Recalage GPS / SIG / Video, et synthèse de textures de bâtiments

Gaël Sourimant, Thomas Colleu, Vincent Jantet, Luce Morin

Recalage GPS / SIG / Video, et synthèse de textures de bâtiments

G. Sourimant[1] T. Colleu[1] V. Jantet[1] L. Morin[2]

[1] Irisa / Inria Rennes Bretagne Atlantique, Campus Universitaire de Beaulieu, 35042 Rennes Cedex

[2] Insa Rennes, 20, Avenue des Buttes de Coësmes, 35043 Rennes Cedex

{gael.sourimant, thomas.colleu, vincent.jantet}@irisa.fr

luce.morin@insa-rennes.fr

Résumé

Dans le contexte du recalage de données SIG de bâtiments avec des vidéos — par exemple pour des applications de réalité augmentée — nous présentons une solution à un des problèmes les plus critiques, à savoir l'initialisation de ce recalage. La méthode proposée exploite d'une part les informations sémantiques que l'on peut associer aux primitives extraites des images, et d'autre part le principe même de l'algorithme robuste RANSAC pour trouver automatiquement la pose initiale de la caméra d'acquisition.
Nous montrons également comment ce recalage peut être exploité pour enrichir la base SIG visualisée par des textures réelles, calculées à partir des images acquises au sol, et ce de façon tout aussi automatique.

Mots clefs

Recalage 2D/3D, GPS, SIG, RANSAC, Modélisation Urbaine, Synthèse de Textures.

1 Introduction

La génération de modèles 3D d'environnements urbains a de nombreuses applications en réalité virtuelle ou augmentée. On a d'une part une demande de plus en plus forte pour avoir accès à des modèles 3D de qualité pour des applications de navigation virtuelle, comme on peut s'en rendre compte avec le succès de Google Earth ou Virtual Earth. D'autre part, on peut également chercher à rajouter des informations virtuelles sur des vidéos réelles, par exemple dans le cadre d'une navigation à l'aide d'un GPS utilisant les images de ce que voit réellement l'utilisateur plutôt que des cartes virtuelles.

La fusion de données géométriques synthétiques de bâtiments avec des images réelles est le point central de ces deux approches complémentaires. Notre étude se place dans ce cadre : on dispose d'un modèle synthétique géo-référencé de bâtiments (issu d'un SIG, pour Système d'Information Géographique), qui contient pour chaque bâtiment son empreinte au sol et son élévation, ainsi que des vidéos acquises en milieu urbain conjointement avec des mesures GPS. Nous montrons qu'un recalage entre les modèles 3D de bâtiments avec les vidéos est possible — recalage qui pourrait alors être utilisé pour des applications de réalité augmentée — et que ce recalage peut être exploité pour améliorer les modèles 3D existants, par exemple en extrayant les textures réelles des façades des bâtiments — qui peuvent alors être utilisées dans des outils de navigation virtuelle.

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 2D/3D. 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 [1, 2, 3] l'utilisateur indique lui-même les correspondances. Dans [4, 5], 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 [5] ou un modèle provenant d'un scanner 3D [6]. 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 [7], et 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 [8], et 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. Tout d'abord, nous calculons une pose approximative de la caméra permettant d'obtenir pour le modèle SIG projeté les mêmes primitives que celles présentes dans l'image. Cette première estimation utilise uniquement les

données GPS et la vidéo et ne fait pas intervenir le modèle SIG. Puis nous estimons simultanément des correspondances 2D/3D et la pose en utilisant une méthode basée RANSAC. Cette estimation utilise le modèle SIG et la première image de la séquence vidéo. La pose est alors suivie dans les images suivantes en utilisant un algorithme d'asservissement visuel virtuel robuste.

Une fois le recalage avec le modèle effectué, nous proposons de l'exploiter pour extraire de la vidéo non pas une texture mais un ensemble de textures pour chaque façade de bâtiment. Elles sont définies dans l'espace du plan principal de celle-ci [9, 10] (par opposition au choix d'une image de référence [11, 12]). A partir de la pile de textures obtenue, nous proposons un algorithme de fusion pixel à pixel pour calculer l'image finale. Les problèmes généralement rencontrés pour le calcul de textures de façades sont de plusieurs ordres. Un bâtiment n'est pas forcément visible complètement dans chaque image, et peut être masqué partiellement par un autre bâtiment. On parle alors d'occultations modélisables, qui peuvent être traitées via l'utilisation de masques calculés à partir du recalage avec le modèle 3D [13, 9, 10]. Les occultations non modélisables (*i.e.* générées par des objets non modélisés dans la base 3D) sont quant à elles généralement supprimées en utilisant soit des outils robustes basés sur des mesures de luminances médianes [11, 12], soit un processus itératif basé sur des masques de corrélation [10]. De façon plus anecdotique, certains travaux se penchent également sur les différences de résolution spatiale des images de la pile [10, 9, 13], le remplissage des zones inconnues [14, 12] ou les variations d'illumination [10].

Dans la section 2, nous présentons notre méthode de recalage automatique entre des données SIG et vidéo. Nous montrons dans la section 3 comment ce recalage peut-être exploité pour le calcul de textures photoréalistes de façades, avant de conclure et d'indiquer quelques perspectives (section 4).

2 Recalage SIG - Vidéo

Nous présentons dans cette partie une méthode automatique permettant de calculer de façon précise la pose de la caméra pour toutes les images de la vidéo considérée. Ce calcul de pose se décompose en deux étapes principales :

1. *Calcul de pose pour la première image.* Un algorithme basé sur les premières images et les mesures GPS permet de calculer précisément la pose pour la première image de la vidéo, pour laquelle on ne connait à ce stade qu'une approximation de la position.

2. *Suivi de la pose.* La mise en correspondance entre des primitives extraites des images et le modèle 3D, associée au suivi de ces primitives dans la vidéo, permet de suivre la projection du modèle recalée pour la première image.

2.1 Initialisation du recalage

Recalage initial exploitant une image clé de la vidéo. Pour initialiser le recalage entre le modèle 3D issu du SIG et la vidéo acquise, la seule donnée dont on dispose de prime abord est l'ensemble des mesures de positionnement GPS associées au trajet effectué par la caméra. Nous sommes donc capables à ce point de nous positionner approximativement au sein du modèle SIG, mais pas de nous orienter pour savoir quelle direction avait la caméra. Pour lever cette ambiguité, nous proposons de découper une fois encore le problème en deux parties :
Estimation aproximative de la pose. Le mouvement approximatif entre deux positions données de la caméra est estimé en utilisant les images seules. La translation estimée est alors mise en correspondance avec la translation mesurée par GPS pour donner l'orientation approximative.
Raffinement de la pose. La pose approchée est exploitée pour détecter des droites 3D dans le modèle, et les mettre en correspondance avec des droites 2D extraites des images (contraintes par le contexte de l'image : droites au sol, verticales, ou à la limite avec le ciel). La pose est calculée *via* un RANSAC qui recherche le meilleur jeu de correspondances qui minimise l'erreur entre les droites extraites des images et les droites 3D projetées.
Ces travaux ont déjà été décrits avec plus de détails dans [15]. La principale limitation de cette approche est qu'elle nécessite de choisir au sein de la vidéo une *image clé*, utilisée pour estimer la pose *relative* entre la première image et celle-ci, dont la translation est mise en correspondance avec celle mesurée par GPS. Dans [15], l'image clé est sélectionnée manuellement.

Recalage initial automatique. Nous souhaitons minimiser encore l'intervention de l'utilisateur, pour tendre vers une procédure complètement automatique. Le choix arbitraire de l'image clé n'est en effet pas satisfaisant car trop dépendant des données : les mesures GPS sont souvent trop bruitées en milieu urbain, entraînant de fait une fausse estimation approximative de la pose, dont on ne peut extraire de correspondances 2D/3D pour calculer la pose de façon précise.
Nous proposons alors un algorithme *supervisé* pour le recalage initial, où l'intervention de l'utilisateur se limite à valider ou non une pose calculée. En cas de rejet, une nouvelle pose est proposée à l'utilisateur. Cette méthode se base sur le postulat suivant : étant donnée une image clé, quels sont les critères permettant de déterminer que la pose estimée (ou estimable) est viable ? La procédure de recalage devient alors séquentielle, dans le sens où l'on teste potentiellement toutes les images de la vidéo comme image clé, l'une après l'autre, en utilisant l'algorithme décrit dans [15]. Plusieurs critères sont utilisés pour déterminer si une image est une image clé valide pour le calcul de pose. Dès que l'un de ces critères est invalidé, l'image suivante de la vidéo est utilisée comme image clé.
Géométrie épipolaire. Tout d'abord, nous évaluons la

quantité de résidu épipolaire induit par la matrice fondamentale estimée $\mathbf{F}$, calculée par un algorithme robuste basé sur RANSAC. Si ce résidu est trop élevé, la géométrie épipolaire est considérée comme trop mal estimée pour calculer la pose relative des caméras. Sinon, on calcule la pose approximative par identification avec la translation GPS.
Extraction des primitives. La pose approximative étant estimée, l'image est conservée si suffisament de droites 3D projetées peuvent être extraites et mises en correspondances avec les droites 2D pour calculer la pose. Le nombre de droites et leur configuration géométrique nécessaires sont décrites dans [15].
Mise en correspondance robuste. A partir des droites 2D et 3D, un algorithme basé RANSAC itère sur l'ensemble de correspondances possibles pour supprimer les outliers et ne conserver que celles qui sont valides. Le succès de cette phase peut être mesuré par le nombre d'itérations RANSAC et le nombre de correspondances trouvées. En effet, ce nombre décroît quand la probabilité de trouver une solution valide augmente. Un simple seuillage sur ce nombre permet ou non de valider l'image clé sélectionnée.

Si une image passe ces trois tests successifs, alors la pose trouvée est soumise à l'utilisateur, qui la valide ou non, auquel cas, une nouvelle image clé est déterminée et la procédure itérée. Un exemple de recalage proposé par notre méthode est illustré sur la figure 5. Il s'agit ici de la première proposition de solution faite par l'algorithme.

2.2 Suivi du recalage

Une fois la pose de la caméra estimée pour la première image de la vidéo, on souhaite la calculer pour toutes les images restantes. Sous l'hypothèse de faibles déplacements inter-images, ce calcul de pose revient à un suivi du recalage estimé pour la première image. Nous présentons ici un rapide résumé de notre méthode. Plus de détails peuvent être trouvés dans [16].
Plusieurs approches pour effectuer un tel suivi sont décrites dans la littérature (par exemple [17]). La méthode que nous avons retenue est une variation robuste de l'algorithme d'asservissement visuel virtuel [18]. Le calcul de pose par asservissement nécessite un ensemble de correspondances entre des primitives 3D appartenant au modèle à recaler, et la projection 2D de ces primitives dans les images. De par la simplicité de leur modélisation, de leur extraction et de leur suivi, nous utilisons des points pour cette mise en correspondance de primitives. Pour assurer le suivi du modèle 3D tout au long de la séquence d'images, nous utilisons un schéma de transfert de points d'une image à l'autre. Les points utilisables sont a priori ceux présents à la fois dans les images et le modèle 3D, c'est-à-dire ceux appartenant aux façades.
Pour assurer un suivi robuste et minimiser la dérive de la pose estimée de la caméra au cours du temps, nous proposons d'extraire en plus des points de façades des points appartenant au sol. Les informations 3D associées à ces points sont (en l'absence ici de DEM) estimées à partir d'une triangulation de Delaunay des empreintes au sol des bâtiments. De plus, la loi de commande d'asservissement utilisée pour calculer la pose est augmentée d'un M-Estimateur comme le propose [18]. La fonctionnelle à minimiser est alors :

$$\mathbf{v} = -\lambda(\mathbf{DL})^{+}\mathbf{D}(\mathcal{P}(\mathbf{X}) - \mathbf{x}), \tag{1}$$

avec $\mathbf{D} = diag(w_1, ..., w_N)$ l'ensemble des poids calculés lors de la M-Estimation (*via* une fonction de coût robuste de Cauchy), $\mathbf{v}$ le vecteur décrivant la pose de la caméra recherchée, $\mathcal{P}(\mathbf{X}) - \mathbf{x}$ la différence pour un point donné entre sa position mesurée dans l'image et la projection de son correspondant 3D à la pose donnée, et $\mathbf{L}$ la matrice d'interaction dépendant des primitives projetées et de la profondeur relative entre la caméra et l'objet visualisé.
Des résultats de suivi pour deux séquences sont présentés sur la figure 4. On remarque que même en présence d'objets occultants, ou de bâtiments qui disparaissent puis réapparaissent, le suivi reste satisfaisant.

3 Extraction des textures de façades

Le problème que l'on cherche à résoudre ici consiste à exploiter le recalage entre les images source et la projection du modèle 3D correspondante pour calculer la texture finale $\mathcal{T}$ de chaque façade visible du modèle. Pour cela, chaque façade f visible dans chaque image I_k permet de générer une texture $\mathcal{T}_k^f$, qui sera généralement incomplète, masquée par des objets du premier plan mais non modélisés dans la base SIG, et plus ou moins floue selon la méthode d'extraction et la configuration géométrique de la caméra. Le calcul de $\mathcal{T}$ se fait alors en deux étapes : extraction des textures $\mathcal{T}_k^f$ et construction de la pile d'images correspondante, puis calcul $\mathcal{T}$ par fusion texel à texel[1] de la pile d'images.

3.1 Extraction des textures

Soit une image I_k ($k \in \{1..n\}$) de la séquence d'origine. On suppose que m façades sont visibles dans I_k. On cherche alors à calculer les m textures $\mathcal{T}_k^f$ correspondant à une image fronto-parallèle des-dites façades. Le ratio des dimensions de $\mathcal{T}_k^f$ respecte celui des dimensions de la façade dans le modèle. Nous utilisons donc un facteur d'échelle η pour passer du domaine métrique au domaine texel.
Les données connues, une fois le recalage image-modèle effectué, sont les coordonnées dans le repère image des quatre coins de chaque façade $\mathbf{x} = [u_i\ v_i]^\top$, $i \in \{1..4\}$. Pour calculer la transformation homographique permettant de passer dans le repère texture $\mathcal{T}_k^f$, on met ces points en correspondance avec les quatre coins de $\mathcal{T}_k^f$, de coordonnées $\mathbf{x}' = [0/u'_j\ 0/v'_j]^\top$, $j \in \{1..4\}$. Si w (resp. h) est la largeur (resp. la hauteur) de la façade f, on a alors $u'_j = \eta w$ et $v'_j = \eta h$. On notera également que la connaissance de la

[1] On différentie dans le texte les *pixels*, qui sont les unités de base des images, des *texels* qui sont les unités de base dans les textures, pour mieux différentier les deux représentations.

pose de la caméra par rapport au modèle 3D n'impose pas aux points de $\mathbf{x}$ de se situer dans les limites de l'image. Ces correspondances sont illustrées sur la figure 1.

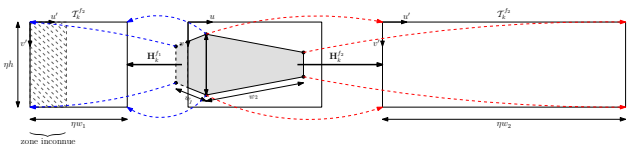

Figure 1 – *Utilisation des coordonnées de sommets de façades pour calculer les textures* $\mathcal{T}_k^f$

La correspondance entre $\mathbf{x}$ et $\mathbf{x}'$ est formalisée par la relation homographique $x' \sim \mathbf{H}_k^f \mathbf{x}$. On déduit de cette équation un système linéaire de la forme $\mathbf{Ah} = \mathbf{x}'$, $\mathbf{h}$ étant un vecteur contenant les entrées de l'homographie souhaitée. Une fois les homographies $\mathbf{H}_k^f$ estimées, les textures $\mathcal{T}_k^f$ sont calculées en utilisant la transformée inverse $\mathbf{H}_k^{f\,-1}$, la couleur de chaque texel $[u'\ v']^\top$ de $\mathcal{T}_k^f$ étant donnée par interpolation bicubique de son correspondant $[u\ v]^\top$ dans I_k.

3.2 Fusion des textures

On dispose désormais d'une pile de textures, et le calcul de la texture finale $\mathcal{T}^f$ se fait texel à texel. Pour chaque pile de texels, la couleur finale est calculée comme la somme pondérée des couleurs de la pile, le poids associé à chaque texel de la pile étant défini par $w^{f,u,v}$. La mise en place de ces poids est explicitée dans les paragraphes suivants.

Occultations modélisables. C'est lors du calcul de la couleur de chaque texel $\mathcal{T}^f_{k\ u,v}$ par transformation inverse puis interpolation que sont gérées les occultations modélisables. A ce stade, chaque texel se voit attribuer un poids $w^{f,u,v}_{occ.mod.}$, qui vaut 1 si le texel est visible et 0 sinon. Si le pixel de coordonnées $[u\ v\ 1]^\top \sim \mathbf{H}_k^{f\,-1}[u'\ v'\ 1]^\top$ est en dehors de l'image I_k, alors le texel n'est pas visible. De plus, si le pixel est dans l'image, la pose de la caméra est utilisée pour déterminer par rétro-projection la façade f' à laquelle appartient ce pixel. Si $f \neq f'$, alors il n'est pas non plus visible (voir figure 2).

$$\begin{cases} w^{f,u,v}_{occ.mod} = 1 & \text{si le texel est visible} \\ w^{f,u,v}_{occ.mod} = 0 & \text{sinon} \end{cases} \tag{2}$$

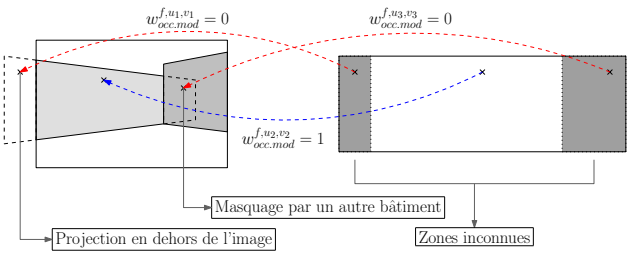

Figure 2 – *Gestion des occultations modélisables lors de la construction des textures*

Occultations non modélisables. Soit $\mathcal{T}^f_{u,v}$ la pile de texels de position $[u\ v]^\top$ dans $\mathcal{T}^f$. Certains de ces texels sont entachés d'erreur (au sens de la couleur) du fait des objets occultants non modélisés (outliers). On souhaite ne conserver que les texels correspondant à la façade (inliers) pour calculer la couleur finale du texel dans $\mathcal{T}$. Si on suppose que pour $\mathcal{T}^f_{u,v}$ les outliers représentent moins de 50% des échantillons, alors le texel $\mathcal{T}^f_{u,v\,j}$ dont la couleur est la médiane des couleurs de la pile est un inlier. Ceux dont la couleur est suffisamment proche sont également considérés comme des inliers. Soit $C(\mathcal{T}^f_{u,v\,j})$ la couleur du $j^{\text{ième}}$ texel de la pile. La couleur médiane est donnée par $C(\mathcal{T}^f_{u,v})_{\text{med}} = \underset{j}{\text{med}}(C(\mathcal{T}^f_{u,v\,j}))$. On peut calculer l'écart des inliers à cette couleur de façon robuste en prenant la médiane de la valeur absolue de l'écart des couleurs à $C(\mathcal{T}^f_{u,v})_{\text{med}}$:

$$\begin{aligned} \Delta C(\mathcal{T}^f_{u,v}) &= \text{MAD}(C(\mathcal{T}^f_{u,v})) \\ &= \underset{j}{\text{med}}(|C(\mathcal{T}^f_{u,v\,j}) - \underset{k}{\text{med}}(C(\mathcal{T}^f_{u,v\,k}))|) \end{aligned} \tag{3}$$

On peut considérer que les inliers conservés $\widehat{\mathcal{T}^f_{u,v}}$ seront tous ceux dont l'écart à $C(\mathcal{T}^f_{u,v})_{\text{med}}$ est inférieur à $\lambda\Delta C(\mathcal{T}^f_{u,v})$, λ étant un scalaire fixé à 2 dans notre cas :

$$\forall k, \mathcal{T}^f_{u,v\,k} \in \widehat{\mathcal{T}^f_{u,v}} \Leftrightarrow |C(\mathcal{T}^f_{u,v\,k}) - C(\mathcal{T}^f_{u,v})_{\text{med}}| \leq \lambda\Delta C(\mathcal{T}^f_{u,v}) \tag{4}$$

La suppression des outliers se fait en leur attribuant un poids $w^{f,u,v}_{occ.n.mod.}$ nul :

$$\begin{cases} w^{f,u,v}_{occ.n.mod.} = 1 \text{ si } \mathcal{T}^f_{u,v\,k} \in \widehat{\mathcal{T}^f_{u,v}} \\ w^{f,u,v}_{occ.n.mod.} = 0 \text{ sinon} \end{cases} \tag{5}$$

Résolution Spatiale. A partir de la liste d'inliers $\widehat{\mathcal{T}^f_{u,v}}$, on cherche désormais à calculer la couleur finale du texel $\mathcal{T}_{u,v}$. Nous attribuons un poids à chaque inlier de la pile de telle sorte que l'influence des texels de plus haute résolution soit plus importante que ceux de basse résolution. Plusieurs critères peuvent être utilisés pour mesurer cette résolution, soit en utilisant la configuration géométrique de la scène, soit en considérant les images en entrée elles-mêmes.

Distance et angle Dans [10], l'angle θ entre la ligne de vue du pixel et celle du projeté du centre optique sur la façade est utilisé. Plus cet angle est important, plus le poids $w^{f,u,v}_{angle}$ est faible. Si on considère que $\theta \in]\frac{-\pi}{2}; \frac{\pi}{2}[$, alors on peut poser :

$$w^{f,u,v}_{angle} = \cos|\theta| \tag{6}$$

La distance entre la façade et la caméra étant également déterminante pour la résolution finale des texels, nous proposons en plus de leur assigner un poids $w^{f,u,v}_{dist}$, défini comme la distance entre le centre optique de la caméra

et le pixel considéré. Cette distance est calculée en utilisant le z-buffer[2] pour l'image k, au point de coordonnées $[u'\ v'\ 1]^\top \sim \mathbf{H}_k^{f\,-1}[u\ v\ 1]^\top$.

$$w_{dist}^{f,u,v} = 1/\text{z-buffer}(k, u', v') \tag{7}$$

Aire de projection Un autre moyen de mesurer la résolution des texels en utilisant la géométrie de la scène est de calculer, pour chacun d'entre eux, l'aire du quadrilatère correspondant dans l'image d'origine (voir figure 3). Plus l'aire est grande, plus le texel apporte une information visuelle pertinente. Si $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ et $\mathbf{d}$ sont les coordonnées des "sommets" du texel dans la texture $\mathcal{T}^f$, alors les coordonnées correspondantes $\mathbf{a}'$, $\mathbf{b}'$, $\mathbf{c}'$ et $\mathbf{d}'$ dans I_k^f sont calculées à l'aide de l'homographie $\mathbf{H}_k^{f\,-1}$. Le poids $w_{aire}^{f,u,v}$ est alors calculé comme l'aire du quadrilatère $(\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{d}')$, c'est-à-dire comme la moitié de la norme du produit vectoriel de ses diagonales :

$$w_{aire}^{f,u,v} = \frac{1}{2}\|(\mathbf{a}' - \mathbf{c}') \times (\mathbf{b}' - \mathbf{d}')\| \tag{8}$$

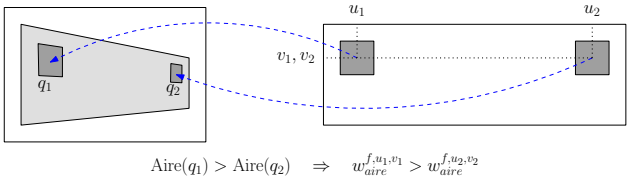

Figure 3 – *Mesure de l'aire des texels projetés*

3.3 Résultats

Sur la figure 6 on peut voir deux textures extraites d'une video de synthèse pour laquelle le recalage est connu, ainsi que la texture reconstruite à partir des textures élémentaires extraites (la mesure d'*aire* est utilisée pour la résolution). On peut voir que les objets occultants sont bien supprimés, que la résolution spatiale de l'image est conservée, et que les spécularités dans les fenêtres sont également absentes de la texture finale. On fait exactement le même constat pour les images réelles (figure 7). Une vue totalement virtuelle calculée en utilisant une unique vidéo est présentée sur la figure 8.
Lors de nos tests, nous n'avons pas pu déterminer quelle méthode entre la mesure d'aire ou de distance-angle était la meilleure pour conserver la résolution spatiale. Nous préconisons donc la première par souci de simplicité et d'efficacité d'implémentation.

4 Conclusion

Nous avons présenté dans cet article une méthodologie permettant le recalage — même initial — d'un modèle 3D basé SIG avec une vidéo, en exploitant des mesures GPS. Dans le cadre du recalage initial, nous proposons une solution automatique mais supervisée contrairement à la plupart des approches existantes. Nous montrons également comment un tel recalage peut-être exploité pour calculer automatiquement des textures de façades photoréalistes à partir d'images acquises au niveau du sol.
Dans un cadre plus contraint, où l'on a une confiance plus importante dans la précision des mesures GPS, on pourra lever la contrainte de supervision pour avoir une méthode complètement automatique de recalage SIG / Vidéo.
De plus, pour rendre le recalage au cours du temps encore plus robuste, nous souhaiterions d'une part intégrer certains outils du recalage initial dans le suivi (calcul de pose à partir de droites), et d'autre part intégrer des résultats de *Structure from Motion* pour mettre en correspondance non plus uniquement des primitives 2D/3D, mais également 3D/3D.

[2] Le z-buffer contient la distance de chaque pixel à un objet 3D affiché, ici les bâtiments SIG.

Références

[1] Heung-Yeung Shum, Mei Han, et Rick Szeliski. Interactive construction of 3d models from panoramic mosaics. Dans *Proc. of CVPR'98*, June 1998.

[2] Paul E. Debevec, Camillo J. Taylor, et Jitendra Malik. Modeling and rendering architecture from photographs : A hybrid geometry- and image-based approach. *Computer Graphics*, 30 :11–20, 1996.

[3] K. Karner, J. Bauer, A. Klaus, et K. Schindler. Metropogis : a city information system. Dans *ICIP02*, pages III : 533–536, 2002.

[4] Seth Teller, Matthew Antone, Zachary Bodnar, Michael Bosse, Satyan Coorg, Manish Jethwa, et Neel Master. Calibrated, registered images of an extended urban area. *Int. J. Comput. Vision*, 53(1), 2003.

[5] Gerhard Reitmayr et Tom Drummond. Going out : robust model-based tracking for outdoor augmented reality. Dans *ISMAR*, pages 109–118, 2006.

[6] Lingyun Liu et Ioannis Stamos. Automatic 3d to 2d registration for the photorealistic rendering of urban scenes. Dans *CVPR '05*, pages 137–143, 2005.

[7] Martin A. Fischler et 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.

[8] Philip David, Daniel DeMenthon, Ramani Duraiswami, et Hanan Samet. Softposit : Simultaneous pose and correspondence determination. Dans *ECCV (3)*, pages 698–714, 2002.

[9] Heinz Mayer, Alexander Bornik, Joachim Bauer, Konrad F. Karner, et Franz Leberl. Multiresolution texture for photorealistic rendering. Dans *Spring Conference on Computer Graphics*, 2001.

[10] Xiaoguang Wang, Stefano Totaro, Franck Taillandier, Allen Hanson, et Seth Teller. Recovering facade texture and microstructure from real-world images. Dans *European Conference on Computer Vision*, 2002.

[11] Diego Ortin et Fabio Remondino. Occlusion-free image generation for realistic texture mapping. Dans *3D-ARCH*

2005 : Virtual Reconstruction and Visualization of Complex Architectures, 2005.

[12] Thommen Korah et Christopher Rasmussen. Improving spatiotemporal inpainting with layer appearance models. Dans *International Symposium on Visual Computing*, 2006.

[13] Pierre Poulin, Mathieu Ouimet, et Marie-Claude Frasson. Interactively modeling with photogrammetry. Dans *Proceedings of Eurographics Workshop on Rendering 98*, pages 93–104, 1998.

[14] J.Y. Rau, T.A. Teo, L.C. Chen, F. Tsai, K.H. Hsiao, et W.C. Hsu. Integration of gps, gis and photogrammetry for texture mapping in photo-realistic city modeling. Dans *Pacific-Rim Symposium on Image and Video Technology*, pages 1283–1292, 2006.

[15] Thomas Colleu, Gael Sourimant, et Luce Morin. Automatic initialization for the registration of gis and video data. Dans *3DTV 2008*, Istambul, Turkey, 2008.

[16] G. Sourimant, L. Morin, et Kadi Bouatouch. Gps, gis and video registration for building reconstruction. Dans *ICIP 2007, 14th IEEE International Conference on Image Processing*, San Antonio, USA, 2007.

[17] Philip David, Daniel Dementhon, Ramani Duraiswami, et Hanan Samet. Softposit : Simultaneous pose and correspondence determination. *Int. J. Comput. Vision*, 59(3) :259–284, 2004.

[18] A.I. Comport. *Robust real-time 3D tracking of rigid and articulated objects for augmented reality and robotics*. Thèse de doctorat, Université de Rennes 1, Mention informatique, September 2005.

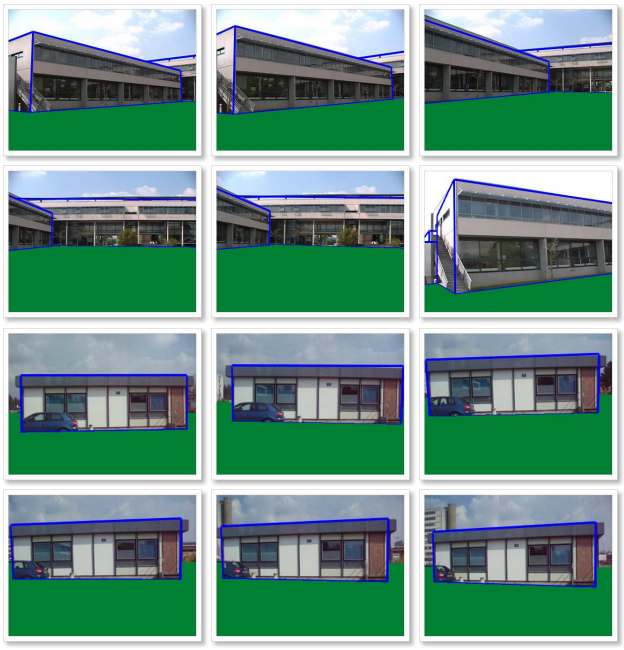

Figure 4 – *Résultats de suivi pour deux séquences réelles, avec superposition du modèle estimé du sol (en vert)*

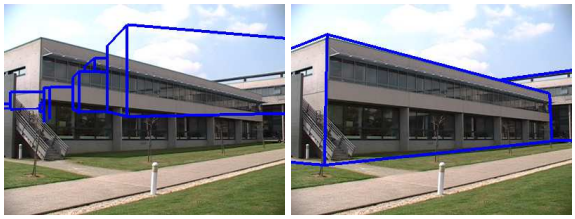

Figure 5 – *Recalage initial automatique : initialisation arbitraire de la pose (gauche) et résultat de recalage obtenu (droite)*

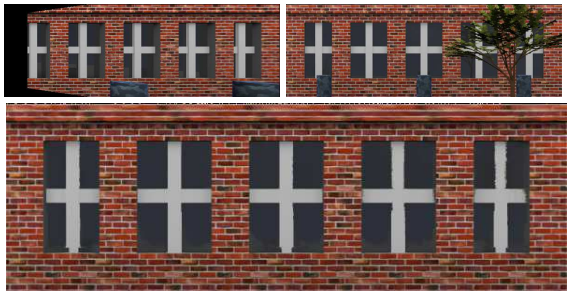

Figure 6 – *Première (haut gauche) et* 50^e *texture (haut droite) d'une séquence de synthèse, et texture reconstruite (bas)*

Figure 7 – *Comparaison entre vue réelle (gauche) et vue avec modèle texturé superposé (droite) pour des images réelles*

Figure 8 – *Rendu d'une vue virtuelle de la base SIG texturée*