

Vers une reconstruction 3D de zones urbaines : mise en correspondance de données Gps, Sig et Vidéo

Gaël SOURIMANT, Luce MORIN, Kadi BOUATOUCH

Irisa / Inria, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France

gael.sourimant@irisa.fr, luce.morin@irisa.fr, kadi.bouatouch@irisa.fr

Résumé – La modélisation en 3D d’environnements urbains est un sujet largement étudié depuis plusieurs années, son attrait étant lié aux applications diverses d’une telle modélisation : navigation virtuelle, réalité augmentée, planification architecturale, etc. L’une des difficultés à ce jour dans ce contexte reste l’acquisition et le traitement de données à grande échelle si l’on cherche à obtenir une reconstruction précise non seulement géométriquement, mais également photométriquement (on veut les véritables textures de chaque bâtiment). Nous présentons dans cet article un système permettant de calculer les positions géo-référencées et les orientations d’images de bâtiments issues de séquences vidéo non calibrées, en tant que préalable indispensable au bon conditionnement de la reconstruction 3D précise d’environnements urbains à grande échelle, notre méthode étant basée sur la fusion de données multimodales, et plus précisément de positions GPS, de modèles 3D polyédriques simples de bâtiments ainsi que de séquences d’images de ces bâtiments.

Abstract – 3D reconstruction of urban environments is a widely studied subject since several years, since it can lead to many useful applications: virtual navigation, augmented reality, architectural planification, etc. One of the most difficult problem nowadays in this context is acquisition and treatment of data if very large scale and precise reconstruction is aimed. In this paper we present a system for computing geo-referenced positions and orientations if images of buildings from non calibrated videos. Providing such information is a mandatory step to well conditioned large scale and precise 3D reconstruction of urban areas. Our method is based on the fusion of multimodal datasets, namely GPS measures, video sequences and rough 3D models of buildings.

1 Introduction

Le succès récent de Google Earth montre que l’ajout de textures photo-réalistes sur une carte 2D ajoute beaucoup d’information pour l’utilisateur par rapport à une carte symbolique traditionnelle. Les fonctionnalités 3D offertes par cet outil, telles que la navigation ou la représentation en 3D des bâtiments est une autre raison de son succès. Cependant, les modèles 3D fournis sont peu réalistes (ce sont des parallélépipèdes gris). Il serait intéressant de pouvoir combiner modèles des 3D photo-réalistes avec des cartes 2D texturés avec des images aériennes. La modélisation 3D d’environnements urbains a d’autres applications, telles que les jeux, le tourisme virtuel, le géo-positionnement ou la réalité virtuelle. Malheureusement, la modélisation manuelle par un graphiste est un processus long qui ne peut être appliqué à la modélisation à grande échelle d’environnements urbains.

Nous présentons un système permettant de calculer des positions et orientations géo-référencées d’images de bâtiments. Notre approche est basée sur la fusion de données multimodales, à savoir des images prises au sol de bâtiments acquises avec des mesures GPS, ainsi qu’une base de donnée de type SIG composée d’un ensemble de modèles 3D de bâtiments décrits par leur empreinte au sol et leur élévation. Si ce type de modélisation est pertinent pour une visualisation aérienne, il n’est pas satisfaisant pour une navigation 3D au niveau du sol. La vidéo et la base SIG contiennent des informations complémentaires : la vidéo fournit le photo-réalisme et les détails géométriques des bâtiments, tandis que les modèles SIG donnent une géométrie "propre" et complète de la scène, structurée en bâtiments individuels. Des mesures GPS sont également ac-

quises de façon synchronisée avec la vidéo. Afin de combiner ces types de données différents, la première étape est de les mettre en correspondance dans le même système de coordonnées. La mise en correspondance est de fait le point sensible du système, étant donnée qu’elle requiert des correspondances géométriques entre des données de type tout à fait différent.

Au cours des dernières années, de nombreuses méthodes pour la reconstruction de zones urbaines ont été développées, mais peu d’entre elles traitent à la fois le problème d’une reconstruction photoréaliste contenant les détails géométriques des façades *et* le problème d’une reconstruction à grande échelle. Dans le projet *MIT City Scanning* [8], des images hémisphériques calibrées sont utilisées pour extraire des plans correspondant aux façades, qui sont alors raffinés et texturés en utilisant des techniques de reconnaissance des formes et de vision par ordinateur. Dans le projet *UrbanScape* [1], un système complètement automatique pour une reconstruction temps-réel et précise à partir de flux vidéos est présenté, en utilisant à la fois le CPU et le GPU. Le projet 4D *Cities* [6] cherche à créer des modèles 3D variant avec le temps à partir d’une collection d’images prises d’endroit différents, à des époques également différentes. Dans le cadre du projet *Fast 3D City Model Generation* [2], le centre-ville de Berkeley est reconstruit précisément en utilisant une caméra laser verticale pour mesurer la structure de bâtiments, une caméra laser horizontale pour le calcul de pose, et une caméra vidéo pour texturer les modèles obtenus, le tout étant monté et synchronisé sur un véhicule. La principale limitation de ces approches est qu’elles nécessitent des équipements très spécifiques et lourds pour l’acquisition des données.

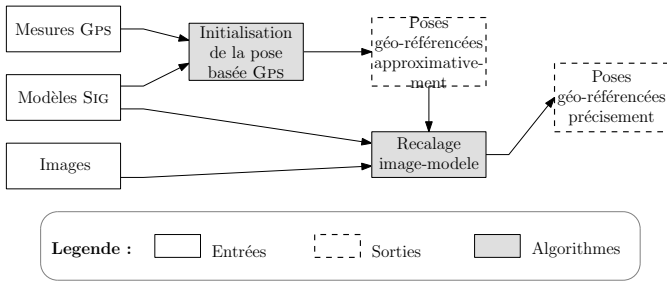


FIG. 1 – Principe du recalage.

Nous résolvons ce problème avec une approche de type raffinement. Nous partons de modèles 3D existants qui sont simples géométriquement et non texturés, auxquels nous ajoutons graduellement de l'information géométrique et photométrique en utilisant des données extraites de vidéos, avec des algorithmes issus de la robotique et de la vision. La figure 1 schématise les différentes étapes de notre algorithme, qui sont explicitées dans les sections 2 et 3.

2 Recalage initial de la caméra

Nous utilisons en entrée de notre méthode de reconstruction de zones urbaines une base de données SIG qui fournit les modèles 3D bruts de bâtiments géo-référencés, et des vidéos desquelles sont extraites des images RVB pour le texturage ainsi que des images de luminance pour l'extraction et le suivi de points. Nous utilisons également des mesures GPS qui sont enregistrées simultanément avec le flux vidéo, et qui fournissent une première approximation pour la géo-localisation des différentes images. Le but de l'approche est de trouver, pour chaque image I de la vidéo, la pose $[\mathbf{R}|\mathbf{t}]$ (orientation et translation) de la caméra pour laquelle le modèle 3D SIG se projette en s'alignant exactement sur les images extraites de la vidéo.

La première étape de ce recalage consiste à trouver une pose $[\mathbf{R}|\mathbf{t}]$ approximative de la caméra par rapport au SIG, en utilisant uniquement les mesures de position données par GPS, acquises en même temps que les flux vidéo. Aucune donnée d'orientation n'étant disponible à l'acquisition, si gps_{t1} et gps_{t2} sont deux mesures de position GPS temporellement successives, l'orientation de la caméra pour l'image I_{t1} est initialisée au vecteur $(gps_{t2} - gps_{t1})$. L'orientation \mathbf{R} étant initialisée pour chaque image, il reste la translation \mathbf{t} à estimer à partir des mesures GPS. Celles-ci étant données en latitude / longitude / altitude, elles sont tout d'abord converties dans le repère UTM pour être exprimées dans le même repère que les modèles 3D SIG. La précision de ces mesures est de 5 mètres dans 95% des cas au niveau du sol, mais l'estimation de l'altitude est beaucoup moins précise. Pour chaque image I , on initialise donc les paramètres t_x et t_y de la caméra directement avec les mesures GPS. L'altitude t_z est calculée quant à elle en positionnant la caméra à une hauteur arbitraire (1,5 mètres) au dessus d'une estimation du sol autour des bâtiments. Cette estimation, un maillage 3D, est calculée en utilisant les points de l'empreinte au sol des bâtiments comme nœuds d'une triangulation de Delaunay.

Nous avons calculé à ce point une pose initiale pour chaque image de la caméra à partir des mesures GPS. Cependant cette

pose est trop approximative pour permettre au modèles 3D de se projeter exactement sur leurs images respectives. Nous allons donc essayer de raffiner cette pose pour toutes les images de la vidéo.

3 Raffinement et suivi de la pose

La trajectoire grossière obtenue par les données GPS est raffinée à l'aide des données vidéo. Le recalage entre données vidéo et modèle SIG consiste à déterminer les paramètres caméra qui permettent de superposer le modèle SIG avec les images des bâtiments présents dans la vidéo. Pour la première image de la séquence, le recalage est réalisé à l'aide d'une procédure semi-automatique. Pour les images suivantes, le recalage est réalisé automatiquement par extraction et suivi de points d'intérêt et par asservissement visuel virtuel.

3.1 Fondements théoriques.

L'alignement entre le projeté d'un objet 3D et l'image de ce même objet a été étudié dans le domaine de la vision par ordinateur et de la robotique. Marchand a proposé une méthode de recalage basée sur l'asservissement visuel [4], qui consiste à estimer la pose de l'objet observé dans l'image, en modifiant la pose d'une caméra virtuelle observant le modèle 3D. Dans le cas général, l'estimation de pose peut être considérée comme un problème d'estimation non-linéaire, faisant intervenir un ensemble de primitives 3D et leur projections 2D dans le plan image. L'objectif est de minimiser l'erreur de projection entre les données observées \mathbf{s}^* dans l'image et la position de ces mêmes données \mathbf{s} , calculée par projection sur le plan image des primitives 3D correspondantes. La matrice de pose ${}^c\mathbf{M}_o$ est calculée itérativement en utilisant la loi de commande :

$$\mathbf{v} = -\lambda(\mathbf{L}_s)^+(\mathbf{s}({}^c\mathbf{M}_o) - \mathbf{s}^*) \quad (1)$$

où \mathbf{v} est un vecteur définissant la pose et fonction de \mathbf{R} et \mathbf{t} , λ est un scalaire et \mathbf{L}_s est le Jacobien de la fonction à minimiser. Cette méthode est générique par rapport au type des primitives utilisées (points, lignes, ellipse, ...), à condition que l'erreur puisse être calculée à partir des données image.

Dans notre cas, les primitives utilisées sont un ensemble de points d'intérêt. \mathbf{s}^* représente alors un ensemble de points 2D \mathbf{p}_i , et \mathbf{s} est l'ensemble des points 3D correspondants \mathbf{P}_i projetés dans l'image pour une pose donnée ${}^c\mathbf{M}_o$. De fait, si nous pouvons produire un ensemble de correspondance entre points 2D dans l'image courante et points 3D du modèle de la base SIG, alors il est possible d'estimer la pose de la caméra pour cette image, exprimée dans le repère du SIG.

La précision de la pose estimée par asservissement visuel virtuel est très sensible aux erreurs introduites par l'extraction des primitives (bruit dans les images, variations d'illumination, occultations, position 3D des points du modèle, ...). L'introduction d'une estimation robuste au niveau de la loi de commande, sous la forme d'un M-estimateur, permet de quantifier la confiance associée à chaque information géométrique utilisée. La nouvelle loi de commande s'écrit alors :

$$\mathbf{v} = -\lambda(\mathbf{D}\mathbf{L}_s)^+\mathbf{D}(\mathbf{s}({}^c\mathbf{M}_o) - \mathbf{s}^*) \quad (2)$$

où $\mathbf{D} = \text{diag}(w_1, w_2, \dots, w_N)$ est une matrice diagonale contenant les poids w_i correspondant à la mesure de confiance asso-

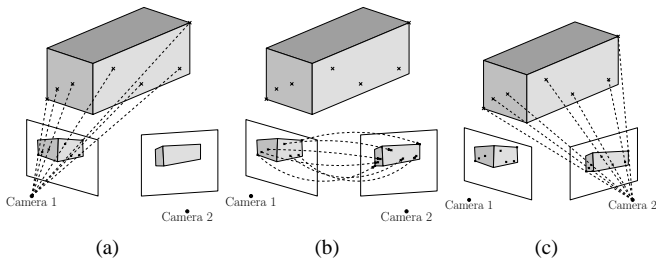


FIG. 2 – Suivi de la pose au long de la vidéo

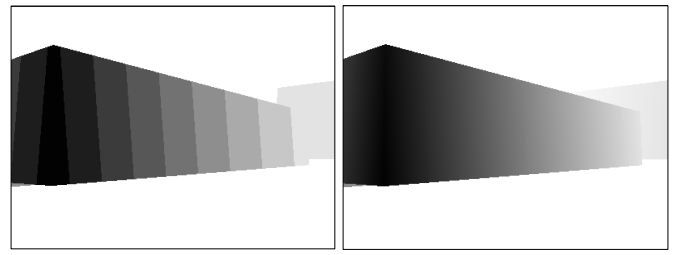
ciée à chaque information visuelle. Ils sont calculés par la fonction robuste de Cauchy. Pour assurer qu'un nombre suffisant de primitives n'est pas rejeté par l'estimateur robuste, on vérifie que la matrice DL_s est de rang plein, (*i.e.* rang 6 puisque la pose a 6 degrés de liberté : 3 pour la position et 3 pour l'orientation), grâce à la décomposition SVD utilisée lors du calcul de la pseudo-inverse $(DL_s)^+$.

3.2 Calcul de la pose pour la 1^{ère} image.

Le SIG est tout d'abord recalé avec la première image de la séquence de façon semi-automatique. A ce point, seules une position et une orientation approximatives de la caméra sont connues pour la première image. L'utilisateur corrige tout d'abord ces valeurs grâce à une interface OpenGL, qui affiche à la fois l'image et la projection du modèle SIG des bâtiments visualisés. Les seuls points 3D qui peuvent être extraits de manière fiable des modèles SIG, pour calculer la pose, sont les coins des bâtiments (*i.e.* les points au niveau du sol et du toit qui appartiennent à l'empreinte des bâtiments). Ceux qui sont visibles sont automatiquement détectés en utilisant une procédure de *color coding*. Ceux qui se projettent en dehors de l'image ou qui sont occultés par une autre façade sont éliminés automatiquement. Pour chaque point 3D X_i sélectionné, l'interface affiche un marqueur dans le modèle SIG, et attend que l'utilisateur fournisse en cliquant sur l'image son correspondant 2D x_i . Une fois que toutes les correspondances 2D-3D sont effectuées, la pose est calculée automatiquement en utilisant un algorithme d'asservissement visuel virtuel à partir de l'équation 1. Au moins quatre de ces correspondances sont nécessaires pour calculer la pose, le résultat étant plus pertinent dans le cas de points non coplanaires.

3.3 Suivi de la pose.

Une fois que la pose a été calculée pour la première image de la vidéo, recalé le modèle SIG avec les images suivantes devient un problème de suivi, qui est traité ici de façon automatique en utilisant également une approche basée asservissement. Soit I_t une image pour laquelle le recalage avec le modèle SIG est effectué, et I_{t+1} l'image suivante pour laquelle nous cherchons à calculer la pose. Comme pour le recalage de la première image, nous avons besoin pour cette image I_{t+1} de correspondances 2D-3D. Celles-ci sont effectuées en se basant sur un schéma de *transfert de points* qui utilise les données extraites de I_t . La procédure complète est illustrée sur la figure 2. Tout d'abord, des points 2D sont extraits de l'image I_t . Pour l'extraction et le suivi de points d'intérêt, nous utilisons une



(a) $\pi_n = 1$ et $\pi_f = 75$

(b) $\pi_n = 20$ et $\pi_f = 75$

FIG. 3 – Influence des plans de clipping sur l'estimation de la profondeur

implémentation du tracker de Kanade-Lucas-Tomasi (KLT)¹. Etant donné que tous les points extraits n'appartiennent pas à un bâtiment, ils sont classifiés en points leur appartenant ou non. Aucune estimation explicite de la profondeur des points 2D extraits n'est effectuée pour savoir s'ils intersectent ou non le modèle SIG. Celle qui leur est assignée est celle du tampon de profondeur, qui a déjà été calculée par OpenGL pour afficher le modèle 3D recalé avec l'image I_t (voir figure 2(a)). Si la valeur assignée est nulle alors le point est classifié comme n'appartenant pas à une façade, et inversement. Nous avons donc à ce point des correspondances 2D-3D pour l'image I_t , *qui est déjà recalée avec le modèle*. On voit d'ailleurs que l'on n'est plus limité à l'utilisation des coins de bâtiments pour avoir une information 3D, puisque que la correspondance image-modèle donne potentiellement une information de profondeur pour tout pixel se situant à l'intérieur de la projection du modèle.

Dans l'optique de mieux conditionner le suivi, et puisque la majorité des points appartiennent souvent à une unique façade, le modèle estimé du sol est également utilisé pour introduire de nouvelles correspondances de primitives qui sont situées globalement sur un plan orthogonal aux plans de façades. De plus, il faut prendre en compte le fonctionnement du tampon de profondeur lors l'estimation des points 3D pour que leur mesure soit suffisamment précise. Dans notre cas, peu de précision est allouée pour les façades si l'on utilise des valeurs génériques pour les plans de clipping. Nous laissons alors le soin à l'utilisateur de définir la distance au plan de clipping lointain (π_f), mais celle du plan de clipping proche (π_n) est déplacée automatiquement à la valeur correspondant au point de bâtiment visible le plus proche de la caméra. Une comparaison des valeurs stockées dans le tampon est représentée sur la figure 3, pour une valeur fixe de π_f et différentes valeurs de π_n . Comme on peut le voir, plus π_n est éloigné de la caméra, plus la précision allouée aux points 3D des façades sera grande (le point le plus proche du bâtiment est situé à environ 23 mètres sur la figure).

En utilisant le KLT, on suit les points d'intérêt entre les images I_t et I_{t+1} (voir figure 2(b)). Si \mathbf{x}_t représente l'ensemble des points 2D extraits de I_t et \mathbf{X} leur position 3D correspondante, puisque l'on connaît des correspondances entre \mathbf{x}_t et \mathbf{x}_{t+1} on peut calculer des correspondances 2D-3D pour I_{t+1} , entre \mathbf{x}_{t+1} et \mathbf{X} . En les utilisant dans l'équation 2 on peut alors calculer la pose de la caméra pour I_{t+1} (figure 2(c)). Cependant, le tracker KLT perd des points au cours du recalage. Pour faire face à ce problème, on introduit une mesure sur le nombre de points perdus. Si à un instant t on estime avoir perdu trop de points (typi-

¹<http://www.ces.clemson.edu/~stb/klt/>

quement 60%), on extrait de nouveaux points d'intérêt en lisant leur profondeur de nouveau dans le tampon de profondeur en utilisant la dernière image recalée (I_{t-1}). On garde cependant les points que l'on avait pas perdu, et on contraint les nouveaux points à être suffisamment distants dans l'image des anciens.

4 Résultats

Nous présentons dans cette section des expérimentations de notre méthode sur plusieurs façades de bâtiments. Après avoir donné quelques détails sur le calibrage de la caméra, des résultats de recalage sont présentés pour deux séquences de test. Les résultats présentés ont été obtenus sur un Pentium IV cadencé à 2.5 GHz avec 512 Mo de RAM, et en utilisant une carte graphique nVidia Quadro2 EX pour le rendu.

Calibrage de la caméra. Dans notre contexte, et grâce au rapport entre la taille des pixels et celle des objets visualisés, nous n'avons pas besoin d'un calibrage extrêmement précis de la caméra (voir également [3]). Le centre de projection est initialisé à $[0\ 0]^T$, et pour la focale nous pouvons utiliser celle donnée par les paramètres constructeur ou des données EXIF² contenues dans les images, comme dans [7].

Résultats de recalage. La séquence de test présentée dans cette section est composée d'images basse résolution (400×300 pixels). Elle a été acquise avec une caméra vidéo numérique du commerce, et contient 650 images où l'on voit plusieurs façades. Le mouvement de la caméra est générique, et ne vise à suivre aucune façade en particulier, ce qui rend le tracking d'autant plus difficile. Des résultats de recalage pour d'autres séquences sont disponibles en ligne³. Deux résultats de suivi sont présentés. Tout d'abord une version simple du tracker a été utilisée (on parlera de version *non robuste* par la suite). Seuls les points de façade extractibles sont pris en compte, aucune optimisation du z-buffer n'est calculée, et la version originale de la loi de commande (équation 1) est utilisée. Bien que cet algorithme donne de bons résultats quand la caméra reste pointée sur l'objet à suivre, une dérive importante apparaît quand cet objet est seulement partiellement visible, disparaît dans quelques images, ou quand par exemple de nombreuses spécularités sont présentes dans la scène. Nous présentons donc des résultats de suivi utilisant la version *robuste* du tracker présentée dans la section 3. Une fois que les correspondances sont données pour la première image, le calcul de la pose est effectué en approximativement 0,2 secondes. Un rendu du modèle SIG en surimpression avec les images correspondantes est illustré sur la figure 4 (le modèle 3D estimé du sol est représenté en gris transparent). Le suivi est calculé en 171 secondes pour la version non robuste, et en 302 secondes pour la version robuste. On peut noter que les différentes améliorations apportées rendent le suivi beaucoup moins sensible à la dérive que dans la version classique (non robuste) de l'algorithme d'asservissement visuel. On notera toutefois que, bien que grandement diminuée, une dérive dans l'estimation de la pose est encore légèrement visible, et doit être supprimée.

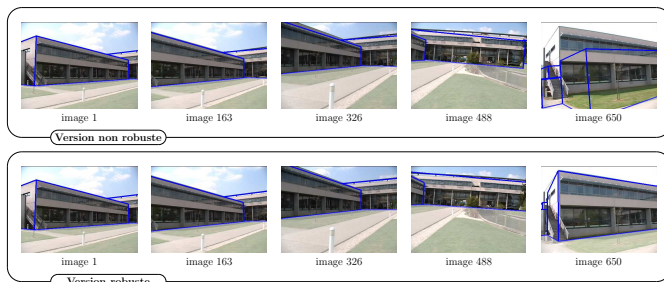


FIG. 4 – Résultats de suivi avec modèle 3D en surimpression

5 Conclusion

Nous avons présenté dans cet article une méthodologie permettant de mettre en correspondance différents types de données, en tant qu'étape obligatoire à une reconstruction à grande échelle de modèles urbains, en interprétant des mesures GPS par rapport à une base de données SIG pour donner une première approximation de la position de la caméra ayant acquis des données au sol, puis en raffinant l'estimée de la pose de cette caméra en utilisant des algorithmes d'asservissement visuel virtuel appropriés. Nous calculons alors une position et une orientation géo-référencées pour chaque image de la vidéo. Dans un futur proche, nous envisageons d'exploiter ces images géo-référencées avec les modèles SIG pour raffiner leur géométrie et leurs textures.

Cependant, des améliorations peuvent encore être apportées à cette méthode. Nous voudrions tout d'abord supprimer la partie manuelle du processus pour la première image, en développant une procédure automatique qui fasse ce premier recalage. De plus, avec une telle procédure, nous pourrions réduire la dérive lors du processus de recalage en l'appelant en fonction de critères d'erreurs à définir. De tels travaux sont actuellement en cours d'étude. Ils se basent sur une estimation de la pose de la première caméra par *ego-motion*, puis une mise en correspondance de lignes extraites des images avec la projection de segments issus du SIG. Il sera alors intéressant de comparer ces travaux avec ceux de Drummond et al. [5].

Références

- [1] A. Akbarzadeh et al. Towards urban 3d reconstruction from video. In *3DPVT*, 2006.
- [2] A. Frueh and A. Zakhor. Data processing algorithms for generating textured 3d building facade meshes from laser scans and camera images. *IJCV*, 2005.
- [3] J.-F. Viguera Gomez, G. Simon, and M.-O. Berger. Calibration errors in augmented reality : A practical study. In *IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2005.
- [4] E. Marchand. *Commande d'une caméra réelle ou virtuelle dans des mondes réels ou virtuels*. Habilitation à diriger les recherches, Université de Rennes 1, Mention informatique, 2004.
- [5] T.W. Reitmayr, G. ; Drummond. Going out : robust model-based tracking for outdoor augmented reality. *IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2006.
- [6] G. Schindler, P. Krishnamurthy, and F. Dellaert. Line-based structure from motion for urban environments. In *3DPVT*, 2006.
- [7] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism : exploring photo collections in 3d. In *ACM SIGGRAPH 2006*, 2006.
- [8] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master. Calibrated, registered images of an extended urban area. *Int. J. Comput. Vision*, 2003.

²Exchangeable Image File Format

³<http://www.irisa.fr/temics/staff/sourimant/tracking>