

# Une méthode d'initialisation automatique pour le recalage de données SIG et vidéo

T. Colleu<sup>1</sup>

G.Sourimant<sup>1</sup>

L.Morin<sup>1</sup>

<sup>1</sup> Laboratoire IRISA/INRIA/Université de Rennes 1

Campus Universitaire de Beaulieu  
Avenue du Général Leclerc, 35042 RENNES Cedex - France

{thomas.colleu, gael.sourimant, luce.morin}@irisa.fr

## Résumé

*Cet article se situe dans un contexte de génération de modèles 3D d'environnements urbains. La méthodologie présentée doit permettre le recalage initial d'un modèle SIG avec une vidéo, de façon automatique. Elle se décompose en deux parties. Tout d'abord, un recalage approximatif est effectué grâce aux données GPS disponibles et en utilisant la théorie de la géométrie épipolaire. Ensuite, la pose (position et orientation de la caméra dans le repère du SIG) ainsi que des correspondances 3D/2D sont estimées simultanément à l'aide d'une méthode basée RANSAC. Les tests sur deux séquences réelles ont validé cette approche et montré des résultats de recalage satisfaisants.*

## Mots clefs

Recalage 3D/2D, modélisation urbaine, géométrie épipolaire, GPS, Ransac.

## 1 introduction

La génération de modèles 3D d'environnements urbains a de nombreuses applications en réalité virtuelle et augmentée, telles que les visites virtuelles (télé-tourisme), le géo-positionnement, l'urbanisme, ou la cartographie. Comme le montre le récent succès de Google Earth, l'ajout de données photométriques sur une carte 2D apporte un net bénéfice pour l'utilisateur. De la même façon, lorsqu'on dispose d'un modèle 3D de bâtiments existants, des textures synthétiques sont insuffisantes pour un rendu photoréaliste, en particulier lors d'une navigation au niveau du sol.

Il est donc intéressant de combiner des informations géométriques synthétiques et des textures réelles (issues de photos ou de vidéo) pour construire un modèle 3D d'environnement urbain. Notre étude se place dans ce cadre : on dispose d'un modèle synthétique géo-référencé de bâtiments (SIG pour Système d'Information Géographique) qui contient pour chaque bâtiment son empreinte au sol (un polygone) et son élévation. On souhaite enrichir ce modèle SIG par des données vidéos qui fourni-

ront la texture des bâtiments et permettront de raffiner sa géométrie. Une acquisition vidéo est réalisée et synchronisée avec une acquisition GPS.

Les données GPS fournissent une position approximative de la caméra dans un repère géo-référencé. Pour fusionner les informations de la vidéo et les informations du SIG, les données vidéo et SIG doivent être recalées : pour chaque image de la séquence, on doit déterminer la position et l'orientation de la caméra dans le repère géo-référencé, de telle sorte que la projection perspective du SIG dans le plan de la caméra soit alignée avec les contours des bâtiments dans l'image.

L'initialisation du recalage consiste à estimer simultanément la pose de la caméra pour la première image et un ensemble de primitives en correspondances 3D/2D. C'est un problème délicat pour lequel on trouve de nombreuses contributions dans la littérature. Une solution consiste à éliminer une des deux inconnues (correspondance ou pose) grâce à une intervention manuelle ou à du matériel de mesure. Ainsi, dans [2] et [6] l'utilisateur indique lui-même les correspondances. Dans [10] et [9], c'est la pose qui est directement mesurée à l'aide d'un matériel de navigation (GPS + centrale inertielle). D'autres solutions sont proposées à partir de modèles plus riches comme un modèle texturé [11] ou un modèle provenant d'un scanner 3D [7]. Le modèle 3D dont nous disposons ne contient que les contours des bâtiments. Dans ce cas, deux méthodes existent, ayant chacune leur inconvénient et à condition qu'une pose approximative soit disponible. La première utilise l'algorithme RANSAC [3], elle est efficace si l'ensemble de primitives est petit et possède peu d'outliers. La deuxième méthode est basée sur la minimisation d'une fonction d'énergie [1], cette méthode peut ne pas converger du fait de la non-linéarité de la fonction de coût.

Dans cet article, nous effectuons le recalage initial en deux étapes :

- une estimation approximative de la pose permettant d'obtenir pour le modèle SIG projeté les mêmes primitives que celles présentes dans l'image, et suffisamment proches de celles-ci. Cette première estimation utilise

uniquement les données GPS et la vidéo et ne fait pas intervenir le modèle SIG.

- une estimation simultanée de la pose et des correspondances en utilisant la méthode basée RANSAC dans laquelle nous nous attachons à réduire le nombre de correspondances possibles. Cette estimation utilise le modèle SIG et la première image de la séquence vidéo.

Ces deux étapes sont présentées dans les sections 2 et 3 respectivement. La section 4 présente les résultats de recalage obtenus sur des données réelles.

## 2 Estimation approximative de la pose de la caméra

### 2.1 Principe

Soit  $P_i = K[R_i|t_i]$ , la matrice de projection de la caméra à l'instant  $i$ , exprimée dans le repère UTM, avec :

- $K$  une matrice  $3 \times 3$  contenant les paramètres intrinsèques de la caméra ; elle est obtenue par calibrage préalable ;
- $t_i$  la translation du repère UTM au repère caméra ;
- $R_i$  la rotation du repère UTM au repère caméra.

On cherche à estimer la pose initiale de la caméra, c'est-à-dire  $R_1$  et  $t_1$ . Les données GPS fournissent une position approximative  $\tilde{t}_1$ , mais elles ne fournissent aucune information d'orientation. On cherche donc une estimation  $\tilde{R}_1$  de l'orientation initiale de la caméra.

La solution proposée consiste à faire coïncider les informations de translation contenues dans la séquence vidéo avec celles contenues dans les données GPS. Tout d'abord, une image  $i$  (appelée "image clé") est sélectionnée dans la séquence vidéo, puis la translation relative, appelée  $t_{relative}$ , entre la caméra  $C_i$  et la caméra  $C_1$  est estimée grâce aux images  $i$  et 1. Ensuite, la translation  $t_{GPS}$  est extraite des mesures GPS. Enfin, la rotation  $\tilde{R}_1$  est calculée avec l'angle entre  $t_{relative}$  et  $t_{GPS}$ .

### 2.2 Explication géométrique

Considérons, sur la figure 1, la translation  $t_{relative}$  entre la caméra  $C_i$  et la caméra  $C_1$ . Cette translation peut être calculée lorsque le repère de la caméra  $C_1$  est pris arbitrairement comme repère absolu, c'est donc une translation relative au repère de la caméra  $C_1$ . Considérons de plus la translation  $t_{GPS}$  donnée par le GPS entre la caméra  $C_i$  et la caméra  $C_1$  dans le repère UTM. Puisque les poses des caméras  $C_i$  et  $C_1$  forment un système rigide, on comprend alors intuitivement que si l'on peut trouver la rotation permettant d'aligner la direction de  $t_{relative}$  sur celle de  $t_{GPS}$ , alors on pourra en déduire la rotation  $R_1$  entre le repère UTM et le repère caméra.

Il reste toutefois un degré de liberté qui est la rotation autour de l'axe de  $t_{GPS}$ . En effet, même si les directions des vecteurs  $t_{relative}$  et  $t_{GPS}$  sont alignées, la caméra peut néanmoins tourner autour de cet axe. Pour éliminer ce degré de liberté, l'orientation de la caméra est contrainte à une rotation plane autour de l'axe vertical. La rotation entre  $t_{relative}$  et  $t_{GPS}$  est ainsi déterminée par l'angle  $\theta$

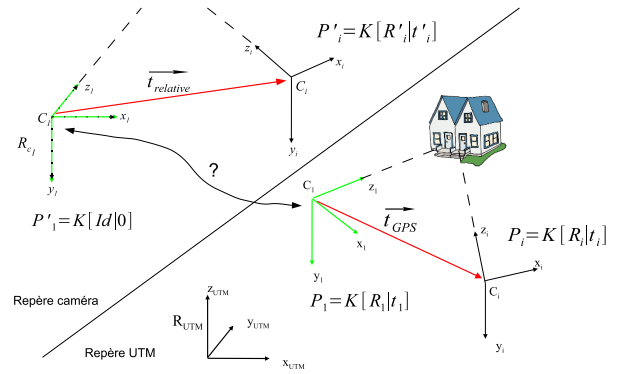


Figure 1 – Identification des vecteurs  $t_{relative}$  et  $t_{GPS}$  pour estimer  $R_1$

décrit par les composantes horizontales de ces vecteurs.

### 2.3 Calcul de $t_{relative}$

Soit  $P'_i = K[R'_i|t'_i]$ , la matrice de projection de la caméra  $C_i$ , exprimée dans le repère de la caméra  $C_1$ .  $R'_i$  et  $t'_i$  expriment donc la pose relative des deux caméras. Le changement de repère entre les repères caméra est donné par :

$$\mathcal{R}_{C_1} \mathcal{T}_{\mathcal{R}_{C_i}} = \begin{bmatrix} R'_i{}^{-1} & -R'_i{}^{-1}t'_i \\ 0 & 1 \end{bmatrix} \quad (1)$$

Comme  $t_{relative}$  est la position de  $C_i$  exprimé dans  $\mathcal{R}_{C_1}$  :

$$t_{relative} = C_{i\mathcal{R}_{C_1}} = \mathcal{R}_{C_1} \mathcal{T}_{\mathcal{R}_{C_i}} C_{i\mathcal{R}_{C_i}} = -R'_i{}^{-1}t'_i \quad (2)$$

La pose  $(R'_i, t'_i)$  est estimée par une approche de type SFM (Structure from motion) [4]. La méthode se décompose en 4 étapes :

1. Choix de l'image clé  $i$  correspondant à la pose  $i$  de la caméra.
2. Détection et suivi de points d'intérêts entre la première et la  $i^{\text{ème}}$  image.
3. Calcul de la matrice essentielle  $E$ .
4. Décomposition de la matrice essentielle  $E$  en  $t'_i$  et  $R'_i$ .

La matrice fondamentale  $F$  est la représentation algébrique de la géométrie épipolaire entre deux vues. Elle permet de retrouver les poses relatives des caméras.

Pour estimer  $F$ , un ensemble des correspondances de points  $x_1 \leftrightarrow x_2$  entre les deux vues est requis. La solution adoptée ici est celle du suivi de points proposée par Kanade, Lucas et Tomasi [1]. Elle consiste à détecter des points caractéristiques dans la première image puis à les suivre à travers la séquence vidéo jusqu'à atteindre l'image clé  $i$ .

Dans l'étape 3, la matrice essentielle est calculée à partir de  $F$  et de la matrice des paramètres intrinsèques  $K$  par  $E = K^T F K$ .

<sup>1</sup>implémentation disponible à l'adresse <http://www.ces.clemson.edu/stb/klf/>

Dans l'étape 4, la translation relative  $t'_i$  et la rotation relative  $R'_i$  sont retrouvées à partir de l'équation  $E = [t'_i] \times R'_i$ , par une décomposition en valeurs singulières de  $E$ [4](pages 257-259).

## 2.4 Identification de $t_{relative}$ et $t_{GPS}$

$C_i$  peut aussi être exprimé dans  $\mathcal{R}_{C_1}$  grâce aux données GPS :

$$C_{i_{\mathcal{R}_{C_1}}} = \mathcal{R}_{C_1} \mathcal{T}_{\mathcal{R}_{UTM}} C_{i_{\mathcal{R}_{UTM}}} = R_1 C_{i_{\mathcal{R}_{UTM}}} + t_1 \quad (3)$$

car

$$\mathcal{R}_{C_1} \mathcal{T}_{\mathcal{R}_{UTM}} = \begin{bmatrix} R_1 & t_1 \\ 0 & 1 \end{bmatrix} \quad (4)$$

Or  $t_1$  peut se déduire de :

$$\begin{aligned} C_{1_{\mathcal{R}_{UTM}}} &= \mathcal{R}_{UTM} \mathcal{T}_{\mathcal{R}_{C_1}} C_{1_{\mathcal{R}_{C_1}}} = -R_1^{-1} t_1 \\ \Leftrightarrow t_1 &= -R_1 C_{1_{\mathcal{R}_{UTM}}} \end{aligned} \quad (5)$$

En remplaçant  $t_1$  dans l'équation (3) par (5) :

$$\begin{aligned} C_{i_{\mathcal{R}_{C_1}}} &= R_1 C_{i_{\mathcal{R}_{UTM}}} - R_1 C_{1_{\mathcal{R}_{UTM}}} \\ &= R_1 (C_{i_{\mathcal{R}_{UTM}}} - C_{1_{\mathcal{R}_{UTM}}}) = R_1 (t_{GPS}) \end{aligned} \quad (6)$$

Finalement, on obtient avec les équations (2) et (6) :

$$t_{relative} = R_1 t_{GPS} \quad (7)$$

Le vecteur  $t_{GPS}$  est mesuré entre les points  $(X_i, Y_i, Z_i)$  et  $(X_1, Y_1, Z_1)$  qui sont extraits des données GPS et  $t_{relative}$  est calculé avec  $R'_i$  et  $t'_i$  comme expliqué dans la section précédente.

## 2.5 Calcul de $R_1$

On veut à présent calculer  $R_1$ , la rotation entre les vecteurs  $t_{relative}$  et  $t_{GPS}$ . En théorie, il existe un nombre infini de rotations 3D permettant d'aligner les vecteurs  $t_{relative}$  et  $t_{GPS}$ . Pour supprimer cette ambiguïté, on considère que  $R_1$  est une rotation plane dans le plan horizontal. L'angle entre composantes horizontales des vecteurs  $t_{relative}$  et  $t_{GPS}$  définit alors l'angle de la rotation  $R_1$ .

L'axe vertical est l'axe  $\vec{y}$  dans le repère caméra, et l'axe  $\vec{z}$  dans le repère UTM. De ce fait  $R_1$  est estimée à partir de  $t_{relative} = (t_{r_x}, t_{r_y}, t_{r_z})$  et  $t_{GPS} = (t_{GPS_x}, t_{GPS_y}, t_{GPS_z})$  par :

$$\tilde{R}_1 = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ 0 & 0 & -1 \\ -\sin \theta & \cos \theta & 0 \end{bmatrix} \quad (8)$$

$$\text{avec } \theta = \arccos(t_{r_x} t_{GPS_x} + t_{r_z} t_{GPS_z})$$

## 3 Recalage précis modèle/image

Lorsque la pose approximative de la caméra est connue, il est possible d'estimer simultanément la pose précise de la caméra et des correspondances 3D/2D de primir.

## 3.1 Détection et labellisation de primitives

Nous avons choisi d'utiliser comme primitives des lignes car elles sont plus stables que les points. En particulier, il y a moins de chances qu'elles soient produites par du bruit et elles résistent mieux aux occlusions.

Dans l'image, les lignes sont détectées par une méthode classique utilisant l'algorithme de Canny et la transformées de Hough.

Dans le modèle 3D, les lignes sont facilement détectées en projetant les coins des bâtiments dans le plan image puis en reliant deux à deux les points formant les contours des façades.

Le nombre de correspondances 2D/3D doit être réduit pour éviter un temps de calcul trop long et pour avoir une proportion d'*outliers* compatible avec l'algorithme RANSAC. Pour éviter des seuils ad hoc de distance entre lignes, nous proposons de réduire le nombre de correspondance possibles par labellisation des lignes.

Pour cela, une classification de l'image en 3 zones (ciel, bâtiment, sol) est réalisée avec la méthode d'estimation du contexte géométrique proposée dans [5]. Les lignes proches des frontières ciel/bâtiment et sol/bâtiment sont sélectionnées et se voient attribuer un label *ciel* ou *sol*. Toutes les lignes verticales détectées sont conservées et associées au label *verticale*. Les lignes du modèle 3D projeté sont également labellisées en *ciel*, *sol* et *verticale*.

On considère alors toutes les correspondances 2D/3D possibles entre lignes de label identique *ciel* ou *sol*. Pour les lignes verticales, un seuil sur la distance limite les correspondances. Ce seuil est fixe et égal à  $0.5 \times \text{Largeur Image}$ .

## 3.2 Estimation robuste avec RANSAC

L'algorithme d'estimation robuste RANSAC est utilisé pour combiner de façon itérative la recherche de correspondances avec le calcul de pose. La procédure suivie peut être résumée ainsi :

1. Sélectionner aléatoirement un sous-ensemble de correspondances
2. Calculer la pose du modèle à partir du sous-ensemble sélectionné.
3. Tester la nouvelle pose en recherchant les *inliers* sur tout l'ensemble de correspondances. La pose produisant le plus grand nombre d'*inliers* est conservée.
4. Ré-itérer les étapes 1 à 3  $N$  fois.
5. Calculer précisément la pose avec l'ensemble d'*inliers* trouvé.

Le nombre d'itérations  $N$  est déterminé de façon adaptative en suivant la méthode présentée dans [4] page 117.

La sélection des *inliers* à l'étape 3 utilise la représentation des lignes en coordonnées polaires  $(\rho, \theta)$ . Une bonne correspondance (*inliers*) est telle que :

$$|\rho_{M3d} - \rho_{Img}| < s_{\rho_{min}} \text{ et } |\theta_{M3d} - \theta_{Img}| < s_{\theta_{min}}$$

Le seuil ( $s_{\rho_{min}}, s_{\theta_{min}}$ ) est de manière classique fixé à la valeur de l'écart-type sur les erreurs de mesure.

### 3.3 Calcul de pose

La méthode d'asservissement visuel virtuel présentée dans [8] est utilisée ici. Le but est de calculer le mouvement de la pose permettant de réduire l'erreur d'alignement obtenue à l'étape 2 de RANSAC. Ce calcul est ré-itéré jusqu'à ce que l'erreur soit suffisamment faible.

L'inconnue recherchée est la vélocité de la pose :

$$v = [t_x \ t_y \ t_z \ r_x \ r_y \ r_z]^T$$

où les  $t$ . correspondent au mouvement translationnel et les  $r$ . au mouvement rotationnel. Pour obtenir une décroissance exponentielle de l'erreur :

$$\dot{e} = -\lambda e$$

où  $\lambda$  est un gain scalaire, alors La loi de commande reliant l'erreur d'alignement,  $e$ , et l'inconnue,  $v$ , est donnée par :

$$v = -\lambda \bar{L}_{pm}^+ e$$

où  $\lambda$  est un facteur de décroissance exponentielle et  $L_{pm}$  est appelée matrice d'interaction ou encore matrice jacobienne.  $\bar{L}_{pm}$  est une estimation de  $L_{pm}$ , et  $\bar{L}_{pm}^+$  est la pseudo-inverse de  $\bar{L}_{pm}$ .

Cette matrice d'interaction 2x6, bien connue dans le domaine de l'asservissement visuel, est obtenue à partir des primitives sélectionnées dans l'étape 1 de RANSAC. Le choix de ces primitives est important car il doit permettre d'obtenir une matrice d'interaction de rang plein 6. En théorie, 3 lignes placées en configuration non-dégénérées suffisent, mais en pratique et du fait du bruit présent dans l'image, un minimum de 4 lignes est nécessaire pour le calcul précis de la pose.

## 4 Résultats

### 4.1 Données utilisées

Présentons tout d'abord les données ainsi que leur acquisition. Les tests ont été effectués avec une vidéo du Campus de l'université de Rennes, au format 400x300, filmée avec une caméra numérique grand public portée à l'épaule. Les mesures GPS sont enregistrées simultanément avec le flux vidéo. Une expérimentation dans des conditions peu favorables à une acquisition GPS (au pied d'un bâtiment, sous un temps couvert) a permis de déterminer une précision horizontale du GPS d'environ 5m. Ces données horizontales sont interpolées linéairement pour fournir une position à chaque instant temporel correspondant à une image de la séquence vidéo. De plus, la précision verticale étant très peu fiable, une meilleure estimation est donnée par une initialisation arbitraire à une altitude de 1,5m au dessus du sol. Pour cela, comme les données SIG ne contiennent pas la surface du sol, celle-ci est modélisée par interpolation par triangulation de Delaunay sur les sommets au sol des bâtiments. On obtient ainsi, pour chaque image de la séquence, une position  $p_t = (X_t, Y_t, Z_t)$ .

### 4.2 Valeur des paramètres

Les paramètres de la méthode sont fixes pour toutes les séquences ou calculés à partir des données. Le nombre initial de points caractéristiques suivis est de 50 ; le seuil de sélection des *inliers* prend la valeur  $(s_{\theta}^l, s_{\rho}^l) = (0.03, 2)$  ; le nombre d'itération  $N$  est déterminé adaptativement en fonction du nombre d'*inliers* trouvés, en fixant à 0.99 la probabilité  $p$  d'avoir au moins un tirage sans *outlier*. Le choix de l'image  $i$  est pour l'instant manuel, mais les tests réalisés ont montré qu'un large intervalle correspond à des images  $i$  valides (typiquement de 20 à 30 images successives).

### 4.3 Alignement obtenu

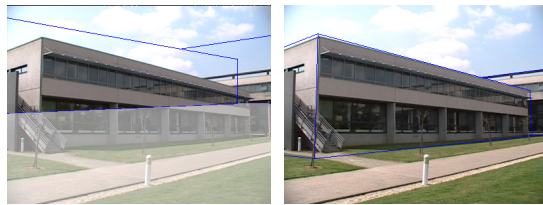
La figure 2 montre les résultats obtenus pour la séquence *ifsic*. La figure 2.a montre l'alignement obtenu après la première partie de l'algorithme. On peut y voir la première image avec en surimpression le modèle SIG projeté en mode fil de fer (en bleu), la surface semi-transparente représentant le sol dans le SIG. On remarque tout d'abord que les façades visibles du modèle SIG correspondent à celles visibles dans l'image. L'objectif de la première partie est donc atteint car une mise en correspondance est possible. Il apparaît que la façade principale du modèle SIG est plus longue et moins haute que celle de l'image. Cela est dû à une erreur sur la position de la caméra donnée par le GPS.

La figure 2.b montre le recalage final obtenu, en 958 itérations de l'algorithme Ransac avec un ensemble de 19 correspondances initiales. Les lignes de l'image, issues de la détection de Canny (2.c) puis d'une transformée de Hough, sélectionnées et labélisées à l'aide du contexte géométrique (2.d), sont montrées sur la figure 2.e. Les lignes extraites du modèle 3D projeté sont montrées sur la figure 2.f.

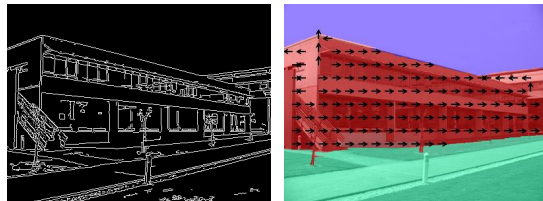
La figure 3 montre des résultats sur une autre séquence.

### 4.4 Interprétation

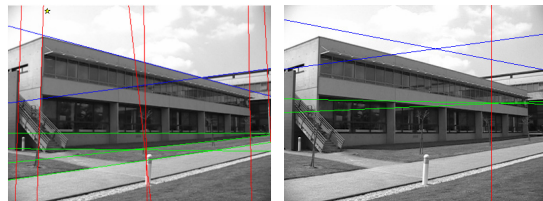
Les résultats obtenus valident l'approche proposée. La bonne réalisation de la première étape de l'algorithme est assurée malgré les erreurs de mesure GPS, les erreurs d'estimation du déplacement relatif et l'hypothèse sur la position verticale de la caméra. Les primitives attendues se projettent dans l'image. Dans la deuxième partie de la méthode, la difficulté consiste à extraire un nombre suffisant de bonnes correspondances, et de les discriminer parmi les fausses correspondances. La première image doit pour cela contenir des arêtes du modèle 3D en nombre suffisant, et si possible non toutes coplanaires. Ces arêtes sont parfois difficiles à extraire car peu contrastées et proches d'autres lignes de l'image : par exemple, l'arête inférieure du bâtiment, qui est de plus proche d'autres lignes de l'image (ombre, contour des fenêtres). Lorsque les arêtes de bâtiment sont détectées et sélectionnées dans l'image, la pose estimée est correcte, même à partir d'une pose approximative très éloignée.



(a) recalage approximatif (b) recalage final ( $i = 80$ )



(c) contours de Cany (d) contexte géométrique



(e) lignes de l'image (f) lignes du modèle 3D

Figure 2 – Résultats pour la séquence ifsic



(a) recalage approximatif (b) recalage final ( $i = 20$ )

Figure 3 – Résultats pour la séquence beaulieu

## 5 Conclusion et travaux futurs

Nous avons présenté dans cet article une méthodologie permettant le recalage initial d'un modèle SIG avec une vidéo, de façon automatique. La solution proposée se décompose en deux parties qui sont, premièrement, l'estimation approximative de la pose du modèle en utilisant deux images et les données GPS, et deuxièmement, l'estimation précise de la pose par la mise en correspondance des lignes du modèle avec les lignes de l'image initiale. Les premiers résultats ont montré que l'estimation approximative de la

pose permet bien de superposer les champs de vue image et modèle SIG projeté dans l'optique d'établir une mise en correspondance de lignes. En ce qui concerne le recalage précis du modèle dans la seconde partie, un alignement correct a été obtenu pour l'orientation et la position. Peu de paramètres interviennent et leur valeur est fixe pour toutes les séquences. La suite de cette étude portera sur le choix automatique d'une image clé valide et sur la suppression de l'hypothèse d'une rotation autour de l'axe vertical. On envisage également d'améliorer la mise-en-correspondance 2D/3D en utilisant les techniques de *Sfm* pour extraire des informations 3D sur les primitives images.

## Références

- [1] Philip David, Daniel DeMenthon, Ramani Duraiswami, and Hanan Samet. Softposit : Simultaneous pose and correspondence determination. In *ECCV (3)*, pages 698–714, 2002.
- [2] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs : A hybrid geometry- and image-based approach. *Computer Graphics*, 30(Annual Conference Series) :11–20, 1996.
- [3] Martin A. Fischler and Robert C. Bolles. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6) :381–395, 1981.
- [4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN : 0521540518, second edition, 2004.
- [5] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Geometric context from a single image. In *ICCV*, volume 1, pages 654 – 661. IEEE, October 2005.
- [6] K. Karam, J. Bauer, A. Klaus, and K. Schindler. Metropogis : a city information system. In *ICIP02*, pages III : 533–536, 2002.
- [7] Lingyun Liu and Ioannis Stamos. Automatic 3d to 2d registration for the photorealistic rendering of urban scenes. In *CVPR '05 - Volume 2*, pages 137–143, Washington, DC, USA, 2005. IEEE Computer Society.
- [8] E. Marchand and F. Chaumette. Virtual visual ser-voing : a framework for real-time augmented reality. *Computer Graphics Forum*, 21(3) :289–298, September 2002.
- [9] Gerhard Reitmayr and Tom Drummond. Going out : robust model-based tracking for outdoor augmented reality. In *ISMAR*, pages 109–118, 2006.
- [10] Seth Teller, Matthew Antone, Zachary Bodnar, Michael Bosse, Satyan Coorg, Manish Jethwa, and Neel Master. Calibrated, registered images of an extended urban area. *Int. J. Comput. Vision*, 53(1) :93–107, 2003.