

Calcul de pose d'un réseau de caméras hybrides à partir de droites

Sang Ly, Cédric Demonceaux, Pascal Vasseur, Claude Pegard

▶ **To cite this version:**

Sang Ly, Cédric Demonceaux, Pascal Vasseur, Claude Pegard. Calcul de pose d'un réseau de caméras hybrides à partir de droites. ORASIS - Congrès des jeunes chercheurs en vision par ordinateur, INRIA Grenoble Rhône-Alpes, Jun 2011, Praz-sur-Arly, France. inria-00595485

HAL Id: inria-00595485
https://inria.hal.science/inria-00595485

Submitted on 24 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Calcul de pose d'un réseau de caméras hybrides à partir de droites

Sang Ly [1] Cédric Demonceaux [2] Pascal Vasseur [3] Claude Pégard [1]

[1] MIS, Université de Picardie Jules Verne, Amiens
[2] Le2i - UMR CNRS 5158, Université de Bourgogne, Le Creusot
[3] LITIS, Université de Rouen, Rouen

Résumé

Nous présentons une méthode de calcul de pose pour un réseau de caméras hybrides à partir de droites. Ce réseau est composé d'au moins trois caméras centrales (perspectives, catadioptriques ou fish-eyes) qui observent un ensemble de droites 3D. Nous décomposons le calcul de la pose des caméras en trois étapes. Dans un premier temps, nous estimons les rotations à partir de points de fuite de droites parallèles dans la scène, puis nous calculons les translations à partir de ces mêmes droites. Une fois la pose obtenue, nous pouvons reconstruire les droites 3D par triangulation des plans 3D. Enfin, nous proposons un ajustement de faisceaux afin d'affiner la pose et la reconstruction 3D. Cette technique peut être utilisée pour déterminer la topologie d'un réseau de caméras hybrides et aussi être étendue à la localisation de robots. Les principales contributions de ce travail concernent tout d'abord l'utilisation unique des droites de l'environnement pour le calcul de pose ainsi que l'application possible de la méthode à tout réseau composé de caméras à point de vue unique. De plus, nous présentons un ajustement de faisceaux sur la position des caméras et la reconstruction des droites 3D en minimisant une mesure d'erreur basée sur les normales aux droites sur les images sphériques. Nous avons validé cette méthode en simulation ainsi que sur les images réelles.

Mots Clef

Calcul de pose, droite, caméras hybrides.

Abstract

We present a pose estimation approach for a setup of multiple hybrid cameras using line features. From line images across at least three single viewpoint cameras such as perspective, central catadioptric or fish-eye ones, the camera poses are recovered by a decoupling of rotations and translations : rotations are estimated from vanishing points of parallel lines and translations are calculated from known rotations and line images. The 3D lines are reconstructed by the triangulation of projective planes passing through line correspondences. The camera poses and 3D scene are finally refined by bundle adjustment. This technique can be used to recover the topology of a multiple hybrid camera system or extended to vision-based robot localization. The main contributions of this approach are the sole use of lines for the estimation, rather than point features and that it can handle different camera types in a unified manner. Moreover, we present an efficient bundle adjustment on camera and 3D line parameters by minimising the error of the line normals on spherical images. The proposed algorithm has been validated on simulated data and real images.

Keywords

Pose estimation, line, hybrid cameras.

1 Introduction

Estimer la pose consiste à déterminer l'orientation et la position relative des caméras et des objets 3D de la scène. Résoudre ce problème est très utile dans de nombreuses applications en vision par ordinateur : surveillance de zone, reconstruction d'une scène 3D, calcul du déplacement d'un robot mobile, estimation de la position d'un visage relativement à une caméra, etc. Pour chacun de ces problèmes, il peut être judicieux d'utiliser différents types de caméras. Par exemple, pour surveiller une zone et reconstruire sans grande précision les objets en déplacements, une caméra catadioptrique peut suffire. A l'inverse, pour une reconstruction 3D plus fine, il est préférable d'utiliser une ou des caméras de grande résolution. Notons que parfois la combinaison de différents types de caméras (perspective, fish-eye, catadioptrique) peut s'avérer intéressante pour ces applications. En effet, les caméras à large champ de vue permettent d'obtenir une vue importante de la scène mais avec une faible résolution, par contre les caméras perspectives ont une résolution importante mais un champ de vue limité. C'est pourquoi, nous nous intéressons ici au calcul de pose d'un ensemble de caméras hétérogènes.

Dans ce travail, seules les droites sont utilisées comme primitives. Ces primitives sont plus stables que les points et moins susceptibles d'être produites par hasard (bruit par exemple), surtout dans des environnements urbains. Comparées aux points, les droites sont moins nombreuses mais portent plus d'informations. De plus, les droites sont moins affectées par occultations car chaque droite peut être recontruite à partir de ses segments dans plusieurs images. L'inconvénient majeur des droites pour le calcul de pose concerne la nécessité d'utiliser au moins trois images là où ldeux images suffisent pour les points.

Il existe de nombreux travaux de calcul de pose utilisant

les droites [11, 17, 21, 18, 7, 15, 2, 4]. Les méthodes de calcul de pose peuvent être classées en 2 approches : les approches factorisation et les approches séquentielles. Les premières, sans aucune hypothèse sur le calibrage de caméra et les droites 3D, la pose de caméras et les droites sont estimées dans un espace projectif [19, 14] ou affine [16]. Néanmoins, cette technique requiert que les droites soient visibles dans toutes les vues. Il existe aussi des *méthodes séquentielles* qui décomposent le problème en trois étapes : (i) estimation du déplacement de la caméra, (ii) triangulation pour obtenir la structure 3D (iii) optimisation par ajustement de faisceaux [8, §18], [13, §11.3]. La première étape peut être résolue à l'aide d'un tenseur trifocal reliant les informations géométriques des droites dans trois vues [20, 7, 5]. Concernant les deuxième et troisième étapes, les principales différences entre les méthodes proposées résident dans la modélisation paramétrique des droites 3D. En effet, différentes représentations des droites peuvent être utilisées, par exemple une projection des droites sur les plans $x = 0$ and $y = 0$ [17], une paramétrisation à partir de deux points 3D [7], un point et une direction [21, 18] ou encore les coordonnées de Plücker [2]. Notons que ces différentes représentations et leurs caractéristiques sont détaillées dans [2]. Pour la phase d'ajustement de faisceaux, la structure 3D des droites et la position des caméras sont simultanément affinées par une minimisation non linéaire d'une erreur de reprojection. Hartley [7] utilise une représentation à partir de 2 images de la droite et propose un ajustement de faisceaux basé sur l'algorithme de Levenberg-Marquardt. Bartoli et al [2] proposent une représentation orthonormale des droites 3D qui permet d'estimer seulement 4 paramètres à chaque fois.

La méthode de calcul de pose que nous proposons ici se décompose aussi en trois étapes. Nous supposons que les caméras sont calibrées et que les droites sont mises en correspondance dans au moins trois vues. Dans un premier temps, la position relative des caméras est déterminée en découplant l'estimation des rotations et des translations. La rotation est calculée à partir de points de fuite de droites parallèles de la scène, puis les translations sont estimées à partir des droites de la scène en connaissant les rotations. Dans un deuxième temps, les droites de l'environnement sont reconstruites par intersection des plans 3D. Enfin, nous proposons une phase optionnelle d'ajustement de faisceaux sur les droites permettant d'affiner les résultats. A travers ces trois étapes, nous proposons d'utiliser une modélisation sphérique de nos caméras. Ce modèle permet d'englober les caméras à point de vue unique (perspectives, catadioptriques) mais aussi fish-eyes [6, 1, 22]. Cette méthode peut être utilisée pour déterminer la topologie d'un réseau de caméras hybrides et aussi étendue à la navigation automobile de robots. De plus, l'utilisation unique des droites pour l'estimation de rotations et translations optimise le temps de calcul d'un technique de découplage. En effet Bazin et al. [3] utilisent des droites pour estimer la rotation et des points pour la translation. Enfin, nous proposons un nouvel ajustement de faisceaux sur la position des caméras et la reconstruction des droites 3D en minimisant une mesure d'erreur basée sur les normales aux droites sur les images sphériques.
Dans les sections suivantes, nous présentons notre méthode de calcul de pose, de reconstruction 3D et d'ajustement de faisceaux. Enfin, nous montrons des résultats expérimentaux sur des données de synthèse et réelles avant de conclure.

2 Calcul linéaire de la pose

Dans cette section, nous présentons une méthode linéaire pour calculer la pose des caméras en découplant les rotations et les translations. Dans un premier temps, les rotations sont obtenues à partir des points de fuite de droites parallèles dans la scène. Dans un second temps, nous estimons les translations à partir des rotations désormais connues et des droites.
Notations : Dans la suite, nous noterons les matrices par une police Sans Serif, les vecteurs en caractères gras et les scalaires en italiques.
Supposons que la scène est composée de m caméras centrales $\mathbf{C}_i$ $(i = 1...m)$ comme illustrées figure 1. Considérons le centre de la première caméra comme origine du repère monde. Notons $[\mathsf{R}_i|\mathbf{t}_i]$ la [Rotation|translation] entre $\mathbf{C}_i$ et l'origine, ainsi $[\mathsf{R}_1|\mathbf{t}_1] = [\mathrm{I}|\mathbf{0}]$. La projection sur les images sphériques d'une droite $\mathbf{L}$ de l'espace 3D est un grand cercle $\mathbf{l}_i$. Nous pouvons exprimer $\mathbf{L}$ vectoriellement par $\mathbf{L} = \mathbf{X}_0 + \mu\mathbf{d}$ où $\mathbf{L}, \mathbf{X}_0, \mathbf{d} \in \mathbb{R}^3$ et $\mu \in \mathbb{R}$. Enfin, nous notons $\mathbf{n}_i \in \mathbb{R}^3$ les normales aux plans passant la droite $\mathbf{L}$ et le centre $\mathbf{C}_i$.

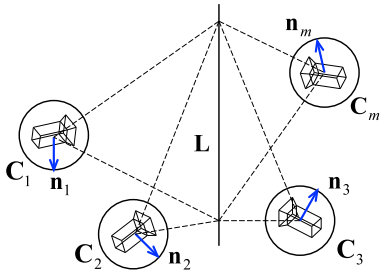


FIGURE 1 – Projection sphérique d'une droite 3D

Une approche de détection de droites dans les images catadioptriques est proposée dans [3]. Elle est composée d'une division et d'une fusion de segments de droites dans l'image sphérique. Nous avons étendu cette méthode à une technique de détection de droites applicable à tous types de caméras centrales.

2.1 Estimation de la rotation

La rotation entre deux caméras peut être estimée à partir des points de fuite dans les images [3]. Ainsi, après la détection des points de fuite $\mathbf{V}_i$ dans les images, chaque rotation R_i peut être obtenue linéairement par [9] :

$$\mathbf{V}_i = \mathsf{R}_i\mathbf{V}_1 \tag{1}$$

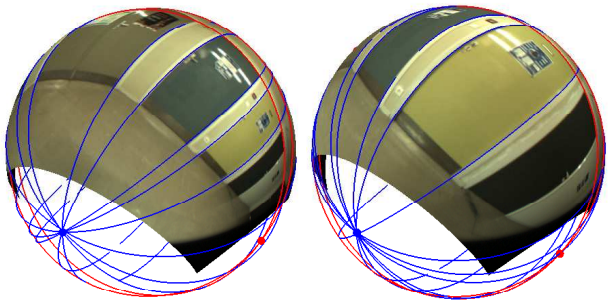


FIGURE 2 – Points de fuite des droites parallèles

2.2 Estimation de la translation

Pour une droite 3D de normale $\mathbf{n}_i$ dans trois images $i = \{1, a, b\}$ (où $2 \leq a, b \leq m$ et $a \neq b$), nous avons montré dans [12] la relation suivante à partir de ces vues :

$$[\mathbf{n}_1]_\times \mathsf{R}_a^T \mathbf{n}_a \mathbf{n}_b^T \mathbf{t}_b - [\mathbf{n}_1]_\times \mathsf{R}_b^T \mathbf{n}_b \mathbf{n}_a^T \mathbf{t}_a = 0 \qquad (2)$$

Ainsi, avec m caméras, nous obtenons $(m-1)(m-2)/2$ triplets de vues $\{1, a, b\}$ soit autant de relation trilinéaire du type (2). Ces relations trilinéaires définissent alors un système linéaire dont les inconnues sont les translations $\mathbf{t}_{2..m}$ en fonction des rotations $\mathsf{R}_{2..m}$ et des normales $\mathbf{n}_{1..m}$. Par exemple, avec un réseau de 4 caméras, nous obtenons le système suivant :

$$\mathsf{AX} = 0 \qquad (3)$$

où

$$\mathsf{A} = \begin{bmatrix} -[\mathbf{n}_1]_\times \mathsf{R}_3^T \mathbf{n}_3 \mathbf{n}_2^T & [\mathbf{n}_1]_\times \mathsf{R}_2^T \mathbf{n}_2 \mathbf{n}_3^T & 0 \\ -[\mathbf{n}_1]_\times \mathsf{R}_4^T \mathbf{n}_4 \mathbf{n}_2^T & 0 & [\mathbf{n}_1]_\times \mathsf{R}_2^T \mathbf{n}_2 \mathbf{n}_4^T \\ 0 & -[\mathbf{n}_1]_\times \mathsf{R}_4^T \mathbf{n}_4 \mathbf{n}_3^T & [\mathbf{n}_1]_\times \mathsf{R}_3^T \mathbf{n}_3 \mathbf{n}_4^T \end{bmatrix}$$

et

$$\mathbf{X} = (\mathbf{t}_2^T \mathbf{t}_3^T \mathbf{t}_4^T)^T$$

De façon générale, à partir d'une droite de normale $\mathbf{n}_{1..m}$ dans les m vues, A est une matrice de taille $3(m-1)(m-2)/2 \times 3(m-1)$ et $\mathbf{X} = (\mathbf{t}_2^T \mathbf{t}_3^T \dots \mathbf{t}_m^T)^T$. Chaque droite supplémentaire augmente la matrice A de $3(m-1)(m-2)/2$ droites. Ainsi, n droites sur m vues définissent une matrice A de dimension $3(m-1)(m-2)n/2 \times 3(m-1)$. Le nombre minimum de droites est donc $n > \frac{2}{m-2}$. Par exemple, il faut au moins 3 droites pour estimer la translation à partir de 3 vues.

3 Reconstruction 3D

Chaque droite de la scène est reconstruite à partir de l'intersection des plans 3D définis par les normales $\mathbf{n}_{1..m}$ des m vues. Ainsi, connaissant la pose des caméras et les normales, la droite 3D vérifie le système suivant :

$$\mathsf{B}\hat{\mathbf{L}} = 0 \qquad (4)$$

où

$$\mathsf{B} = \begin{bmatrix} \mathbf{n}_1^T & 0 \\ \mathbf{n}_2^T \mathsf{R}_2 & \mathbf{n}_2^T \mathbf{t}_2 \\ \vdots & \vdots \\ \mathbf{n}_m^T \mathsf{R}_m & \mathbf{n}_m^T \mathbf{t}_m \end{bmatrix} \quad \text{et} \quad \hat{\mathbf{L}} = (\mathbf{L}^T, 1)^T$$

A l'aide d'une décomposition en valeurs singulières de $\mathsf{B} = \mathsf{UDV}^T$, les deux colonnes de V correspondant aux deux plus grandes valeurs singulières peuvent être utilisées pour obtenir l'intersection de ces plans [8, §12.7].

4 Ajustement des faisceaux

Cette étape optionnelle permet d'affiner le calcul de pose des caméras et la structure 3D en minimisant l'erreur de reprojection des droites dans les images sphériques.
Chaque caméra i $(i = 1...m)$ est modélisée par un vecteur de taille 7 $\mathbf{c}_i$=$(r_0, r_1, r_2, r_3, t_x, t_y, t_z)_i$ où (r_0, r_1, r_2, r_3) est la représentation quaternaire de la rotation et (t_x, t_y, t_z) la translation. Chaque droite 3D j $(j = 1...n)$ est représentée par un vecteur de taille 6 $\mathbf{l}_j$=$(e_x^1, e_y^1, e_z^1, e_x^2, e_y^2, e_z^2)_j$ défini par deux points 3D e^1 et e^2 sur la droite.
Le vecteur de paramètres à optimiser est défini par tous les paramètres décrivant les m caméras et n droites 3D $\mathbf{Q} = (\mathbf{c}_1 \dots \mathbf{c}_m, \mathbf{l}_1 \dots \mathbf{l}_n)$.
L'ajustement de faisceaux consiste alors à minimiser l'erreur de reprojection suivante :

$$\min_{\mathbf{c}_i, \mathbf{l}_j} \sum_{i=1}^{m} \sum_{j=1}^{n} d(\hat{\mathbf{n}}_{ij}, \mathtt{P}(\mathbf{c}_i, \mathbf{l}_j)) \qquad (5)$$

avec

$$d = \hat{\mathbf{n}}_{ij} \times [(\mathsf{R}_i \mathbf{e}_j^1 + \mathbf{t}_i) \times (\mathsf{R}_i \mathbf{e}_j^2 + \mathbf{t}_i)]$$

où $\mathtt{P}$ représente la projection sphérique d'une droite $\mathbf{l}_j$ sur la caméra $\mathbf{c}_i$ et $\hat{\mathbf{n}}_{ij}$ la projection sphérique inverse de la droite j sur la caméra i.
Cette minimisation est réalisée à l'aide de l'algorithme d'optimisation non linéaire de Levenberg-Marquardt dans lequel nous utilisons les paramètres obtenus précédemment lors de la phase linéaire d'estimation et de reconstruction 3D. Chaque ligne de la matrice Jacobienne de dimension $mn \times (7m + 6n)$ est calculée comme suit :

$$\frac{\partial d}{\partial \mathbf{Q}} = [\frac{\partial d}{\partial \mathbf{c}_1} \dots \frac{\partial d}{\partial \mathbf{c}_m}, \frac{\partial d}{\partial \mathbf{l}_1} \dots \frac{\partial d}{\partial \mathbf{l}_n}] \qquad (6)$$

Remarquons que $\frac{\partial d_{ij}}{\partial \mathbf{c}_k} = 0 \; \forall k \neq i$ et que $\frac{\partial d_{ij}}{\partial \mathbf{l}_k} = 0 \; \forall k \neq j$. Ainsi en notant $\frac{\partial d_{ij}}{\partial \mathbf{c}_i}$ et $\frac{\partial d_{ij}}{\partial \mathbf{l}_j}$ par $\mathbf{A}_{ij}$ et $\mathbf{B}_{ij}$ respectivement, le Jacobien se présente alors par l'équation 7.
Cette technique d'optimisation est simple et rapide. Notons que ce schéma est similaire à un ajustement de faisceaux minimisant l'erreur de projection des points 3D sur l'image dans une approche basée sur les points. Néanmoins, comme les droites sont moins nombreuses mais apporte plus d'informations, l'ajustement de faisceaux sur les droites 3D est donc plus rapide et plus efficace que l'ajustement basé points.

$$\mathsf{J} = \left[\begin{array}{cccc|cccc} \mathbf{A}_{11} & 0 & \dots & 0 & \mathbf{B}_{11} & 0 & \dots & 0 \\ 0 & \mathbf{A}_{21} & \dots & 0 & \mathbf{B}_{21} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{A}_{m1} & \mathbf{B}_{m1} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}_{1n} & 0 & \dots & 0 & 0 & 0 & \dots & \mathbf{B}_{1n} \\ 0 & \mathbf{A}_{2n} & \dots & 0 & 0 & 0 & \dots & \mathbf{B}_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{A}_{mn} & 0 & 0 & \dots & \mathbf{B}_{mn} \end{array}\right] \quad (7)$$

5 Résultats expérimentaux

5.1 Simulation

Une approche similaire à notre méthode peut être trouvée dans [10]. Dans cet article, la rotation est supposée connue et la translation est calculée sur les images sphériques à partir d'une correspondance de points. C'est pourquoi, nous proposons d'évaluer et de comparer notre méthode d'estimation basée droites avec cette méthode basée sur les points. Pour ce faire, nous avons créé une scène virtuelle composée de 150 points et 150 droites observés par 4 caméras. Les points, les droites et la position de caméras sont générés aléatoirement. La baseline moyenne entre les caméras est de 0.5 mètre et la distance moyenne entre les droites et les caméras est de 10 mètres. Les translations $\mathbf{t}_{2..4}$ des 4 caméras sont estimées à partir de l'approche basée points de [10] et notre approche basée droites.

Comme les points de la méthode [10] et nos normales sont définis sur une sphère unitaire, nous pouvons les paramétrer par les angle d'azimut et d'élévation. Pour vérifier la sensibilité des méthodes aux bruits de mesure, nous avons donc ajouté un bruit blanc Gaussien à chaque angle. De plus, pour simuler l'imprécision dans l'estimation de la rotation, nous avons perturbé chaque angle de rotation par un bruit blanc Gaussien. Les mêmes rotations (avec ou sans bruit) sont utilisées pour estimer la translation à partir des points et des droites. La figure 3 montre l'erreur angulaire moyenne des trois translations après 1000 essais. Nous pouvons remarquer que la méthode basée sur les droites est plus robuste en présence de bruit dans l'estimation des rotations. Ceci montre un apport important de notre méthode par rapport aux méthodes basées sur les points car expérimentalement, les rotations ne sont jamais connues précisément.

5.2 Séquence Hall

Un hall est filmé à l'aide de 3 caméras perspectives et de 3 caméras fish-eyes (figure 4). Ces caméras ont été préalablement calibrées par la toolbox *http ://www.robots.ox.ac.uk/∼cmei/Toolbox.html*. La pose des caméras est calculée à partir de 13 droites en correspondance à travers les 6 vues, l'ajustement de faisceaux

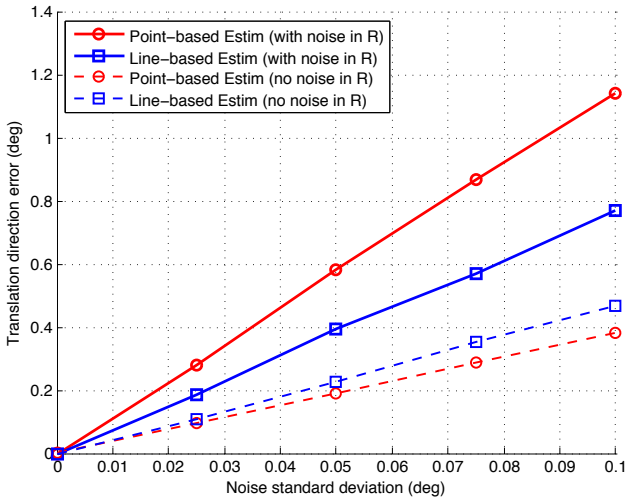


FIGURE 3 – Erreur d'estimation sur la translation de la méthode basée points [10] et notre approche basée droites

converge au bout de 3 itérations. La figure 5 montre la reconstruction 3D réalisée à partir de notre méthode.

Les 4 portes dont les droites sont extraites dans la figure 4 sont reconstruites et nommées D1 à D4 de la droite vers la gauche. A partir du rapport hauteur/largeur de chaque porte obtenu par reconstruction, nous déduisons la hauteur de chaque porte à partir de la largeur réelle (tableau 1). Les résultats avant et après ajustement de faisceaux sont respectivement notés Hauteur LIN et Hauteur LIN + BA. L'erreur finale est inférieur à 0.5% pour les portes D2-D3-D4. Cependant, l'erreur pour la porte D1 est plus importante à cause des distorsions importantes dans les images fish-eyes, puisque cette porte se trouve à la périphérie de ces images. La reprojection des droites 3D sur l'une des images fish-eyes est illustrée dans la figure 6. Comme nous pouvons le remarquer, l'erreur de reprojection est plus faible après la phase d'ajustement de faisceaux.

5.3 La séquence échiquier

Une séquence de 20 images d'un échiquier prise avec un fish-eye est proposée pour valider notre méthode. La figure 7 montre 6 images de cette séquence, nous déterminons la position des caméras et la reconstruction 3D de l'échiquier. Nos résultats sont comparés aux paramètres extrinsèques du calibrage par la toolbox *http ://www.robots.ox.ac.uk/∼cmei/Toolbox.html*. La différence entre nos résultats et le calibrage est présentée dans le tableau 2. La reconstruction est illustrée figure 8.

6 Conclusions

Nous avons présenté dans ce papier une méthode de calcul de pose à partir des droites de la scène pour des caméras hybrides (perspectives, catadioptriques, fish-eyes). L'approche proposée consiste dans un premier temps à estimer les rotations entre les caméras puis connaissant ces



FIGURE 4 – Images fish-eyes et perspectives avec détection des droites

	D1	D2	D3	D4
Hauteur LIN (cm)	231	207	205	208
Hauteur LIN + BA (cm)	228	204	204	203
Hauteur réel (cm)	203	203	203	203
Largeur réelle (cm)	80	144	90	80

TABLE 1 – Résultat de reconstruction de la séquence Hall

rotations, nous calculons les translations. Ces deux étapes se font par des méthodes linéaires basées exclusivement sur les droites de la scène. De plus, un ajustement de faisceaux est présenté pour affiner les résultats. Notre méthode a été validée en simulation ainsi que sur des séquences réelles prises avec des caméras perspectives et fish-eyes.

Références

[1] J.P. Barreto and H. Araujo. Issues on the geometry of central catadioptric image formation. In *CVPR*, pages II :422–427, 2001.

[2] A.E. Bartoli and P.F. Sturm. Structure-from-motion using lines : Representation, triangulation, and bundle adjustment. *CVIU*, 100(3) :416–441, December 2005.

[3] J.C. Bazin, C. Demonceaux, P. Vasseur, and I. Kweon. Motion estimation by decoupling rotation and translation in catadioptric vision. *CVIU*, 114(2) :254–273, February 2010.

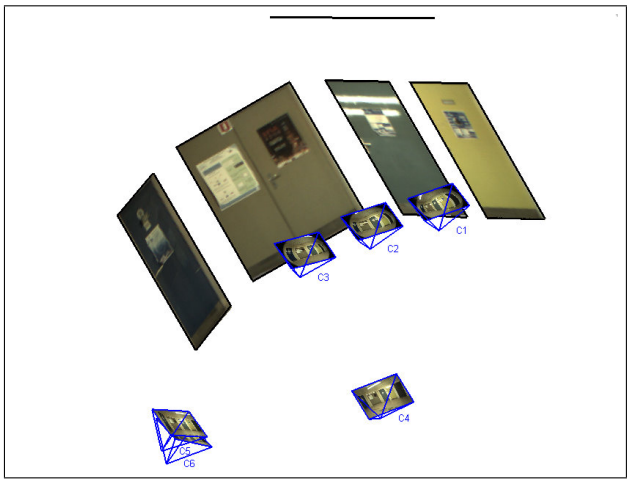


FIGURE 5 – Reconstruction 3D



FIGURE 6 – Reprojection des droites 3D après reconstruction (en rouge) et ajustement de faisceaux (en bleu)

[4] M. Chandraker, J.W. Lim, and D. Kriegman. Moving in stereo : Efficient structure and motion using lines. In *ICCV*, pages 1741–1748, 2009.

[5] A.W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *ECCV*, page I : 311, 1998.

[6] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical implications. In *ECCV*, pages II : 445–461, 2000.

[7] R.I. Hartley. Lines and points in three views and the trifocal tensor. *IJCV*, 22(2) :125–140, March 1997.

[8] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, June 2004.

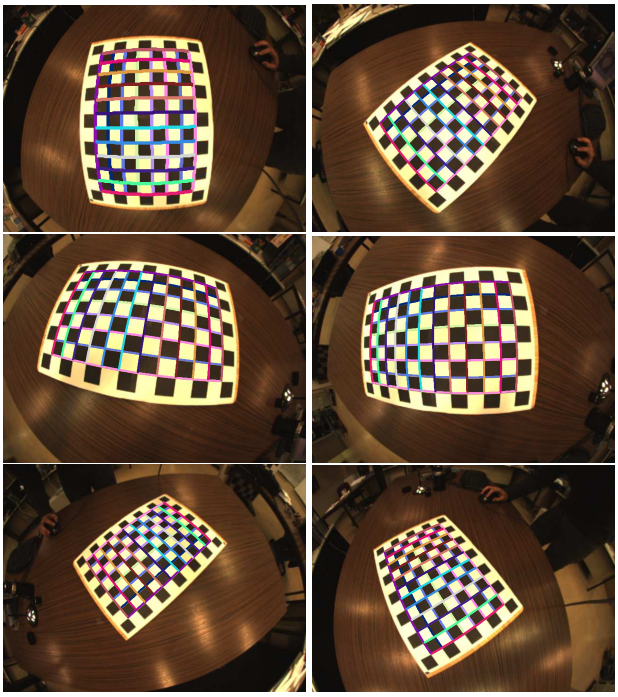


FIGURE 7 – Images de l'échiquier par une caméra fish-eye

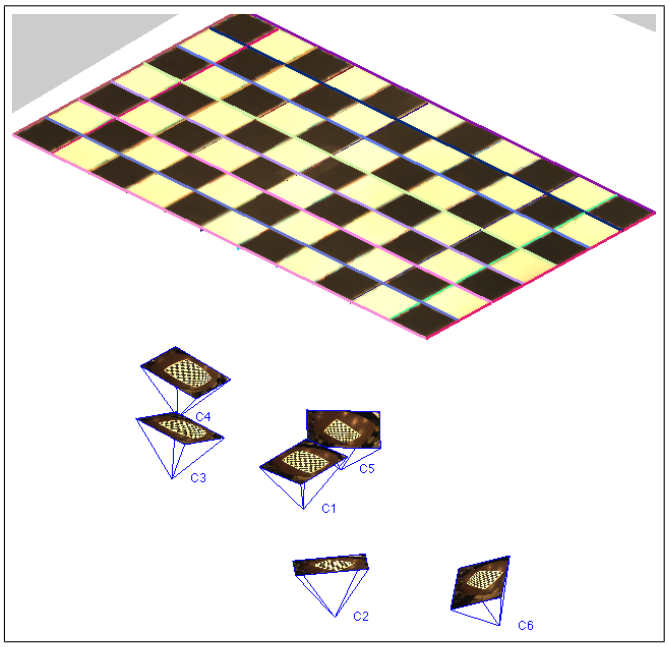


FIGURE 8 – Reconstruction de l'échiquier

R_i	R axis (deg)	R angle (deg)	$\mathbf{t}_i$	**t** direction (deg)
R_2	0.62	0.59	$\mathbf{t}_2$	0.82
R_3	0.82	0.17	$\mathbf{t}_3$	0.94
R_4	0.97	0.54	$\mathbf{t}_4$	1.53
R_5	0.07	0.21	$\mathbf{t}_5$	0.19
R_6	0.19	0.30	$\mathbf{t}_6$	1.02

TABLE 2 – Pose de caméras pendant la séquence échiquier : différence entre notre estimation et un calibrage extrinsèque à partir de l'échiquier

[9] Berthold K. P. Horn, H.M. Hilden, and Shariar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOURNAL OF THE OPTICAL SOCIETY AMERICA*, 5(7) :1127–1135, 1988.

[10] J.H. Kim and R.I. Hartley. Translation estimation from omnidirectional images. In *DICTA*, page 22, Washington, DC, USA, 2005. IEEE Computer Society.

[11] Y.C. Liu and T.S. Huang. A linear algorithm for motion estimation using straight line correspondences. *CVGIP*, 44(1) :35–57, October 1988.

[12] S. Ly, C. Demonceaux, and P. Vasseur. Translation estimation for single viewpoint cameras using lines. In *ICRA*, pages 1928–1933, 2010.

[13] Y. Ma, S. Soatto, J. Kosecka, and S.S. Sastry. *An Invitation to 3-D Vision : From Images to Geometric Models*. Springer-Verlag, 2005.

[14] D. Martinec and T. Pajdla. Line reconstruction from many perspective images by factorization. In *CVPR*, pages I : 497–502, 2003.

[15] N. Navab, Y. Genc, and M. Appel. Lines in one orthographic and two perspective views. *PAMI*, 25(7) :912–917, July 2003.

[16] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *PAMI*, 19(8) :834–845, August 1997.

[17] M.E. Spetsakis and Y. Aloimonos. Structure from motion using line correspondences. *IJCV*, 4(3) :171–183, 1990.

[18] C.J. Taylor and D.J. Kriegman. Structure and motion from line segments in multiple images. *PAMI*, 17(11) :1021–1032, November 1995.

[19] B. Triggs. Factorization methods for projective structure and motion. In *CVPR*, pages 845–851, 1996.

[20] B. Triggs. Linear projective reconstruction from matching tensors. *IVC*, 15(8) :617–625, August 1997.

[21] J.Y. Weng, T.S. Huang, and N. Ahuja. Motion and structure from line correspondences : Closed-form solution, uniqueness, and optimization. *PAMI*, 14(3) :318–336, March 1992.

[22] X.H. Ying and Z.Y. Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In *ECCV*, pages Vol I : 442–455, 2004.