



HAL
open science

Calcul du mouvement et de la structure a partir de points et de droites

Olivier Faugeras, F. Lustman, G. Toscani

► **To cite this version:**

Olivier Faugeras, F. Lustman, G. Toscani. Calcul du mouvement et de la structure a partir de points et de droites. RR-0661, INRIA. 1987. inria-00075892

HAL Id: inria-00075892

<https://inria.hal.science/inria-00075892>

Submitted on 24 May 2006

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.

IRIA

UNITÉ DE RECHERCHE
IRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105

78153 Le Chesnay Cedex
France

Tél. (1) 39 63 55 11

Rapports de Recherche

N° 661

CALCUL DU MOUVEMENT ET DE LA STRUCTURE À PARTIR DE POINTS ET DE DROITES

Olivier FAUGERAS
Francis LUSTMAN
Giorgio TOSCANI

Mai 1987

**Motion and Structure from Motion
from
Point and Line Matches¹**

Olivier D. Faugeras, Francis Lustman, Giorgio Toscani

INRIA
Domaine de Voluceau-Rocquencourt
BP 105
78153 Le Chesnay Cedex
FRANCE

ABSTRACT

We discuss the general problem of estimating camera displacements from point and line matches between frames when the environment is static. In the case of points, we develop a closed form analytical solution for the displacement which is a generalization of the ideas of Longuet-Higgins [Longuet-Higgins 1981]. The technique appears to be quite robust to noise. We also present an analysis of the degenerate case where the points are coplanar and establish the theoretical link between the technique of Longuet-Higgins and that of Tsai and Huang [Tsai and Huang 1982]. In the case of lines, we prove that at least three views are necessary and derive the equations satisfied by the displacements using the idea of a pencil of planes. We show that six lines are sufficient to estimate the displacement parameters and show that they can be recovered robustly even in the presence of noise using Extended Kalman Filtering. We also describe a technique for reconstructing 3D lines from three projections after the camera motion has been estimated which is also based on the idea of a pencil of planes. Experimental results are presented on both synthetic and real data.

Keywords: Motion, Structure from Motion, Eight-Point Algorithm, 3D Lines, Pencils of Planes, Extended Kalman Filtering

¹ This work was developed in part under ESPRIT contract P940



CALCUL DU MOUVEMENT ET DE LA STRUCTURE A PARTIR DE POINTS ET DE DROITES

Olivier Faugeras, Francis Lustman, et Giorgio Toscani
INRIA
Domaine de Voluceau-Rocquencourt
BP 105
78153 Le Chesnay Cedex

Résumé

Nous traitons le problème général de l'estimation des déplacements d'une caméra à partir de points et de droites appariés entre différentes vues, dans un environnement statique. Dans le cas de points, nous trouvons une formule explicite pour le déplacement, qui est une généralisation des idées de Longuet-Higgins [1]. Nous présentons également une analyse du cas dégénéré où les points sont coplanaires et établissons le lien entre la technique de Longuet-Higgins et celle de Tsai et Huang [2]. Dans le cas des droites, nous prouvons qu'au moins trois vues sont nécessaires et obtenons, à l'aide de faisceaux de plans, les équations satisfaites par les paramètres des déplacements. Nous montrons que six droites sont nécessaires pour estimer les paramètres du déplacement et que ceux-ci peuvent être recouverts de manière robuste, même en présence de bruit, à l'aide du filtre de Kalman étendu. Nous décrivons également une technique permettant de reconstruire des droites 3D à partir de plusieurs projections après le calcul du mouvement de la caméra, elle aussi fondée sur l'utilisation des faisceaux de plans. Des résultats expérimentaux sont présentés sur des données réelles et synthétiques.

1 Introduction

Nous étudions dans cet article l'application des techniques d'appariement de primitives à l'estimation du mouvement et de la structure à partir d'un nombre donné de vues. Nous supposons qu'une caméra mobile filme un environnement 3D statique et que son mouvement est inconnu. Le problème est donc, à partir d'une séquence de telles images, de recouvrer le mouvement de la caméra et la structure de l'environnement.

Nous ne nous attaquons pas au problème de recouvrer le mouvement instantané mais à celui de recouvrer, à partir d'une séquence F_1, \dots, F_n de vues, les déplacements rigides $D_{1,2}, \dots, D_{n-1,n}$. Ce qui arrive lorsque les intervalles de temps $\Delta t_{i,i+1}$ entre les vues tendent vers zéro n'est pas traité ici mais l'on peut se référer à [3] pour un cas particulier.

On peut grossièrement classer les approches du problème en deux catégories. L'approche du flot optique s'appuie sur le fait qu'il existe un ensemble de relations liant la position et le mouvement dans l'espace aux positions des images et au flot optique, ou vitesse apparente des motifs d'intensité. La solution de ces équations ne se comporte cependant numériquement pas bien. Une autre approche est d'extraire des primitives dans les images de la séquence, de les appairer d'image à image, et de recouvrer le mouvement et la structure de l'environnement. Nous pensons que cette approche est plus robuste que celle du flot optique,

L'approche par appariement de primitives nécessite cinq étapes:

1. Choix des primitives à calculer dans les images.
2. Calcul de ces primitives.
3. Appariement des primitives entre les images.
4. Calcul du mouvement 3D de la caméra.
5. Reconstruction de la structure de l'environnement.

Si nous voulons que les appariements soient significatifs, il faut que les primitives que nous calculons dans les images soient liées aux objets de la scène. Par liées, nous entendons que les primitives devraient correspondre à des réalités physiques telles que frontières de surfaces, ombres ou discontinuités de profondeur si nous sommes intéressés par les changements, ou les régions uniformes relativement à un certain critère, comme la lumière reflétée par une surface lambertienne, si nous sommes intéressés par l'absence de changements. Cela implique que nos primitives vont tomber dans deux grandes classes:

- Les points de contour et les structures dérivées comme les points d'intérêt (angles, maximum de courbure) ou les courbes.
- Les régions.

Dans cette article, nous nous concentrons sur les points.

Le calcul des contours et des régions uniformes dans les images est toujours un domaine de recherche active en Vision par Ordinateur mais il existe maintenant une bonne analyse théorique de ce que font les détecteurs de contours [4] et des détecteurs optimaux (selon certains critères) ont été proposés et longuement testés [5] [6] [7].

Le problème de l'appariement de primitives entre images n'a pas été étudié intensivement dans le cas du mouvement, où la position des caméras est inconnue. Il a, par contre, été très étudié dans

le domaine de la stéréovision où ces positions sont supposées connues et où la contrainte épipolaire réduit l'espace de recherche d'un espace de dimension 2 à un espace de dimension 1 [8] [9] [10] [11] [12]. Nous ne nous intéressons pas ici à ce problème.

Le sujet abordé dans cet article est, étant donné un certain nombre d'appariements entre un certain nombre d'images, de calculer le mouvement de la caméra et de reconstruire la structure de l'environnement. Nous sommes très intéressés par le problème de l'incertitude et des erreurs, et voulons développer des méthodes de calcul du mouvement et de la reconstruction qui soient robustes au bruit et qui permettent d'obtenir une estimation de l'incertitude finale sur les paramètres calculés.

Plus précisément, nous présentons une analyse orientée par ces critères de l'algorithme des huit points proposé par Longuet-Higgins [1] [13] dans le cas général et dans le cas dégénéré où les points sont coplanaires. Dans le cas général, nous proposons une expression explicite aux moindres carrés, qui possède de bonnes propriétés de robustesse; dans le cas planaire, nous établissons la relation entre l'algorithme des huit points et l'analyse de Tsai et Huang [2] et Faugeras et Lustman [14]. Nous traitons ensuite le problème de l'appariement de lignes entre plusieurs images, qui implique l'utilisation de plus de deux images. Nous montrons qu'au moins six lignes sont nécessaires pour recouvrer le déplacement et proposons une méthode d'estimation fondée sur l'utilisation du filtre de Kalman étendu. Nous comparons notre approche à celle de Mitiche, Seida et Aggarwal [15] [16] et à celle de Liu et Huang [17] [18].

Nous discutons ensuite le problème, étant donné ces estimations des déplacements rigides des caméras, de reconstruire l'environnement 3D et d'obtenir des estimations de l'incertitude. Nous traitons le cas des lignes.

2 Retour à l'algorithme des huit points

Etablissons tout d'abord un certain nombre de notations. Le modèle de caméra utilisé est le modèle simple à *sténopé* décrit en Figure 1.

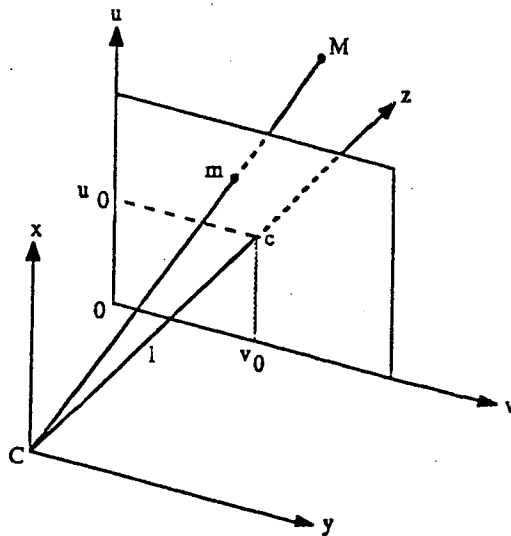


Figure 1 :

Un point M de l'espace 3D forme une image m dans le plan rétinien (O, u, v) par une transformation perspective de centre C , centre optique de la caméra. L'axe optique Cz perce le plan rétinien au point c de coordonnées (u_0, v_0) . La distance Cc est la distance focale de la caméra. Du fait de la manière dont les images sont obtenues électroniquement, il existe en général deux facteurs d'échelle k_u et k_v correspondant aux échelles selon les axes u et v . Les paramètres accessibles par calibration optique sont les produits $\alpha_u = fk_u$ et $\alpha_v = fk_v$ [19] [20]. En conséquence, on prend la distance focale f égale à 1. L'ensemble $(u_0, v_0, \alpha_u, \alpha_v)$ représente les paramètres intrinsèques de la caméra et est obtenu une fois pour toutes par calibration.

Nous négligeons certains effets importants tels que la distortion de la lentille mais ils peuvent de toute manière être éliminés [20] et ne changent donc pas le reste de notre analyse. Pour un pixel de coordonnées (u, v) le vecteur Cm exprimé dans le repère $Cxyz$ du système est égal à

$$\left(\frac{u - u_0}{\alpha_u}, \frac{v - v_0}{\alpha_v}, 1 \right)^t$$

Nous allons considérer le système de la Figure 2 où la caméra a été déplacée de sa première position, 1, à sa deuxième position, 2, selon une transformation D .

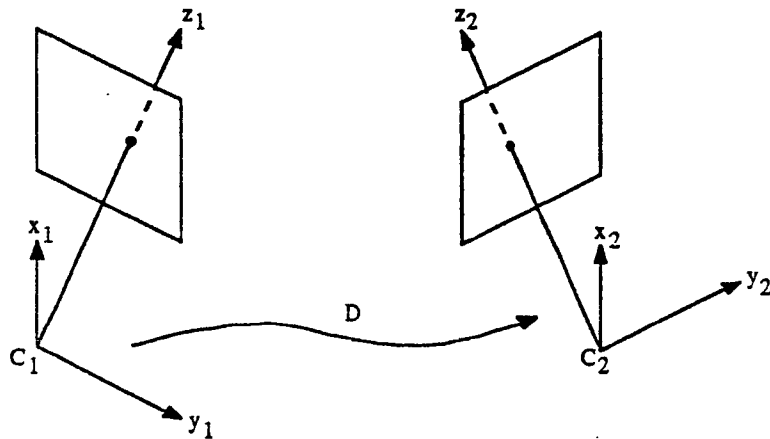


Figure 2 : Déplacement de la caméra de la position 1 à la position 2

Nous allons décomposer D comme le produit d'une rotation et d'une translation. Il y a une infinité de façons d'obtenir une telle décomposition selon le choix de l'axe de rotation. Si on pose la condition que l'axe de rotation doit passer par l'origine C_1 du repère lié à la caméra dans sa première position, alors la décomposition est unique et la translation qui suit la rotation R est représentée par le vecteur $C_1C_2 = t$.

La question que nous nous posons maintenant est, étant donné un certain nombre de points (m_{1i}, m_{2i}) mis en correspondance entre les images 1 et 2, de retrouver la rotation et la direction de la translation t . Le problème a été résolu par [1] dans le cas d'une solution exacte, sans bruit. En présence de bruit, la méthode est très sensible. Nous proposons ici une méthode qui peut faire face d'une façon robuste au problème du bruit. Elle est basée sur la contrainte de coplanarité utilisée par Longuet-Higgins.

Observons maintenant la Figure 3 où m_1 et m_2 sont les images dans les deux caméras du même point physique m : les trois vecteurs C_1m_1 , t et C_2m_2 sont coplanaires et leur déterminant est

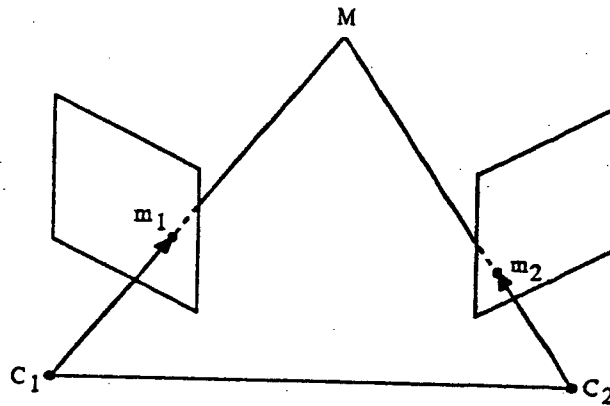


Figure 3 : La contrainte de coplanarité

donc nul. Ce que nous pouvons mesurer sont les coordonnées de $C_1 m_1$ et $C_2 m_2$ dans le repère de la Figure 2. Le vecteur $C_2 m_2$ exprimé dans le repère 1 est égal à $R \cdot C_2 m_2$. Dans le repère 1 nous avons ainsi:

$$C_1 m_1 \cdot (t \wedge R C_2 m_2) = 0 \quad (1)$$

En introduisant la matrice antisymétrique T :

$$T = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

l'équation (1) peut être réécrite:

$$C_1 m_1^t E C_2 m_2 = 0 \quad (2)$$

avec : $E = TR$.

E est définie seulement à une constante près, car on ne connaît que la direction de t . En posant $\|t\| = 1$ on obtient $\|E\|^2 = 2$.

Remarquons que (2) est une équation linéaire dans les coefficients de la matrice E . Soient X_1^t , X_2^t , et X_3^t ses vecteurs ligne et X le vecteur (9×1) $[X_1^t, X_2^t, X_3^t]^t$, nous pouvons écrire (2):

$$a^t X = 0 \quad (3)$$

où :

$$a^t = [x_1 x_2, x_1 y_2, x_1, y_1 x_2, y_1 y_2, y_1, x_2, y_2, 1]^t \quad (4)$$

et les x'_i et les y'_i sont les coordonnées normalisées de $C_1 m_1$ et $C_2 m_2$ dans les repères 1 et 2 respectivement. Avec n correspondances on obtient n équations et donc un système linéaire:

$$A_n X = 0$$

où A_n est une matrice $(9 \times n)$. Longuet-Higgins résoud ce système pour $n = 8$ quand le rang de A_8 est 8. Le noyau est donc de dimension 1 et X est le vecteur du noyau de norme $\sqrt{2}$. Les cas

dégénérés quand $\text{rang}(\mathbf{A}_8) < 8$ sont analysés dans [13] [21]. Nous discutons plus loin le cas où les points 3D sont coplanaires, ce qui donne aussi une matrice \mathbf{A}_8 dégénéré.

Une amélioration évidente de l'algorithme de Longuet-Higgins est de résoudre le système 2 en utilisant des techniques de moindres carrés. En remarquant que $\|\mathbf{X}\|^2 = 2$ et $\|\mathbf{t}\|^2 = 2$, ceci est équivalent à résoudre :

$$\min_{\mathbf{X}} \|\mathbf{A}_n \mathbf{X}\|$$

sous la condition :

$$\|\mathbf{X}^2\| = 2$$

La solution est le vecteur propre de la matrice $\mathbf{A}_n^t \mathbf{A}_n$ de norme $\sqrt{2}$ correspondante à la plus petite valeur propre.

Le problème suivant est, étant donnée la solution \mathbf{X} , de retrouver \mathbf{R} et \mathbf{t} . Commençons d'abord avec \mathbf{t} . Nous allons présenter deux approches.

La première consiste à remarquer que la matrice \mathbf{E} satisfait l'équation suivante:

$$\mathbf{E}\mathbf{E}^t = \mathbf{T}\mathbf{R}\mathbf{R}^t\mathbf{T}^t = -\mathbf{T}^2 = \begin{bmatrix} t_y^2 + t_z^2 & -t_x t_y & -t_x t_z \\ -t_x t_y & t_x^2 + t_z^2 & -t_y t_z \\ -t_x t_z & -t_y t_z & t_x^2 + t_y^2 \end{bmatrix}$$

La matrice $\mathbf{E}\mathbf{E}^t$ étant connue d'après l'estimation de \mathbf{X} , on peut résoudre le problème de moindres carrés suivant:

$$\min_{\mathbf{t}} \|\mathbf{E}\mathbf{E}^t + \mathbf{T}^2\|$$

sous la condition:

$$\|\mathbf{t}^2\| = 1$$

La solution est le vecteur propre correspondant à la plus petite valeur propre de la matrice $\mathbf{E}\mathbf{E}^t - \mathbf{I}$. La preuve se trouve dans l'appendice A.

Une autre approche [3] consiste à remarquer que $\mathbf{t}^t \mathbf{E} = \mathbf{t}^t \mathbf{T}\mathbf{R} = \mathbf{0}$. \mathbf{t} est donc la solution du problème de moindres carrés suivant:

$$\min_{\mathbf{t}} \|\mathbf{E}^t \mathbf{t}\|$$

sous la condition:

$$\|\mathbf{t}^2\| = 1$$

La solution est le vecteur propre correspondant à la plus petite valeur propre de la matrice $\mathbf{E}\mathbf{E}^t$. La solution est la même que dans le cas précédent car les matrices $\mathbf{E}\mathbf{E}^t - \mathbf{I}$ et $\mathbf{E}\mathbf{E}^t$ ont les mêmes vecteurs propres.

Nous allons maintenant estimer la matrice de rotation \mathbf{R} . Celle-ci est obtenue en minimisant par rapport à \mathbf{R} le critère $\|\mathbf{E} - \mathbf{T}\mathbf{R}\|$ qui peut être résolu facilement en utilisant, par exemple, la représentation de la rotation par des quaternions [19].

On a ainsi obtenu une solution analytique pour la rotation et la direction de la translation dans le sens des moindres carrés. Pour tester la validité de l'approche et sa sensibilité au bruit, nous avons utilisé des données synthétiques et réelles, et comparé nos résultats avec ceux obtenus par la méthode de Longuet-Higgins.

3 Cas Planaire

Lorsque les points utilisés lors de la précédente estimation se trouvent dans un même plan, l'analyse précédente se trouve invalidée parce que le rang de la matrice A_n devient strictement inférieur à 8. Nous prouvons même que, dans ce cas, $\text{rang}(A_n) = 6$.

Rappelons tout d'abord que, dans le cas planaire, il existe une relation homographique entre les coordonnées (x_2, y_2) et (x_1, y_1) des images d'un point 3D, qu'on peut écrire, en coordonnées projectives:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} a & b & c \\ e & f & g \\ k & l & m \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = A \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}$$

Remplacer x_2 et y_2 par leurs valeurs dans l'équation (4) nous permet de réécrire la matrice A_n sous la forme: $A_n = D_n A'_n$ où D_n est une matrice diagonale égale à $\text{diag}(\frac{1}{kx_1^2 + ly_1^2 + m})$ et A'_n est une matrice $n \times 9$.

Les termes de la diagonale de D_n sont non nuls, donc $\text{rang}(A'_n) = \text{rang}(A_n)$.

Un simple examen de la matrice montre en outre que les trois vecteurs

$$\begin{aligned} Y_1^t &= (0 & -k & e & 0 & -l & f & 0 & -m & g) \\ Y_2^t &= (k & 0 & -a & l & 0 & -b & m & 0 & -c) \\ Y_3^t &= (-e & a & 0 & -f & b & 0 & -g & c & 0) \end{aligned}$$

appartiennent au noyau de A'_n . En conséquence, $\text{rang}(A_n) \leq 6$.

Pour déterminer le rang de A'_n , notons Z_i ses 9 vecteurs colonnes. Nous devons résoudre le système linéaire en λ_i :

$$\sum_{i=1}^9 \lambda_i Z_i = 0 \quad (5)$$

Les équations (5) sont équivalentes à

$$\forall x_1, \forall y_1 \quad \lambda^t a = 0$$

avec $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \lambda_8, \lambda_9)^t$

Plus explicitement, on obtient :

$$\begin{aligned} \lambda^t a &= x_1^2(\lambda_1 a + \lambda_2 e + \lambda_3 k) + y_1^2(\lambda_4 b - \lambda_5 f + \lambda_6 l) + \\ & x_1 y_1(\lambda_1 b + \lambda_2 f + \lambda_3 l + \lambda_4 a + \lambda_5 e + \lambda_6 k) + \\ & x_1(\lambda_1 c + \lambda_2 g + \lambda_3 m + \lambda_7 a + \lambda_8 e + \lambda_9 k) + y_1(\lambda_4 c + \lambda_5 g + \lambda_6 m + \lambda_7 b + \lambda_8 f + \lambda_9 l) + \\ & \lambda_7 c + \lambda_8 g + \lambda_9 m \end{aligned}$$

Si cette expression doit être nulle quels que soient x_1 et y_1 , alors:

$$B\lambda = 0$$

la matrice B étant égale à

$$\begin{pmatrix} a & e & k & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b & f & l & 0 & 0 & 0 \\ b & f & l & a & e & k & 0 & 0 & 0 \\ c & g & m & 0 & 0 & 0 & a & e & k \\ 0 & 0 & 0 & c & g & m & b & f & l \\ 0 & 0 & 0 & 0 & 0 & 0 & c & g & m \end{pmatrix}$$

Considérons tout d'abord le rang de la matrice A de l'équation 3. Il est connu [2] que la matrice A s'écrit :

$$A = dR + tn^t$$

où $n = (n_1, n_2, n_3)^t$ et d représentent respectivement la normale et la distance du plan exprimé dans le repère de la figure 1. Un vecteur x du noyau de A doit satisfaire l'équation :

$$(dI + C)x = 0$$

où la matrice C est égale à (n_1R^t, n_2R^t, n_3R^t) . x est donc un vecteur propre de la matrice C associé à la valeur propre $-d$. C possède 0 comme valeur propre double et t^tRn comme valeur propre simple. La première de ces valeurs signifie que le plan passe par l'origine et la seconde qu'il est dans une position particulière. Ces singularités mises à part, le rang de A est en général 3.

Considérons maintenant les trois premiers vecteurs colonne de B : le rang de A étant égal à 3, ils engendrent le même sous-espace de R^6 que (e_1, e_3, e_4) , où e_i représente la base canonique de R^6 . Pour la même raison, les trois colonnes suivantes engendrent le même sous-espace que (e_2, e_3, e_5) et les trois dernières colonnes le même sous-espace que (e_4, e_5, e_6) . Il en résulte donc immédiatement que les neuf vecteurs colonne de B engendrent R^6 , ce qui prouve que le rang de A'_n est 6.

Il en résulte que la solution X des équations 3 peut s'écrire

$$X = \alpha_1 Y_1 + \alpha_2 Y_2 + \alpha_3 Y_3$$

En utilisant les expressions de Y_1, Y_2, Y_3 , on trouve que la matrice E s'écrit $E = c(\alpha)A$ où $c(\alpha)$ est une matrice 3×3 représentant le produit vectoriel par le vecteur $\alpha = (\alpha_1, \alpha_2, \alpha_3)^t$, c'est-à-dire $c(\alpha)x = \alpha \wedge x$. Comme nous savons que $t^tE = 0$, il en résulte que $t^tc(\alpha)A = 0$

Comme $\text{rang}(A) = 3$, $c(\alpha)t = 0$ et α est donc parallèle à t : $t = d\alpha$

En écrivant que $EE^t = -T^2$, nous obtenons l'équation :

$$TAA^tT = d^2T^2 \quad (6)$$

Si on écrit les équations (6) dans le repère propre de AA^t , on obtient :

$$T \text{diag}(d_1^2, d_2^2, d_3^2)T = d^2T^2$$

où les d_i sont les valeurs singulières de A . Ceci nous donne alors le système d'équations :

$$\begin{cases} (d^2 - d_3^2)x_2^2 + (d^2 - d_2^2)x_3^2 = 0 \\ (d^2 - d_3^2)x_1^2 + (d^2 - d_1^2)x_3^2 = 0 \\ (d^2 - d_2^2)x_1^2 + (d^2 - d_1^2)x_2^2 = 0 \\ (d^2 - d_1^2)x_2x_3 = 0 \\ (d^2 - d_2^2)x_3x_1 = 0 \\ (d^2 - d_3^2)x_1x_2 = 0 \end{cases}$$

où $x = U^t t$, et U est la matrice orthogonale des vecteurs propres de AA^t . Le système des trois premières équations est identique à celui de [14]. La solution est donc la même et ne dépend que de l'ordre de multiplicité des valeurs singulières. Rappelons que, si $d_1 \neq d_2 \neq d_3$, il y a deux solutions, et une seulement si deux parmi les trois valeurs singulières sont égales. Ceci nous fournit le vecteur t , et nous pouvons alors calculer la matrice de rotation R en résolvant le système :

$$dR = TA$$

d, T et A sont connus, et R peut être obtenue en utilisant la technique proposée dans [1] et [22]. Remarquons que 3 est exacte, c'est-à-dire nous n'avons pas à résoudre un problème de moindres carrés comme dans la section précédente. Connaissant R , nous pouvons obtenir la normale:

$$\mathbf{n} = \mathbf{A}^t \mathbf{t} - d \mathbf{R}^t \mathbf{t}$$

Cette méthode de calcul des paramètres du mouvement et du plan met en évidence la relation existant entre les méthodes de Longuet-Higgins [1] et de Tsai et Huang [2].

4 Appariement de droites

L'utilisation de droites comme primitives d'appariement a été récemment proposée par Mitiche, Seida et Aggarwal [15] [16] et Liu et Huang [17] [18]. Les segments de droites peuvent être extraits des images de manière fiable par approximation polygonale et forment donc une source de primitives robustes pour un grand nombre de procédés d'appariement utilisés en Vision par Ordinateur (reconnaissance et positionnement d'objets, stéréovision). Contrairement au cas des points qui a été traité précédemment, appairer deux lignes dans deux vues comme étant l'image d'une même droite 3D ne contraint pas le déplacement rigide entre ces vues, ainsi que l'illustre la Figure 4.

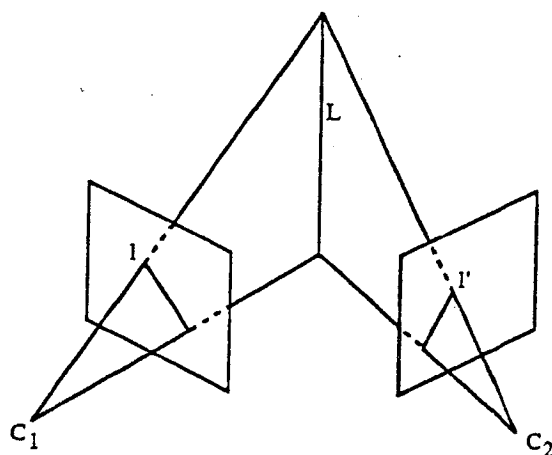


Figure 4

En fait, appairer l et l' définit simplement une droite 3D L mais n'impose pas de contrainte sur les positions de C_1 et C_2 (comparer avec la Figure 3). Il en résulte donc, comme la remarque en est faite dans [15] [16] [17] et [18], que nous devons utiliser au moins trois vues.

Considérons donc la situation décrite en Figure 5.

Nous voulons obtenir les équations nécessaires et suffisantes que doivent vérifier les déplacements rigides D_{12} et D_{23} pour que les droites l , l' et l'' dans les rétines 1, 2 et 3 soient les images d'une même droite 3D L (condition H).

Définissons les trois plans (C_1, l) , (C_2, l') et (C_3, l'') . Dans un repère donné où l'équation d'un plan est $ux + vy + wz + d = 0$, nous représentons ce plan par un vecteur de dimension 4. Le vecteur $\mathbf{n} = (u, v, w, d)^t$ est normal au plan $P = (u, v, w, d)^t$

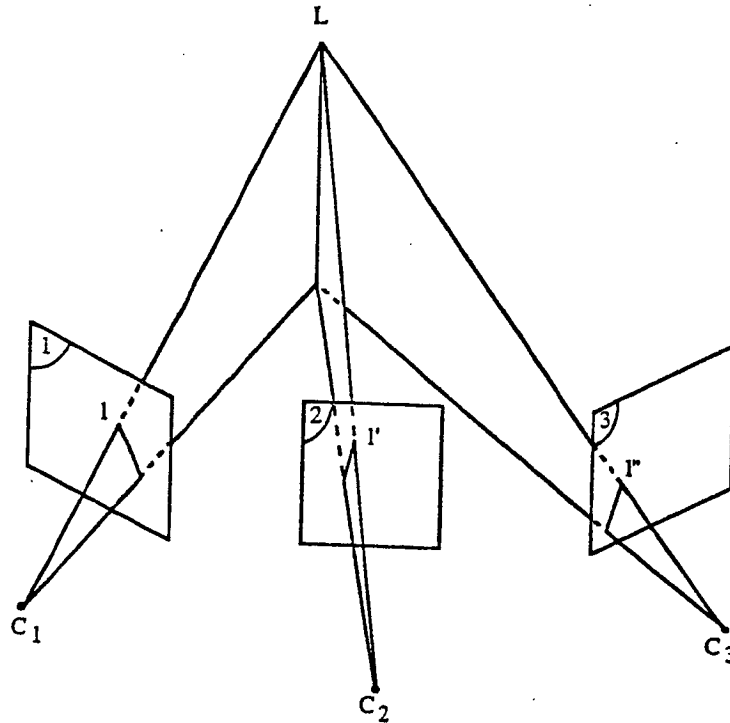


Figure 5

Etant donnés deux plans P_1 et P_2 , l'ensemble des plans passant par leur intersection, ou faisceau linéaire de base P_1 et P_2 , est l'ensemble des plans représentés par les vecteurs $\lambda_1 P_1 + \lambda_2 P_2$, pour des réels quelconques λ_1, λ_2 . Si les deux plans sont parallèles, on obtient un faisceau de plans parallèles.

Appelons P_1, P_2 , et P_3 les plans $(C_1, l), (C_2, l')$ et (C_3, l'') dans un repère donné quelconque. La condition H est équivalente à :

les plans P_i appartiennent au même faisceau

qui est à son tour équivalente à :

il existe λ_1, λ_2 et λ_3 non tous nuls tels que :

$$\lambda_1 P_1 + \lambda_2 P_2 + \lambda_3 P_3 = 0$$

ou encore :

le rang de la matrice 4×3 $A = (P_1 \ P_2 \ P_3)$ est inférieur ou égal à 2

enfin équivalent à :

les quatre déterminants de taille 3 extraits de la matrice A sont tous nuls.

Pour calculer ces déterminants, choisissons le repère attaché à la première rétine. Dans ce repère, D_{12} est représenté par une matrice de rotation R_1 autour d'un axe passant par C_1 et une translation

t_1 ; de même, D_{23} est représenté par une matrice de rotation R_2 autour d'un axe passant par C_2 et une translation t_2 . Notons n , n' et n'' les normales au trois plans (C_1, l) , (C_2, l') et (C_3, l'') dans les repères respectivement attachés aux rétines 1, 2, et 3. Ces normales se calculent aisément en fonction des segments de droites l , l' et l'' et des paramètres intrinsèques de la caméra. Il est aisé de montrer que, dans le repère de la rétine 1, les trois plans sont représentés par les vecteurs

$$\begin{aligned} P_1 &= (\quad n^t \quad \quad \quad 0 \quad \quad) \\ P_2 &= (\quad n'^t R_1^t \quad \quad \quad -t_1^t R_1 n' \quad) \\ P_3 &= (\quad n''^t R_2^t R_1^t \quad \quad \quad -(t_1 + R_1 t_2)^t R_1 R_2 n'' \quad) \end{aligned}$$

Parmi les quatre déterminants, le premier ne fait intervenir que les rotations : c'est

$$D_1 = \begin{vmatrix} n & R_1 n' & R_1 R_2 n'' \end{vmatrix}$$

Plus géométriquement :

$$D_1 = n \cdot (R_1 n' \wedge R_1 R_2 n'') = R_1^t n \cdot (n' \wedge R_2 n'')$$

C'est l'expression trouvée par Liu et Huang [17] [18].

Il y a trois autres déterminants qui font intervenir à la fois les translations et les rotations. Il s'avère en fait qu'ils sont tous proportionnels (cf. appendice C). Appelons donc D_2 l'un d'entre eux.

Nous avons prouvé que :

La condition (H) est équivalente aux 2 équations $D_1 = D_2 = 0$.

Plusieurs commentaires peuvent être effectués à ce point. Tout d'abord, le nombre d'inconnues est de 6 pour les rotations et 5 pour les translations (puisque l'échelle absolue est inconnue, ce qu'on retrouve d'ailleurs dans les équations car $D_2 = 0$ est homogène en t_1 et t_2). En conséquence, le nombre minimum de droites nécessaires à l'estimation des paramètres du mouvement est 6.

Nous avons montré que :

Pour recouvrer le mouvement 3D en appariant des droites entre trois vues, il est nécessaire d'apparier au moins 6 points

Cela contredit un résultat d'Aggarwal et Mitiche [15] [16] qui affirment que 4 lignes sont suffisantes. Leur méthode est fondée sur l'introduction d'une inconnue par droite projetée et l'expression de la contrainte de rigidité de la scène comme la conservation des angles des droites entre elles. Ces angles ne font pas intervenir les rotations et les translations. Chaque paire de lignes fournit deux équations et on obtient donc $2C_n^2$ équations et $3n$ inconnues pour n droites. Pour $n = 4$, le nombre d'équations égale le nombre d'inconnues, mais notre résultat montre que les équations ainsi obtenues ne peuvent être indépendantes.

Une seconde remarque est que nous pouvons découpler l'estimation des rotations de celle des translations, puisque D_1 ne fait pas intervenir les translations.

Penchons nous maintenant sur le problème de l'estimation des déplacements à partir d'un nombre donné de triplets de droites 2D.

5 Estimation des déplacements par appariement de droites

La technique utilisée est celle du filtre de Kalman étendu que nous avons proposé [23] comme outil général de résolution d'un certain nombre de problèmes en Vision. Nous représentons une rotation par un vecteur \mathbf{r} de dimension 3 dont la direction est celle de l'axe de rotation et la norme l'angle de rotation. On peut alors écrire:

$$\mathbf{R}\mathbf{u} = \mathbf{u} + \frac{\sin \theta}{\theta} \mathbf{r} \wedge \mathbf{u} + \frac{1 - \cos \theta}{\theta} \mathbf{r} \wedge (\mathbf{r} \wedge \mathbf{u})$$

Rappelons brièvement le cas général puis appliquons le à notre problème. Nous obtenons une mesure $\mathbf{x} \in \mathbf{R}^m$ qui dépend d'un paramètre $\mathbf{a} \in \mathbf{R}^n$ d'une manière non linéaire qui peut s'exprimer par la relation

$$f(\mathbf{x}, \mathbf{a}) = 0$$

où f est une application de $\mathbf{R}^m \times \mathbf{R}^n$ dans \mathbf{R}^p . Nous supposons que l'observation \mathbf{x} est bruitée par un bruit gaussien de moyenne nulle:

$$\mathbf{x} = \mathbf{x}' + \epsilon$$

$$\mathbf{E}(\epsilon) = 0$$

$$\mathbf{E}(\epsilon\epsilon^t) = \Lambda$$

Le problème est, étant donné un certain nombre d'observations \mathbf{x}_i , de déterminer le vecteur \mathbf{a} qui satisfait *au mieux* les relations $f(\mathbf{x}_i, \mathbf{a}_i) = 0$. Le sens de *au mieux* sera explicité dans un moment.

Supposons que nous possédions une *bonne* estimation \mathbf{a}^* de \mathbf{a} . Nous pouvons alors linéariser f au voisinage de $(\mathbf{x}', \mathbf{a}^*)$:

$$f(\mathbf{x}', \mathbf{a}) = 0 \approx f(\mathbf{x}, \mathbf{a}^*) + \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{a}^*)(\mathbf{x}' - \mathbf{x}) + \frac{\partial f}{\partial \mathbf{a}}(\mathbf{x}, \mathbf{a}^*)(\mathbf{a} - \mathbf{a}^*)$$

où $\frac{\partial f}{\partial \mathbf{x}}$ est une matrice $p \times m$ et $\frac{\partial f}{\partial \mathbf{a}}$ une matrice $p \times n$. Cette expression peut se réécrire (on abandonne le signe \approx):

$$-f(\mathbf{x}, \mathbf{a}^*) + \frac{\partial f}{\partial \mathbf{a}} \mathbf{a}^* = \frac{\partial f}{\partial \mathbf{a}}(\mathbf{x}, \mathbf{a}^*)\mathbf{a} - \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{a}^*)\epsilon$$

qui est une équation de mesure linéaire:

$$\mathbf{y} = \mathbf{M}\mathbf{a} + \mathbf{u}$$

où

$$\mathbf{y} = -f(\mathbf{x}, \mathbf{a}^*) + \frac{\partial f}{\partial \mathbf{a}}(\mathbf{x}, \mathbf{a}^*)\mathbf{a}^*$$

$$\mathbf{M} = \frac{\partial f}{\partial \mathbf{a}}(\mathbf{x}, \mathbf{a}^*)$$

et

$$\mathbf{u} = \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{a}^*)\epsilon$$

Remarquons que \mathbf{y} et \mathbf{M} sont connus ainsi que les statistiques du second ordre sur \mathbf{u} :

$$\mathbf{E}(\mathbf{u}) = 0$$

$$W = E(uu^t) = \frac{\partial f}{\partial x}(x, a^*) \Lambda \frac{\partial f}{\partial x}(x, a^*)^t$$

Nous mesurons y car nous connaissons f et a^* et observons x , version altérée de x' . Pour les mêmes raisons, M et $\frac{\partial f}{\partial x}(x, a^*)$ sont connus. Donc, si nous avons n mesures x_1, \dots, x_n satisfaisant $f(x_i, a_i) = 0$ et si nous avons une estimée initiale \hat{a}_0 de a et de sa matrice de covariance associée $S_0 = E((\hat{a}_0 - a)(\hat{a}_0 - a)^t)$, nous pouvons utiliser l'approche du filtre de Kalman pour déduire récursivement une estimée \hat{a}_n de a et de sa matrice de covariance $S_n = E((\hat{a}_n - a)(\hat{a}_n - a)^t)$, après avoir pris en compte n observations. Les équations récursives correspondantes sont les équations standard du filtre de Kalman et sont rappelées dans l'appendice D.

Nous pouvons maintenant donner un sens précis à l'expression *au mieux*. a_n est le vecteur qui minimise le critère:

$$(a - a_0)^t S_0^{-1} (a - a_0) + \sum_{i=1}^n (y_i - M_i a)^t W_i^{-1} (y_i - M_i a)$$

Cette équation est importante car elle montre comment le filtre de Kalman prend explicitement en compte le bruit de mesure et le pondère. Plus le bruit sur la i -ième mesure est grand, plus l'inverse W_i^{-1} de la matrice de covariance est petite, et, par conséquent, moins le i -ième terme contribue, dans le critère précédent, à l'estimée finale.

Etudions maintenant en détail les applications de cette technique à notre problème. D_1 est une équation non linéaire $f(x, a)$ qui lie les mesures $x = (\begin{smallmatrix} n^t & n^{tt} & n^{ttt} \end{smallmatrix})^t$ au paramètre inconnu $a = (\begin{smallmatrix} r_1^t & r_2^t \end{smallmatrix})^t$.

Etant parti d'une estimée initiale \hat{a}_0 des rotations et de leur matrice de covariance S_0 , et après avoir utilisé les n mesures x_1, \dots, x_n , décrivant les appariements, le filtre de Kalman étendu fournit la *meilleure* estimation a_n des rotations et leur matrice de covariance S_n .

Le mot *meilleur* est pris au sens de la minimisation du critère précédent qui ne fait intervenir que les dérivées partielles de f par rapport à x et à a .

Ces dérivées partielles peuvent être calculées en utilisant, par exemple, les résultats décrits dans [23].

Une fois r_1 et r_2 calculés, nous pouvons calculer t_1 et t_2 très simplement à l'aide de l'équation D_2 qui est linéaire en t_1 et t_2 .

6 Calcul de la structure de la scène

Ayant obtenu des estimations de R_1 , R_2 , t_1 et t_2 , nous voulons maintenant reconstruire les droites 3D dans un repère quelconque. A cette intention, nous devons choisir une représentation des droites 3D. Quatre paramètres suffisent à décrire une droite 3D. Une représentation minimale possible en fonction de quatre paramètres (a, b, p, q) est la suivante:

$$\begin{cases} x = az + p \\ y = bz + q \end{cases} \quad (7)$$

Dans ce cas, la droite est paramétrée en fonction de z , ce qui exclut le cas des droites perpendiculaires à l'axe des z . Pour ces lignes, une représentation en x ou en y sera choisie.

Revenant à la Figure 5, nous voulons représenter la droite L dans les équations (7) étant donné les déplacements rigides D_{12} et D_{23} et les droites d , d' et d'' . En quoi cela représente-t-il un problème? A cause du bruit présent dans nos mesures, les trois plans (C_1, l) , (C_2, l') et (C_3, l'')

ne s'intersectent pas exactement selon une droite de sorte que nous devons déterminer la *meilleure* droite d'intersection. Nous pouvons faire cela en utilisant nos idées précédentes : les faisceaux de plans et le filtre de Kalman.

En effet, les équations (7) représentent deux plans $P = (1, 0, -a, -p)^t$ et $R = (0, 1, -b, -q)^t$. Si nous considérons le faisceau engendré par Q et R , nous constatons qu'il doit contenir les plans P_1 , P_2 et P_3 . En conséquence, les matrices $H_1 = (P_1 \ Q \ R)$, $H_2 = (P_2 \ Q \ R)$ et $H_3 = (P_3 \ Q \ R)$ doivent toutes trois être de rang inférieur ou égal à 2. Comme dans la section précédente, ceci est équivalent à écrire que, pour chaque matrice H_i , les quatre sous-déterminants de taille 3 sont nuls. De nouveau, on montre que deux d'entre eux seulement sont indépendants (cf. appendice E) et nous obtenons donc 6 équations linéaires en nos inconnues a , b , p et q :

$$\begin{cases} ap_i + bq_i + r_i = 0 \\ pp_i + qq_i + s_i = 0 \end{cases} \quad (8)$$

où $P_i = (p_i, q_i, r_i, s_i)^t$

En conséquence, nous pouvons résoudre (8) aux moindres carrés ou bien, si nous sommes intéressés à obtenir l'incertitude sur les paramètres, à l'aide du filtre de Kalman, la covariance sur les P_i ayant été obtenue comme résultat de la section précédente.

7 Implémentation

Nous avons implémenté la méthode décrite dans la section 2 sur des données réelles et sur des données synthétiques, et l'avons comparée avec la méthode originale de Longuet-Higgins. Pour que la comparaison soit équitable, le vecteur X_9 est estimé par la même méthode dans les deux cas, c'est-à-dire par moindres carrés comme il est expliqué dans la section 2. Il est intéressant de remarquer que nous avons également implémenté la technique, proposée par Huang [22], de résoudre $A_n X = 0$ en imposant une des coordonnées de X égale à 1 et des moindres carrés sur l'équation $A_n' X' + b_n = 0$ où A_n' est la matrice $n \times 8$ obtenue en éliminant la i -ième colonne b_n de A_n (i est le numéro de la coordonnée égale à '1'), mais il ne semble pas exister de bonne corrélation entre nombre de conditionnement de la matrice $A_n'^t A_n$ et la qualité des résultats obtenus. Ce que nous entendons par *qualité* est défini dans la suite.

L'image de la scène 3D test utilisée est montrée en Figure 6. Elle est constituée de 10 points situés à une distance d'environ 4 mètres de la caméra. La taille de l'objet est également d'environ 4 mètres. La matrice de transformation perspective que nous utilisons pour la projection des points 3D a été obtenue par calibration des caméras réelles [19]. Nous avons simulé un certain nombre de déplacements des caméras et du bruit dans les mesures des pixels. La résolution est 512×512 est le bruit varie de 1 à 5 pixels. Une fois estimée la rotation R et la direction t de la translation, nous pouvons reconstruire les points 3D aux moindres carrés et les reprojeter sur les deux rétines avant et après mouvement. Nous pouvons alors calculer la distance moyenne entre les pixels réels et reconstruits. La somme de ces deux mesures constitue notre critère de qualité.

La Table 1 montre quelques résultats typiques pour chacune des deux techniques. Deux remarques peuvent être faites. Tout d'abord, les deux techniques apparaissent assez robustes à des distortions assez importantes, contrairement à ce qui a été jusqu'à maintenant rapporté dans la littérature. Ensuite, notre méthode se comporte au moins aussi bien que celle de Longuet-Higgins dans tous les cas et significativement mieux dans certains cas. La Figure 7 montre les points recalculés et reprojétés sur les deux rétines pour notre méthode dans le cas de la ligne 2 de la Table 1.

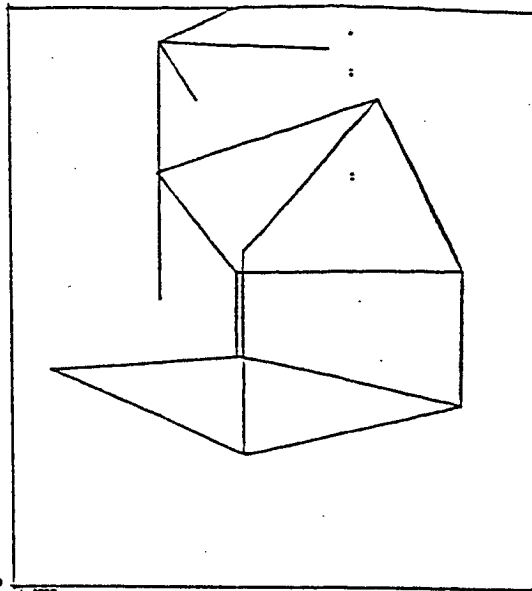


Figure 6 : Image des points synthétiques originaux

Nous avons également fait des expériences sur des données réelles qui ont été utilisées dans les expériences de calibration des caméras. Nous avons pris quatre vues stéréo d'une grille (cf. Figure 8) constituée de lignes noires sur fond blanc à une distance de 3, 3.5, 4 et 4.5 mètres. Nous en avons extrait automatiquement les droites dans les quatre images et calculé avec une grande précision les coordonnées en pixels de leurs intersections, ce qui nous a fourni 113 points dans chaque image. Notre méthode de calibration stéréo, décrite dans [19] nous donne le déplacement (rotation R et translation t) de la caméra 1 à la caméra 2, en utilisant au moins deux plans de données. Nous avons alors utilisé notre technique sur les points des deux autres plans et estimé R et la direction de la translation t . Les résultats se trouvent en Table 2. Ils montrent que cette technique d'estimation du déplacement de la caméra donne d'excellents résultats.

Nous avons aussi implémenté la méthode décrite dans les sections 4 et 5 sur des données réelles et sur des données synthétiques. Nos données synthétiques sont les mêmes que celles utilisées pour l'expérience précédente, excepté le fait qu'au lieu d'utiliser les points nous avons utilisé certaines des droites les reliant. De nouveau, nous avons simulé un certain nombre de déplacements des caméras et du bruit dans les mesures. Le bruit est ajouté sur les mesures des pixels définissant les extrémités des segments et varie entre 0.1 et 3 pixels (c'est-à-dire un bruit uniforme dans un intervalle de longueur double de ces valeurs). Le filtre de Kalman étendu n'a pour le moment été utilisé que pour l'estimation de la rotation, en utilisant l'équation $D_1 = 0$, la translation étant estimée aux moindres carrés. Les résultats se trouvent en Table 3.

Pour ce qui est des données réelles, nous avons pris un ensemble de 8 triplets de vues stéréo, chacune décalée de la première d'une rotation de 5 degrés environ, à l'aide d'une tourelle commandable en rotation. Les appariements ont été obtenus par une procédure d'appariement automatique de droites 3D obtenues par un algorithme de stéréovision trinoculaire [24]. Les résultats se trouvent en Table 4.

La stabilité de la valeur de l'angle, autour d'une valeur de 5.3 degrés, et de la direction de l'axe de rotation, semblent montrer que notre méthode est assez précise pour corriger les imprécisions

axe de rotation : (1,1,1)

direction de la translation: (1,0,0)

θ	t	∂p	av	sd	θ'	$\theta\%$	α	β	
5	10	1	57.8	7.2	5.89	17.8	24.59	26.24	(1)
					5.36	7.2	11.48	27.58	(2)
5	50	1	91.8	14.3	15.18	203.6	42.21	4.53	(1)
					5.72	14.5	6.78	1.78	(2)
5	100	1	139.6	31.1	11.03	120.7	37.31	3.09	(1)
					5.74	14.8	6.30	0.82	(2)
5	10	5	57.7	8.1	6.06	21.2	47.34	72.43	(1)
					5.04	0.8	72.61	(2)	
5	50	5	91.5	14.7	20.01	300.2	115.43	24.59	(1)
					8.37	67.3	37.77	11.48	(2)
5	100	5	139.4	31.1	55.9	1018	121	23.2	(1)
					9.1	82	22.3	5.06	(2)
15	50	5	189.1	20.4	34.6	130.7	35.6	6.68	(1)
					19.5	30.3	8.5	7.84	(2)
15	100	5	232.3	30.3	25.15	67.6	24.98	4.19	(1)
					19.73	31.6	5.37	1.93	(2)

θ : angle exact de rotation (en degrés)

θ' : angle calculé de rotation (en degrés)

t : norme de la translation exacte

$\theta\%$: erreur sur l'angle de rotation (en pourcents)

∂p : bruit de mesure (en pixels)

av : moyenne des déplacements en pixels image

sd : écart type des déplacements en pixels image

α : erreur angulaire absolue sur l'axe

β : erreur angulaire absolue sur la direction de la translation

(1) = Longuet-Higgins amélioré

(2) = nouvelle technique

Table 1

de la commande.

Notre méthode semble donc bien se comporter en présence de données bruitées. Remarquons enfin que notre estimée initiale a toujours été la matrice de rotation identité mais qu'un autre avantage important de notre méthode est qu'elle permet l'utilisation de connaissances *a priori* sur la rotation, par les paramètres initiaux a_0 et S_0 .

Nous avons, enfin, implémenté notre technique de reconstruction et l'avons appliquée aux droites issues de l'algorithme de stéréovision trinoculaire [24]. Nous montrons les résultats obtenus sur une scène de bureau, en vue de face (Figure 9) et de dessus (Figure 10).

8 Conclusion

La conclusion principale de cet article est qu'il semble possible d'extraire le mouvement et la structure de manière fiable par appariement de points et de droites entre plusieurs vues. Il faut insister sur deux points : les techniques de moindres carrés et du filtre de Kalman étendu peuvent

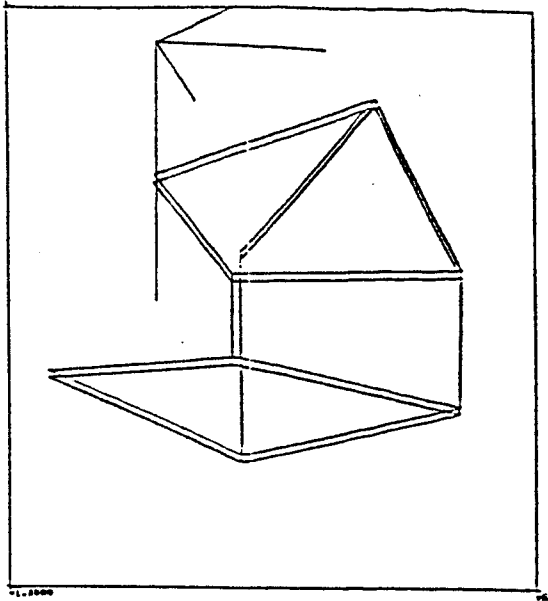


Figure 7 : Images des points originaux et reconstruits superposés

effectivement être utilisées en présence de bruit dans les images, et les idées de la géométrie projective, par exemple les faisceaux de plans, constituent un outil puissant de résolution de nombreux problèmes d'analyse du mouvement. Nous prédisons même que ces techniques vont s'avérer encore plus fructueuses dans un avenir proche.

De nombreux problèmes doivent, bien sûr, encore être résolus. Premièrement, les propriétés de convergence du filtre de Kalman doivent être mieux comprises. Enfin, et ce n'est pas le moindre problème, des techniques permettant d'obtenir des appariements corrects doivent être développées.

A

Nous voulons déterminer le vecteur t solution du problème de minimisation :

$$\min_{\|t^2\|=1} \|\mathbf{E}\mathbf{E}^t + \mathbf{T}^2\|$$

Par définition de la norme matricielle :

$$\|\mathbf{E}\mathbf{E}^t + \mathbf{T}^2\|^2 = \text{Tr}((\mathbf{E}\mathbf{E}^t + \mathbf{T}^2)^2)$$

La contrainte nous donne par ailleurs : $\mathbf{T}^2 = -\mathbf{I} + \mathbf{t}\mathbf{t}^t$ Nous obtenons donc, en notant $\mathbf{F} = \mathbf{E}\mathbf{E}^t - \mathbf{I}$:

$$\begin{aligned} \text{Tr}((\mathbf{E}\mathbf{E}^t + \mathbf{T}^2)^2) &= \text{Tr}(\mathbf{F}^2) + \text{Tr}(\mathbf{t}\mathbf{t}^t\mathbf{t}\mathbf{t}^t) + 2\text{Tr}(\mathbf{F}\mathbf{t}\mathbf{t}^t) \\ &= \text{Tr}(\mathbf{F}^2) + \|\mathbf{t}\|^4 + 2\mathbf{t}^t\mathbf{F}\mathbf{t} \end{aligned}$$

Comme $\text{Tr}(\mathbf{F}^2)$ ne dépend pas de t , et que $\|\mathbf{t}^2\| = 1$, t est le vecteur propre normé de \mathbf{F} correspondant à la plus petite valeur propre.

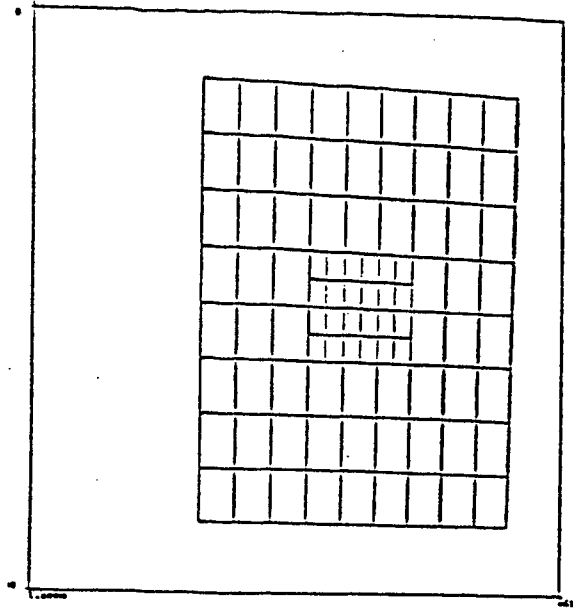


Figure 8 : Lignes extraites de la grille

B

Nous montrons ici comment trouver la meilleure rotation R qui minimise l'expression :

$$\|E - TR\|^2 \quad (9)$$

lorsque E et T sont connus. Introduisons les trois vecteurs

$$\begin{aligned} m_1 &= (0 \quad t_z \quad -t_y) \\ m_2 &= (-t_z \quad 0 \quad -t_x) \\ m_3 &= (t_y \quad -t_x \quad 0) \end{aligned} \quad (10)$$

L'expression 9 peut se réécrire :

$$\sum_{i=1}^3 \|R^t x_i - m_i\|^2 \quad (11)$$

où les vecteurs x_i sont les colonnes de E .

La représentation des rotations sous forme de quaternions nous permet de trouver la solution exacte comme solution d'un problème de vecteurs propres. Si nous représentons un quaternion q comme une paire (s, v) où s est un réel et v un vecteur de \mathbf{R}^3 , nous pouvons définir le produit de deux quaternions par :

$$q * q' = (ss' - v \cdot v', sv' + s'v + v \wedge v')$$

Le conjugué \bar{q} d'un quaternion q est défini comme :

$$\bar{q} = (s, -v)$$

et la norme d'un quaternion par :

$$|q|^2 = q * \bar{q} = \bar{q} * q = s^2 + \|v\|^2$$

axe de rotation : (0.04,0.036,0.005)

translation : (0.57,0.45,0.68) θ : angle exact de rotation (en degrés)

θ	t	av	sd	θ'	$\theta\%$	α	β
3.48	8	13.4	6.2	3.37	3.1	0.75	0.91

θ' : angle calculé de rotation (en degrés)

t : norme de la translation exacte

$\theta\%$: erreur sur l'angle de rotation (en pourcents)

av : moyenne des déplacements en pixels image

sd : écart type des déplacements en pixels image

α : erreur angulaire absolue sur l'axe

β : erreur angulaire absolue sur la direction de la translation

Table 2

Ces définitions étant posées, on peut aisément vérifier que $|q * q'| = |q' * q| = |q||q'|$. Une rotation d'axe w (vecteur unitaire) et d'angle θ est représentée par les deux quaternions q et $-q$, où :

$$q = \left(\cos \frac{\theta}{2}, \sin \frac{\theta}{2} w \right)$$

Notons que

$$|q|^2 = 1$$

On a également la propriété suivante :

$$Rx = q * x * \bar{q}$$

où nous avons utilisé l'identification *naturelle* des vecteurs de \mathbf{R}^3 et des quaternions $(0, x)$. Tout cela nous permet de réécrire l'expression 11 :

$$\sum_{i=1}^3 |\bar{q} * x_i * q - m_i|^2 = \sum_{i=1}^3 |\bar{q} * (x_i * q - q * m_i)|^2 = \sum_{i=1}^3 |x_i * q - q * m_i|^2$$

Le truc consiste alors à reconnaître dans cette expression $x_i * q - q * m_i$ une fonction linéaire de q , pouvant donc s'écrire $A_i q$ où A_i est une matrice 4×4 et où nous avons identifié les quaternions avec les vecteurs normés de \mathbf{R}^4 . Le critère s'écrit finalement :

$$q B q^t$$

avec $B = \sum_{i=1}^3 A_i A_i^t$, matrice définie positive. La solution est par conséquent le vecteur propre de B associé à la plus petite valeur propre.

C

Dans cet appendice, nous voulons calculer les trois déterminants D_1 , D_2 et D_3 , qui font intervenir les translations. Pour simplifier les calculs, réécrivons les vecteurs P_1 , P_2 et P_3 comme suit :

$$P_1 = (a^t, 0)^t \quad P_2 = (b^t, \alpha)^t \quad P_3 = (c^t, \beta)^t$$

Translation 1 : 50 0 25 (cms)
 Translation 2 : 0 100 0 (cms)
 Axe de rotation 1 : 1 1 1
 Axe de rotation 2 : 0 1 1
 Point de départ : 0 pour tous les paramètres

1	2	3	4	5	6	7	8	9	10
5	5	0.1	0.61	0.081	0.47	1.5	0.064	0.29	0.44
5	5	0.5	4.4	1.1	3.2	10	0.71	1.8	3.4
5	5	1	12	9.4	7.8	24	2.4	4.9	10
30	5	0.1	0.029	0.16	0.027	0.58	0.059	0.24	0.24
30	5	0.5	0.32	1.2	0.26	5.2	0.33	1.8	2.9
30	5	1	1.1	5.4	0.79	15	1.9	6	10
30	30	0.5	0.39	0.51	0.29	1.1	0.1	1.4	2.5
30	30	1	1.1	1.2	0.82	2.7	1.1	4	7.5
30	30	3	3.8	5.8	2.8	8.9	5.4	23	41
10	20	0.5	2.4	0.61	1.7	2.7	0.74	1.7	3.3
10	20	1	6.1	2	4.2	6.4	2.4	4.6	9.1

1 (resp. 2) : angle rotation 1 (resp. 2) en degrés.
 3 : erreur pixel eps (bruit uniforme dans un intervalle de longueur 2 eps).
 4 (resp. 5) : erreur relative sur angle 1 (resp. 2) en pourcents.
 6 (resp. 7) : erreur absolue sur angle de l'axe 1 (resp. 2) en degrés.
 8 (resp. 9) : erreur absolue sur la direction de la translation 1 (resp. 2) en degrés.
 10 : erreur relative sur rapport des normes des translations 1 et 2.

Table 3

avec

$$a = (a_1, a_2, a_3)^t \quad b = (b_1, b_2, b_3)^t \quad c = (c_1, c_2, c_3)^t$$

Avec ces notations :

$$D_1 = \begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix} = a_1(b_2c_3 - b_3c_2) - b_1(a_2c_3 - a_3c_2) + c_1(a_2b_3 - a_3b_2) \quad (12)$$

Remarquons que, par définition, α et β sont combinaison linéaire des coordonnées de b et c respectivement :

$$\alpha = \alpha_1 b_1 + \alpha_2 b_2 + \alpha_3 b_3 \quad (13)$$

$$\beta = (\alpha_1 + \beta_1)c_1 + (\alpha_2 + \beta_2)c_2 + (\alpha_3 + \beta_3)c_3 \quad (14)$$

Calculons maintenant les trois déterminants :

$$D_2 = \begin{vmatrix} a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ 0 & \alpha & \beta \end{vmatrix} \quad D_3 = \begin{vmatrix} a_3 & b_3 & c_3 \\ 0 & \alpha & \beta \\ a_1 & b_1 & c_1 \end{vmatrix} \quad D_4 = \begin{vmatrix} 0 & \alpha & \beta \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix}$$

1	2		3		4	5
0-1-2	206	0.02168	0.996340	-0.85454	5.2	4.9
		0.03186	0.996462	-0.83988	5.4	4.8
1-2-3	158	0.001352	0.997401	-0.072036	5.3	4.1
		0.000696	0.993526	-0.113600	5.2	6.5
2-3-4	108	0.002404	0.995785	-0.091683	5.3	5.3
		0.002502	0.997182	-0.074976	5.3	4.3
3-4-5	63	-0.039143	0.997057	-0.065924	5.3	4.4
		0.002230	0.995984	-0.089499	5.3	5.1
4-5-6	10	0.001226	0.996201	-0.087078	5.2	5
		0.002527	0.997028	-0.077000	5.2	4.4
5-6-7	13	0.014293	0.998154	-0.059027	5.3	3.5
		0.003504	0.995266	-0.097128	5.2	5.6

1: triplet d'images utilisé.

2: nombre de segments appariés.

3: axe de rotation (première ligne de l'image 1 à l'image 2, seconde ligne de l'image 2 à l'image 3)

4: angle de rotation.

5: angle de l'axe relativement à l'axe vertical du repère absolu.

Table 4

D_2 , D_3 et D_4 peuvent se réécrire :

$$\begin{aligned}
 D_2 &= -\alpha(a_2c_3 - a_3c_2) + \beta(a_2b_3 - a_3b_2) \\
 D_3 &= \alpha(a_3c_1 - a_1c_3) - \beta(a_3b_1 - a_1b_3) \\
 D_4 &= -\alpha(a_1c_2 - a_2c_1) + \beta(a_1b_2 - a_2b_1)
 \end{aligned} \tag{15}$$

En utilisant les équations (12), (13) et (14), on peut développer les équations (15) :

$$\begin{aligned}
 D_2 &= \alpha_1[c_1(a_2b_3 - a_3b_2) - b_1(a_2c_3 - a_3c_2)] + \\
 &\alpha_2[c_2(a_2b_3 - a_3b_2) - b_2(a_2c_3 - a_3c_2)] + \\
 &\alpha_3[c_3(a_2b_3 - a_3b_2) - b_3(a_2c_3 - a_3c_2)] + \\
 &(\beta_1c_1 + \beta_2c_2 + \beta_3c_3)(a_2b_3 - a_3b_2) = \\
 &\alpha_1D_1 + (\alpha_1a_1 + \alpha_2a_2 + \alpha_3a_3)(b_3c_2 - b_2c_3) + (\beta_1c_1 + \beta_2c_2 + \beta_3c_3)(a_2b_3 - a_3b_2)
 \end{aligned}$$

On trouve de même :

$$D_3 = \alpha_2D_1 + (\alpha_1a_1 + \alpha_2a_2 + \alpha_3a_3)(b_3c_1 - b_1c_3) + (\beta_1c_1 + \beta_2c_2 + \beta_3c_3)(a_3b_1 - a_1b_3)$$

$$D_4 = \alpha_3D_1 + (\alpha_1a_1 + \alpha_2a_2 + \alpha_3a_3)(b_2c_1 - b_1c_2) + (\beta_1c_1 + \beta_2c_2 + \beta_3c_3)(a_1b_2 - a_2b_1)$$

Par conséquent, pour prouver que ces trois équations sont équivalentes, nous devons juste montrer que :

$$\begin{aligned}
 \frac{b_3c_2 - b_2c_3}{b_3c_1 - b_1c_3} &= \frac{a_2b_3 - a_3b_2}{a_1b_3 - a_3b_1} \\
 \frac{b_3c_1 - b_1c_3}{b_2c_1 - b_1c_2} &= \frac{a_1b_3 - a_3b_1}{a_1b_2 - a_2b_1}
 \end{aligned}$$

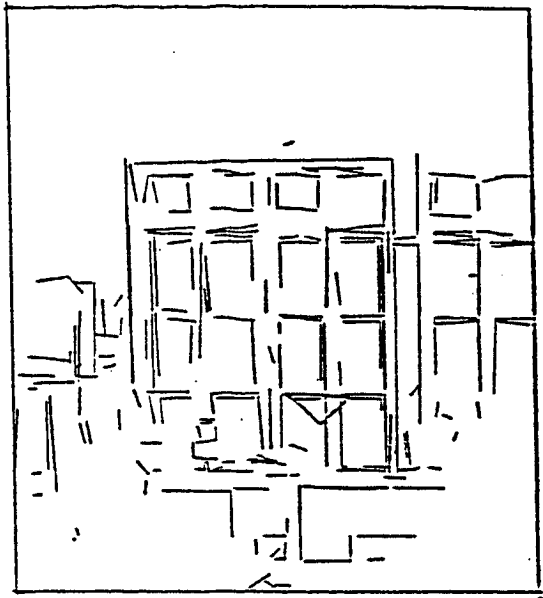


Figure 9 : Vue de face d'une scène de bureau reconstruite

Comme :

$$\begin{aligned}
 (b_3c_2 - b_2c_3)(a_1b_3 - a_3b_1) - (b_3c_1 - b_1c_3)(a_2b_3 - a_3b_2) &= \\
 a_1b_3(b_2c_3 - b_3c_2) + a_2b_3(b_3c_1 - b_1c_3) + a_3b_3(b_1c_2 - b_2c_1) &= \\
 b_3D_1 &
 \end{aligned}$$

et :

$$\begin{aligned}
 (b_3c_1 - b_1c_3)(a_1b_2 - a_2b_1) - (b_2c_1 - b_1c_2)(a_3b_1 - a_1b_3) &= \\
 a_1b_1(b_2c_3 - b_3c_2) + a_2b_1(b_1c_3 - b_3c_1) + a_3b_1(b_2c_1 - b_1c_2) &= \\
 -b_1D_1 &
 \end{aligned}$$

Nous avons donc montré que, si D_1 est nul, D_2 , D_3 et D_4 sont équivalents.

D

L'équation :

$$y_i = M_i a + u_i$$

de la section 5 est, dans la terminologie du filtre de Kalman, une équation de mesure du processus a constant par rapport à i :

$$a_i = a_{i-1}$$

D'après les équations du filtre de Kalman, une nouvelle estimée et une estimation de la matrice de covariance sont données par :

$$\begin{aligned}
 \hat{a}_i &= \hat{a}_{i-1} + K_i(y_i - M_i \hat{a}_{i-1}) \\
 K_i &= S_{i-1} M_i^t (W_i + M_i S_{i-1} M_i^t)^{-1} \\
 S_i &= (I - K_i M_i) S_{i-1}
 \end{aligned}$$

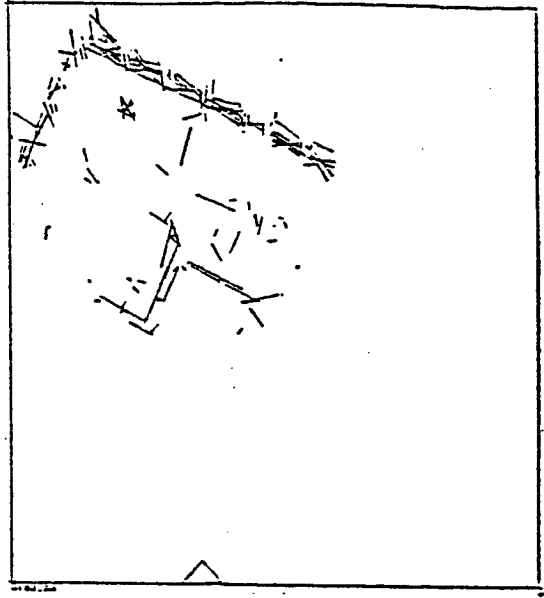


Figure 10 : Vue de dessus de la scène de bureau

ou, de manière équivalente :

$$S_i^{-1} = S_{i-1}^{-1} + M_i^t W_i^{-1} M_i$$

où $W_i = E(u_i u_i^t)$.

Lorsque toutes les mesures ont été utilisées le paramètre a est connu par son estimée *a posteriori* \hat{a}_n et la matrice de covariance correspondante S_n .

E

Nous voulons prouver que chacune des matrices $H_i (i = 1, 2, 3)$ de la section 6 fournit seulement deux équations en les inconnues a, b, p, q représentant la droite 3D à reconstruire. En écrivant que $P_i = (u_i, v_i, w_i)^t$ pour chacun des trois plans déterminés par les droites l, l' et l'' de la Figure 5, nous devons calculer :

$$D_1 = \begin{vmatrix} u_i & 1 & 0 \\ v_i & 0 & 1 \\ w_i & -a & -b \end{vmatrix} \quad D_2 = \begin{vmatrix} v_i & 0 & 1 \\ w_i & -a & -b \\ r_i & -p & -q \end{vmatrix} \quad D_3 = \begin{vmatrix} w_i & -a & -b \\ r_i & -p & -q \\ u_i & 1 & 0 \end{vmatrix} \quad D_4 = \begin{vmatrix} r_i & -p & -q \\ u_i & 1 & 0 \\ v_i & 0 & 1 \end{vmatrix}$$

On trouve aisément que :

$$\begin{aligned} D_1 &= w_i + au_i + bv_i & D_2 &= ar_i - pw_i + v_i(aq - bp) \\ D_3 &= -br_i + qw_i + u_i(aq - bp) & D_4 &= r_i + pu_i + qv_i \end{aligned}$$

Comme :

$$-pD_1 + aD_4 = D_2 \quad \text{et} \quad pD_1 - bD_4 = D_2$$

les quatre équations précédentes sont dépendantes et nous pouvons utiliser les deux plus simples, $D_1 = D_4 = 0$.

References

- [1] H.C. LONGUET-HIGGINS. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133-135, 1981.
- [2] R. TSAI and T.S. HUANG. Estimating three-dimensional motion parameters of a rigid planar patch, ii: singular value decomposition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 30, 1982.
- [3] Hans-Hellmut Nagel. Image sequences - ten (octal) years- from phenomenology towards a theoretical foundation. In *Proceedings 8th ICPR, Paris, France*, pages 1174-1185, IEEE, October 1986.
- [4] V. TORRE and T. POGGIO. On edge detection. *IEEE Transactions on PAMI*, 8, 1986.
- [5] E.C. HILDRETH. The detection of intensity changes by computer and biological vision systems. *Computer Vision Graphics and Image Processing*, 22:1-27, 1983.
- [6] J. CANNY. A computational approach to edge detection. *IEEE Transactions on PAMI*, 8 No6:679-698, 1986.
- [7] DERICHE R. Using canny's criteria to derive an optimal edge detector recursively implemented. *International Journal of Computer Vision*, 1, No2, 1987.
- [8] D. MARR and T. POGGIO. Cooperative computation of stereo disparity. *Science*, 194:283-287, 1976.
- [9] GRIMSON W.E.L. Computational experiments with a feature based stereo algorithm. *IEEE Transactions on PAMI*, 7, No 1:17-34, 1985.
- [10] H. BAKER and T.O. BINFORD. Depth from edge and intensity based stereo. In *Proceedings 7th Joint Conference on Artificial Intelligence, Vancouver, Canada*, pages 631-636, August 1981.
- [11] Y. OHTA and T. KANADE. Stereo by intra- and inter-scanline search. *IEEE Transactions on PAMI*, 7(2):139-154, 1985.
- [12] N. AYACHE and B. FAVERJON. Efficient registration of stereo images by matching graph descriptions of edge segments. *International Journal of Computer Vision*, 1(2), April 1987.
- [13] H.C. LONGUET-HIGGINS. The reconstruction of a scene from two projections - configurations that defeat the 8-point algorithm. In *Proceedings First Conference on Artificial Intelligence Applications, Denver, Colorado*, pages 395-397, 1984.
- [14] O.D. FAUGERAS and F. LUSTMAN. Let us suppose that the world is piecewise planar. In O. D. Faugeras and Georges Giralt, editors, *Robotics Research, The Third International Symposium*, pages 33-40, MIT Press, 1986.
- [15] A. MITICHE, S. SEIDA, and J.K. AGGARWAL. Line based computation of structure and motion using angular invariance. In *Proceedings Workshop on Motion: Representation and Analysis, Charleston, South Carolina, USA*, pages 175-180, IEEE, May 1986.

- [16] A. MITICHE, S. SEIDA, and J.K. AGGARWAL. Interpretation of structure and motion using straight line correspondences. In *Proceedings 8th ICPR, Paris, France*, pages 1110-1112, IEEE, October 1986.
- [17] Y. LIU and T.S. HUANG. Estimation of rigid body motion using straight line correspondences. In *Proceedings Workshop on Motion: Representation and Analysis, Charleston, South Carolina, USA*, pages 47-51, IEEE, May 1986.
- [18] Y. LIU and T.S. HUANG. Estimation of rigid body motion using straight line correspondences: further results. In *Proceedings 8th ICPR, Paris, France*, pages 306-307, IEEE, October 1986.
- [19] O.D. FAUGERAS and G. TOSCANI. The calibration problem for stereo. In *Proceedings of CVPR '86, Miami Beach, Florida*, pages 15-20, IEEE, June 1986.
- [20] R. TSAI. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of CVPR '86, Miami Beach, Florida*, pages 364-374, IEEE, June 1986.
- [21] X. ZHUANG and R.M. HARALICK. Two view motion analysis. In *Proceedings CVPR '85, San Francisco, California*, pages 686-690, June 1985.
- [22] T.S. HUANG. Determining three-dimensional motion and structure from two perspective views. In *Handbook of Pattern Recognition and Image Processing*, chapter 14, Academic Press, 1986.
- [23] O.D. FAUGERAS, N. AYACHE, and B. FAVERJON. Building visual maps by combining noisy stereo measurements. In *Proceedings IEEE Conference on Robotics and Automation, San Francisco, California, USA*, pages 1433-1438, April 1986.
- [24] N. AYACHE and F. LUSTMAN. Fast and reliable passive trinocular stereovision. In *Proceedings ICCV '87, IEEE, June 1987*.

I) INTRODUCTION

We investigate in this article the application of token matching techniques to the estimation of motion and structure from a number of views. We assume that a moving camera is looking at an otherwise static 3D environment and that the motion is unknown. The problem is then, from a sequence of such images, to recover the motion of the camera and the structure of the environment.

We do not attack the problem of recovering the instantaneous motion but rather from the sequence F_1, \dots, F_n of frames we want to recover the sequence of rigid displacement D_{12}, \dots, D_{n-1n} . The question of what happens when the time intervals $\Delta t_{i,i+1}$ between frames go to zero is not addressed here but see [Nagel 86] for a special case.

The approaches to the problem of computing motion and structure fall into broadly two categories. The optical flow approach relies on the fact that there exist a set of equations relating position and motion in space to image positions and optical flow, i.e. the apparent velocity of intensity patterns. However, the solution of these equations is not well behaved numerically.

Another approach is to extract tokens from the images in the sequence, match them from image to image, and recover the motion of the camera and the structure of the environment. We believe that this approach is more robust than the optical flow approach.

The token matching approach implies five steps :

- i) choice of tokens to be computed in the images
- ii) compute the tokens
- iii) match the tokens between images
- iv) compute the 3D motion of the camera
- v) reconstruct the structure of the environment.

The tokens we compute in the images should be closely related to objects in the scene if we want the matches to be meaningful. By closely related, we mean that tokens should signify physical events in the scene such as surface markings, shadows, depth discontinuities if we are interested in changes or uniform patches of some perspective invariant quantity such as the light reflected by a Lambertian surface if we are interested in the absence of changes. This implies that our tokens are going to fall into two broad classes : edge points and structures derived from these such as points of interest (corners, maximum curvature) or curves, and regions. In this article we concentrate on edge points.

The computation of edges and uniform regions in images is still an active research area in Computer Vision but there are now good theoretical analysis of what edge detectors do [Torre and Poggio 1986] and some optimal (from a number of standpoints) detectors have been proposed and fairly widely tested [Hildreth 1983, Canny 1986, Deriche 1987].

The problem of matching tokens between images has not been studied widely in the case of motion where the positions of the cameras are unknown. On the other hand it has been fairly well studied in the work on stereo where these positions are assumed to be known and where the epipolar constraint reduces the search space from two to one dimension [Marr and Poggio 1976, Grimson 1985, Baker and Binford 1981, Ohta and Kanade 1985, Ayache and Faucher 1987]. We do not address this problem in this article.

The subject discussed here is, given a number of token matches between a number of

images, how do we compute the motion of the camera and reconstruct the structure of the environment. We are very much concerned with the problem of uncertainty and errors and insist on having methods of estimation motion/reconstructing the environment which are both robust to noise and yield an estimation of the final uncertainty on the estimated parameters.

Specifically, we present an analysis along these lines of the eight-point algorithm proposed by Longuet-Higgins [Longuet-Higgins 1981, 1984] in the general case and in the degenerate case where the points are coplanar ; in the general case we propose a closed form mean-square solution that has some desirable robustness properties ; in the planar case we establish the relationship between the eight-point algorithm and the analysis of Tsai and Huang [Tsai and Huang 1982] and Faugeras and Lustman [Faugeras and Lustman 1987]. We then discuss the problem of matching lines between images. This implies using more than two images. We prove that at least six lines are necessary to recover the displacement, and propose a method based on Extended Kalman Filtering to estimate it. We compare our approach to that of Mitiche, Seida, and Aggarwal [Mitiche, Seida, and Aggarwal 1986a and b], and of Liu and Huang [Liu and Huang 1986a and b].

We then discuss the problem of, given the estimates of the rigid displacements between cameras, to reconstruct the 3D environment and obtain estimates of the uncertainty. We treat the case of lines.

II) THE EIGHT POINT ALGORITHM REVISITED

We first establish a number of notations. The camera model we use is the simple pinhole model described in Figure 1.

A point M in 3D space forms an image m in the retina plane $(O u v)$ by a perspective transformation of center C , the optical center of the camera. The optical axis Cz pierces the retina plane at point c of coordinates (u_0, v_0) . The distance Cc is the focal length f of the

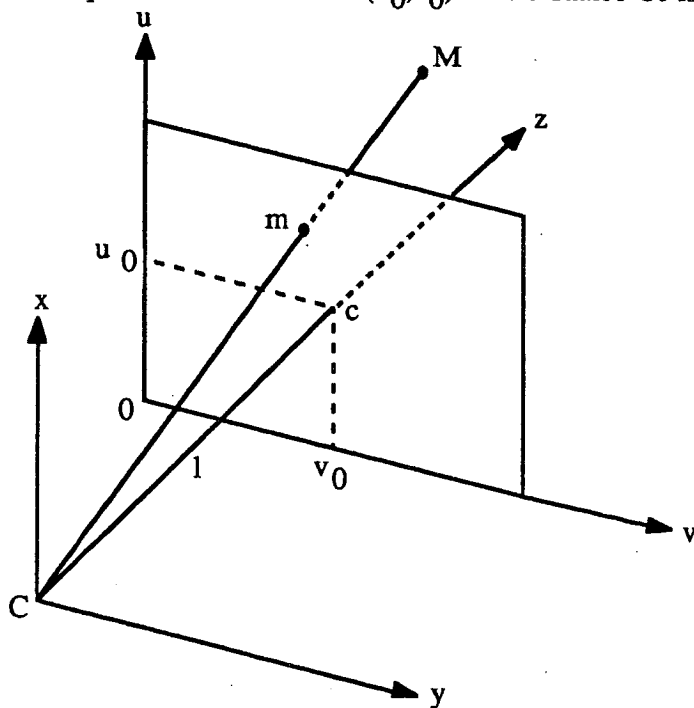


Figure 1

camera. Because of the way images are obtained electronically, there exist in general two scale factors k_u , and k_v corresponding to different scales along the u and v axis. The parameters which are accessible by optical calibration are the products $\alpha_u = f k_u$ and $\alpha_v = f k_v$ [Faugeras and Toscani 1986, Tsai 1986]. Therefore, f is taken to be one. The set of four parameters $(u_0, v_0, \alpha_u, \alpha_v)$ are the intrinsic parameters of the camera and are obtained once for all by calibration. We neglect some important effects like lens distortion but they can be eliminated anyway [Tsai 1986] and therefore do not change the rest of our analysis.

For a pixel of coordinates u, v , the vector Cm expressed in the coordinate system $Cxyz$ is equal to

$$\left[\frac{u-u_0}{\alpha_u}, \frac{v-v_0}{\alpha_v}, 1 \right]^t$$

We now consider the system of Figure 2 where the camera has moved from its first position, 1, to its second position, 2, by a rigid transformation D .

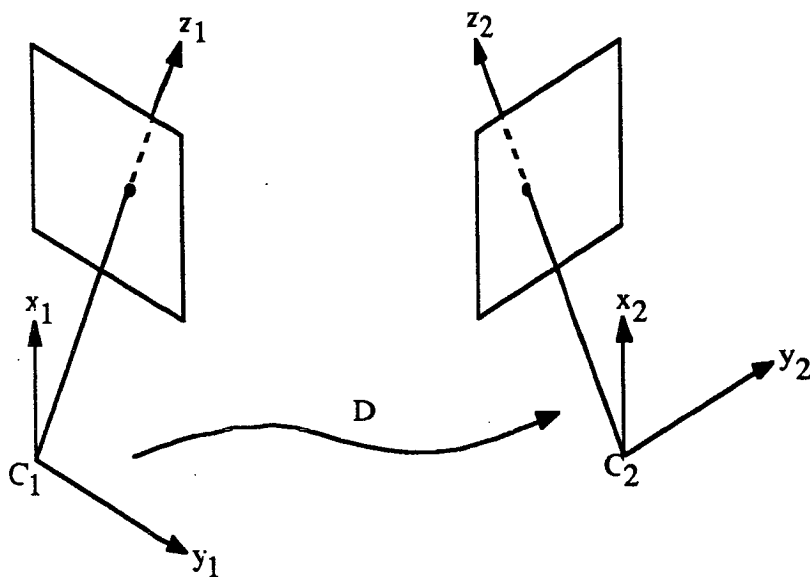


Figure 2

We decompose D as the product of a rotation and a translation. There are infinitely many ways of achieving such a decomposition depending on how we choose the rotation axis. If we assume that it goes through the origin C_1 of the coordinate system attached to the camera in its first position, then the decomposition is unique and the translation that follows the rotation R is represented by the vector $C_1 C_2 = t$.

The question we address now is that of, given a number of point matches (m_{1i}, m_{2i}) between images 1 and 2, to recover the rotation and the direction of the translation t . The problem has been solved by Longuet - Higgins [Longuet-Higgins 1981] in the exact case, assuming no noise. When noise is present, the method is quite sensitive. We propose here a method which can cope robustly with the noise problem. It is based on the exploitation of the planarity constraint used by Longuet - Higgins.

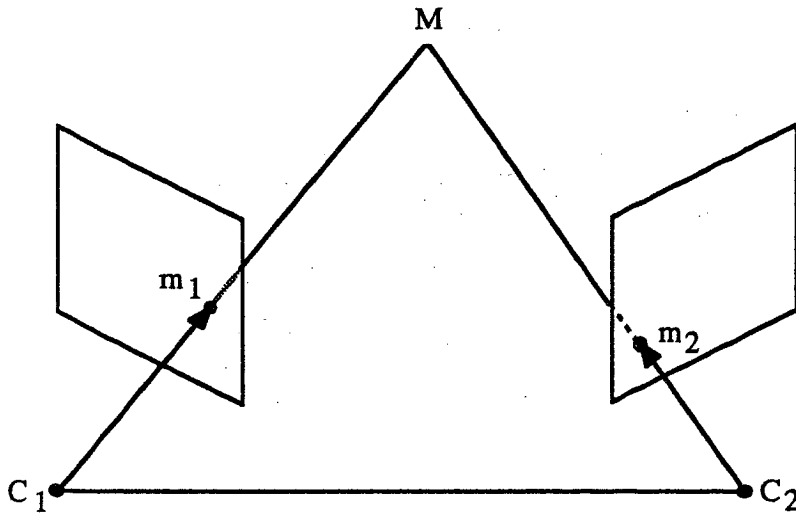


Figure 3

Let us look at Figure 3 where m_1 and m_2 are the images of the physical point M in the two cameras. The three vectors C_1m_1 , t , and C_2m_2 are coplanar and therefore their determinant is zero. What we can measure are the coordinates of C_1m_1 and C_2m_2 in the coordinate systems of Figure 2. C_2m_2 expressed in the coordinate system 1 is equal to $R C_2m_2$. Therefore, we have, in coordinate system 1 :

$$C_1m_1 \cdot (t \wedge R C_2m_2) = 0 \quad (1)$$

Introducing the antisymmetric matrix T :

$$T = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

equation (1) can be rewritten as :

$$C_1m_1^t E C_2m_2 = 0 \quad (2)$$

with : $E = TR$

Since only the direction of t is known, E is defined up to a scale factor. We normalize things by assuming that $\|t\| = 1$. This implies that $\|E\|^2 = 2$.

Notice that (2) is a linear equation in the coefficient of matrix E . Denoting its row vectors by X_1^t, X_2^t, X_3^t and by X the 9×1 vector $[X_1^t, X_2^t, X_3^t]^t$, we can write (2) as :

$$a^t X = 0 \quad (3)$$

where : $a = [x_1x_2, x_1y_2, x_1y_1x_2, y_1y_2, y_1, x_2, y_2, 1]^t$ (3')

and the x_i 's and y_i 's are the normalized coordinates of C_1m_1 and C_2m_2 in coordinate systems 1 and 2.

If we have n matches, we have n such equations, and a linear system :

$$A_n X = 0 \quad (4)$$

where A_n is a $n \times 9$ matrix. Longuet - Higgins solves this system for $n = 8$ when the rank of A_8 is 8. The null space is then of dimension 1 and X is the vector of norm $\sqrt{2}$ in the null space. Degenerate cases occur when $\text{rank}(A_8) < 8$ and are analyzed in Longuet-Higgins [Longuet-Higgins 1984] and Zhuang and Haralick [Zhuang and Haralick 1985]. We discuss later the case where the 3D points are coplanar which also yields a degenerate matrix A_8 .

An obvious improvement of the method of Longuet - Higgins, is to solve the system (4) using mean-square techniques. Noticing that $\|X\|^2 = 2\|t\|^2 = 2$, this is equivalent to solving :

$$\underset{X}{\text{Min}} \|A_n X\| \quad \text{subject to} \quad \|X\|^2 = 2$$

The solution is the eigenvector of matrix $A_n^t A_n$ of norm $\sqrt{2}$ corresponding to the smallest eigenvalue.

The next problem is, given the solution X , to recover R and t . Let us start with t . We present two approaches.

The first consists in noticing that matrix E satisfies the following equation :

$$E E^t = T R R^t T^t = -T^2 = \begin{pmatrix} t_y^2 + t_z^2 & -t_y t_x & -t_z t_x \\ -t_y t_x & t_z^2 + t_x^2 & -t_y t_z \\ -t_z t_x & -t_y t_z & t_x^2 + t_y^2 \end{pmatrix}$$

Since the matrix $E E^t$ is known after the estimation of X , we can solve the following meansquare problem :

$$\underset{t}{\text{Min}} \|E E^t + T^2\|^2 \quad \text{subject to} \quad \|t\|^2 = 1$$

The solution is the eigenvector corresponding to the smallest eigenvalue of matrix $E E^t - I$. The proof is presented in Appendix A.

Another approach [Nagel 86] consists in noticing that $t^t E = t^t T R = 0$. Therefore t is the solution of the following meansquare problem :

$$\underset{t}{\text{Min}} \|E^t t\|^2 \quad \text{subject to} \quad \|t\|^2 = 1$$

The solution is the eigenvector corresponding to the smallest eigenvalue of matrix $E E^t$. Therefore, the solution is the same as in the previous case since matrixes $E E^t$ and $E E^t - I$ have the same eigenvectors.

The ambiguity on the sign of t are resolved later by the condition that the forward coordinates of any point must both be positive.

We now turn to the problem of estimating the rotation matrix R . This is achieved by minimizing with respect to R the criterion $\|E - T R\|$ which can be solved easily using, for

example, the quaternion representation of 3D rotations [Faugeras and Hébert 1986, Faugeras and Toscani 1986]. The details are presented in Appendix B.

We have therefore obtained a closed form analytical solution for the rotation and direction of translation in the mean-square sense. In order to test the validity of the approach and its sensitivity to noise, we have experimented with both synthetic and real data and compared our results with those obtained by the Longuet-Higgins method.

III) PLANAR CASE

When the points used in the previous estimation are in a plane, the previous analysis fails because the rank of matrix A_n becomes strictly less than 8. In fact we can prove that $\text{rank}(A_n) = 6$.

In order to do it, we must remind the reader of the fact that for the planar case, the relationship between the coordinates (x_2, y_2) and (x_1, y_1) of the images of a 3D point can be written [Tsai and Huang 1982] as :

$$\begin{matrix} sx_2 & a & b & c & x_1 & x_1 \\ sy_2 & e & f & g & y_1 & = A & y_1 \\ s & 1 & m & n & 1 & 1 \end{matrix}$$

Replacing x_2 and y_2 by their values in equation (3') allows us to rewrite matrix A_n as :

$$A_n = D_n A_n'$$

where D_n is a diagonal matrix equal to $\text{diag} \left(\frac{1}{kx_1^{(i)} + ly_1^{(i)} + m} \right)$
and A_n' is an $n \times 9$ matrix.

The terms on the diagonal of D_n are non zero, therefore $\text{rank}(A_n) = \text{rank}(A_n')$. We can prove by simple inspection that the three vectors

$$\begin{aligned} Y_1 &= [0 - k e 0 -l f 0 -m g]^t \\ Y_2 &= [k 0 -a 1 0 -b m 0 -c]^t \\ Y_3 &= [-e a 0 -f b 0 -g c 0]^t \end{aligned}$$

are in the null space of A_n' , i.e. satisfy $A_n' Y_i = 0$ ($i = 1, 2, 3$). Therefore, $\text{rank}(A_n') \leq 6$.

In order to determine the rank of A_n' , let us denote by Z_i its 9 column vectors. We have to solve the system of linear equations in λ_i :

$$\sum_{i=1}^9 \lambda_i Z_i = 0 \quad (5)$$

Equations (5) are equivalent to

$$\lambda^t a = 0 \quad \text{for all } x_1, y_1$$

where $\lambda = [\lambda_1, \dots, \lambda_9]^t$. Expressing $\lambda^t a$ yields :

$$\begin{aligned} \lambda^t a = & x_1^2(\lambda_1 a + \lambda_2 e + \lambda_3 k) + y_1^2(\lambda_4 b + \lambda_5 f + \lambda_6 l) + \\ & x_1 y_1(\lambda_1 b + \lambda_2 f + \lambda_3 l + \lambda_4 a + \lambda_5 e + \lambda_6 k) + \\ & x_1(\lambda_1 c + \lambda_2 g + \lambda_3 m + \lambda_7 a + \lambda_8 e + \lambda_9 k) + \\ & y_1(\lambda_4 c + \lambda_5 g + \lambda_6 m + \lambda_7 b + \lambda_8 f + \lambda_9 l) + \lambda_7 c + \lambda_8 g + \lambda_9 m \end{aligned}$$

Writing that this expression equals 0 for all x_1, y_1 yields the following system :

$$B\lambda = 0$$

where matrix **B** is given by :

$$B = \begin{bmatrix} a e k & 0 0 0 & 0 0 0 \\ 0 0 0 & b f l & 0 0 0 \\ b f l & a e k & 0 0 0 \\ c g m & 0 0 0 & a e k \\ 0 0 0 & c g m & b f l \\ 0 0 0 & 0 0 0 & c g m \end{bmatrix}$$

Let us consider the rank of **A** first. It is well known [Tsai and Huang 1982], that matrix **A** can be written as:

$$A = dR + tn^t$$

where **n** and **d** are the normal and distance to the plane expressed in the coordinate system of figure 1. A vector **x** in the nullspace of **A** must satisfy the linear equation:

$$(dI + C)x = 0$$

where matrix **C** is equal to $[n_1 R^t \quad n_2 R^t \quad n_3 R^t]$. Therefore, **x** is an eigenvector of matrix **C** and **-d** is an eigenvalue. **C** has two eigenvalues equal to 0 and one equal to $t^t R n$. The first two eigenvalues imply that the plane goes through the origin, a possibility that we can readily eliminate under the general viewpoint assumption. The third eigenvalue also implies that our camera is at a very specific distance from the observed plane, again a possibility that can be eliminated under the general viewpoint assumption. Therefore, the rank of **A** is 3.

Considering the first three column vectors, because the rank of matrix **A** is equal to 3, they span the same subspace of R^6 as e_1, e_3, e_4 , where e_i is the standard basis vector with all coordinates equal to 0 except the *i*th which is equal to 1. For the same reasons the next three column vectors of **B** span the same subspace of R^6 as e_2, e_3, e_5 , and the last three column vectors of **B** span R^6 which proves that $\text{rank}(A'_n) = 6$.

The solution **X** of equations (4) can therefore be written as :

$$X = \alpha_1 Y_1 + \alpha_2 Y_2 + \alpha_3 Y_3$$

Using the expressions for Y_1, Y_2 and Y_3 , this implies that the corresponding matrix **E** can be written as :

$$E = c(\alpha)A$$

where $c(\alpha)$ is a 3×3 matrix representing the vector product with the vector $\alpha = [\alpha_1 \ \alpha_2 \ \alpha_3]^t$, i.e. $c(\alpha)x = \alpha \wedge x$. But we also know that :

$$t^t E = 0$$

therefore :

$$t^t c(\alpha)A = 0$$

and since $\text{rank}(A) = 3$ this implies $c(\alpha)t = 0$ and therefore α is parallel to t , i.e. $t = d\alpha$.

Writing that $E E^t = -T^2$, we obtain the equation :

$$T A A^t T = d^2 T^2 \quad (6)$$

Writing equation (6) in the coordinate system defined by the eigenvectors of matrix $A A^t$, we obtain :

$$T \text{diag}(d_1^2, d_2^2, d_3^2) T = d^2 T^2$$

where the d_i 's are the singular values of matrix A . This yields the following system of linear equations :

$$\begin{aligned} (d^2 - d_3^2)x_1^2 + (d^2 - d_3^2)x_2^2 + (d^2 - d_2^2)x_3^2 &= 0 & (7) \\ (d^2 - d_3^2)x_1^2 + (d^2 - d_2^2)x_3^2 &= 0 & (8) \\ (d^2 - d_2^2)x_1^2 + (d^2 - d_1^2)x_2^2 &= 0 & (9) \\ (d^2 - d_1^2)x_2 x_3 &= 0 & (10) \\ (d^2 - d_2^2)x_3 x_1 &= 0 & (11) \\ (d^2 - d_3^2)x_1 x_2 &= 0 & (12) \end{aligned}$$

where $x = U^t t$, and U is the orthogonal matrix constructed with the eigenvectors of matrix $A A^t$. The system of equations (7) - (9) is the same as the one discussed in Faugeras and Lustman [Faugeras and Lustman 1987]. Therefore, the solution is the same and depends upon the multiplicity of the singular values. We remind the reader that when $d_1 \neq d_2 \neq d_3$ there are two solutions, and one solution when two or three singular values are equal. This provides a solution for the vector t . We can then compute the rotation matrix by solving the equation :

$$d T R = T A \quad (13)$$

d , T , and A are known, and the solution for R can be readily obtained using the technique proposed in [Longuet - Higgins 1981] and [Huang's book]. Notice that (13) is exact, i.e., we do not have to solve a meanspace problem as in the previous Section. Knowing R , we then solve for the normal as :

$$A = d R + t n^t$$

Multiplying on the left by t^t :

$$t^t A = d t^t R + n^t$$

and :

$$n = A^t t - d R^t t$$

This method for computing the motion parameters and the plane equation shows the relationship between the method of Longuet-Higgins [Longuet - Higgins 1981] and that of Tsai and Huang [Tsai and Huang 1982].

IV) MATCHING LINES :

Using lines as tokens in the matching process has been recently proposed by Mitiche, Seida, and Aggarwal, and Liu and Huang [Mitiche, Seida, and Aggarwal 1986a and b, Liu and Huang 1986a and b]. Edge segments can be reliably extracted from images through polygonal approximation and form a robust source of tokens in a number of matching processes which have been found useful in Vision [object recognition and location, stereo]. Unlike in the case of points which was treated previously, matching two lines in two frames as being the images of the same 3D line, does not constrain the rigid displacement between these frames, as shown in Figure 4.

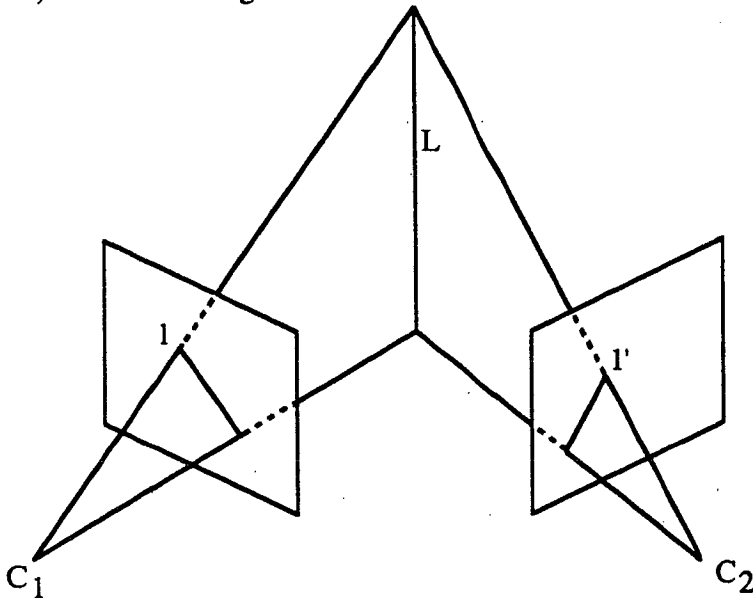


Figure 4

Indeed, matching l and l' simply defines a 3D line L but puts no constraint on C_1 and C_2 (compare with Figure 3). Therefore, as pointed out in [Mitiche, Seida, and Aggarwal 1986a and b, Liu and Huang 1986a and b], we need at least three views. Let us consider the situation described in Figure 5. We want to derive the necessary and sufficient condition that must be satisfied by the rigid displacements D_{12} and D_{23} if the lines l , l' , and l'' in retinas 1, 2 and 3 are the images of the same 3D line L (condition H).

The idea we introduce to answer this question comes directly from projective geometry and is used later to solve a related problem.

In order to proceed, we define the three planes (C_1, l) , (C_2, l') , and (C_3, l'') . In a given coordinate system, we represent a plane by a four dimensional vector $\mathbf{P} = [u, v, w, d]^t$, the

equation of the plane being $ux + vy + wz + d = 0$. In particular, the three-dimensional vector $\mathbf{n} = [u, v, w]^t$ is normal to the plane.

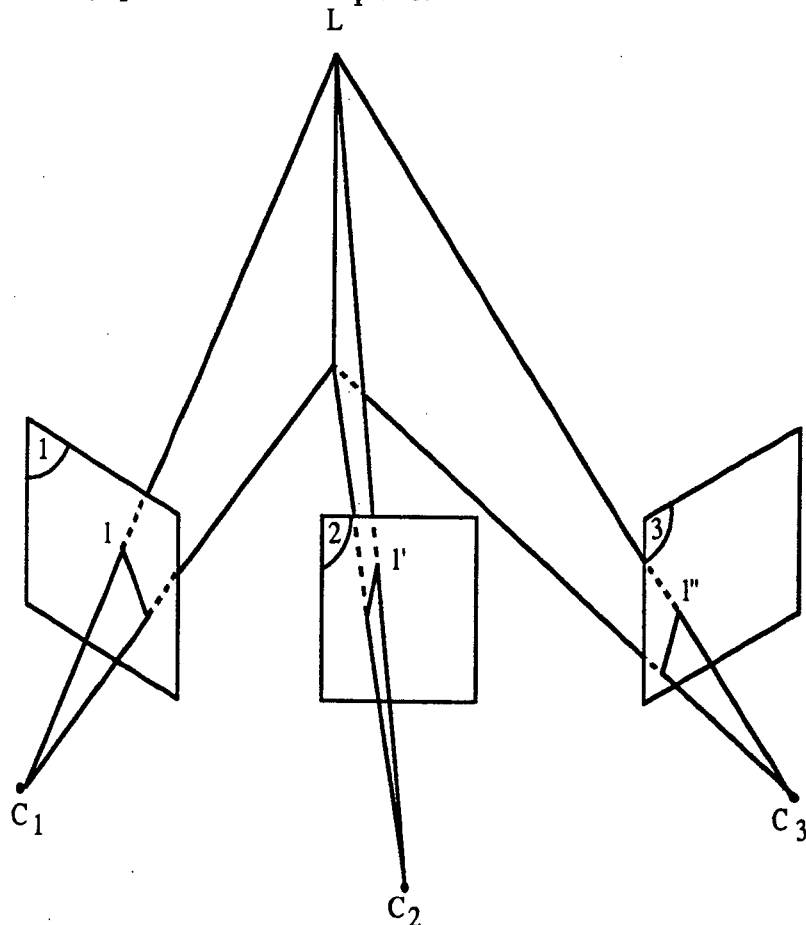


Figure 5

Given two planes represented by \mathbf{P}_1 and \mathbf{P}_2 , the set of vectors $\lambda_1 \mathbf{P}_1 + \lambda_2 \mathbf{P}_2$ represents the set of planes (called a pencil) going through the intersection of the two planes represented by \mathbf{P}_1 and \mathbf{P}_2 , when λ_1 and λ_2 take real values.

This is true even if their intersection is at ∞ , i.e. if the two planes are parallel.

Let us denote by \mathbf{P}_1 , \mathbf{P}_2 , and \mathbf{P}_3 the representations of the three planes $(C_1, 1)$, $(C_2, 1')$, and $(C_3, 1''')$ in some coordinate system. Condition H is equivalent to :

\mathbf{P}_1 , \mathbf{P}_2 , and \mathbf{P}_3 belong to a same pencil of planes

which is in turn equivalent to :

there exist λ_1 , λ_2 , and λ_3 (not all equal to zero) such that :

$$\lambda_1 \mathbf{P}_1 + \lambda_2 \mathbf{P}_2 + \lambda_3 \mathbf{P}_3 = \mathbf{0}$$

which is in turn equivalent to

the rank of the 4×3 matrix $A = [P_1, P_2, P_3]$ is less than or equal to 2

which is in turn equivalent to

the four determinants of size 3 that can be extracted from matrix A are all equal to zero

In order to compute these determinants, we choose the coordinate system attached to retina 1. In this coordinate system D_{12} is represented by the rotation matrix R_1 with respect to an axis passing through C_1 and the translation vector t_1 ; D_{23} is represented by the rotation matrix R_2 with respect to an axis passing through C_2 and the translation vector t_2 . Let us denote by n , n' , and n'' the normals to the three planes (C_1, l) , (C_2, l') , and (C_3, l'') in the coordinate systems attached to retinas 1, 2, and 3. These normals are easily computed from the line segments l , l' , and l'' , and the intrinsic parameters of the camera. It is fairly easy to show that in the coordinate system attached to retina 1, the three planes are represented by the vectors

$$\begin{aligned} P_1 &= [n^t, 0]^t, \\ P_2 &= [n^t R_1^t, -t_1^t R_1^t n'^t]^t, \\ \text{and } P_3 &= [n^t R_2^t R_1^t, -(t_1 + R_1 t_2)^t R_1^t R_2^t n''^t]^t. \end{aligned}$$

Among the four previous determinants, there is one which involves only the rotations :

$$D_1 = n \cdot (R_1 n' \wedge R_1 R_2 n'') = R_1^t n \cdot (n' \wedge R_2 n'')$$

This is the expression found in [Liu and Huang 1986a and b].

There are three determinants which involve the translation terms. It turns out that they are all proportional (see Appendix C). Let us denote by D_2 one of them.

We have proven :

$$\text{Condition (H) is equivalent to the two equations } D_1 = D_2 = 0.$$

Several comments can be made at this point. First, the number of unknowns is 6 for the rotations and 5 for the translations since absolute scale is unknown. Indeed, equation $D_2 = 0$ is homogeneous in t_1, t_2 , i.e. $D_2(\lambda t_1, \lambda t_2) = \lambda D_2(t_1, t_2)$. Therefore, the minimum number of lines we need to estimate the motion parameters is 6.

We have proven the following result :

In order to recover 3D motion from matching straight lines in three frames, a minimum of 6 lines is required.

This contradicts a result by Aggarwal and Mitiche [Mitiche, Seida, and Aggarwal 1986a and b] who made the claim that 4 lines were sufficient. Their method is based upon introducing one unknown per projected line and expressing the rigidity of the scene as the conservation of angles between lines. These angles do not involve the unknown rotations and translation vectors. Each pair of lines provides 2 equations, therefore given n lines, they obtain $2 C_n^2$ equations for $3n$ unknowns. Thus the claim that n equals 4 is sufficient. Our result shows that their equations cannot be independent.

A second comment is that we can decouple the estimation of rotations from that of the translation vectors since D_1 does not contain any translation terms. We now turn to the question of estimating the displacements from a number of triples of 2D lines.

V) ESTIMATING THE DISPLACEMENTS FROM LINE MATCHES

The technique we use is that of the Extended Kalman Filter which we proposed as a general tool for solving a number of Vision problems in [Faugeras Ayache and Faverjon 1986]. We represent a rotation by a three dimensional vector r whose direction is the axis of rotation and norm is the angle of rotation. The rotation matrix can be written as $\exp(c(r))$ [Faugeras Ayache and Faverjon 1986].

Let us discuss briefly the general case and then specialize to our problem. We deal with an observation x in \mathbb{R}^m that depends on a parameter a in \mathbb{R}^n in a nonlinear fashion that can be expressed as a relation

$$f(x,a) = 0$$

where f maps $\mathbb{R}^m \times \mathbb{R}^n$ into \mathbb{R}^p . We assume that the observation x is corrupted with noise which we model as an additive zero mean Gaussian noise :

$$x = x' + \epsilon \quad \text{with} \quad E(\epsilon) = 0 \quad \text{and} \quad E(\epsilon\epsilon^t) = \Lambda$$

The problem is, given a number of observations x_i , to find the parameter vector a that satisfies best the relations $f_i(x_i,a)=0$. Best is to be made more precise in a moment.

Supposing that we know a "good" estimate a^* of a , then we can use the idea of linearization and expand f in the vicinity of (x',a) :

$$f(x',a) = 0 \approx f(x,a^*) + \frac{\partial f}{\partial x}(x,a^*) (x' - x) + \frac{\partial f}{\partial a}(x,a^*) (a - a^*)$$

As usual, $\partial f/\partial x$ is a $p \times m$ matrix and $\partial f/\partial a$ a $p \times n$ matrix. This expression can be rewritten as (dropping the \approx sign):

$$-f(x,a^*) + \frac{\partial f}{\partial a}(x,a^*) a^* = \frac{\partial f}{\partial a}(x,a^*) a - \frac{\partial f}{\partial x}(x,a^*) \epsilon$$

which is a linear measurement equation :

$$y = M a + u$$

where:

$$y = -f(x,a^*) + \frac{\partial f}{\partial a}(x,a^*) a^*$$

$$\mathbf{M} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{a}^*)}{\partial \mathbf{a}} \text{ and } \mathbf{u} = - \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{a}^*)}{\partial \mathbf{x}} \boldsymbol{\varepsilon}$$

Notice that \mathbf{y} and \mathbf{M} are known and \mathbf{u} 's second order statistics are also known :

$$E(\mathbf{u}) = \mathbf{0}$$

$$\mathbf{W} = E(\mathbf{u}\mathbf{u}^t) = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{a}^*)}{\partial \mathbf{x}} \boldsymbol{\Lambda} \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{a}^*)}{\partial \mathbf{x}}^t$$

We "measure" \mathbf{y} because we know \mathbf{f} and \mathbf{a}^* , and observe \mathbf{x} , the corrupted version of \mathbf{x}' . For the same reasons, matrixes \mathbf{M} and $\partial \mathbf{f}(\mathbf{x}, \mathbf{a}^*)/\partial \mathbf{x}$ are known. So, if we have n measurements $\mathbf{x}_1, \dots, \mathbf{x}_n$ satisfying $\mathbf{f}_i(\mathbf{x}_i, \mathbf{a}) = \mathbf{0}$ and if we start with an initial estimate $\hat{\mathbf{a}}_0$ of \mathbf{a} and its associated covariance matrix $\mathbf{S}_0 = E((\hat{\mathbf{a}}_0 - \mathbf{a})(\hat{\mathbf{a}}_0 - \mathbf{a})^t)$ we can use the Kalman filtering approach to deduce recursively an estimate $\hat{\mathbf{a}}_n$ of \mathbf{a} and its covariance matrix $\mathbf{S}_n = E((\hat{\mathbf{a}}_n - \mathbf{a})(\hat{\mathbf{a}}_n - \mathbf{a})^t)$ after taking into account n observations. The corresponding recursive equations are the standard Kalman equations and are given in Appendix D. We can now give a precise meaning to the word best. $\hat{\mathbf{a}}_n$ is the parameter vector that minimizes the criterion:

$$(\mathbf{a} - \hat{\mathbf{a}}_0)^t \mathbf{S}_0^{-1} (\mathbf{a} - \hat{\mathbf{a}}_0) + \sum_{i=1}^n (\mathbf{y}_i - \mathbf{M}_i \mathbf{a})^t \mathbf{W}_i^{-1} (\mathbf{y}_i - \mathbf{M}_i \mathbf{a}) \quad (14)$$

This equation is important, because it shows how the Kalman filtering explicitly takes into account the noise in the measurements and weighs them accordingly. The more noise we have on the i th measurement, the "smaller" the inverse covariance matrix \mathbf{W}_i^{-1} is, and therefore the less the i th term in the above criterion contributes to the final estimate.

Let us now study in details the application of this technique to our problem. D_1 is a nonlinear equation $\mathbf{f}(\mathbf{x}, \mathbf{a})$ that relates the measurements $\mathbf{x} = [\mathbf{n}^t, \mathbf{n}''^t, \mathbf{n}'''^t]^t$ to the unknown $\mathbf{a} = [\mathbf{r}_1^t, \mathbf{r}_2^t]^t$. Starting from an initial estimate $\hat{\mathbf{a}}_0$ of the rotations and its covariance matrix \mathbf{S}_0 , and going through the measurements $\mathbf{x}_1, \dots, \mathbf{x}_n$ describing the line matches, the Extended Kalman provides the best estimate $\hat{\mathbf{a}}_n$ of the rotations and its covariance matrix \mathbf{S}_n . The word best is taken in the sense of the minimization of the previous criterion involving the partial derivations of \mathbf{f} with respect to \mathbf{x} and \mathbf{a} . These derivatives can be computed using, for example, the results described in [Faugeras Ayache and Faverjon 1986].

Once \mathbf{r}_1 and \mathbf{r}_2 have been estimated, we can compute \mathbf{t}_1 and \mathbf{t}_2 using equations D_2 very simply since it is linear in $\mathbf{t}_1, \mathbf{t}_2$.

VI) COMPUTING THE STRUCTURE OF THE SCENE

Having obtained estimates of $\mathbf{R}_1, \mathbf{R}_2, \mathbf{t}_1$, and \mathbf{t}_2 , we now want to reconstruct the 3D lines in some coordinate system. In order to do this, we must choose a representation for 3D lines. Four parameters are sufficient to describe 3D lines. A possible minimal representation is in terms of four parameters (a, b, p, q) such that the line is the intersection of the two planes :

$$\begin{aligned} x &= az + p \\ y &= bz + q \end{aligned} \quad (15)$$

Here, the line is parameterized as a function of the z coordinate. The vector $[a, b, 1]^T$ is parallel to the line and the point of coordinates $[p, q, 0]^T$ is on the line. Problems occur for lines perpendicular to the z -axis for which another coordinate (x or y) must be used as a parameter.

Referring to Figure 5, we want to represent line L as in equations (15) given the estimates of the rigid displacements D_{12} and D_{23} and the lines d , d' , and d'' . Why is it a problem? Because noise is present in our measurements, the three planes ($C_{1,l}$), ($C_{2,l'}$), and ($C_{3,l''}$) do not intersect exactly along a line and we must determine the "best" line of intersection. We can do this by using two of the previous ideas: the pencil of planes and the Kalman Filter.

Indeed, equations (15) represent two planes $Q = [1, 0, -a, -p]^T$, and $R = [0, 1, -b, -q]^T$. If we consider the pencil generated by Q and R , P_1 , P_2 , and P_3 must belong to this pencil. Therefore, the three 4×3 matrixes $H_1 = [P_1 \ Q \ R]$, $H_2 = [P_2 \ Q \ R]$, and $H_3 = [P_3 \ Q \ R]$ must be of rank less than or equal to 2. Just as in the previous Section, this is equivalent to write that for each matrix H_i , $i = 1, 2, 3$, the four determinants of size 3 are equal to 0. Again, only two of these determinants are independent (see Appendix E) and we therefore obtain six equations in our unknowns a, b, p, q , which turn out to be linear:

$$\begin{aligned} ap_i + bq_i + r_i &= 0 \\ pp_i + qq_i + s_i &= 0 \end{aligned} \quad (16)$$

where $P_i = [p_i, q_i, r_i, s_i]^T$ and $i = 1, 2, 3$.

Therefore, we can either solve (16) by least-squares or, if we are interested in getting an estimate of the uncertainty on a, b, p, q by Kalman Filtering, the covariances on the P_i 's being obtained as a result of the previous Section.

VII) IMPLEMENTATION

We have implemented the method described in Section II on both synthetic and real data and compared it with the original Longuet-Higgins method. To make the comparison fair, in both methods, the vector X_0 is estimated in the same way, i.e by meansquare as explained in Section II. It might be interesting to note that we have also implemented the technique proposed by Huang [Huang 1986]. There, it is proposed to solve $A_n X = 0$ by setting one coordinate of X equal to 1 and to solve by meansquare the system $A_n' X' + b_n = 0$, where A_n' is the $n \times 8$ matrix obtained in eliminating the i th column b_n of A_n (i is the number of the coordinate of X set to 1), but there does not seem to exist a good correlation between the condition number of matrix $A_n'^t A_n'$ and the quality of the results obtained, where what we mean by quality is defined in the sequel.

The image of the 3D test scene we have used is shown in figure 6. It is made of 10 points situated at a distance of 4 meters from our camera. The size of the object is also about four

meters. The perspective transformation matrix we use to project the 3D points has been obtained by calibration of a real camera [Faugeras and Toscani 1986]. We have simulated a number of camera displacements and noise in pixel measurements. The resolution is 512x512 pixels, and noise varies from 1 to 5 pixels. Once we have estimated the rotation R and the direction of the translation t , we can reconstruct the 3D points by meansquare and project them back in the two retinas before and after motion. We then compute the average distance between the actual and the reconstructed pixels. The sum of these two measurements constitute our measure of quality.

Table 1 shows some typical results for both techniques. Two main remarks can be made at this point. First, both techniques appear to be quite robust to quite large noise distortions contrarily to what has been reported in the literature so far. Second, our method performs at least as well as the Longuet-Higgins one, and significantly better in many cases. Figure 7 shows the projected reconstructed points in the two retinas for our method in the case of line 2 of Table 1.

We have also experimented with real data which have been used in camera calibration experiments. We have taken four stereo views of a special pattern displayed in figure 8 made of lines painted black on a sheet of white paper at distances of 3, 3.5, 4, and 4.5 meters. We have extracted automatically those lines from the four images and computed with great accuracy the pixel coordinates of their intersections, thus yielding 113 points in each image. Our method of stereo calibration described in [Faugeras and Toscani 1986] gives us the displacement (rotation R and translation t) from camera 1 to camera 2 by using at least two planes of data. We have then used our technique on the points in the other two planes and estimated R and the direction of translation t . Results are shown in Table 2. They show that this technique for determining camera displacement yields excellent results.

We have also implemented the technique described in Sections IV and V both on synthetic and real data.

Our synthetic data is the same as the one used in the previous experiment, except that instead of using the points, we have used some of the lines defined by them. Again, we have simulated a number of camera motions and measurement noise. The noise is added on the pixels defining the endpoints of the projected segments and varies between 0.1 and 1 pixel. Results are presented in table 3.

With a 3-camera system commandable in rotation we took 7 stereo triplets of the same scene, each one differing from the previous by a rotation of 5 degrees around a vertical axis. The matches were obtained by using a 3D matching procedure between two views following the trinocular stereo algorithm described in [Ayache and Lustman 1987]. The results appear in table 4. The deviation of the angle from its predicted value of 5 degrees and of the axis from a perfect verticality being quite stable, we believe that our computations are more accurate than the command of the system. Furthermore, it indicates that our method can cope with noisy data.

They somewhat contradict the results reported by Liu and Huang [Liu and Huang 1986a and b] who reported a large sensitivity of their method to noise and convergence toward false minima. The Extended Kalman Filtering approach seems to be much more immune to these problems since in all cases our initial guess was the identity matrix for the rotation. Another big advantage of this method is that it allows us to easily inject any a priori knowledge about the rotation as constraints on the initial covariance matrix S_0 .

We have also implemented our reconstruction technique and applied it to lines obtained from stereo done with three cameras [Ayache and Lustman 1987]. We show the reconstructed 3D lines for an office scene viewed from the front (Figure 9), and from above (Figure 10).

VIII) DISCUSSION

The main conclusion of this article is that it seems possible to extract motion and structure from motion fairly reliably by matching points and lines across frames. Two points need to be stressed: Mean Square Techniques and Extended Kalman Filtering can effectively cope with the noise present in image measurements, and Projective Geometry ideas such as pencils of planes are powerful tools for solving many problems in motion analysis. We predict that these techniques will prove even more useful in the foreseeable future.

Of course, many things still need to be investigated. First, the convergence properties of the Kalman Filter must be explored further, second we should see if translations can be incorporated directly when line matches are involved, and last but not least, we must investigate techniques for obtaining the correct matches.

Appendix A

We want to find the vector \mathbf{t} that minimizes:

$$\| \mathbf{E} \mathbf{E}^t + \mathbf{T}^2 \|^2 \quad \text{subject to } \|\mathbf{t}\|^2 = 1$$

We have by definition of the matrix norm:

$$\| \mathbf{E} \mathbf{E}^t + \mathbf{T}^2 \|^2 = \text{Tr}((\mathbf{E} \mathbf{E}^t + \mathbf{T}^2)^2)$$

Also, because of the constraint: $\mathbf{T}^2 = -\mathbf{I} + \mathbf{t}\mathbf{t}^t$

Therefore, denoting $\mathbf{E} \mathbf{E}^t - \mathbf{I}$ by \mathbf{F} , we have:

$$\text{Tr}((\mathbf{E} \mathbf{E}^t + \mathbf{T}^2)^2) = \text{Tr}(\mathbf{F}^2) + \text{Tr}(\mathbf{t}\mathbf{t}^t\mathbf{t}\mathbf{t}^t) + 2\text{Tr}(\mathbf{F}\mathbf{t}\mathbf{t}^t) = \text{Tr}(\mathbf{F}^2) + \|\mathbf{t}\|^4 + 2\mathbf{t}^t\mathbf{F}\mathbf{t}$$

Since $\text{Tr}(\mathbf{F}^2)$ does not depend on \mathbf{t} , and that $\|\mathbf{t}\|^2=1$, this proves the result that \mathbf{t} is the eigenvector of unit norm of matrix \mathbf{F} corresponding to its smallest eigenvalue.

Appendix B

We show here how to find the best rotation R that minimizes the matrix norm:

$$\|E - TR\|^2 \quad (B1)$$

when E and T are known. Introducing the three vectors $m_1 = [0, t_y, -t_x]^t$, $m_2 = [-t_z, 0, t_x]^t$, and $m_3 = [t_y, -t_x, 0]^t$, equation B1 can be rewritten as:

$$\sum_{i=1}^3 \|R^t x_i - m_i\|^2 \quad (B2)$$

where the vectors x_i are the column vectors of matrix E .

The representation of rotations by quaternions [Hamilton, Pervin and Webb, Hebert], allows to find the exact solution as the solution of an eigenvector problem. If we represent a quaternion q as a pair (s, v) , where s is a real and v a vector of R^3 , then we define the product of two quaternions q and q' as:

$$q * q' = (ss' - v \cdot v', sv' + s'v + v \wedge v')$$

The conjugate q of a quaternion q is defined as: $q = (s, -v)$

and the norm of a quaternion q as: $|q|^2 = q * q = q * q = s^2 + \|v\|^2$

From these definitions, it can be easily verified that $q * q' = |q| |q'|$.

A rotation of axis w (a unit vector) and of angle θ , is represented by the two quaternions q and $-q$, where:

$$q = (\cos(\theta/2), \sin(\theta/2)w)$$

Notice that $|q|^2 = 1$.

The following property also holds: $Rx = q * x * q$

where we have used the "natural" identification between vectors of R^3 and quaternions $(0, x)$. Using these definitions, equation B2 can be rewritten as:

$$\sum |q * x_i * q - m_i|^2 = \sum |q * (x_i * q - q * m_i)|^2 = \sum |x_i * q - q * m_i|^2$$

The trick is then to recognize that the expression $x_i * q - q * m_i$ is a linear function of q which can be written $A_i q$ where A_i is a 4×4 matrix and we have identified quaternions with vectors of R^4 of unit norm. The criterion eventually can be written as: $q B q^t$

with $B = \sum A_i A_i^t$, a symmetric positive matrix. The solution is therefore the eigenvector of matrix B of unit norm associated to the smallest eigenvalue.

Appendix C

In this appendix we want to compute the three determinants D_2 , D_3 , and D_4 involving the translation terms. In order to simplify the computation, let us rewrite the vectors P_1 , P_2 , and P_3 as follows:

$$P_1 = [a^t, 0] \quad P_2 = [b^t, \alpha]^t \quad P_3 = [c^t, \beta]^t$$

with:

$$a = [a_1, a_2, a_3]^t \quad b = [b_1, b_2, b_3]^t \quad c = [c_1, c_2, c_3]^t$$

With these notations:

$$D_1 = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = a_1(b_2c_3 - b_3c_2) - b_1(a_2c_3 - a_3c_2) + c_1(a_2b_3 - a_3b_2) \quad (C1)$$

Notice that by definition, α and β are linear combinations of the coordinates of b and c , respectively:

$$\alpha = \alpha_1 b_1 + \alpha_2 b_2 + \alpha_3 b_3 \quad (C2)$$

$$\beta = (\alpha_1 + \beta_1)c_1 + (\alpha_2 + \beta_2)c_2 + (\alpha_3 + \beta_3)c_3 \quad (C3)$$

Let us now compute the three determinants:

$$D_2 = \begin{vmatrix} a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ 0 & \alpha & \beta \end{vmatrix} \quad D_3 = \begin{vmatrix} a_3 & b_3 & c_3 \\ 0 & \alpha & \beta \\ a_1 & b_1 & c_1 \end{vmatrix} \quad D_4 = \begin{vmatrix} 0 & \alpha & \beta \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix}$$

D_2 , D_3 , and D_4 can be rewritten as:

$$D_2 = -\alpha(a_2c_3 - a_3c_2) + \beta(a_2b_3 - a_3b_2) \quad (C4)$$

$$D_3 = \alpha(a_3c_1 - a_1c_3) - \beta(a_3b_1 - a_1b_3) \quad (C5)$$

$$D_4 = -\alpha(a_1c_2 - a_2c_1) + \beta(a_1b_2 - a_2b_1) \quad (C6)$$

Using Equations A1-A3, we can expand A4-A6:

$$\begin{aligned} D_2 &= \alpha_1[c_1(a_2b_3 - a_3b_2) - b_1(a_2c_3 - a_3c_2)] + \\ &\quad \alpha_2[c_2(a_2b_3 - a_3b_2) - b_2(a_2c_3 - a_3c_2)] + \\ &\quad \alpha_3[c_3(a_2b_3 - a_3b_2) - b_3(a_2c_3 - a_3c_2)] + \\ &\quad (\beta_1c_1 + \beta_2c_2 + \beta_3c_3)(a_2b_3 - a_3b_2) = \\ &\alpha_1 D_1 + (\alpha_1 a_1 + \alpha_2 a_2 + \alpha_3 a_3)(b_3c_2 - b_2c_3) + \\ &(\beta_1c_1 + \beta_2c_2 + \beta_3c_3)(a_2b_3 - a_3b_2) \end{aligned}$$

In the same way we find:

$$D_3 = \alpha_2 D_1 + (\alpha_1 a_1 + \alpha_2 a_2 + \alpha_3 a_3)(b_3c_1 - b_1c_3) + (\beta_1c_1 + \beta_2c_2 + \beta_3c_3)(a_3b_1 - a_1b_3)$$

$$D_4 = \alpha_3 D_1 + (\alpha_1 a_1 + \alpha_2 a_2 + \alpha_3 a_3)(b_2c_1 - b_1c_2) + (\beta_1c_1 + \beta_2c_2 + \beta_3c_3)(a_1b_2 - a_2b_1)$$

Therefore, if $D_1=0$, in order to prove that these three equations are equivalent, we just have to show that:

$$\frac{b_3c_2 - b_2c_3}{b_3c_1 - b_1c_3} = - \frac{a_2b_3 - a_3b_2}{a_3b_1 - a_1b_3} \quad (C7)$$

$$\frac{b_3c_1 - b_1c_3}{b_2c_1 - b_1c_2} = - \frac{a_3b_1 - a_1b_3}{a_1b_2 - a_2b_1} \quad (C8)$$

Indeed:

$$\begin{aligned} (b_3c_2 - b_2c_3)(a_3b_1 - a_1b_3) + (b_3c_1 - b_1c_3)(a_2b_3 - a_3b_2) = \\ a_1b_3(b_2c_3 - b_3c_2) - a_2b_3(b_1c_3 - b_3c_1) + a_3b_3(b_1c_2 - b_2c_1) = \\ b_3D_1 \end{aligned}$$

and:

$$\begin{aligned} (b_3c_1 - b_1c_3)(a_1b_2 - a_2b_1) + (b_2c_1 - b_1c_2)(a_3b_1 - a_1b_3) = \\ -a_1b_1(b_2c_3 - b_3c_2) + a_2b_1(b_1c_3 - b_3c_1) - a_3b_1(b_1c_2 - b_2c_1) = \\ -b_1D_1 \end{aligned}$$

Therefore, we have proven that, if D_1 is equal to 0, then D_2 , D_3 , and D_4 are equivalent.

Appendix D

Equation : $y_i = M_i a + u_i$ of Section V is, in the terminology of the Kalman Filter (Jazwinsky 1970) a measurement equation on the process a constant with respect to i : $a_i = a_{i-1}$

Using the equations of the Kalman Filter, the new estimate and estimation covariance matrix are given by:

$$\begin{aligned}\hat{a}_i &= \hat{a}_{i-1} + K_i(y_i - M_i \hat{a}_{i-1}) \\ K_i &= S_{i-1} M_i^t (W_i + M_i S_{i-1} M_i^t)^{-1} \\ S_i &= (I - K_i M_i) S_{i-1}\end{aligned}$$

or equivalently :

$$S_i^{-1} = S_{i-1}^{-1} + M_i^t W_i^{-1} M_i$$

where : $W_i = E(u_i u_i^t)$

When all measurements have been processed, parameter a is known by its a-posteriori estimate \hat{a}_n and the corresponding covariance matrix S_n .

Appendix E

We want to prove that each matrix H_i , $i=1, 2, 3$ of Section VI provides only two equations in the unknowns a, b, p, q representing the 3D line to be reconstructed. Writing $P_i=[u_i, v_i, w_i, r_i]^t$ for each of the three planes determined by the lines l, l' , and l'' of Figure 5, we have to compute:

$$D_1 = \begin{matrix} u_i & 1 & 0 \\ v_i & 0 & 1 \\ w_i & -a & -b \end{matrix} \quad D_2 = \begin{matrix} v_i & 0 & 1 \\ w_i & -a & -b \\ r_i & -p & -q \end{matrix}$$

$$D_3 = \begin{matrix} w_i & -a & -b \\ r_i & -p & -q \\ u_i & 1 & 0 \end{matrix} \quad D_4 = \begin{matrix} r_i & -p & -q \\ u_i & 1 & 0 \\ v_i & 0 & 1 \end{matrix}$$

It is easily found that:

$$D_1 = w_i + au_i + bv_i \quad (E1)$$

$$D_2 = ar_i - pw_i + v_i(aq - bp) \quad (E2)$$

$$D_3 = -br_i + w_iq + u_i(aq - pb) \quad (E3)$$

$$D_4 = r_i + pu_i + qv_i \quad (E4)$$

Since: $-pD_1 + aD_4 = D_2$ and $qD_1 - bD_4 = D_3$
the four equations D_1 - D_4 are not independent, and we can use the two simplest ones, $D_1=D_4=0$.

References

- Ayache, Nicholas, and Lustman, Francis 1987. "Fast and Reliable Passive Trinocular Stereovision," ICCV '87.
- Ayache, Nicholas, and Faverjon, Bernard 1987. "Efficient Registration of Stereo Images by Matching Graph Descriptions of Edge Segments," to appear in the *International Journal of Computer Vision*.
- Baker, Harleen, and Binford, Thomas O. 1981 (August, Vancouver, Canada). "Depth from Edge and Intensity Based Stereo", *Proceedings 7th Joint Conference on Artificial Intelligence*, 631-636.
- Canny, John 1986. "A computational approach to edge detection," *IEEE Transactions on PAMI*, VOL. 8, NO. 6, 679-698.
- Deriche, Rachid 1987. "Using Canny's criteria to derive an optimal edge detector recursively implemented," to appear in *The International Journal of Computer Vision*, VOL. 1, NO. 2.
- Faugeras, Olivier D. and Toscani, Giorgio 1986 (June 22-26, Miami Beach, Florida, USA). "The Calibration Problem for Stereo," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition CVPR-86*, 15-20.
- Faugeras, Olivier D. and Hebert, Martial 1986. The Representation, Recognition, and Locating of 3D Shapes from Range Data. *The International Journal of Robotics Research*, Vol 5, No 3, 27-52.
- Faugeras, Olivier D., Ayache, Nicholas, and Faverjon, Bernard 1986 (April 7-10, San Francisco, California, USA). Building Visual Maps by Combining Noisy Stereo Measurements. *Proceedings IEEE Conference on Robotics and Automation*, 1433-1438.
- Faugeras, Olivier D., and Lustman, Francis 1987. "Let us suppose that the world is piecewise planar," *Robotics Research, The Third International Symposium*, edited by O. D. Faugeras and Georges Giralt, MIT Press.
- Grimson, W. Eric L. 1985. "Computational Experiments with a Feature Based Stereo Algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, VOL. PAMI-7, NO. 1, 17-34.
- Hildreth, Ellen C. 1983. "The detection of intensity changes by computer and biological vision systems," *Computer Vision Graphics and Image Processing* 22, 1-27.
- Huang, Thomas S. 1986. "Determining Three-Dimensional Motion and Structure from Two Perspective Views" in *Handbook of Pattern Recognition and Image Processing*, Chapter 14, Academic Press.
- Liu, Yuncai, and Huang, Thomas S. 1986 (May, Charleston, South Carolina, USA). "Estimation of rigid body motion using straight line correspondences," *Proceedings Workshop on Motion: Representation and Analysis*, IEEE Computer Society, 47-51.
- Liu, Yuncai, and Huang, Thomas S. 1986 (October 27-31, Paris, France). "Estimation of rigid body motion using straight line correspondences, further results," *Proceedings ICPR 1986*, 306-307.
- Longuet-Higgins, H.C. 1981. "A Computer Algorithm for Reconstructing a Scene from Two Projections," *Nature* 293, 133-135.
- Longuet-Higgins, H.C. 1984 (December 5-7, Denver, Colorado, USA). "The Reconstruction of a scene from Two Projections - Configurations that defeat the 8-Point Algorithm," *Proceedings First Conference on Artificial Intelligence Applications*, 395-397.
- Marr, David, and Poggio, Tommy 1976. "Cooperative Computation of Stereo Disparity," *Science* 194, 283-287.
- Mitiche, Amar, Seida, Steven, and Aggarwal, J.K. 1986 (May, Charleston, South Carolina, USA). "Line based computation of structure and motion using angular invariance," *Proceedings Workshop on Motion: Representation and Analysis*, IEEE Computer Society, 175-180.
- Mitiche, Amar, Seida, Steven, and Aggarwal, J.K. 1986 (October 27-31, Paris, France). "Interpretation of structure and motion using straight line correspondences," *Proceedings ICPR 1986*, 1110-1112.
- Nagel, Hans-Hellmut 1986 (October 27-31, Paris, France). "Image Sequences - Ten (octal) Years- From Phenomenology towards a Theoretical Foundation," *Proceedings ICPR 1986*, 1174-1185.
- Ohta, Y., and Kanade, Takeo 1985. "Stereo by Intra- and Inter-Scanline Search," *IEEE Transactions on PAMI*, VOL-7, NO.2, 139-154.
- Torre, Vincent and Poggio, Tommy 1986. "On edge detection," *IEEE Transactions on PAMI*, VOL-8.
- Tsai, Roger Y. and Huang, Thomas S. 1982. "Estimating three-dimensional motion parameters of a rigid planar patch, II: singular value decomposition," *IEEE Transactions on ASSP*, VOL. 30, NO. 4.
- Tsai, Roger Y. 1986 (June 22-26, Miami Beach, Florida, USA). "An Efficient and Accurate Camera Calibration

Technique for 3D Machine Vision," Proc. CVPR-86, 364-374.
Zhuang, X., and Haralick, Robert M. 1985 (June 19-23, San Francisco, California, USA). "Two View Motion Analysis," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition CVPR-85*, 686-690.

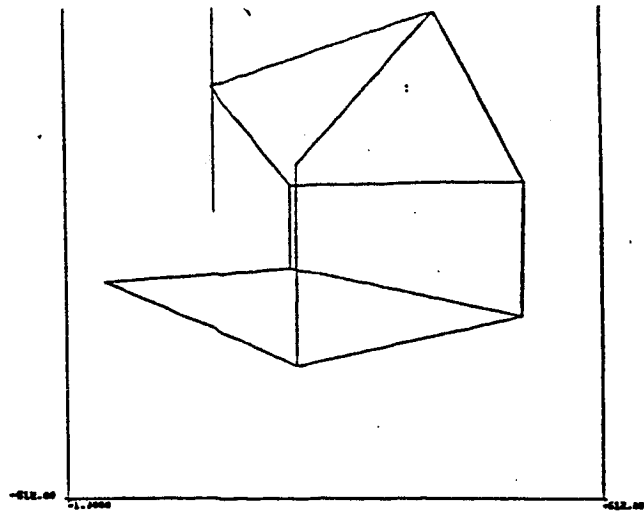


Figure 6 : Image of the original synthetic "house" data.

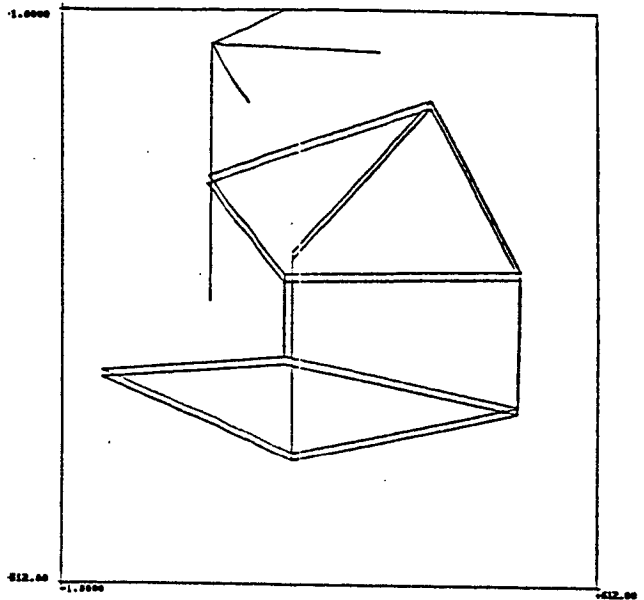


Figure 7 : Images of the original and reconstructed synthetic house data. Displacement has been estimated by the algorithm described in Section II. This corresponds to the experiment of row 2 in table 1.

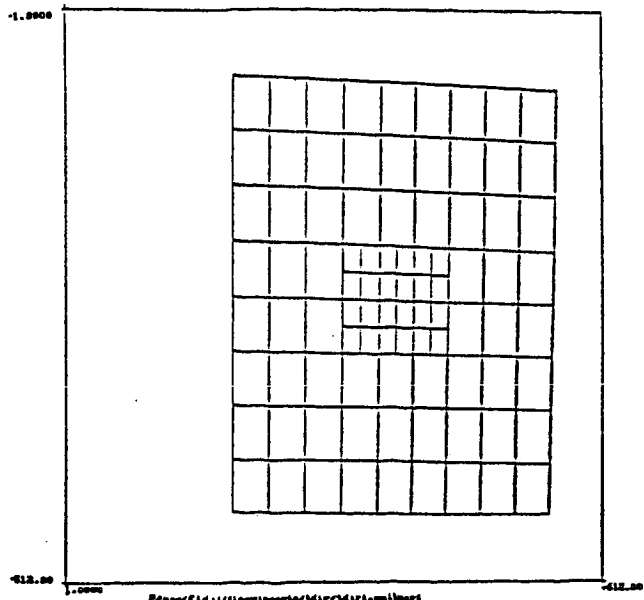


Figure 8 : Lines extracted from the grid pattern and used in the experiment described in table 2.

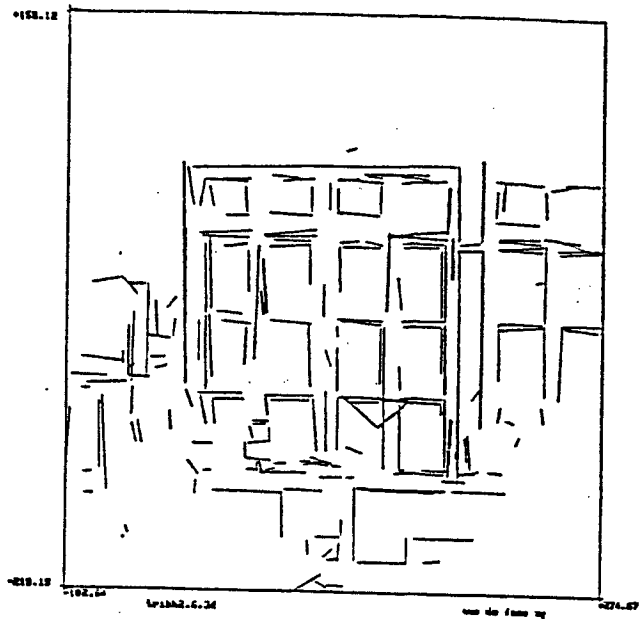


Figure 9 : Front view of the reconstructed office scene.
Lines have been reconstructed using the
technique described in Section VI.

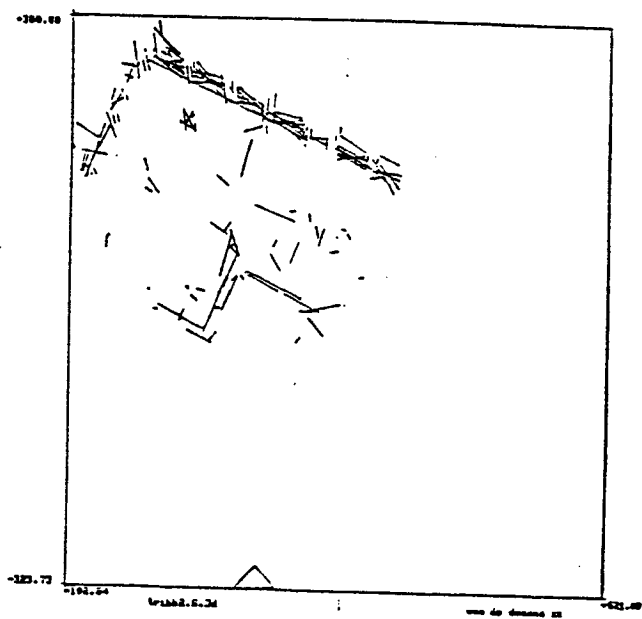


Figure 10 : View from above of the same scene.

TABLE 1

θ	t	∂p	av	sd	θ'	$\theta\%$	α	β	
5	10	1	57.8	7.2	5.89	17.8	24.59	26.24	(1)
					5.36	7.2	11.48	27.58	(2)
5	50	1	91.8	14.3	15.18	203.6	42.21	4.53	(1)
					5.72	14.5	6.78	1.78	(2)
5	100	1	139.6	31.1	11.03	120.7	37.31	3.09	(1)
					5.74	14.8	6.30	0.82	(2)
5	10	5	57.7	8.1	6.06	21.2	47.34	72.43	(1)
					5.04	0.8		72.61	(2)
5	50	5	91.5	14.7	20.01	300.2	115.43	24.59	(1)
					8.37	67.3	37.77	11.48	(2)
5	100	5	139.4	31.1	55.9	1018	121	23.2	(1)
					9.1	82	22.3	5.06	(2)
15	50	5	189.1	20.4	34.6	130.7	35.6	6.68	(1)
					19.5	30.3	8.5	7.84	(2)
15	100	5	232.3	30.3	25.15	67.6	24.98	4.19	(1)
					19.73	31.6	5.37	1.93	(2)

axis of rotation: (1, 1, 1) direction of translation: (1, 0, 0)

θ = angle of exact rotation (in deg) θ' = angle of computed rotation

t = magnitude of exact translation (in cm) $\theta\%$ = rotation angle error (in percent)

∂p = measurement noise (in pixels) α = angle between exact and computed axes of rotation

av = average of image pixel displacements β = angle between exact and computed directions of translation

sd = standard deviation of image pixel displacements (1) = improved Longuet-Higgins

(2) = new technique

TABLE 2

θ	t	av	sd	θ'	$\theta\%$	α	β
3.48	8	13.4	6.2	3.37	3.1	0.75	0.91

axis of rotation: (0.04, 0.036, 0.005) translation: (0.57, 0.45, 0.68)

θ = angle of reference rotation (in deg) θ' = angle of computed rotation

t = magnitude of reference translation (in cm) $\theta\%$ = rotation angle error (in percent)

av = average of image pixel displacements α = angle between reference and computed axes of rotation

sd = standard deviation of image pixel displacements β = angle between reference and computed directions of translation

TABLE 3

Translation 1 : 50 0 25 (cms)
 Translation 2 : 0 100 0 (cms)
 Axis of rotation 1 : 1 1 1
 Axis of rotation 2 : 0 1 1
 Starting point : 0 for all the parameters

1	2	3	4	5	6	7	8	9	10
5	5	0.1	0.61	0.081	0.47	1.5	0.064	0.29	0.44
5	5	0.5	4.4	1.1	3.2	10	0.71	1.8	3.4
5	5	1	12	9.4	7.8	24	2.4	4.9	10
30	5	0.1	0.029	0.16	0.027	0.58	0.059	0.24	0.24
30	5	0.5	0.32	1.2	0.26	5.2	0.33	1.8	2.9
30	5	1	1.1	5.4	0.79	15	1.9	6	10
30	30	0.5	0.39	0.51	0.29	1.1	0.1	1.4	2.5
30	30	1	1.1	1.2	0.82	2.7	1.1	4	7.5
30	30	3	3.8	5.8	2.8	8.9	5.4	23	41
10	20	0.5	2.4	0.61	1.7	2.7	0.74	1.7	3.3
10	20	1	6.1	2	4.2	6.4	2.4	4.6	9.1

1 (resp. 2): angle rotation 1 (resp. 2) in degrees
 3: pixel error eps (uniform noise in an interval of length 2 eps)
 4 (resp. 5): relative error on angle 1 (resp. 2) in percents
 6 (resp. 7): absolute error on angle of axis 1 (resp. 2) in degrees
 8 (resp. 9): absolute error on direction of translation 1 (resp. 2) in degrees.
 10: relative error on ratio of norms of translations 1 and 2

TABLE 4

1	2	3	4	5		
0-1-2	206	-0.02168	0.996340	-0.85454	5.2	4.9
		0.03186	0.996462	-0.83988	5.4	4.8
1-2-3	158	0.001352	0.997401	-0.072036	5.3	4.1
		0.000696	0.993526	-0.113600	5.2	6.5
2-3-4	108	0.002404	0.995785	-0.091683	5.3	5.3
		0.002502	0.997182	-0.074976	5.3	4.3
3-4-5	63	-0.039143	0.997057	-0.065924	5.3	4.4
		0.002230	0.995984	-0.089499	5.3	5.1
4-5-6	10	0.001226	0.996201	-0.087078	5.2	5
		0.002527	0.997028	-0.077000	5.2	4.4
5-6-7	13	0.014293	0.998154	-0.059027	5.3	3.5
		0.003504	0.995266	-0.097128	5.2	5.6

1: triplet of images used
 2: number of matched segments
 3: axis of rotation (first line from image 1 to image 2, second line from image 2 to image 3)
 4: angle of rotation
 5: angle of axis relative to the vertical axis of the absolute coordinate system

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique

1

2

3

4

5

6

7

8

9