z_w et X_w

Supposons que \vec{V}_{ver} est le vecteur qui joint C au point de fuite dans le plan image exprimé dans le système de la caméra, et $\vec{Y}_w(0, 1, 0)$ étant l'axe Y de l'espace objet (figure 2). Nous cherchons à définir la matrice de rotation qui transforme \vec{V}_{ver} dans \vec{Y}_w , pour cela nous déterminons l'axe de la rotation $\vec{\omega}$ et l'angle θ de la rotation de la façon suivante : $\vec{\omega} = \vec{V}_{ver} \otimes \vec{Y}_w$, après simplification et normalisation $\vec{\omega} = [\frac{V_z}{d}, 0, \frac{-V_x}{d}]$, ou $d = \sqrt{V_z^2 + V_x^2}$, après simplification $\theta = \arccos(V_y)$. Utilisant la formule d'Olinde-Rodrigues nous obtenons la matrice de rotation suivante :

$$R_{ver} = I \cos \theta + \sin \theta [\omega]_{\times} + (1 - \cos \theta) \omega \omega^T. \quad [1]$$

Cette rotation (R_{ver}) est appliquée alors à toutes les coordonnées 2D des points obtenus dans chaque image, donc : \hat{m} est remplacé par $R_{ver} \hat{m}$.

5.3. Simplification et réécriture de la contrainte de coplanarité

Rappelons en premier que pour une paire de points homologues \hat{m}^1 et \hat{m}^2 d'une caméra parfaite, la contrainte sur ces 2 points est exprimée par l'équation de coplanarité :

$$[\hat{m}_x^2 \quad \hat{m}_y^2 \quad 1] E \begin{bmatrix} \hat{m}_x^1 \\ \hat{m}_y^1 \\ 1 \end{bmatrix} = 0. \quad [2]$$

où E est la matrice essentielle 3×3 de rang 2 (Hartley *et al.*, 2004). Nous pouvons aussi exprimer cette contrainte par l'équation 3.

$$[\hat{m}_x^2 \quad \hat{m}_y^2 \quad 1] \begin{bmatrix} 0 & T_z & -T_y \\ -T_z & 0 & T_x \\ T_y & -T_x & 0 \end{bmatrix} R \begin{bmatrix} \hat{m}_x^1 \\ \hat{m}_y^1 \\ 1 \end{bmatrix} = 0. \quad [3]$$

Cependant, si nous appliquons la rotation (R_{ver}) obtenue dans l'équation 1 à tous les points homologues, avant que nous ne tenions compte de cette contrainte (équation 3), la rotation R est exprimée de façon plus simple, puisqu'il reste seulement un paramètre à estimer, l'angle ϕ autour de l'axe Y (axe vertical). Donc :

$$R_{\phi} = \begin{bmatrix} \cos \phi & 0 & -\sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{bmatrix} \quad [4]$$

Nous remplaçons $\cos \phi$ par $(1 - t^2)/(1 + t^2)$ et $\sin \phi$ par $2t/(1 + t^2)$. La nouvelle équation de coplanarité sera simplifiée ainsi :

$$\begin{aligned}
 & (-2 \widehat{m}_x^2 T_y t + \widehat{m}_y^2 (T_z (1 - t^2) + 2T_x t) - \\
 & \widehat{m}_z^2 T_y (1 - t^2)) \widehat{m}_x^1 + (\widehat{m}_x^2 (1 + t^2) T_z + \\
 & \widehat{m}_z^2 (1 + t^2) T_x) \widehat{m}_y^1 + (\widehat{m}_x^2 T_y (1t^2) + \\
 & \widehat{m}_y^2 (2T_z t - T_x (1 - t^2))) - 2\widehat{m}_z^2 T_y t \widehat{m}_z^1 = 0.
 \end{aligned} \tag{5}$$

Nous définissons pour 3 paires de points homologues les équations 5, et nous les nommons $\{f_2, f_3, f_4\}$. Les inconnues restantes sont par conséquent T_x, T_y, T_z et t . La base a seulement 2 degrés de liberté, parce que le modèle n'a pas d'échelle. Par conséquent il est nécessaire, ou de fixer une composante de la base à 1, ou d'ajouter la contrainte de normalité. Nous choisissons cette dernière : $f_1 \equiv T_x^2 + T_y^2 + T_z^2 - 1 = 0$. L'avantage est que ceci permet d'obtenir un modèle plus général. Nous avons par conséquent un système de 4 équations polynomiales de degré 3 $\{f_1, f_2, f_3, f_4\}$. Maintenant nous décrivons la résolution directe de ce système polynomial en utilisant les bases de Gröbner.

6. Résolution à l'aide des bases de Gröbner

Nous rappelons tout d'abord les définitions des bases de Gröbner, et aussi le lien entre les bases de Gröbner et l'algèbre linéaire. Puis, nous utilisons ces concepts pour trouver un algorithme spécifique pour calculer la base de Gröbner du système de polynômes défini dans la section 5.3. La notion de base de Gröbner a été introduite par B. Buchberger, qui en a donné le premier algorithme de calcul (cf. (Buchberger, 1965)). Cet algorithme est disponible dans la plupart des logiciels d'algèbre comme MAPLE, MATHEMATICA, SINGULAR (Greuel *et al.*, 2005), MACAULAY2 (Grayson *et al.*, 1996), COCOA (COCOA, n.d.) et le logiciel SALSA (SALSA, n.d.). Soit $R = K[x_1, \dots, x_n]$ un anneau de polynômes où K est un corps arbitraire. Soit $f_1, \dots, f_k \in R$ un ensemble de k polynômes et soit $I = \langle f_1, \dots, f_k \rangle$ un idéal de R produit par les f_i . Nous avons besoin aussi d'un ordre monomial sur R . Nous rappelons ici la définition de l'ordre lexicographique inverse (DRL), dénoté par $<$, qui est un rangement spécial des monômes et présente d'intéressantes propriétés de calcul. Pour ceci nous dénotons respectivement par $\deg(m)$ et $\deg_i(m)$ le degré total et le degré en x_i d'un monôme m . Si m et m' sont des monômes, alors $m < m'$ si et seulement si le dernier terme non nul entré dans la séquence $(\deg_1(m') - \deg_1(m), \dots, \deg_n(m') - \deg_n(m), \deg(m) - \deg(m'))$ est négatif (cf. (Cox *et al.*, 2007)). Soit $\text{in}(f) \in R$ le terme initial (le plus grand terme) d'un polynôme $f \in R$ selon l'ordre $<$ et $\text{in}(I) = \langle \text{in}(f) \mid f \in I \rangle$ l'idéal initial de I .

DÉFINITION 6.1 (Base de Gröbner). — *Un sous-ensemble fini $G \subset I$ est une base de Gröbner de I pour l'ordre $<$ si $\langle \text{in}(G) \rangle = \text{in}(I)$.*

DÉFINITION 6.2 (Base de Gröbner réduite). — *Une base de Gröbner G de I est une base de Gröbner réduite de I si pour tout $g \in G$, g est monique et aucun monôme de g n'est dans $\langle \text{in}(G \setminus \{g\}) \rangle$.*

PROPOSITION 6.1 ((Cox *et al.*, 2007), Proposition 6, page 92). — *Tout idéal a une base de Gröbner réduite et cette base est unique.*

6.1. Matrice de Macaulay

Nous rappelons maintenant la définition d'une matrice de Macaulay et nous expliquons comment nous pourrions l'utiliser pour calculer la base de Gröbner d'un idéal. Avec les notations de la section précédente, nous considérons l'idéal I engendré par les f_i , et \prec , l'ordre DRL pour un ordre monomial. Nous supposons que nous connaissons le degré maximum d des monômes qui paraissent dans la représentation des éléments de la base de Gröbner de I en les f_i . Notons que ce degré est le degré maximal des monômes qui paraissent dans le calcul de la base de Gröbner de I .

Nous pouvons construire la matrice de Macaulay $M_d(f_1, \dots, f_k)$ (pour faire court nous la dénotons par M_d) comme suit : écrivons horizontalement tous les monômes de degré au plus égal à d , rangés selon \prec (le premier est le plus grand). Ainsi, chaque colonne de la matrice est indexée par un monôme de degré au plus d . Multiplions chaque f_i de 1 à k par n'importe quel monôme m de degré au plus $d - \deg(f_i)$, et écrivons les coefficients de mf_i sous leurs monômes correspondants, ce qui donne donc une ligne de la matrice. Les lignes sont rangées: la rangée mf_i est avant uf_j si ou bien $i < j$, ou $i = j$ et $m < u$. Pour toute ligne dans la matrice, considérons l'indexation du monôme de la première colonne non nulle de cette ligne. Il est appelé le monôme de tête de la ligne, et c'est le monôme de tête du polynôme correspondant.

$$M_d = mf_i \begin{pmatrix} \text{les monomes de degré au plus } d \\ \vdots \\ \vdots \end{pmatrix}$$

L'élimination de Gauss appliquée à cette matrice conduit à une base de Gröbner de I (cf. (Lazard, 1983)). En effet, appelons \tilde{M}_d la forme d'élimination de Gauss de M_d , où la seule opération élémentaire possible dans une ligne est l'addition d'une combinaison linéaire des lignes antérieures. Maintenant, considérons tous les polynômes qui correspondent à une ligne dont le terme de tête n'est pas le même dans M_d et dans \tilde{M}_d , alors l'ensemble de ces polynômes est une base de Gröbner de I .

6.2. Construction du solveur algébrique spécifique

Dans cette section nous décrivons un algorithme général pour calculer la base de Gröbner du système de polynômes défini dans la section 5. Il faut bien noter que quand les coordonnées des points en entrée changent, seuls les coefficients des polynômes changent. Donc, en utilisant l'approche de Lazard (cf. section 6.), nous construisons une matrice de Macaulay (et nous pouvons la calculer directement quand les coordonnées des points changent), et une élimination de Gauss sur cette matrice donne la base de Gröbner de l'idéal.

Soit $f_1, \dots, f_4 \in \mathbb{C}[T_x, T_y, T_z, t]$ le système de polynômes comme défini dans la section 5.3. Soit $I = \langle f_1, \dots, f_4 \rangle$. Notre premier défi est de choisir un bon ordre.

Comme bon ordre monomial, nous voulons dire un ordre monomial pour lequel le degré maximum atteint dans le calcul de la base de Gröbner est minimum. Quant à la complexité, nous cherchons un ordre monomial pour lequel le calcul a la complexité optimale. Nous choisissons l'ordre DRL parce qu'il fournit typiquement les calculs de base de Gröbner les plus rapides. Ici, nous étudions deux variantes de cet ordre monomial. En premier, nous considérons $\text{DRL}(T_x, T_y, T_z, t)$. Nous calculons le degré maximum des monômes qui apparaissent dans le calcul de la base de Gröbner de I pour cet ordre monomial. Pour ceci, nous homogénéisons les f_i par rapport à une variable auxiliaire h et nous calculons la base de Gröbner du système homogénéisé pour $\text{DRL}(T_x, T_y, T_z, t, h)$. Le degré maximal des éléments de cette base est 6 et par conséquent le degré maximal des monômes qui paraissent dans le calcul de la base de Gröbner de I sera 6. Considérons un autre ordre monomial. Pour ceci, nous rappelons en premier la définition de l'ordre monomial de degré par bloc. Soit X et Y deux ensembles de variables et $<_X$ (resp. $<_Y$) un rangement de monômes défini sur X (resp. Y). L'ordre monomial de degré par bloc de $<_X$ et $<_Y$ est un rangement de monômes défini sur les monômes en X et Y et nous le dénotons par $<$. Soit $m = xy$ et $m' = x'y'$ deux monômes où x et x' (resp. y et y') sont des monômes en X (resp. Y). Nous définissons que $m < m'$ si $\text{deg}(m) < \text{deg}(m')$, ou $\text{deg}(m) = \text{deg}(m')$ et $x <_X x'$ ou $\text{deg}(m) = \text{deg}(m')$ et $x = x'$ et $y <_Y y'$. Maintenant, considérons l'ordre de degré par bloc de $\text{DRL}(T_x, T_y, T_z)$ et $\text{DRL}(t)$. En utilisant la technique précitée nous pouvons calculer le degré maximal des monômes qui paraissent dans le calcul de la base de Gröbner par rapport à cet ordre monomial, qui est 8. Donc le premier ordre monomial est un meilleur. Notons que pour quelques systèmes un ordre de degré par bloc de DRL est mieux qu'un DRL simple. Nous avons testé quelques autres ordres monomiaux, et il apparaît qu'un DRL simple est le meilleur.

Notre deuxième problème est de construire $M_6(f_1, \dots, f_4)$, soit donc M . Pour calculer une telle matrice, nous devons trouver les produits mf_i , tels qu'une élimination de Gauss sur la matrice de la représentation de ces produits nous mène à la base de Gröbner de I . Pour ceci, nous utilisons le degré maximum atteint dans le calcul de la base de Gröbner, qui est 6. Nous considérons tous les produits mf_i où le m est un monôme de degré au plus $6 - \text{deg}(f_i)$. Cela donne 175 polynômes. Parmi eux, il y a des produits qui sont utiles pour construire M . En utilisant le programme en MAPLE suivant, nous pouvons choisir ceux qui sont utiles :

```
L:=NULL:
AA:=A:
for i from 1 to nops(A) do
  unassign('p');
  X:=AA:
  member(A[i], AA, 'p');
  AA:=subsop(p=NULL,AA):
  if IsGrobner(Macaulay(AA)) then
    L:=L,i;
  else
    AA:=X:
  fi:
od:
```

où `IsGrobner` est un programme pour tester si un ensemble de polynômes forme une base de Gröbner pour I ou non, et `Macaulay` est un programme qui exécute une élimination de Gauss sur la représentation matricielle d'un ensemble de polynômes. Cela donne 65 polynômes de degré au plus 6. Dans ce cas-là, M a une dimension de 65×77 . Voici la liste des 65 polynômes qui ont été trouvés ainsi.

$$\begin{aligned}
& f_4, t f_4, T_z f_4, T_y f_4, T_x f_4, t T_z f_4, t T_y f_4, t T_x f_4, \\
& T_z T_y f_4, T_z T_x f_4, T_y^2 f_4, T_y T_x f_4, T_x^2 f_4, t T_z T_y f_4, \\
& t T_z T_x f_4, t T_y^2 f_4, t T_y T_x f_4, t T_x^2 f_4, f_3, t f_3, T_z f_3, \\
& T_y f_3, T_x f_3, t T_z f_3, t T_y f_3, t T_x f_3, T_z T_y f_3, T_z T_x f_3, \\
& T_y^2 f_3, T_y T_x f_3, T_x^2 f_3, t T_z T_x f_3, t T_y^2 f_3, t T_y T_x f_3, \\
& t T_x^2 f_3, f_2, t f_2, T_z f_2, T_y f_2, T_x f_2, t T_z f_2, t T_y f_2, \\
& t T_x f_2, T_z T_y f_2, T_z T_x f_2, T_y^2 f_2, T_y T_x f_2, T_x^2 f_2, \\
& t T_z T_x f_2, t T_y^2 f_2, t T_y T_x f_2, t T_x^2 f_2, f_1, t f_1, T_z f_1, \\
& T_y f_1, T_x f_1, t^2 f_1, t T_y f_1, t T_x f_1, t^3 f_1, t^2 T_y f_1, \\
& t^2 T_x f_1, t^3 T_y f_1, t^3 T_x f_1
\end{aligned}$$

Grâce à la propriété selon laquelle la division par l'idéal I est bien définie quand nous le faisons par rapport à une base de Gröbner de I , nous pouvons considérer l'espace de tous les restes sur division par I (cf. (Cox *et al.*, 2007)). Cet espace est appelé *l'anneau quotient* de I , et nous le dénotons par $A = \mathbb{C}[T_x, T_y, T_z, t]/I$. C'est une propriété classique, que si I est radical, alors le système $f_1 = \dots = f_4 = 0$ a un nombre fini de solutions N si et seulement si la dimension de A en tant qu'un espace \mathbb{C} -vectoriel est N (cf. (Cox *et al.*, 2007), Proposition 8 page 235). On peut vérifier facilement par la fonction `IsRadical` de MAPLE que I est radical. Une base pour A en tant qu'un espace vectoriel peut être obtenu en utilisant `in(I)` par ((Cox *et al.*, 2007), Théorème 6, page 234)

$$B = \{m \mid m \text{ est un monôme et } m \notin \text{in}(I)\}$$

Par le calcul d'une base de Gröbner de I , nous pouvons calculer `in(I)` qui est égal à `in(I) = \langle T_x, T_y, T_z^2, t^6 \rangle` et donc l'ensemble

$$B = \{1, t, t^2, t^3, t^4, t^5, T_z, T_z t, T_z t^2, T_z t^3, T_z t^4, T_z t^5\}$$

est une base pour A comme un espace vectoriel sur \mathbb{C} . Par conséquent, nous pouvons conclure que le système $f_1 = \dots = f_4 = 0$ a 12 solutions. Notons que nous avons obtenu ces résultats pour un jeu particulier de coordonnées de points en entrée. Discutons maintenant ces résultats pour tout ensemble de points. Considérons $\widehat{m}_x^1, \dots, \widehat{m}_z^3$ comme étant les paramètres, et I comme un idéal de l'anneau $\mathbb{C}(\widehat{m}_x^1, \dots, \widehat{m}_z^3)[T_x, T_y, T_z, t]$. Soit G une base de Gröbner de

$$I \subset K[\widehat{m}_x^1, \dots, \widehat{m}_z^3][T_x, T_y, T_z, t]$$

pour un ordre monomial de degré par bloc de DRL(T_x, T_y, T_z, t) et DRL($\widehat{m}_x^1, \dots, \widehat{m}_z^3$), et soit C_i le produit des coefficients initiaux de ses éléments considérés comme polynômes en T_x, T_y, T_z, t avec ses coefficients dans $K[\widehat{m}_x^1, \dots, \widehat{m}_z^3]$. C'est un résultat classique à prouver, si $C_i(\widehat{m}_x^1, \dots, \widehat{m}_z^3) \neq 0$, alors la substitution des paramètres par leur valeurs dans G_i donne une base de Gröbner de I (cf. (Cox *et al.*, 2007) p. 288 par exemple). Donc toutes propriétés géométriques de l'idéal obtenues pour des points particuliers (que nous avons étudiés précédemment) sont vraies pour presque tout idéal obtenu par un autre jeu de points, et nous disons que ces propriétés sont *génériquement* vraies. Par conséquent, tout idéal ainsi obtenu est radical et son idéal initial est égal à $\langle T_x, T_y, T_z^2, t^6 \rangle$, et son système correspondant a 12 solutions. Nous rappelons ici brièvement la méthode de la valeur propre que nous utilisons pour résoudre le système $f_1 = \dots = f_4 = 0$, cf. (Cox *et al.*, 1998), page 56 pour plus de détails. Pour tout $f \in \mathbb{C}[T_x, T_y, T_z, t]$ nous notons par $[f]$ le coset de f dans A . Nous définissons $m_f : A \rightarrow A$ par la règle suivante :

$$m_f([g]) = [f].[g] = [fg] \in A$$

Donc, l'idéal engendré par les f_i est de dimension zéro, A est un espace vectoriel de \mathbb{C} de dimension finie, nous pouvons alors présenter m_f par une matrice qui est appelée la *matrice action* de f . Pour tout i , si nous prenons $f = x_i$, les valeurs propres des m_{x_i} sont les x_i coordonnées des solutions du système. En utilisant ces valeurs propres pour chaque i , et un test pour vérifier si un $n - \text{uplet}$ de ces valeurs propres annule ou pas les f_i , nous pouvons trouver les solutions du système. Une méthode plus efficace est d'utiliser les vecteurs propres. Soit f une forme linéaire générique dans A , alors nous pouvons prendre directement toutes les solutions du système parmi les vecteurs propres de m_f , cf. (Cox *et al.*, 1998), page 64.

6.3. Calcul de l'orientation relative finale

Après la résolution du système polynomial, et l'obtention des paramètres T_x, T_y, T_z et t , il est possible de calculer l'orientation relative finale entre les deux images. Si nous supposons que R_{ver}^1 est la matrice de rotation définie dans la section 5.2 pour l'image 1, R_{ver}^2 la même pour l'image 2, et R_ϕ la matrice de la rotation définie par t (équation 4), l'orientation relative définitive entre les images 1 et 2 est :

$$R_{final} = R_{ver}^2 {}^t R_\phi R_{ver}^1, \tag{6}$$

$$\overrightarrow{T}_{final} = R_{ver}^2 {}^t \overrightarrow{T}, \text{ avec } \overrightarrow{T} = [T_x, T_y, T_z]^t.$$

7. Tests expérimentaux

La précision du calcul de l'orientation relative, en utilisant un point de fuite vertical et 3 points de liaison, vient de trois facteurs :

- 1) la précision de la résolution polynomiale des paramètres de la translation ($T_x T_y T_z$), et de la rotation autour de l'axe Y en utilisant les bases de Gröbner,
- 2) la précision géométrique de l'estimation de la direction verticale,
- 3) la précision de l'algorithme sur les points de liaison en présence de bruit.

Pour évaluer ces différents aspects, nous avons dans un premier temps travaillé sur des données synthétiques, dans la Section 7.1, puis nous avons utilisé de vraies données, dans la section 7.2.

7.1. Performances en présence de bruit

Dans cette section, la performance de la méthode des 3 points en présence de bruit a été étudiée et comparée à l'algorithme de 5 points (Stewenius *et al.*, 2006) en utilisant le logiciel fourni par les auteurs (Stewenius, n.d.). Le dispositif expérimental employé est semblable à (Nistér, 2004). La distance au volume de la scène est utilisée comme unité de mesure, et la longueur de la ligne de base est de 0.3. L'écart type du bruit est exprimé en pixels, pour une image 352×288 , avec $\sigma = 1.0$ pixel. Le champ de vue est égal à 45 degrés. La profondeur varie entre 0 et 2. Deux valeurs différentes de translation ont été traitées, une en X (mouvement transverse) et une en Z (mouvement avant-arrière). Les tests impliquent 2 500 tirages aléatoires de points homologues. Pour chaque tirage, nous déterminons l'angle entre la ligne de base

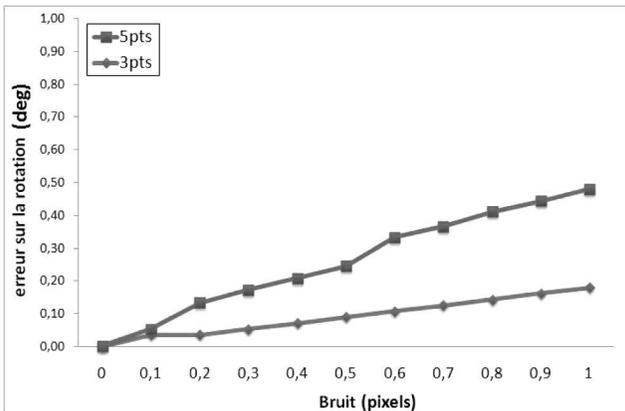


Figure 4. Erreur sur la rotation de la base (en degrés, mouvement transverse)

estimée et le vecteur de la ligne de base vraie. Cet angle est appelé ici erreur translationnelle, et est exprimé en degrés. Pour l'estimation de l'erreur sur la matrice de la rotation, l'angle de $(R_{true}^T R_{estimate})$ est calculé, et la valeur moyenne résultant des 2 500 tirages aléatoires pour chaque niveau de bruit est donnée. D'après les figures 4, 5, 6 et 7, nous voyons que l'algorithme de 3 points est plus robuste, si on tient compte des erreurs causées par le bruit, tant en mouvement transverse qu'en mouvement avant-arrière, pour l'estimation de la rotation et de la translation.

Maintenant, nous comparons l'algorithme des 3 points et l'algorithme des 5 points sur une scène plane. Dans cette configuration tous les points de l'espace objet ont le même Z (ici égal à 2). Les résultats pour l'estimation de la rotation (figure 8) montrent que les deux algorithmes fournissent une bonne détermination de la

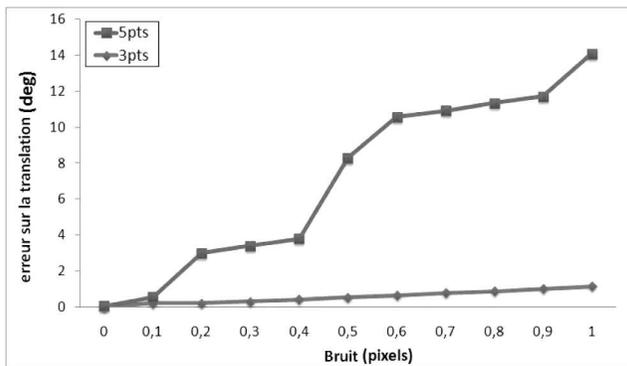


Figure 5. Erreur sur l'orientation de la base (en degrés, mouvement transverse)

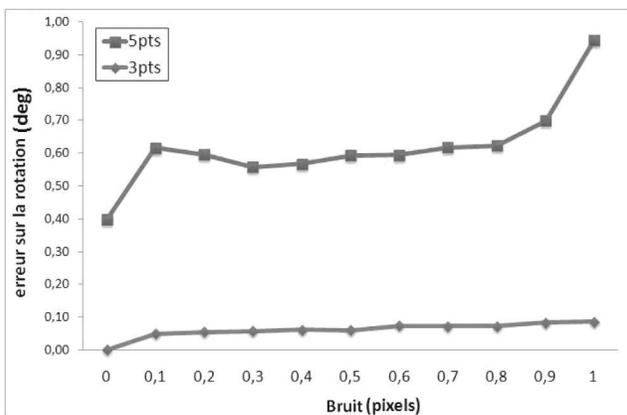


Figure 6. Erreur sur la rotation (en degrés, mouvement avant-arrière)

rotation, mais la méthode des 3 points donne des résultats bien meilleurs que celle des 5 points pour la détermination de l'orientation de la base en mouvement transverse (figure 9). Cette faiblesse de l'algorithme des 5 points pour une scène plane a été discutée dans (Segvic *et al.*, 2007).

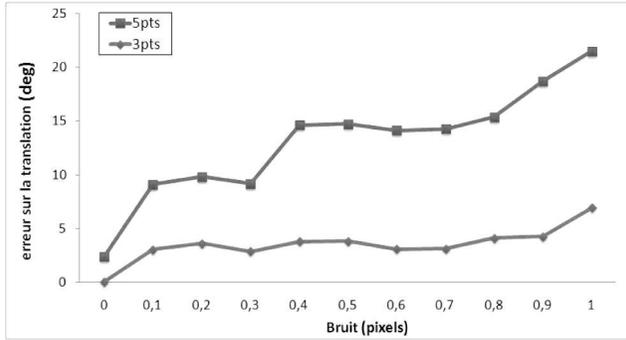


Figure 7. Erreur sur l'orientation de la base (en degrés, mouvement avant-arrière)

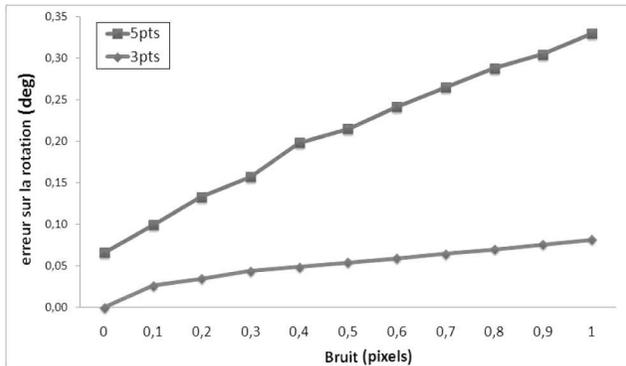


Figure 8. Erreur sur la rotation (en degrés) en configuration plane (mouvement transverse)

Impact de la précision de la direction verticale sur l'estimation d'orientation relative

Nous avons introduit une erreur de 0 à $0,5^\circ$ sur la précision angulaire de la direction verticale. Aujourd'hui par exemple, un dispositif inertiel à bas prix tel que le Xsens-MTi ([http : \www.xsens.com/](http://www.xsens.com/), n.d.) donne une précision autour de $0,5^\circ$ sur l'angle de rotation autour des axes X ou Z (la direction verticale étant selon l'axe Y). Bien sûr, il existe des centrales inertielles de grande précision, elles peuvent atteindre mieux que $0,01^\circ$ sur les angles d'orientation quand elles sont associées correctement avec d'autres capteurs capables de compenser leur dérive (par exemple le GPS). En

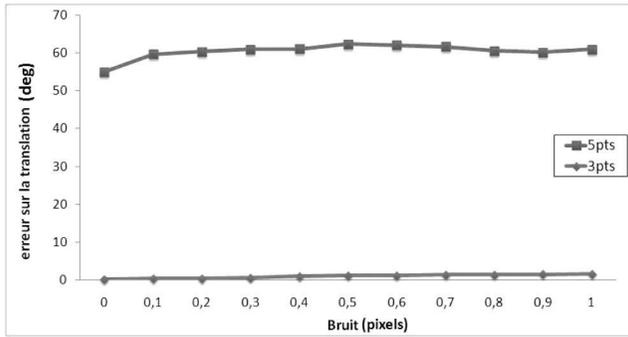


Figure 9. Erreur sur l'orientation de la base (en degrés, mouvement transverse)

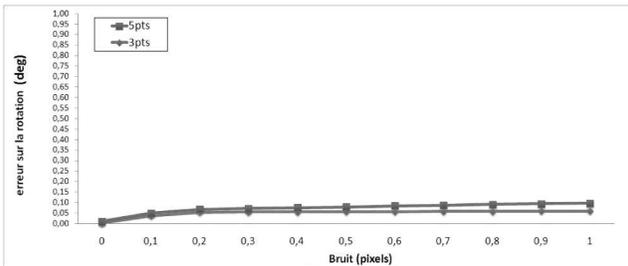


Figure 10. Erreur sur la rotation (en degrés) en configuration plane (mouvement avant-arrière)

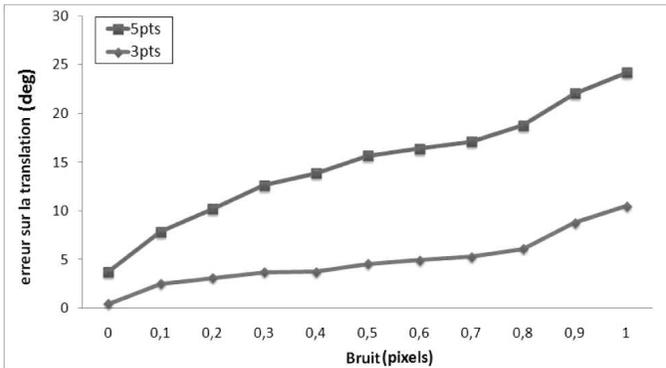


Figure 11. Erreur sur l'orientation de la base (en degrés) en configuration plane (mouvement avant-arrière)

utilisant l'extraction automatique du point de fuite vertical, particulièrement dans une scène urbaine, nous obtenons une direction verticale très précise (bien mieux que $0,01^\circ$), comme ce sera montré plus loin. Nous avons vérifié l'impact de cette précision sur la détermination de la rotation et de la base (figures 12 et 13).

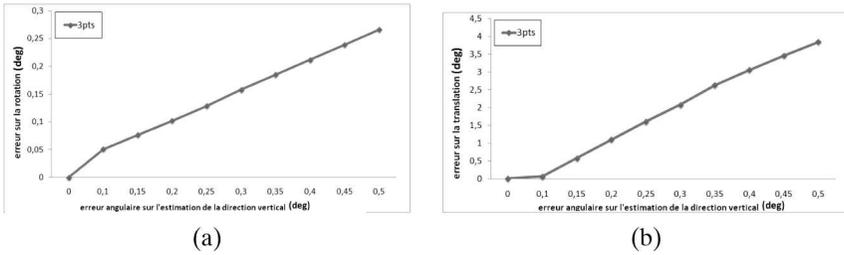


Figure 12. Impact de la précision géométrique sur la direction verticale sur l'estimation de a) la rotation (en degrés), et b) l'orientation de la base (en degrés) en mouvement transverse

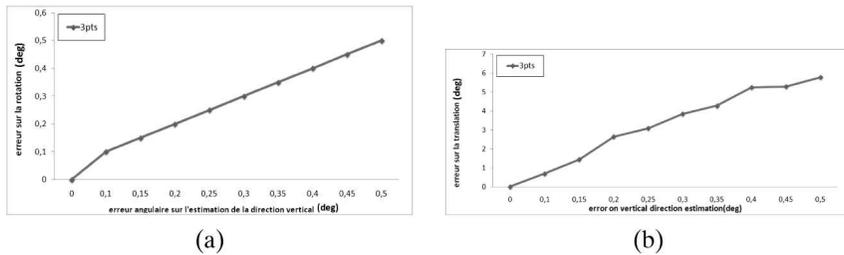


Figure 13. Impact de la précision géométrique sur la direction verticale sur l'estimation de a) la rotation (en degrés), et b) l'orientation de la base (en degrés) en mouvement avant-arrière

7.2. Exemple avec des données réelles

Afin de fournir un exemple numérique sur de vraies images, nous avons choisi de travailler sur la séquence de 9 images « entry-P10 » de la base de données en ligne (Strecha *et al.*, 2008). Dans cette base de données nous disposons de tous les paramètres intrinsèques et externes. En premier, nous avons extrait les points de fuite sur chaque image. Nous avons utilisé l'algorithme de (Kalantari *et al.*, 2008) parce qu'au-delà de sa grande rapidité, il permet une propagation d'erreur sur les points de fuite à partir de l'erreur sur la détection des segments. Nous exprimons cette erreur en termes d'angles, et les résultats sont présentés dans le tableau 1. Comme on peut le voir, la détermination du point de fuite vertical est très précise et d'après les figures 12 et 13 elle induit une erreur proche de zéro.

Ensuite, nous avons calculé l'orientation relative pour 3 images consécutives (chaque fois, 2 couples d'images successives). Les points d'intérêt sont extraits en utilisant l'algorithme SIFT (Lowe, 2004). Les résultats sont présentés dans la figure 14. La valeur moyenne des erreurs angulaires sur la rotation s'élève à 0,82 degré. Pour l'estimation de la direction de la base, cette erreur monte à 1,33 degré. Ces résultats montrent clairement l'efficacité et la robustesse de la méthode.

Tableau 1. Résultats : détection de la direction verticale en utilisant le point de fuite

Image	Erreur angulaire sur la direction verticale, en degrés
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

Error on the rotation (°) / Error on the translation (°)	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009
0000 	—	0.39 / 0.30	0.19 / 1.75							
0001 		—	0.30 / 1.94	0.35 / 1.61						
0002 			—	0.17 / 1.99	1.51 / 2.81					
0003 				—	0.61 / 0.001	0.58 / 1.81				
0004 					—	0.35 / 1.10	0.64 / 0.31			
0005 						—	0.54 / 0.65	1.16 / 1.79		
0006 							—	1.94 / 0.66	0.86 / 1.02	
0007 								—	0.74 / 0.89	2.21 / 1.21
0008 									—	1.42 / 2.82
0009 										—

Figure 14. Résultat sur la séquence « entry-P10 ». Chaque cellule contient l'erreur sur la rotation en degrés (en haut à gauche) et celle sur l'orientation de la base (en bas à droite)

7.3. Performances en temps de calcul

La résolution du système polynômial et l'extraction automatique du point de fuite ont été écrites en C++. Avec un PC à 1,60 GHz, avec 2 Go de RAM le temps de chaque résolution est approximativement de $2 \mu s$, permettant des applications temps réel. Nous pouvons noter, sans surprise, que le processus de sélection qui utilise RanSac (Fischler *et al.*, 1981) parmi les points SIFT est considérablement plus rapide sur 3 points qu'avec l'algorithme de 5 points.

8. Résumé et conclusions

Aujourd'hui les appareils personnels, comme les téléphones cellulaires dont les prix sont de plus en plus bas, incluent des systèmes MEMS et inertiels en plus d'un appareil photo, permettant de fournir très facilement la direction de la verticale dans l'image. En outre, si besoin est, l'extraction automatique du point de fuite vertical par traitement d'image offre une alternative de très bonne précision. Nous avons donc démontré l'avantage d'utiliser la connaissance de la direction verticale, et présenté un algorithme effectif pour résoudre le problème de l'orientation relative à partir de cette information. En plus d'une vitesse de calcul bien plus élevée, par comparaison avec la solution classique des 5 points, notre algorithme fournit une précision nettement meilleure pour l'estimation de la base. Une autre amélioration intéressante a été démontrée : les scènes planes ne posent plus de problème pour l'estimation de la base. Ce résultat intéressant, particulièrement dans les scènes urbaines où le cas se présente très souvent, est dû à une formulation du problème plus appropriée, qui utilise de façon explicite les paramètres significatifs de l'orientation relative (paramètres de rotation et de translation).

Remerciements

*A. Hashemi remercie le CEAMA, de l'université technique d'Isfahan.
M. Kalantari remercie ici Arnaud Le Bris, du Laboratoire MATIS, pour l'aide apportée à l'acquisition des images ayant servi aux tests et aux calculs de mise en œuvre de la méthode SIFT.*

Bibliographie

- Antone M., Teller S. (2000). « Automatic Recovery of Relative Camera Rotations for Urban Scenes », vol. 02, *IEEE Computer Society*, Los Alamitos, CA, USA, p. 282-289.
- Barnard S. T. (1983). « Interpreting Perspective Images », *Artificial Intelligence*, vol. 21, p. 435-462.
- Batra D., Nabbe B., Hebert M. (2007). « An Alternative Formulation for Five Point Relative Pose Problem », p. 21-21.
- Buchberger B. (1965). Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal, PhD thesis, Universität Innsbruck.

- CocOA, A System for doing Computations in Commutative Algebra, Technical report, n.d.
<http://cocoa.dima.unige.it>.
- Cox D., Little J., O’Shea D. (1998). *Using algebraic geometry*, vol. 185 of *Graduate Texts in Mathematics*, Springer-Verlag, New York.
- Cox D., Little J., O’Shea D. (2007). *Ideals, varieties, and algorithms*, Undergraduate Texts in Mathematics, third edn, Springer-Verlag, New York. An introduction to computational algebraic geometry and commutative algebra.
- Demazure M. (1988). Sur Deux Problemes de Reconstruction, Technical Report n° 882, INRIA.
- Faugeras O. (1993). *Three-Dimensional Computer Vision : A Geometric Viewpoint*, MIT Press.
- Faugeras O. D., Maybank S. (1990). « Motion from point matches : multiplicity of solutions », *International Journal of Computer Vision*, vol. 4, n° 3, p. 225-246.
- Fischler M., Bolles R. (1981). « Random Sample Consensus : A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography », *Comm. of the ACM*, vol. 24, n° 6, p. 381-395, June.
- Grayson R. D., Stillman E. M. (1996). « Macaulay 2, a Software System for Research in Algebraic Geometry », Available at <http://www.math.uiuc.edu/Macaulay2>.
- Greuel G.-M., Pfister G., Schönemann H. (2005). SINGULAR 3.0, A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern. <http://www.singular.uni-kl.de>.
- Hartley R. I., Zisserman A. (2004). *Multiple View Geometry in Computer Vision*, second edn, Cambridge University Press, ISBN : 0521540518.
- Heyden A., Sparr G. (1999). « Reconstruction from Calibrated Cameras-A New Proof of the Kruppa-Demazure Theorem », *Journal of Mathematical Imaging and Vision*, vol. 10, p. 123- 142(20), March. <http://www.xsens.com/n.d>.
- Kalantari M., Jung F., Paparoditis N., Guédon J. (2008). « Robust and Automatic Vanishing Points Detection with their Uncertainties from a Single Uncalibrated Image, by Planes Extraction on the Unit Sphere », *IAPRS*, vol. 37 (Part 3A), Beijing, China, p. 203-208, jul.
- Kasser M., Egels Y. (2001). *Digital Photogrammetry*, Taylor & Francis, Inc., Bristol, PA, USA.
- Kruppa E. (1913). « Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung », *Other*, vol., p. 1939-1948.
- Kukelova Z., Bujnak M., Pajdla T. (2008). « Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems ».
- Lazard D. (1983). « Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations », *Computer algebra (London, 1983)*, vol. 162 of *Lecture Notes in Comput. Sci.*, Springer, Berlin, p. 146-156.
- Li H., Hartley R. (2006). « Five-Point Motion Estimation Made Easy », p. I : 630-633.
- Lobo J., Dias J. (2003). « Vision and Inertial Sensor Cooperation Using Gravity as a Vertical Reference », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, n° 12, p. 1597-1608.
- Longuet Higgins H. (1981). « A Computer Algorithm for Reconstructing a Scene from Two Projections », *Nature*.
- Lowe D. (2004). « Distinctive Image Features from Scale-Invariant Keypoints », *International Journal of Computer Vision*, vol. 60, n° 2, p. 91-110, November.

- Lutton E., Maitre H., Lopez-Krahe J. (1994). « Contribution to the Determination of Vanishing Points Using Hough Transform », *IEEE Trans. Pattern Anal. Mach. Intell.*, avril.
- Ma Y., Soatto S., Kosecka J., Sastry S. (2003). *An invitation to 3D vision, from images to models*, Springer-Verlag.
- Maybank S. (1992). *Theory of Reconstruction from Image Motion*, Springer-Verlag.
- Nistér D. (2004). « An Efficient Solution to the Five-Point Relative Pose Problem », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, n° 6, p. 756-777, June.
- Philip J. (1996). « A non-iterative algorithm for determining all essential matrices corresponding to five point pairs », *Photogrammetric Record*, vol. 15, n° 88, p. 589-599.
- SALSA, Solvers for ALgebraic Systems and Applications, Technical report, n.d. <http://fgbrs.lip6.fr/salsa/>.
- Schaffalitzky F., Zisserman A. (2000). « Planar Grouping for Automatic Detection of Vanishing Lines and Points », *Image and Vision Computing*, vol. 18, p. 647-658.
- Segvic M., Schweighofer G., Pinz A. (2007). « Performance evaluation of the five-point relative pose with emphasis on planar scenes », *Performance Evaluation for Computer Vision*, Workshop of the Austrian Association for Pattern Recognition, Austria, p. 33-40.
- Shufelt J. A. (1999). « Performance Evaluation and Analysis of Vanishing Point Detection Techniques », *IEEE transactions PAMI*, vol. 21, n° 3, p. 282-288, March.
- Stewenius H., « Matlab code for solving the fivepoint problem », n.d. <http://vis.uky.edu/~stewe/FIVEPOINT/>.
- Stewenius H., Engels C., Nistér D. (2006). « Recent Developments on Direct Relative Orientation », *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, n° 4, p. 284-294.
- Strecha C., von Hansen W., Van Gool L., Fua P., Thoennessen U. (2008). « On benchmarking camera calibration and multi-view stereo for high resolution imagery », p. 1-8.
- Sturm R. (1869). « Das Problem der Projektivitat und seine Anwendung auf die Flächen Zweiten Grades », *Math. Annalen*, vol. 1, p. 533-573.
- Triggs B. (2000). Routines for Relative Pose of Two Calibrated Cameras from 5 Points, Technical report, INRIA.
- van den Heuvel F. A. (1998). « Vanishing Point Detection For Architectural Photogrammetry », *International Archives of Photogrammetry and Remote Sensing*, vol. 32, n° 5, p. 652-659.
- Vieville T., Clergue E., Facao P. (1995). « Computation of Ego Motion Using the Vertical Cue », vol. 8, n° 1, p. 41-52.
- Wolf P., de Witt B. (2000). *Elements of Photogrammetry with Applications in GIS*, McGraw-Hill, February.



Mahzad Kalantari, née en 1982 à Ispahan (Iran), a obtenu son diplôme d'ingénieur en géomatique à l'université d'Ispahan, puis son grade de Master en information géographique à l'Ecole Nationale des Sciences Géographiques – université Paris Est en 2005. Sa thèse de doctorat en informatique, soutenue en 2009, portait sur la photogrammétrie et la vision par ordinateur. Elle suit actuellement une spécialisation post-doctorale en électronique.



Amir Hashemi est né en 1975. Il a obtenu sa licence en mathématiques à l'université de technologie de Sharif (Iran), son master en mathématiques à l'université de technologie d'AmirKabir (Iran), et son doctorat en informatique à l'université de Paris 6. Ses activités de recherche sont consacrées à l'amélioration de l'efficacité du calcul des bases de Gröbner et à la résolution des systèmes polynomiaux, et leur application. Il est actuellement professeur assistant à l'université de technologie d'Isfahan en Iran.



Franck Jung est né en 1970 à Strasbourg. Diplômé de l'école polytechnique, il a obtenu un doctorat en mathématiques appliquées sur des problématiques d'apprentissage appliquées au traitement d'images. Après avoir dirigé l'école supérieure des géomètres et topographes (ESGT) au Mans, il occupe actuellement le poste de sous-directeur en charge de l'animation scientifique et technique au ministère de l'écologie. Ses centres d'intérêts portent sur les problématiques d'apprentissage, d'inférence statistique et les techniques d'optimisation.



Jeanpierre Guédon est né à Grosbreuil (85) en 1962. Après des études en mathématiques et informatique à l'université de Nantes, il a obtenu un master en automatique et informatique et un doctorat de l'ECN-UN sur la tomographie. Après un postdoc à la FDA aux USA, il est devenu enseignant chercheur à l'école polytechnique de l'université de Nantes. Son champ principal d'activité est la transformée Mojette (trans. de Radon discrete exacte) qu'il a développé depuis plus de quinze ans. Il a lancé le jeu Mojette en 2010.