Le Calcul de Pose: de nouvelles méthodes matricielles

Marc-André Ameller, Long Quan, Bill Triggs

Le Calcul de Pose : de nouvelles méthodes matricielles

Camera Pose Revisited : new linear algorithms

Marc-André Ameller Long Quan Bill Triggs

INRIA Rhônes-Alpes

INRIA Rhône-Alpes, 655, av. de l'Europe
38334 St. Ismier cedex, France.
prénom.nom@inria.fr

Résumé

Estimer la pose d'une caméra signifie calculer sa position et son oritentation, à l'aide de ses paramètres internes, ainsi que de la connaissance de la position de points 3D de référence et de leurs projections dans l'image. On va brièvement passer en revue différentes méthodes existantes pour le calcul de la pose, puis on introduira 4 nouveaux algorithmes. Tous ces algorithmes sont basés sur l'algèbre linéaire. Le premier est basé sur le calcul de valeur propre d'une matrice 5×5. Elle permet de résoudre le problème de pose avec 3 points et donc, donne plusieurs solutions. Les 3 autres algorithmes, qui permettent le calcul de la pose à partir de quatre points, donne une unique solution, qui est le noyau d'une matrice. Ce noyau est calculé à l'aide d'une SVD. Ces techniques sont basées sur les matrices de résultants : la méthode 24×24 est une méthode de résultant de Macaulay, et les méthodes 12×12 et 9×9 sont des versions compressées, obtenues après élimination gaussienne. Un des avantages de ces méthodes est leur simplicité. En particulier, les coefficients des matrices ne sont que des fonctions simples des données. Les expériences numériques donnent une comparaison entre les nouveaux algorithmes et ceux déja existant.

Mots clés : Calibration, Estimation de la pose, Résolution de systèmes de polynômes, Matrices de résultant, Résection.

Abstract

Camera pose estimation is the problem of determining the position and orientation of an internally calibrated camera from known 3D reference points and their images. We briefly survey existing methods for pose estimation, then introduce four new algorithms based on efficient matrix computations. The first is based on eigendecomposition of a 5×5 matrix and returns the four intrinsic solutions to the problem of pose from 3 points. The remaining three methods give a unique linear solution from four points by SVD null space computation on resultant matrices. The 24×24 method is the raw resultant matrix, and the 12×12 and 9×9 methods are compressed versions of this obtained by Gaussian elimination with pivoting on constant entries. All of these methods are simple to implement. In particular, the matrix entries are simple functions of the input data. Numerical experiments are given comparing the performance of the new algorithms with several existing methods.

Keywords : Calibration, Pose Estimation, Resection, Polynomial Solving, Resultant Matrices.

1 Introduction

L'estimation de la pose d'une caméra consiste à déterminer la position et l'orientation de la caméra à partir de la calibration de celle-ci, ainsi que de la donnée de coordonnées de points de référence et de la position respective de leurs projections sur l'image. Ce problème est aussi appelé *resection* par les photogramètres, et a été bien étudié dans le passé. Avec 3 points, le problème de pose possède dans le cas général 4 solutions. De nombreuses méthodes sont connues pour calculer ces solutions. La plus ancienne est probablement due à Lagrange en 1795. Voir [20, 7, 8, 9, 17]. Fischler & Bolles [7] ont donné une méthode devenue populaire en vision par ordinateur, lorsqu'ils ont introduit la méthode RANSAC pour la détection des valeurs abérrantes dans les données initiales. Haralick *et al* [9] a montré différentes variations, nouvelles et anciennes, de la méthode basique utilisant 3 points, et étudié les différentes stabilités suivants la manière de procéder aux substitutions et éliminations. Pour les problèmes redondants, des méthodes itératives ont été développées [13, 21, 3]. Des méthodes ont aussi été développées pour le calcul de pose à partir de correspondances de lignes et d'autres primitives. [10, 4, 1, 14, 12].
La méthode de 3 points possède des solutions multiples. Si l'on veut obtenir une solution unique, il faut ajouter des

données. Par exemple, si l'on ajoute un quatrième point de référence, en général, il existe une solution unique. Aussi, dans certains cas dégénérés, même un nombre infini de points ne donne pas de solution unique. Ces *configurations critiques* sont connues précisément. Voir [18, 20] pour les détails. Brièvement, ces configurations correspondent au cas où la caméra est sur une cubique twisté (l'horoptère) dans l'espace, qui est tracé sur un cylindre (Le cylindre dangereux). On peut noter les cas suivants qui sont dégénérés : (1) tous les points sont à l'infini (la translation de la caméra ne peut être estimée); (2) une droite et un cercle, avec la droite orthogonale au plan du cercle. Ce dernier cas est particulièrement ennuyeux, pour le calcul de pose à partir de 3 points, ou à partir de 4 points coplanaires et formant un rectangle, lorsque la caméra est dans le proche voisinage des points. On montrera les effets de ces cas dégénérés dans quelques expériences.

L'article est motivé par le fait qu'il n'y ait que peu de méthodes donnant directement la solution unique du problème de pose dans le cas de données redondantes. Des familles d'algorithmes linéaires sont présentées dans [16, 19]. Malheureusement, ces méthodes possèdent les mêmes inconvénients que les méthodes algébriques en ce sens que les coefficients des matrices utilisées sont compliqués, extraits de polynômes de degré 4, ce qui alourdit considérablement l'implémentation. Dans cet article, on propose : (1) un nouvel algorithme pour le calcul de pose à partir de 3 points, basé sur le calcul des valeurs propres d'une matrice 5×5 obtenue par la méthode de résultant de Bezout-Cayley-Dixon; (2) Trois algorithmes linéaires pour le calcul de pose à partir de 4 points, basé sur le calcul du noyau de matrices 9×9, 12×12 et 24×24, dont les coefficients sont simples.

L'article est organisé comme suis. Dans la section 2, on reprend les bases géométriques du calcul de pose. La section 3 passe en revue une méthode pour résoudre linéairement des systèmes de polynômes. Dans la section 4, on présente une méthode de valeurs propres pour la méthode de 3 points. Les 3 nouveaux algorithmes linéaires sont présentés dans la section 5. La section 6 donne quelques premiers résultats d'expérimentation qui comparent ancienne et nouvelles méthodes, et ceci sur des données simulées. Enfin, la section 7 résume les contributions et donne quelques conclusions.

2 Géométrie pour le calcul de pose à partir de points

Étant donnée une caméra calibrée centrée en $\mathbf{c}$ et des correspondances entre des points de référence 3D $\mathbf{p}_i$ et leurs images $\mathbf{u}_i$. Chaque paire de correspondances de points donne une contrainte sur les distances inconnues entre les points de référence et le centre de la caméra, $x_i = ||\mathbf{p}_i - \mathbf{c}||$. Les

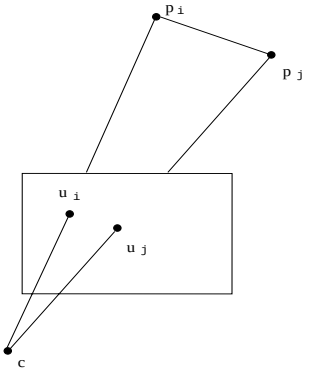

FIG. 1 – *La détermination de la géométrie de base pour le calcul de pose à partir de paires de correspondances de points $\mathbf{p}_i \leftrightarrow \mathbf{u}_i$ et $\mathbf{p}_j \leftrightarrow \mathbf{u}_j$ entre des points 3D et leurs projections dans les images.*

contraintes s'écrivent (cf. Figure 1) :

$$P_{ij}(x_i,x_j) \ :\equiv: {x_i}^2 + {x_j}^2 + c_{ij}\, x_i\, x_j - {d_{ij}}^2 \ :=: 0 \quad (1)$$
$$c_{ij} \ :\equiv: -2\, \cos\theta_{ij} \quad (2)$$

où $d_{ij} = ||\mathbf{p}_i - \mathbf{p}_j||$ est la distance connue entre les points de référence numéro i et j, et θ_{ij} l'angle entre les demi-droites $[\mathbf{cp}_i)$ et $[\mathbf{cp}_j)$. Le cosinus de cet angle peut aisément être déduit de la position des projections des points dans l'image et de la matrice $\mathbf{K}$ de calibration de la caméra de la manière suivante :

$$\cos\theta_{ij} \ :=: \frac{\mathbf{u}_i^T \mathbf{C}\, \mathbf{u}_j}{\sqrt{(\mathbf{u}_i^T \mathbf{C}\, \mathbf{u}_i)(\mathbf{u}_j^T \mathbf{C}\, \mathbf{u}_j)}},$$

où $\mathbf{C} = (\mathbf{K}\, \mathbf{K}^T)^{-1}$.

Pour $n = 3$, on obtient le système de polynômes suivant.

$$\begin{cases} P_{12}(x_1,x_2) = 0, \\ P_{13}(x_1,x_3) = 0, \\ P_{23}(x_2,x_3) = 0 \end{cases}$$

où les trois inconnues sont les distances x_1,x_2,x_3. Le théorème de Bezout [2] borne le nombre de solutions du système à $8 = 2^3$ solutions. Cependant, n'ayant que des termes en des puissances paires des x_i, le système est invariant par la transformation $x_i \mapsto -x_i$. Ainsi, les solutions peuvent être classées par paires de solutions opposées. En utilisant le résultant de Sylvester 2 fois, on peut éliminer les variables x_2 et x_3, de manière à obtenir un polynôme de degré 8 en x_1, dont tout les termes ont un degré pair, c'est-à-dire un polynôme de degré 4 en $x = x_1^2$:.

$$g(x) = a_4\, x^4 + a_3\, x^3 + a_2\, x^2 + a_1\, x + a_0 = 0.$$

Cette équation possède au plus 4 solutions en x et peut-être résolue de manière explicite. Comme les x_i sont positifs, $x_1 = \sqrt{x}$.

Dans la méthode, et pour les autres qui vont suivre, on calcule les distances x_i afin d'estimer les coordonnées des points de référence 3D dans un repère attaché et centré sur la caméra : $\tilde{\mathbf{p}}_i = x_i \mathbf{K}^{-1}\mathbf{u}_i$. Pour trouver la pose de la caméra, il suffit alors d'estimer le mouvement rigide qui aligne le mieux les points dans leurs repères respectifs. Les centres de gravité des deux nuages de points donnent la translation. La rotation s'en déduit facilement par la méthode des quaternions ou par SVD [11, 6].

3 Les méthodes issues de l'algèbre linéaire

Les matrices de résultant L'algèbre linéaire est un outil puissant pour manipuler les polynômes [15, 5, 2]. Un polynôme $P_i(x) \ := : \ \sum_\alpha c_{\alpha,i}\, x^\alpha$ en les variables $x \ := : \ (x_1,\ldots,x_n)$ est une somme finie de *coefficients* $c_{\alpha,i}$ multipliés par des *monômes* $x^\alpha \ := : \ x_1^{\alpha_1}\cdot x_2^{\alpha_2}\cdot \ldots \cdot x_n^{\alpha_n}$, avec $\alpha \ := : \ (\alpha_1,\ldots,\alpha_n) \in \mathbb{Z}^n$ qui est appelé multi-index ou vecteur exposant. L'idée clé est de regarder les polynômes comme produits de vecteurs lignes de coefficients par des vecteurs colonnes de monômes. De manière similaire, on peut voir un système de polynôme comme une matrice de coefficients que multiplie un vecteur colonne de monômes. La structure interne des polynômes n'apparaît ainsi que de manière implicite. C'est une manière de "sur-paramètrer" le problème, mais qui a l'avantage d'être linéaire.

Considérons un système de polynômes quelconque :

$$\begin{cases} P_1(x_1,\ldots,x_n) &=& 0 \\ & \vdots & \\ P_q(x_1,\ldots,x_n) &=& 0, \end{cases}$$

On va résoudre ce système par linéarisation. Pour cela, on va choisir des polynômes $(Q_j)_{j\in\{1,\ldots,n\}}$ de l'idéal engendré par les P_i, de sorte que n soit plus grand que le nombre de monômes distincts présents dans l'écriture des Q_j. Avec les polynômes Q_j, on peut appliquer le procédé expliqué ci-dessus, c'est-à-dire écrire le système de polynômes $\{Q_j = 0, j \in \{1,\ldots,n\}$ sous la forme d'un produit d'une matrice de coefficients par un vecteur colonne de monôme :

$$\mathbf{M}\mathbf{v} = 0.$$

Comme les polynômes Q_j sont choisit en plus grand nombre que les monômes, la matrice $\mathbf{M}$ possède plus de ligne que de colonnes. La matrice $\mathbf{M}$ s'appelle matrice résultante du système de polynômes. Dans les cas favorables, le système possède une unique solution, et le noyau de la matrice résultante est de dimension 1. Dans ce cas, l'unique solution du système de polynômes est facile à extraire. Les polynômes permettant la construction de la matrice résultante, peuvent être soit trouvés à la main, soit en utilisant une méthode automatique [15]. Bien que ces constructions automatiques donnent des solutions convenables, en général, elles ne garantissent pas la minimalité de la taille de la matrice obtenue. Dans notre cas, la matrice a été construite à la main.

La Méthode de Bezout-Cayley-Dixon Une autre technique pour résoudre les systèmes de polynômes est la méthode de Bezout-Cayley-Dixon [15]. Considérons un système de polynômes général :

$$\begin{cases} P_1(x_1,\ldots,x_n) &=& 0 \\ & \vdots & \\ P_{n+1}(x_1,\ldots,x_n) &=& 0. \end{cases}$$

Introduisons les nouvelles variables $y_1,\ldots,y_n$ et construisons la matrice :

$$\mathbf{M} = \begin{pmatrix} P_1(Z_0) & P_1(Z_1) & \ldots & P_1(Z_n) \\ \vdots & \vdots & & \vdots \\ P_{n+1}(Z_0) & P_{n+1}(Z_1) & \ldots & P_{n+1}(Z_n) \end{pmatrix}.$$

Où, $Z_i = (y_1,\ldots,y_i,x_{i+1},\ldots,x_n)$, avec $i \in \{0,\ldots,n\}$. Dans chaque colonne, on convertit un x de plus en un y (Le résultat dépend en général de l'ordre des variables qu'on choisit). Si $(x_1,\ldots,x_n)$ est une solution du système, la première colonne s'annule et ainsi $\det(\mathbf{M}) = 0$. En plus, le déterminant est divisible par $(x_1 - y_1)\ldots(x_n - y_n)$ parce que $P_i(\ldots,x_i,\ldots) - P_i(\ldots,y_i,\ldots)$ est divisible par $(x_i - y_i)$ (s'annule si $y_i = x_i$). Si nous posons :

$$P(x,y) = \frac{\det \mathbf{M}}{\prod_{i=1}^n (x_i - y_i)}$$

$P(x,y)$ est bien un polynôme, et nous avons :

$$P(x,y) \ := : \ \sum c_{\alpha,\beta} x^\alpha y^\beta \ := : \ \mathbf{w}^T \mathbf{C}\, \mathbf{v}$$

où $\mathbf{v} = (\ldots x^\alpha \ldots)^T$ et $\mathbf{w} = (\ldots y^\beta \ldots)^T$ sont des vecteurs de monômes et $\mathbf{C}$ est une matrice de coefficients $c_{\alpha,\beta}$. Résoudre le système polynomial se réduit à la résolution du système linéaire

$$\mathbf{C}\cdot\mathbf{v} = 0.$$

En effet, comme les y peuvent prendre des valeurs arbitraires, chaque monôme y^α est linéairement indépendant de tous les autres.

4 L'algorithme de vecteur propre pour la pose de 3 points

Nous appliquons maintenant la méthode de Bézout-Cayley-Dixon au calcul de pose à partir de 3 points. Pour cela, on divise chacun des P_{ij} par x_1^2. On pose ensuite :

$$u_1 = 1/x1,\, u_2 = x_2/x_1,\, u_3 = x_3/x_1.$$

On obtient ainsi 3 polynômes Q_{12}, Q_{13}, Q_{23} en les variables u_1, u_2, et u_3. On considère maintenant ces polynômes comme étant des polynômes en u_2,u_3 à coefficients

en u_1 et c_{ij},d_{ij}, et on applique la méthode ci-dessus. Ceci donne :

$$\mathbf{w}^T\mathbf{M}\mathbf{v} = 0, \tag{3}$$

avec :

$$\mathbf{v} = \begin{pmatrix} 1 \\ y_2 \\ y_3 \\ y_2\,y_3 \\ y_2^2 \end{pmatrix} \quad \mathbf{w} = \begin{pmatrix} 1 \\ u_2 \\ u_3 \\ u_3^2 \\ u_2\,u_3 \end{pmatrix}$$

Et, $\mathbf{M} = \mathbf{M}_1 + \mathbf{M}_2 u_1^2$, où :

$$\mathbf{M}_2 = \begin{pmatrix} -d_{23}\,c_{12}\,c_{13} & -d_{13}\,c_{12}\,c_{23}{+}c_{13}\,(d_{12}{-}d_{23}) \\ c_{13}\,(d_{12}{-}d_{23}) & -d_{13}\,c_{23} \\ -d_{12}\,c_{13}\,c_{23}{+}c_{12}\,(d_{13}{-}d_{23}) & -d_{23}{+}d_{13}{+}d_{12} \\ -d_{12}\,c_{23} & 0 \\ -d_{23}{+}d_{13}{+}d_{12} & 0 \end{pmatrix}$$

$$\begin{pmatrix} c_{12}\,(-d_{23}{+}d_{13}) & -d_{23}{+}d_{13}{+}d_{12} & -d_{13}\,c_{23} \\ -d_{23}{+}d_{13}{+}d_{12} & 0 & 0 \\ -d_{12}\,c_{23} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{M}_1 = \begin{pmatrix} 0 & c_{12}\,c_{23}{-}c_{13} & -c_{12} & -2 & c_{23} \\ -c_{13} & c_{23}{-}c_{12}\,c_{13} & -2 & -c_{12} & 0 \\ -c_{12}{+}c_{13}\,c_{23} & -2{+}c_{13}\,c_{12}\,c_{23} & c_{23}{-}c_{12}\,c_{13} & -c_{13} & c_{13}\,c_{23} \\ c_{23} & c_{12}\,c_{23} & 0 & 0 & c_{23} \\ -2 & -c_{12} & -c_{13} & -c_{23} & 0 \end{pmatrix}$$

Résoudre l'équation (3) signifie trouver u_1 tel que :

$$(u_1^{-2}\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_2)\mathbf{v} = 0,$$

puisque $\mathbf{M}_1$ est génériquement non-singulière. Mais ceci est équivalent à trouver les valeurs propres de la matrice $\mathbf{M}_1^{-1}\mathbf{M}_2$. Le calcul de $x_1 = 1/u_1$ se ramène donc a un calcul de valeurs propres.

5 Algorithme linéaire pour la pose de 4 points

Maintenant, nous appliquons les méthodes de résultant au problème de pose. Pour 4 points, le système d'entrée possède 6 équations (1) à 4 variables : $\{P_{ij}(x_i,x_j) := : 0 \mid 1 \leq i < j \leq 4\}$. Chaque solution de ce système satisfait donc :

$$\{\, x_i\, P_{jk}(x_j,x_k) = 0 \,|\, i,j < k = 1,2,3,4\}$$

Ce nouveau système contient 24 polynômes en 24 monômes : 4 de la forme x_i^3, 12 de la forme $x_i^2\, x_j\, x_k$ et 4 de la forme $x_i\, x_j\, x_k$ (où i,j,k sont distincts), et finallement 4 x_i. Il s'avère que (si les entrées c_{ij} et d_{ij} sont correctes) la matrice résultante correspondante 24×24 a génériquemenet un noyau de dimension 1, qui contient les deux solutions algébriques possibles x et $-x$, mais aussi la solution trivialle $x = 0$ qui provient de la multiplication de P_{jk} par x_i. Mais il est facile de choisir la solution désirée, puisque les profondeurs x_i doivent être positives. En détail, la matrice 24×24 et la liste des monômes étiquettant ses colonnes sont sur la figure 2.
Les coefficients de la matrice sont de très simples fonctions des données d'entrée. Une fois que le vecteur de monômes $\mathbf{v}$ du noyau est trouvé par SVD de la matrice, la solution pour les profondeurs x peut être calculée comme suit. Nous pouvons simplement prendre quelques ratios fixé des éléments, comme

$$(x_1,x_2,x_3,x_4) := : \left(\sqrt{\frac{\mathbf{v}_1}{\mathbf{v}_{21}}}, \sqrt{\frac{\mathbf{v}_2}{\mathbf{v}_{22}}}, \sqrt{\frac{\mathbf{v}_3}{\mathbf{v}_{23}}}, \sqrt{\frac{\mathbf{v}_4}{\mathbf{v}_{24}}}\right).$$

Une méthode quelque peu plus appropriée est calculer le facteur d'échelle à l'aide des composantes les plus grandes du vecteur solution. En effet, en procédant ainsi, on utilise les composantes qui présentent le moins de risque d'être affectées par le bruit, ce qui donnera le meilleur facteur d'échelle possible. On procède ensuite de même pour le calcul des x_i à partir du facteur d'échelle.

Génériquement, si les coefficients $c_{i,j}$ et $d_{i,j}$ ne correspondent pas à une géométrie 3D cohérente, la matrice va être de rang maximal. Donc le déterminant de la matrice 24×24 est un multiple (non-trivial) du polynôme résultant du système.

5.1 Les méthodes linéaires 12×12 et 9×9

Une matrice résultante peut être réduite en une matrice plus petite par élimination Gaussienne symbolique partielle. Ici, le fait que la matrice d'entrée est très creuse et a un grand nombre de constantes d'entrée qui peuvent être utilisées comme pivots reconnus comme stables, permet l'élimination symbolique pour aller assez loin avant que les coefficients ne deviennent trop compliqués pour une implémentation convenable et/ou que le conditionnement numérique soit perdu. C'est un compromis entre la complexité des entrées et la taille de la matrice résultante. Mais l'élimination ne doit pas aller trop loin pour ne pas rendre trop difficile l'extraction de la solution à partir des monômes restants. De plus, la réduction change la taille de l'erreur effective, de sorte que cela peut affecter la précision des résultats.

Dans ce cas, nous expérimentons avec les réductions 12×12 et 9×9. A chaque étape de la réduction Gaussienne nous choisissons un pivot constant de sorte à ne pas avoir de division polynomiale, et permutons les lignes en conséquence. La version 12×12 élimine tous les termes avec des x_1, x_2 ou x_3 au carré ou au cube. En éliminant aussi les termes x_4^2 et x_4^3 cela donne une matrice 8×8 en:

$$(x_1\,x_2\,x_3, \ldots, x_2\,x_3\,x_4,\, x_1, \ldots, x_4)$$

. Mais nous jugeons les coefficients beaucoup trop complexes, et en éliminant le terme x_4^3 cela rend la récupération de l'échelle de la solution plus difficile, aussi nous préferrons laisser x_4^3 et avoir une matrice résultante 9×9. Les matrices 12×12 et 9×9 sont un peu trop grandes pour être présentées ici, mais peuvent être demandées aux auteurs.

6 Résultats expérimentaux

Cette section présente un comparatif entre des expériences testant le nouvel algorithme développé ici et, quelques al-

$$
\left(\begin{array}{cccc|cccc|cccc|cccc|cccc|cccc}
1 & 0 & 0 & 0 & c_{12} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{12}^2 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & c_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{12}^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{12} & 0 & 0 & 0 & 0 & 0 & d_{12}^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{12} & 0 & 0 & 0 & 0 & 0 & d_{12}^2 \\
\hline
1 & 0 & 0 & 0 & 0 & c_{13} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{13}^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & c_{13} & 0 & 0 & 0 & 0 & d_{13}^2 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & c_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{13}^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & c_{13} & 0 & 0 & 0 & 0 & d_{13}^2 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & c_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & d_{14}^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & c_{14} & 0 & 0 & 0 & d_{14}^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & c_{14} & 0 & 0 & 0 & d_{14}^2 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{14}^2 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & c_{23} & 0 & 0 & 0 & d_{23}^2 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & c_{23} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{23}^2 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & c_{23} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{23}^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & c_{23} & 0 & 0 & 0 & d_{23}^2 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & c_{24} & 0 & 0 & 0 & d_{24}^2 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & c_{24} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{24}^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & c_{24} & 0 & 0 & 0 & d_{24}^2 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & c_{24} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{24}^2 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & c_{34} & 0 & 0 & 0 & d_{34}^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{34} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{34}^2 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & c_{34} & 0 & 0 & 0 & d_{34}^2 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & c_{34} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{34}^2
\end{array}\right)
\left(\begin{array}{c}
x_1^3 \\ x_2^3 \\ x_3^3 \\ x_4^3 \\ \hline
x_1^2 x_2 \\ x_1^2 x_3 \\ x_1^2 x_4 \\ x_2^2 x_1 \\ x_2^2 x_3 \\ x_2^2 x_4 \\ x_3^2 x_2 \\ x_3^2 x_1 \\ x_3^2 x_4 \\ x_4^2 x_2 \\ x_4^2 x_3 \\ x_4^2 x_1 \\ \hline
x_1 x_2 x_3 \\ x_1 x_2 x_4 \\ x_1 x_3 x_4 \\ x_2 x_3 x_4 \\ \hline
-x_1 \\ -x_2 \\ -x_3 \\ -x_4
\end{array}\right) = 0
$$

FIG. 2 – *Le problème linéarisé*

gorithmes plus anciens. Les méthodes portent les noms abrégés comme suit :

4pt 3×3	L'algorithme des 4 points présenté dans [16].
3pt	L'élimination classique basée sur la méthode des 3 points [16].
3pt 5×5	La méthode 5×5 des 3 points du système propre.
4pt 9×9	La méthode 9×9 pour 4 points.
4pt 12×12	La méthode 12×12 pour 4 points.
4pt 24×24	La méthode 24×24 pour 4 points.

Nous avons donc testé une variante de chaque algorithme qui réordonne heuristiquement les points d'entrée de sorte que les points de référence qui sont utilisés par l'algorithme soient aussi dispersés que possible dans l'image.

Dans toutes les expériences, les points sont projetés avec une distance focale de 1024 pixels dans une image 512×512. Un bruit gaussien est additionné, avec un sigma par défaut de 1 pixel.

Chaque point des graphs représente 200 tirages de points 3D généré au hasard dans un nuage gaussien de déviation standard 1 unité, vu par environ 5 unités plus loin. Cependant, dans l'expérience sur les singularités, il y a eu 500 essais par point et seul le bruit a varié entre les essais, pas les points 3D. Dans l'expérience sur la coplanarité, le nuage est d'abord projeté sur un plan. Les résultats représentent les erreurs médianes, relatives à la taille totale de la translation et à la rotation de 1 radian. Le taux d'échec a été mesuré plutôt arbitrairement, comme le pourcentage d'essais total où, soit l'erreur de rotation, soit l'erreur de translation était supérieur à 0.5 unités.

La figure 10 montre le comportement des différentes méthodes avec 4 points d'entrée, quand le niveau de bruit augmente. La figure 3 montre le comportement quand des points supplémentaires sont additionnés à l'ensemble de données, pour les méthodes qui peuvent gérer des points supplémentaires.

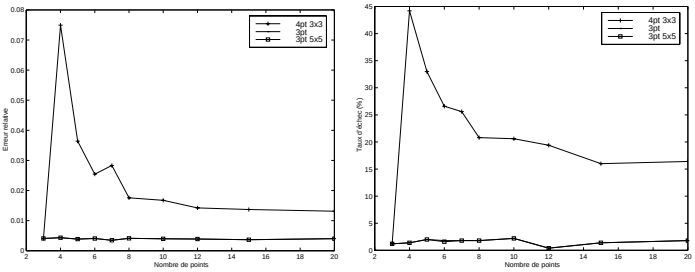

FIG. 3 – *Erreur de translation relative et taux d'échec versus le nombre de points d'entrée.*

Comme mentionné dans l'introduction, le problème de pose possède quelques cas singuliers qui cause souvent des problèmes dans la pratique. La figure 4 montre les résultats des erreurs et taux d'échec pour une telle configuration. Les données sont 4 points coplanaires dans un carré $[-1,1] \times [-1,1]$ et la caméra commence à la 'position=0', en un point singulier (pour la méthode de 4 points) directement au-dessus de leur centre. La caméra bouge ensuite parallèlement au plan des quatres points selon une droite, en se dirigeant vers un des angles du carré. A la 'position= $\sqrt{2}$' unités elle croise le coté du cylindre circulaire vertical qui correspond aux 4 points de donnée, où une autre singularité se produit.

La principale conclusion de ces expériences est que — en se basant sur le fait qu'elles retournent des solutions possibles multiples qui peuvent ne pas convenir — l'algorithme des 3-points surpasse significativement tous les algorithmes de 4 points linéaires. Même si les méthodes linéaires utilisent plus de données et intègrent des redondances, leurs erreurs relatives et leurs taux d'échec sont souvent 2 à 10 fois supérieurs à ceux des méthodes 3 points. La normalisation des données conventionnelles ne semble pas aider ici, mais des méthodes plus sophistiquées sont possibles.

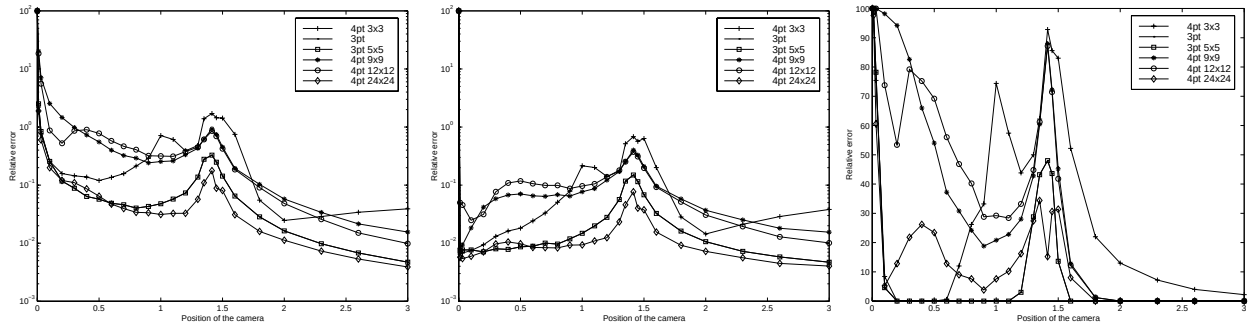

FIG. 4 – *Translation relative et erreur de rotation et taux d'échec quand la caméra bouge d'une configuration critique à une autre pour le problème de pose, à la position parametre 0 et* $\sqrt{2}$.

Les points coplanaires ne sont pas un cas singulier pour aucune des méthodes testées ici, et d'une manière générale leur performances sont similaires au cas non-coplanaires, excepté que l'avantage de performance des méthodes 3 points est décroissant. En augmentant le nombre de points pour les algorithmes linéaires cela améliore légèrement les résultats, mais pas suffisament pour égaler les méthodes des trois points. Pour les méthodes qui traitent les points asymétriquement, en choisissant des points image éparpillés comme points de base, cela semble améliorer les résultat en moyenne. Il existe sûrement de meilleures heuristiques que le choix de points éparpillés pour trouver des configurations de base stable. En particulier, avec de nombreux points dans un nuage gaussien, il y a une légère tendance, en choisissant des points éparpillés, de choisir des configurations proche de la dégénéréscence 'camera au dessus du cercle de points', qui réduit la stabilité moyenne.

L'élimination traditionelle et la nouvelle méthode, basé sur un calcul de vecteur propre, pour le calcul de pose à partir de 3 points ont des performances très proches dans tous les cas testés, la méthode du vecteur propre ayant peut être une toute petite avance.

La performance des méthodes linéaires 24×24, 12×12 et 9×9 est ainsi similaire, avec 24×24 ayant un léger avantage dans la précision globale, mais étant significativement plus lent que la méthode 9×9. Ce ne sont que des tendances statistiques — dans chaque problème il est très difficile de prévoir laquelle va le mieux fonctionner.

Les erreurs, et spécialement les taux d'échec, de toutes les méthodes sont significativement plus élevées que ce que l'on voudrait, spécialement pour les méthodes de 4 points. Ceci est en partie du à la génération alléatoire des données qui est souvent proche d'une configuration particulière.

Nous avons aussi effectué des tests avec des méthodes robustes sur l'algorithme algébrique, et l'algorithme linéaire 24×24. Nous avons comparé ces résultats avec une "méthode combinée" qui calcule la solution avec l'un et l'autre des algorithmes puis choisit celui ayant le moins de valeurs abérrantes entre les deux méthodes sur 300 échantillons aléatoires sur 30 points aléatoires. L'algorithme RANSAC pour la méthode combinée a moitié moins d'itérations que les autres méthodes pour obtenir la même efficacité en temps. Le nombre de valeurs abérrantes est estimé à partir d'un seuil fixé sur les erreurs de reprojection. Nous utilisons 1, 3 et 5 pixels comme seuils.

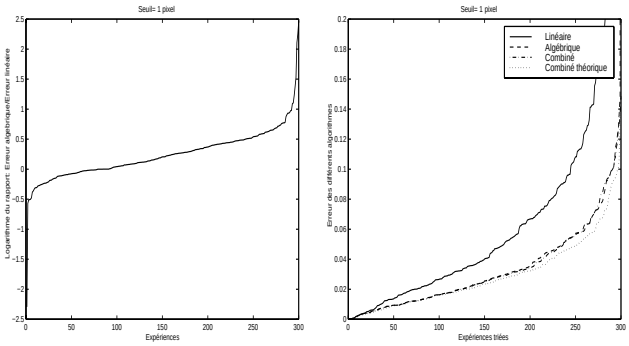

FIG. 5 – *Comparaison des 3 méthodes avec 1 pixel comme seuil de reprojection.*

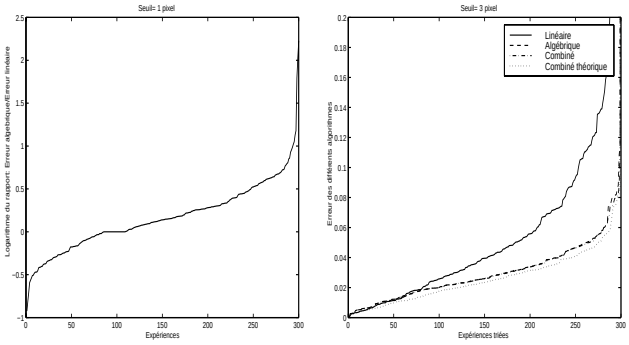

FIG. 6 – *Comparaison des 3 méthodes avec 3 pixels comme seuil de reprojection.*

Nous avons remarqué que l'algorithme combiné est un tout petit peu plus robuste (voir Figure 6). Nous avons effectué 7500 tests pour chacune des trois méthodes, ensuite nous avons trié les erreurs résiduelles et finallement zoomé sur les 100 pires tests de chaque méthode. L'algorithme combiné réduit la zone d'instabilité de l'estimation sur les

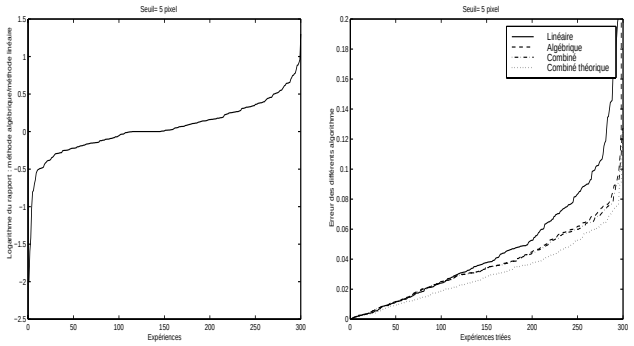

FIG. 7 – *Comparaison des 3 méthodes avec 5 pixels comme seuil de reprojection.*

méthodes individuelles ayant différentes configurations instables.

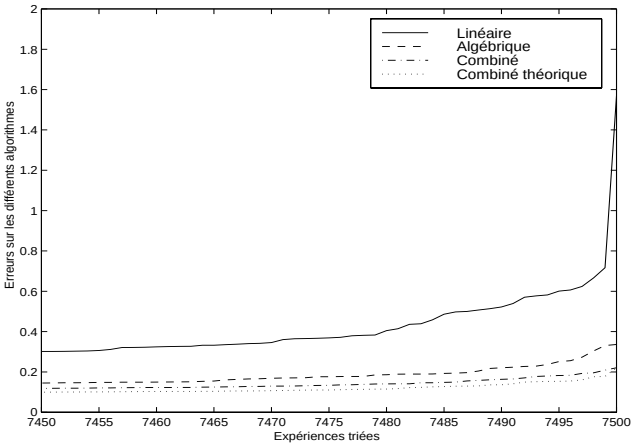

FIG. 8 – *La robustesse de la méthode combiné.*

Les résultats avec les expériences sur des images réelles avec des données de calibration connues sont présentés à la Figure 9.

7 Conclusions

Nous avons présenté une nouvelle méthode pour l'estimation de pose à partir de 3 points, basé sur un calcul de vecteur propre, et trois nouveaux algorithmes linéaires pour l'estimation de pose à partir de 4 points. Le principal avantage des algorithmes linéaires est qu'ils génèrent une solution unique. Aucune des méthodes ne dégénère pour des points (génériques) coplanaires. Toutes les méthodes développées ici sont faciles à implémenter dans le sens où leur matrices sont des fonctions relativement simples des coordonnées d'entrée. La méthode du vecteur propre surpasse légèrement l'élimination traditionnelle et est recommandée pour être utilisée en applications. Les méthodes linéaires à 4 points demandent des méthodes de normalisation des données plus sophistiquées pour être plus pratique.

FIG. 9 – *Les points réprojetés sur l'image originale par (a) la méthode algébrique (b) la méthode linéaire* 24×24 *(c) la méthode linéaire non linéairement optimisée* 24×24.

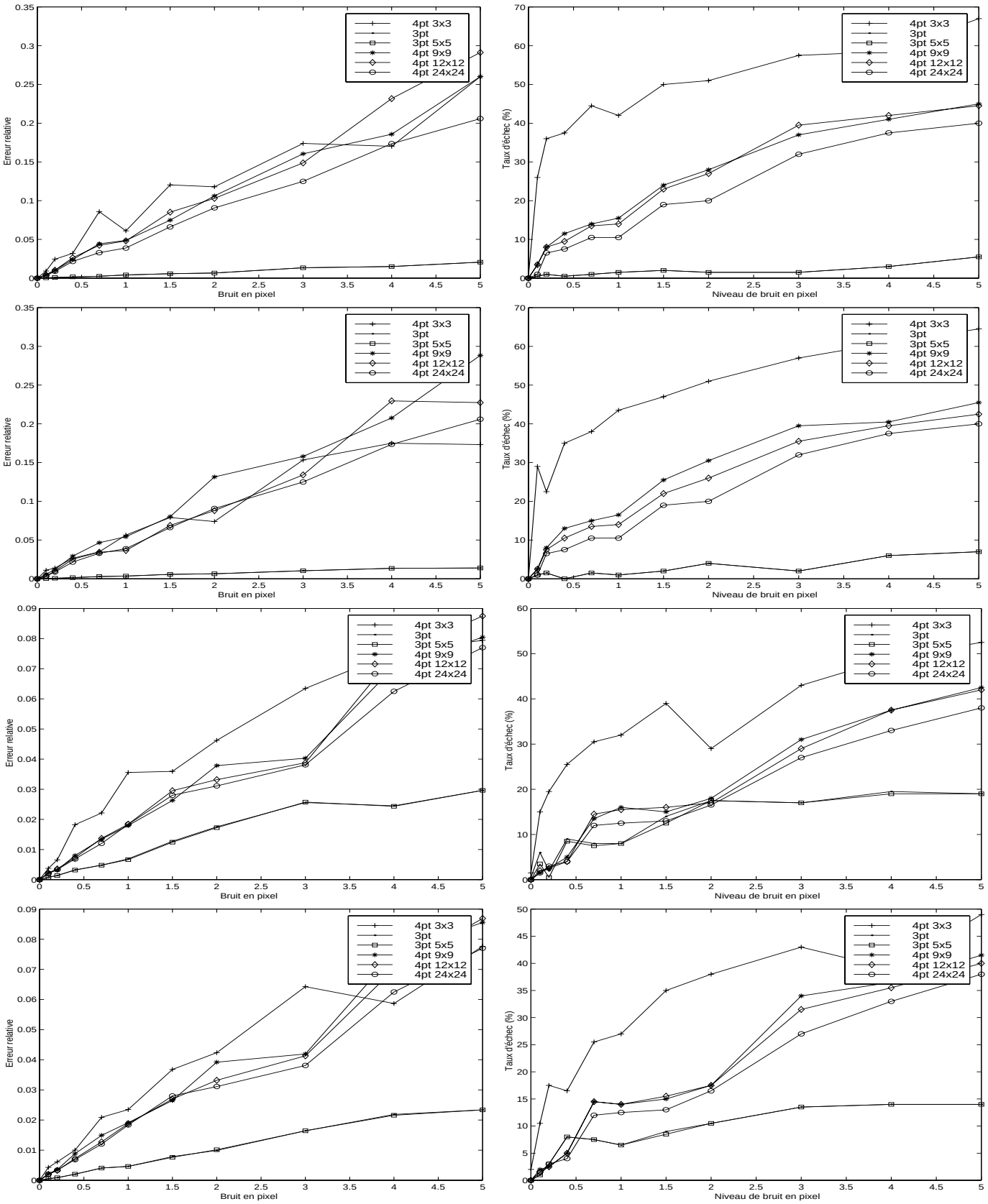

FIG. 10 – *L'erreur de translation relative et le taux d'échec versus le niveau de bruit en pixels, pour 4 points. Les deux premières lignes sont pour les données non coplanaires, les deux suivantes pour les données coplanaires. Les première et troisième lignes utilisent l'ordre des points d'entrée, la seconde et la quatrième réordonnent heuristiquement les points pour une meilleure stabilité avant de lancer les méthodes de pose. La tendance de l'erreur de rotation est très proche de celle de la translation, et n'est pas montrée par manque de place.*

Références

[1] H.H. Chen. Pose determination from line-to-plane correspondence: Existence condition and closed-form solutions. In *Proceedings of the 3rd International Conference on Computer Vision, Osaka, Japan*, pages 374–378, 1990.

[2] D.Cox, J.Little, and D.O'Shea. *Using Algebraic Geometry*. Springer, 1998.

[3] D. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1/2):123–141, 1995.

[4] M. Dhome, M. Richetin, J.T. Lapresté, and G. Rives. Determination of the attitude of 3D objects from a single perspective view. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, December 1989.

[5] I.Z. Emeris. A general solver based on sparse resultants: Numerical issues and kinematic applications. Technical Report RR-3110, INRIA, 1997.

[6] O. Faugeras and M. Hebert. The representation, recognition, and locating of 3D objects. *The International Journal of Robotics Research*, 5:27–52, 1986.

[7] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6):381 – 395, June 1981.

[8] W. Förstner. Reliability analysis of parameter estimation in linear models with applications to mensuration problems in computer vision. *Computer Vision, Graphics and Image Processing*, 40:273–310, 1987.

[9] R.M. Haralick, C. Lee, K. Ottenberg, and M. Nölle. Analysis and solutions of the three point perspective pose estimation problem. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, USA*, pages 592–598, 1991.

[10] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics and Image Processing*, 47:33–44, 1989.

[11] B.K.P. Horn. Closed form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[12] Y. Liu, T.S. Huang, and O.D. Faugeras. Determination of camera location from 2D to 3D line and point. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 12(1):28–37, January 1990.

[13] D. Lowe. Fitting parameterized three-dimensional models to images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[14] D.G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, Norwell, Massachusets, 1985.

[15] B. Mourrain and V.Y. Pan. Multivariate polynomials, duality and structured matrices. Technical report, INRIA Sophia-Antipolis, October 1998.

[16] L. Quan and Z.D. Lan. Linear n-point camera pose determination. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, August 1999.

[17] C.C. Slama, editor. *Manual of Photogrammetry, Fourth Edition*. American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, USA, 1980.

[18] E.H. Thompson. Space resection: Failure cases. *Photogrammetric Record*, X(27):201–204, 1966.

[19] B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 278–284, Kerkyra, Greece, September 1999.

[20] B.P. Wrobel. Minimum solutions for orientation. In *Proc. of the Workshop on Calibration and Orientation of Cameras in Computer Vision, Washington D.C., USA*. Springer-Verlag, August 1992.

[21] J.S.C. Yuan. A general phogrammetric solution for the determining object position and orientation. IEEE *Transactions on Robotics and Automation*, 5(2):129–142, April 1989.