Recalage de vidéo et de modèle 3D SIG

Bibliographie de Master 2 Recherche Informatique
Université de Rennes 1

Vincent JANTET

Janvier 2008

Résumé

Ce rapport se situe dans le domaine de la génération de modèle 3D d'environnement urbain. Il expose quelques solutions présentées dans l'état de l'art pour résoudre un problème de recalage initial entre une vidéo et un modèle SIG. Le fil conducteur est le rapport de stage de T. Colleu [1, 2] dont les deux grandes étapes sont détaillées précisément. La première étape permet un recalage approximatif, qui permet d'une part de limiter le modèle SIG à la zone restreinte visible dans la vidéo, et d'autre part à servir de point de départ aux algorithmes de convergence [4] utilisés par la suite. Cette étape est faite par l'utilisation de données GPS, combinée à une utilisation de la théorie épipolaire. La seconde partie s'intéresse à extraire de l'image des lignes marquant les contours, et à les faire correspondre aux arêtes issues du modèle. Cette mise en correspondance utilise une estimation du contexte géométrique couplée à un algorithme d'estimateur robuste (RANSAC). Un calcul de pose précis est enfin effectué sur les *inliers* détectés par RANSAC. Ces méthodes fournissent des résultats variables, et parfois non satisfaisants. La dernière partie sera donc consacrée à d'autres solutions issues de la littérature. La plupart sont des méthodes de reconstruction 3D à partir d'une vidéo, mais certaines idées clés peuvent être utilisées dans le cadre du recalage.

1 Introduction

Générer un modèle 3D photo-réaliste d'une zone urbaine est un travail fastidieux s'il est fait à la main. Cependant, de nombreuses applications comme les visites virtuelles ou la cartographie, en sont friandes. En effet, le modèle 3D offre à l'utilisateur un confort supplémentaire et une immersion plus importante. Il serait intéressant de pouvoir générer ces modèles, rapidement et à moindre coût.

Les données SIG (Acronyme pour "Système d'Information Géographique") sont des données géographiques géo-référencées, qui contiennent entre autre les contours au sol, et la hauteur de chaque bâtiment. Cette base de données permet de représenter les bâtiments sous forme de polyèdres simples comme on peut le voir sur la figure 1, mais ne contient ni les textures des façades, ni leur géométrie précise (portes, fenêtres, ...). L'idée suivie par l'équipe TEMICS à l'IRISA est de se baser sur les données SIG, puis de raffiner le modèle en lui appliquant des textures issues d'une vidéo faite autour des bâtiments. Il faut pour cela avoir recalé la vidéo avec le modèle 3D, c'est à dire qu'il faut calculer la position de la caméra exprimée dans le repère des données SIG, pour chaque image de la séquence. Ainsi, la projection perspective du modèle SIG dans les images doit correspondre aux contours réels des bâtiments. L'acquisition des coordonnées GPS de la caméra permet un premier recalage peu précis, mais qui limite l'espace de recherche lors du calcul de pose plus précis. Une fois le recalage effectué, il est possible de raffiner la géométrie par des informations de relief issues de la vidéo, et de raffiner le modèle par une extraction des textures.

Dans un premier temps, ce document s'intéresse aux travaux de G. Sourimant [3] et de T. Colleu [1] concernant le recalage entre un modèle

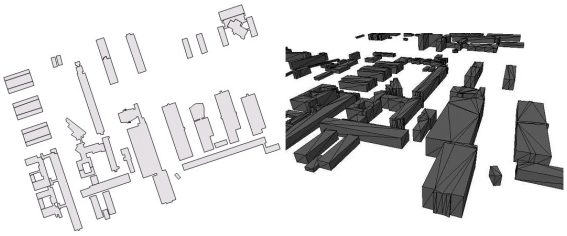

FIG. 1 – Représentation 3D des données SIG.

3D pré-existant, et les images issues d'une vidéo. C'est la phase d'initialisation qui est particulièrement critique pour la suite des traitements, puisque elle n'a aucune connaissance à priori de la pose. Pour la suite de la séquence, la pose estimée à l'instant $t-1$ est une bonne approximation de la pose à l'instant t. Les méthodes décrites se veulent complètement automatisables, et génériques à tout type d'environnement urbain. Dans un second temps, ce document expose trois nouvelles méthodes permettant de résoudre ce problème, principalement issues du rapport de T. Werner et A. Zisserman [10]. La première s'intéresse aux propriétés des points de fuite pour estimer la pose approximative, la seconde opte pour une mise en correspondance des points de contours directement basé sur des descripteurs de Harris, alors que la troisième propose de travailler dans le repère 3D pour faire les mises en correspondance.

Les méthodes qui pourraient être utilisées par la suite pour raffiner le modèle ne sont pas abordées ici, elles peuvent être trouvées dans de nombreuses autres publications [3, 10, 11].

2 Recalage proposé par T. Colleu

Le recalage est en fait un calcul de pose, c'est à dire le calcul des six degrés de liberté de la caméra qui ont permis d'obtenir chaque image. On représente une pose sous la forme d'une matrice $[R \mid t]$ contenant les trois paramètres de la rotation R, et les trois paramètres de translation t, permettant de passer du repère R_C de la caméra au repère R_{UTM} des données SIG.

$$[R \mid t] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} & & & t_x \\ & R & & t_y \\ & & & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Il est alors possible de définir la matrice $P = K[R \mid t]$ représentant la projection sur le plan image, d'un point 3D défini dans le repère SIG. La matrice K représente les paramètres intrinsèques de la caméra. Elle est définie par (u_0, v_0) qui représente la projection dans l'image du centre optique de la caméra, et par les $p_.$ qui sont des paramètres dépendant de la taille des pixels et de la distance focale de la caméra (distance entre le centre optique et le plan image).

$$K = \begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

On suppose ici que la caméra est déjà calibrée, c'est à dire que la matrice K est déjà connue. Si ce n'est pas le cas, il est possible d'effectuer le calibrage à partir de la vidéo, en utilisant les travaux présentés dans [19, 20].

De nombreuses contributions ont été faites dans le domaine du calcul de pose, mais nécessitent pour la plupart une intervention manuelle pour mettre des points en correspondance [12]. Le principe retenu par T. Colleu dans [1, 2] se voulant automatisable, il se décompose en deux étapes :

- La première étape se base sur des idées de [22] pour faire une estimation approximative de la pose. En associant les données GPS avec la vidéo, il est possible d'estimer à la fois la position et l'orientation de la caméra, ce qui permet de limiter le modèle SIG aux seules primitives présentes dans la vidéo.
- La seconde étape utilise l'algorithme RANSAC [9] pour associer les primitives 3D aux contours 2D issues de la première image de la vidéo. Le calcul de la pose précise est ensuite effectué par la méthode décrite par E. Marchand dans [4].

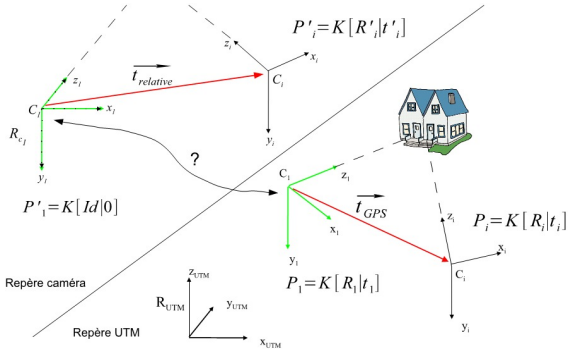

FIG. 2 – Représentation des vecteurs $t_{relatif}$ et t_{GPS}.

2.1 Estimation approximative de la pose

La première étape consiste à estimer la position de la caméra, ainsi que son orientation dans le repère geo-référencé, en minimisant les traitements à faire sur la vidéo. L'idée est de réduire l'espace des poses possibles pour rendre efficace l'algorithme de mise en correspondance effectué à l'étape suivante. Estimer la matrice de projection P_1 correspondant à la première image est le point important, puisque les poses dans les autres images en découlent.

Les données GPS à l'instant initial de la vidéo sont converties dans le repère UTM pour donner une approximation à 5 m près de la position de la caméra t_1 dans le repère R_{UTM}. Pour estimer l'orientation de la caméra, c'est à dire la matrice de rotation R_1, l'idée est de faire coïncider le vecteur de déplacement extrait des données GPS, avec le vecteur de déplacement extrait de deux images de la séquence.

Entre l'instant initial et un autre instant i plus loin dans la séquence, les données GPS fournissent directement le déplacement t_{GPS} de la caméra dans le repère UTM. De ces mêmes instants-clés, on peux extraire le déplacement $t_{relatif}$ de la caméra dans le repère C_1 par une approche de type SFM (Structure from Motion) [13] comme détaillée dans le paragraphe suivant. L'alignement du vecteur t_{GPS} avec le vecteur $t_{relatif}$ est expliqué dans le paragraphe suivant et permet de trouver l'orientation de la caméra dans le repère R_{UTM}.

2.1.1 Extraction de $t_{relatif}$

Sur deux images de la vidéo, des points sont mis en correspondance par la méthodes de suivi de Kanade, Lucas et Tomasi (KLT) [21]. A partir de ces correspondances entre les points m_j de l'image 1 et m'_j de l'image i, il est possible de retrouver la géométrie épipolaire, c'est à dire la matrice fondamentale F qui vérifie les équations :

$$\forall j : {m'_j}^T * F * m_j = 0$$

L'estimation de cette matrice F nécessite au moins sept couples de points en corrélation. A cause des erreurs de suivi et du bruit dans l'image, une estimation robuste est nécessaire, et est faite par une méthode basée RANSAC (RANdom SAmple Consensus) comme expliqué dans [13]. L'idée est de prendre sept couples de points aléatoirement dans l'ensemble des couples mis en corrélation, et de s'en servir pour calculer la matrice F. Si un des couples utilisés est aberrant, la matrice F ainsi calculée ne correspondra pas à la réalité, et elle ne sera pas adaptée pour décrire la plupart des autres couples de points de l'échantillon. On attribue donc une note à la matrice F, en fonction du nombre de couples de l'échantillon qui font partie de l'*ensemble de consensus*, c'est à dire qui sont proches du modèle estimé par F (avec une marge d'erreur ϵ). Si la note est trop basse (il faut définir un seuil), alors la matrice F calculée est considérée comme fausse, et on recommence avec un nouvel ensemble de couples aléatoires. Statistiquement, on finira par tirer un ensemble de couples ne contenant pas d'élément aberrant, et on finira par trouver une matrice F proche de la réalité. On pourra alors utiliser cette matrice F qui maximise l'*ensemble de consensus* pour supprimer les points aberrants (les *outliers*) qui ne suivent pas le modèle.

Puisque l'on a supposé que la matrice des paramètres intrinsèques K de la caméra était connue, il est possible de calculer la matrice essentielle E qui représente le mouvement relatif de la caméra entre les deux images.

$$E = K^T * F * K$$

Il ne reste plus qu'a décomposer cette matrice $E = [R|t]$ pour retrouver les paramètres extrinsèques $t_{relatif}$ et $R_{relatif}$ de la caméra. Une dé-

composition en valeurs singulières (SVD : Singular Value Decomposition), expliquée dans [13], permet de retrouver la translation $t_{relatif}$ que l'on cherche.

2.1.2 Alignement de t_{GPS} et $t_{relatif}$

Les vecteurs de déplacement t_{GPS} et $t_{relatif}$ sont dûs au même déplacement physique de la caméra, mais ils ne sont pas exprimés dans le même repère. Avec la notation ${}^{R_2}R_{R_1}$ pour la rotation entre le repère R_1 et R_2, on peut définir le point C_i dans le repère R_{UTM} de deux façons différentes :

$$\begin{aligned} {C_i}_{R_{UTM}} &= {C_1}_{R_{UTM}} + t_{GPS} \\ &= {}^{R_{UTM}}R_{R_{C_1}} * t_{relatif} + {C_1}_{R_{UTM}} \end{aligned}$$

On a donc :

$$t_{relatif} = {}^{R_{UTM}}R_{R_{C_1}}{}^{-1} * t_{GPS}$$

Mathématiquement, cette équation pourrait être utilisée pour retrouver deux des trois degrés de liberté de la rotation ${}^{R_{C_1}}R_{R_{UTM}}$, mais il n'est pas possible d'estimer les trois DL. Pour y remédier, la solution employée dans [1, 2] consiste à supposer que la caméra est portée "normalement", c'est à dire qu'elle est orientée dans le plan du sol (ni pointée vers le ciel, ni pointée vers le sol) et qu'elle est tenue droite (pas inclinée). Cela peut se traduire par la colinéarité des axes verticaux dans le repère caméra et le repère UTM. Il n'y a alors plus qu'un seul paramètre à estimer, c'est θ, l'angle de la caméra dans la rotation d'axe vertical. Pour cela, on définit t'_{GPS} et $t'_{relatif}$ comme les vecteurs t_{GPS} et $t_{Relatif}$ après projection dans le plan du sol, et normalisation.

$$t'_{GPS} = \begin{bmatrix} t'_{GPSx} \\ t'_{GPSy} \end{bmatrix} \quad t'_{relatif} = \begin{bmatrix} t'_{rx} \\ t'_{rz} \end{bmatrix}$$

L'angle qu'ils forment entre eux (calculé par un produit scalaire) correspond à l'angle θ.

$$\theta = acos(t'_{rx} * t'_{GPSx} + t'_{rz} * t'_{GPSy})$$

Le signe de cette rotation est déterminé par celui du produit vectoriel $t_{GPS} \otimes t_{Relatif}$.

Le résultat de ce recalage approximatif est représenté sur la figure 3. On remarque que le modèle ne se superpose pas parfaitement à l'image. Dans la pratique, l'erreur d'alignement réalisée par cette méthode est suffisamment faible pour que des primitives du modèle SIG projeté, puissent être retrouvées dans l'image. Le calcul de pose plus précis, permettant de superposer les contours de l'image et les arêtes du modèle SIG, fait l'objet de la partie suivante.

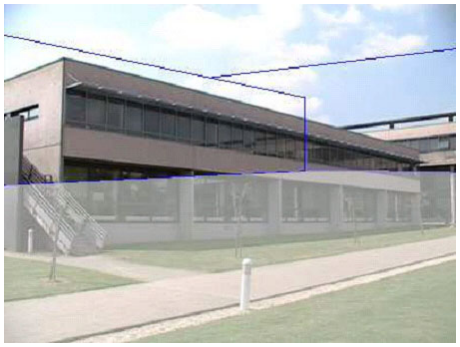

FIG. 3 – Recalage approximatif effectué par identifications du déplacement dans l'image, et du déplacement GPS.

2.2 Recalage précis

A l'aide de la pose estimée dans la partie précédente, il est possible de déterminer quels sont les bâtiments du modèle SIG qui sont visibles dans l'image. Afin d'affiner le calcul de pose, la méthode proposée par T. Colleu [1, 2] se base sur les arêtes des bâtiments, qui ont l'avantage d'être plus stables que les points. Elles sont en particulier moins sensibles aux occlusions, et ne sont pas produites par du bruit. Une correspondance est ensuite faite entre les arêtes SIG projetées dans l'image, et les contours détectés dans l'image, pour pouvoir calculer la pose précise.

Le recalage précis s'effectue donc en trois étapes :

- La première étape consiste à extraire de l'image les contours rectilignes en utilisant un algorithme de Canny [5], suivi d'un algorithme de Hough probabiliste [7].
- La deuxième étape se base sur une estimation du contexte géométrique [8] pour filtrer ces contours rectilignes, et ne garder que ceux correspondant probablement à des arêtes de bâtiments.

– La troisième étape recherche des correspondances parmi ces primitives et le modèle SIG par une approche basée RANSAC, et calcule la pose précise par une méthode de convergence décrite par E. Marchand dans [4].

Ces trois étapes sont décrites en détail dans les paragraphes suivants.

2.2.1 Détection des contours rectilignes

Le principe est d'extraire de l'image les contours, et de chercher ensuite des alignements dans ces contours.

Détecter les contours dans une image revient à chercher les pixels dont le gradient est élevé. Il existe plusieurs algorithmes de détection de contours, et celui qui a été retenu est l'algorithme de Canny [6] qui est un des plus classiques. Son exécution se fait en trois temps :

– Dans un premier temps l'image est lissée par une convolution avec un filtre gaussien. L'image ainsi floutée contient moins de contours provenant simplement du bruit.
– Dans un second temps, le gradient est calculé en chaque pixel, en appliquant à l'image les filtres de Sobel. (*En plus d'obtenir l'intensité du gradient dans les deux directions de l'image, il est possible d'obtenir l'orientation du contour, mais cette valeur ne nous intéresse pas pour la suite*). Seul les maxima locaux du gradient peuvent représenter des contours, les non-maxima sont donc directement éliminés.
– Finalement, la différentiation des contours se fait par un seuillage à hysteresis de l'intensité du gradient. C'est à dire que les gradients de valeurs supérieures à un seuil haut sont considérés comme des contours, ceux inférieurs à un seuil bas sont éliminés, et ceux entre les deux seuils ne sont conservés que s'ils sont connectés à un point de contour.

Le résultat de l'algorithme est donc une image binaire, dont les pixels notés "Vrai" sont les points de contours.

Une fois les contours détectés, l'utilisation de la transformée de Hough permet de repérer des alignements dans ces contours. Elle consiste à représenter l'ensemble des droites passant par un point comme une sinusoïde dans le repère des coordonnées polaires (θ, ρ) et à chercher des intersections

FIG. 4 – Contours détectés par l'algorithme de Canny.

entre ces courbes. Sans rentrer plus dans le détail, le résultat de l'algorithme est un ensemble de droites qui passent par un grand nombre de points de contour.

FIG. 5 – Contours linéaires détectés par l'algorithme de Hough.

Les deux algorithmes de Canny et de Hough sont implémentés dans la librairie libre OpenCV, et ont donc été directement utilisés.

2.2.2 Labélisation des primitives

Parmi toute les droites détectées comme des contours à l'étape précédente, seul un petit nombre correspond effectivement aux arêtes des bâtiments. Il est donc important de filtrer ces lignes et ainsi limiter les correspondances possibles. La phase de mise en correspondance, expliquée par la suite, en sera largement facilitée en raison de l'élimination de certaines lignes aberrantes. Une méthode ingénieuse est proposée pour effectuer ce tri, et se base sur l'estimation

de contexte proposée dans [8], dont l'implémentation est disponible sur Internet. Cet algorithme s'applique à une image, et classifie chaque pixel suivant s'il appartient au ciel, au sol ou à une paroi verticale. Cet algorithme permet également de détecter les zones poreuses, qui représentent généralement la végétation. Le résultat de cette classification est représenté sur la figure 6.

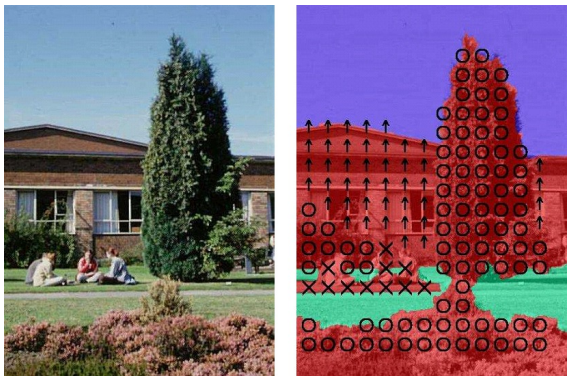

FIG. 6 – Image originale, et après classification par contexte (en bleu le ciel, en vert le sol, en rouge fléché les façades des bâtiments et en rouge marqué d'un autre symbole la végétation et les objets.

A l'aide de cette classification , il devient possible de labéliser les lignes détectées par l'algorithme de Hough en fonction des zones qu'elles délimitent le mieux. Pour cela, on réutilise les algorithmes de détection de contours expliqués précédemment (Canny + Hough) mais cette fois sur la carte représentant chaque classe. On obtient donc une linéarisation des frontières des classes. On calcule ensuite la distance entre chaque ligne de l'image, et les frontières des classes pour déterminer si la ligne représente probablement une arête de ciel (frontière entre une façade et le ciel), une arête de sol (frontière entre une façade et le sol) ou bien une arête verticale (frontière entre deux façades). Les lignes ne se rapprochant d'aucune frontière de classes sont directement supprimées car elles ne représentent probablement rien de géométrique. (*Pour les lignes verticales, il est préférable de ne pas éliminer trop vite celles ne se rapprochant d'aucune frontières de classes, mais de simplement les pondérer par leurs distance aux frontières*). La figure 7 représente les lignes du contour qui ont été conservées après comparaison à la carte des classes, et après labélisation en fonction de ce qu'elles délimitent le plus probablement.

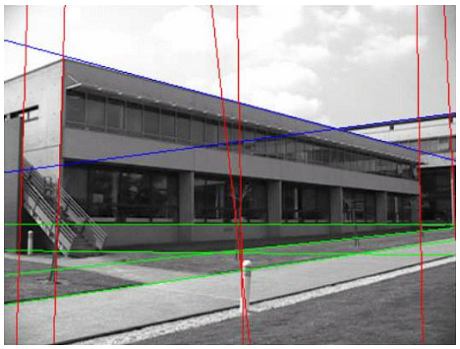

FIG. 7 – Lignes conservées et labélisées : en bleu les arêtes de ciel, en vert les arêtes de sol, et en rouge les arêtes verticales.

Les primitives présentes dans les données SIG qui ont été sélectionnées par l'estimation de la pose, sont projetées dans l'image. Pour pallier au problème des faces masquées, et des occultations partielles, les faces dont la surface après projection n'est pas supérieure à un certain seuil sont éliminées. Les arêtes des faces restantes sont alors labélisées en fonction de ce qu'elles délimitent. Le recalage précis consiste ensuite à mettre en correspondance des lignes de contours et des arêtes, et à calculer la pose permettant une bonne superposition des deux ensembles.

2.2.3 Mise en correspondance

Une fois les contours extraits de l'image et labélisés, ils sont mis en correspondance avec les primitives du modèle SIG par un algorithme basé RANSAC. Un ensemble de quatre couples de lignes est sélectionné aléatoirement, en accordant plus d'importance aux correspondances plus probables (c'est le cas des lignes ciel, et de certaines arêtes verticales). De ces correspondances, il est possible de calculer la pose précise par une approche itérative décrite par E. Marchand dans [4] et détaillée dans la partie suivante (les sélections de lignes en configuration dégénérée sont directement éliminées). Si l'ensemble choisi aléatoirement contenait des données aberrantes, la pose

calculée ne permettrait pas de superposer correctement toutes les primitives du modèle SIG sur des contours de l'image. Pour le savoir, le modèle SIG est de nouveau projeté dans l'image en utilisant la nouvelle pose calculée, et chaque correspondance possible est classifiée en *inlier* ou *outlier* suivant l'erreur d'alignement.

$$\left.\begin{array}{l} \mid \rho_{M3d} - \rho_{Img} \mid < Seuil_{\rho} \\ \mid \theta_{M3d} - \theta_{Img} \mid < Seuil_{\theta} \end{array}\right\} \Leftrightarrow \text{"}Inliers\text{"}$$

Une note est donc donnée à chaque ensemble d'*inliers*, en fonction de son erreur d'alignement globale. L'ensemble le mieux noté est utilisé pour calculer la pose définitive avec le maximum de précision.

Malgré le faible nombre de lignes à mettre en correspondance, cette étape de la méthode est la plus coûteuse en temps, puisque elle fait appel un grand nombre de fois au calcul de pose.

2.2.4 Calcul de pose

Le calcul de pose est un problème non linéaire, très étudié dans le domaine de la vision par ordinateur. En partant d'une pose initiale approximative, et d'un ensemble de primitives en correspondance, le problème revient à calculer le déplacement de la pose qui minimise l'écart au modèle. La solution utilisée est celle expliquée dans [4] qui cherche à faire converger la pose par un asservissement sur sa vélocité (son déplacement). Le déplacement v de la pose est représenté par trois valeurs de translation, et trois valeurs de rotation.

$$v = \begin{bmatrix} t_x & t_y & t_z & r_x & r_y & r_z \end{bmatrix}$$

Ce déplacement est estimé par la loi de commande :

$$v = -\lambda (L_s)^{+} (s(C) - s^{*})$$

La matrice d'interaction L_s est la matrice jacobienne qui s'écrit :

$$L_s = \begin{bmatrix} \lambda_\theta \cos\theta & \lambda_\theta \sin\theta & -\lambda_\theta \rho & -\rho\cos\theta & -\rho\sin\theta & -1 \\ \lambda_\rho \cos\theta & \lambda_\rho \cos\theta & -\lambda_\rho \rho & (1+\rho)^2 \sin\theta & -(1+\rho)^2 \cos\theta & 0 \end{bmatrix}$$

La pose est mise à jour itérativement, en lui appliquant le déplacement v calculé à l'étape précédente, jusqu'à ce que ce déplacement devienne négligeable. En théorie, il y a six paramètres à estimer, ce qui implique que seuls trois couples de droites en correspondance sont nécessaires au calcul de la pose. En pratique, c'est un ensemble de quatre couples qui est utilisé pour éliminer les configurations dégénérées et améliorer le résultat.

2.3 Limites et discussions

Les résultats obtenus par l'association d'un classificateur par contextes géométriques, et d'un algorithme de cherche de correspondances (comme RANDSAC), sont satisfaisants, mais ne sont pas toujours parfaits. On remarque par exemple que le recalage représenté par la figure 8 est incorrect, puisque l'arête au pied du bâtiment s'est recalée avec la frontière de l'ombre. La véritable ligne délimitant le pied du bâtiment n'avait pas été considérée comme importante lors de la détection de contour (Canny et Hough), erreur due à un seuil trop élevé.

FIG. 8 – Résultat final du recalage.

Quelques remarques peuvent être faites sur chacune des deux méthodes.

2.3.1 Recalage approximatif

La méthode d'approximation de la pose décrite précédemment fait la supposition que la caméra est portée "normalement". Cela n'est pas vraiment gênant, puisque c'est la cas dans la plupart des vidéos réelles. Cependant, il est possible de s'affranchir de cette hypothèse, en ne faisant pas coïncider seulement un couple de vecteurs de translation, mais plusieurs couples de vecteurs, définissant deux trajectoires. Si l'on met en relation la trajectoire de la caméra dans le repère R_{C_1}, et celle dans le repère R_{UTM}, il est possible

de définir sans ambiguïté tout les paramètres de pose. Cette amélioration n'aura pas beaucoup d'influence sur la précision de l'alignement (qui est déjà "suffisant"), et demandera beaucoup plus de calculs, puisque il faudra extraire plusieurs $t_{relatif}$. Elle n'a donc pas encore été approfondie.

Une autre particularité de la méthode proposée est de faire intervenir l'image i de la séquence vidéo, qui ne peut pas être choisie complètement au hasard. En effet, une image trop proche de l'image de départ entraîne une grande imprécision sur le calcul de $t_{relatif}$ et t_{GPS}. A l'inverse, une image trop éloignée risque de ne pas contenir assez de points qui peuvent être mis en corrélation avec des points de l'image de départ (occultations, sortie du champ, ...). Le choix de l'instant i est donc primordial, et aucune solution efficace pour le déterminer de façon automatique n'a encore été proposée. Un grand nombre de valeurs de i différentes peuvent être utilisées, c'est pourquoi un choix empirique marche la plupart du temps.

2.4 Recalage précis

Pour effectuer le recalage précis, utiliser les contextes géométriques pour limiter les couples de droites potentiellement en correspondance est une idée innovante, qui n'est que trop rarement exploitée. Cependant, la classification est une des étapes longues de l'exécution du programme, alors que seule une infime partie des résultats qu'elle fournit est utilisée par la suite. En effet, sur les sept classes détectées par l'algorithme, seuls les contours de cinq d'entre elles sont utilisés pour labéliser les lignes.

3 Autres approches au problème de recalage

La méthode proposée par T.Colleu dans [1, 2] fonctionne assez bien, mais il est intéressant de remarquer que d'autres laboratoires ont proposé des méthodes parfois complètement différentes, qui peuvent être utilisées pour résoudre ce même problème. Cette partie est consacrée à la description de trois idées tirées des articles [10, 11], et contient des explications pour les utiliser afin de résoudre un problème de recalage.

- La première méthode utilise les points de fuite pour estimer la pose de la caméra de façon grossière. Elle est donc plutôt à comparer avec la méthode de recalage approximative proposée par T. Colleu.
- La seconde méthode s'intéresse au problème de mise en correspondance des primitives, en utilisant une méthode basée sur des descripteurs locaux. Les méthodes basées descripteurs sont un peu moins utilisées depuis que les algorithmes de suivi de points d'intérêts sont efficaces, mais ils ont fait leurs preuves.
- La dernière méthode cherche à transporter tout le problème du recalage dans l'espace à trois dimensions. A l'inverse des méthodes précédentes qui projettent le modèle 3D dans l'image, nous verrons ici un moyen pour retrouver la géométrie 3D de la scène à partir des images. C'est la méthode décrite dans le papier [10].

3.1 Extraction des points de fuite

De la même façon qu'il est possible de déterminer la pose précise à partir d'une pose approximative et de seulement trois couples de droites en corrélation [4] (en configuration non dégénérée), il est possible de le faire avec trois couples de points. Or, une méthode est expliquée dans [23] pour récupérer des points particuliers de l'image initiale que sont les points de fuite. Dans un environnement urbain, il y a un nombre très restreint de points de fuite, et bien souvent, seulement trois sont vraiment marqués. Il est donc envisageable d'extraire ces trois points de fuite, et de les faire correspondre aux points de fuite récupérés dans le modèle SIG.

Le calcul de pose ainsi obtenu n'est pas précis, mais il peut remplacer la première partie de la méthode décrite précédemment, qui est l'estimation approximative de la pose. Il ne serait donc plus nécessaire de calculer $t_{relatif}$. En effet, la seule connaissance de la position de la caméra (donnée par le GPS) et des points de fuites restreint les poses possibles à seulement une dizaine. D'après [23], récupérer les points de fuite d'une image est une opération coûteuse si on veux la faire avec précision. Cependant, ces points peuvent être intèressants pour d'autres utilisations ultérieures [10, 11].

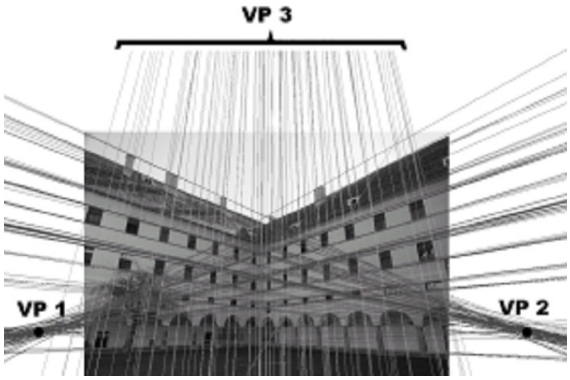

FIG. 9 – Points de fuite extraits d'une photo en zone urbaine.

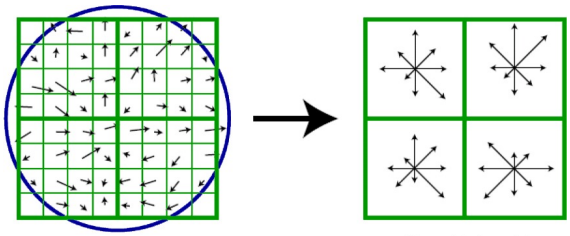

FIG. 10 – La figure de gauche représente la direction du gradient en chaque pixel de l'image, et la figure de droite représente le descripteur de Harris associé à chaque zone.

3.2 Points d'intérêt sur l'image des contours

La sélection des points d'intérêt dans les méthodes de suivi telles que KLT [21], est basée sur des petites fenêtres qui ont une grande disparité fréquentielle. C'est le cas des angles présents dans l'image, et de tous les contours en général. Si l'on fait une recherche des points d'intérêt dans la carte des contours, ainsi que dans la carte des arêtes du modèle SIG projeté, il sera possible de déterminer des correspondances comme cela est fait dans [17]. Les détecteurs de Harris peuvent être utilisés pour sélectionner les points d'intérêt, et les descripteurs qu'ils associent aux points peuvent largement aider à la mise en correspondance. En effet, les descripteurs des points d'intérêt peuvent être vus comme des histogrammes d'orientation du gradient, comme le montre la figure 10. Deux points en correspondance dans les contours issus de l'image, et ceux issus du modèle, auront des descripteurs voisins.

3.3 Modèle 3D des contours

La méthode proposée par T. Colleu [1, 2] consiste à faire les correspondances entre les primitives une fois qu'elles sont toutes projetées dans l'image. Cela implique une perte d'information sur la profondeur des arêtes qui peut mener à des incohérences dans les corrélations détectées (un contour lointain peut être mis en corrélation avec une arête proche). Une approche inverse est proposée par T. Werner et A. Zisserman dans [10], qui consiste faire les correspondances dans le monde 3D. Au lieu de projeter le modèle SIG dans le plan image, ils récupèrent des informations de profondeur de l'image, pour estimer la position 3D des primitives. Leur problématique est plutôt orientée sur le calcul d'un modèle 3D d'une scène, à partir de plusieurs photos, ou d'une vidéo. Les deux premières étapes qu'ils proposent, à savoir, l'extraction d'information de profondeur pour un ensemble de points et la recherche de primitives linéaires ou planaires parmi ces points 3D, s'adaptent cependant très bien au problème du recalage.

3.3.1 Extraction d'un nuage de points 3D

Pour récupérer l'information de profondeur pour un ensemble intéressant de points de l'image, la première étape consiste à définir des points d'intérêt et à les suivre par une méthode TLT [21] sur plusieurs images distinctes de la séquence. Il faut un minimum de deux images assez éloignées pour pouvoir estimer la profondeur, mais l'utilisation de plus d'images, comme proposé dans l'article, permet de détecter plus facilement les points aberrants et permet une meilleure estimation. On considère connue la pose P_i des différents points de vue, mais si ce n'est pas le cas, il est possible de la récupérer relativement au repère d'une des caméras, comme expliqué précédemment [4]). Ensuite, si l'on considère que les points $p_i = (x_i, y_i)$ mis en correspondance, représentent un même point physique M, il est possible de récupérer les coordonnées 3D du point

$M(X, Y, Z)$ en résolvant les $2 * n$ équations :

$$\begin{cases} x_1 & = & P_{1x}(X, Y, Z) \\ y_1 & = & P_{1y}(X, Y, Z) \\ & \vdots & \\ x_n & = & P_{nx}(X, Y, Z) \\ y_n & = & P_{ny}(X, Y, Z) \end{cases}$$

Ces calculs, appliqués à un grand ensemble de points d'intérêt, fournissent comme résultat un nuage de points 3D qui est représenté dans la figure 11. C'est de ce nuage que vont être extraites les primitives qui pourront être utilisées pour la mise en correspondance avec le modèle SIG.

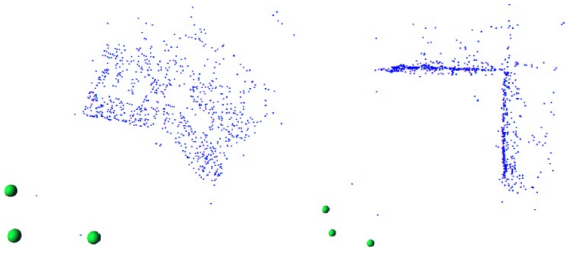

FIG. 11 – Deux vues différentes d'un nuage de points calculé à partir de trois prises de vue.

3.3.2 Recherche de primitive et recalage

L'étape suivante consiste à chercher dans ce nuage de points 3D, des alignements ou des plans de points. Ce sont les mêmes idées utilisées ici que pour détecter des alignements dans les points de contours, mais appliquées cette fois dans un monde en trois dimensions [14]. La figure 12 représente une vue des lignes 3D et des plans qui ont pu être extraits.

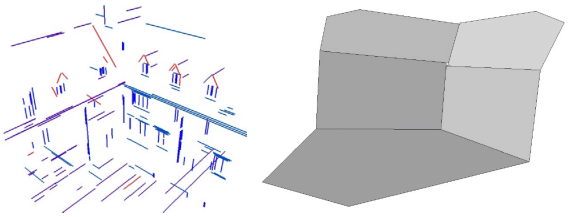

FIG. 12 – Une représentation des lignes 3D, et des plans 3D issue du nuage de points.

Le problème est donc transposé dans un espace à trois dimensions, mais l'article ne poursuit pas le recalage, il s'intéresse plutôt à un aspect construction de modèle. Si le recalage approximatif effectué au préalable est suffisamment précis, une méthode de mise en correspondance des lignes les plus proches, ou des plans les plus proches, peut fournir de bons résultats. Sinon, une approche par un estimateur robuste (RANSAC) comme vue précédemment, peut être envisagée.

4 Conclusion

Nous avons vu la méthodologie suivie par l'équipe TEMICS de l'IRISA, permettant le recalage initial automatique d'un modèle SIG avec une vidéo. La solution proposée se décompose en deux parties. La première estime une pose approximative pour limiter les correspondances possibles entre les lignes du modèle et les contours de l'image. La seconde effectue cette mise en correspondance par un algorithme basé RANSAC afin de déterminer la pose précise. Cette méthode ne présente pas toujours de bons résultats. Il faut par exemple choisir l'image i de façon judicieuse, ce qui est pour le moment fait à la main. De plus, certaines configurations ne permettent pas un bon recalage. C'est le cas si il n'y a pas quatre arêtes de bâtiment clairement visibles dans l'image. Pour tenter de remédier à ces difficultés, nous avons vu d'autres approches au problème de recalage, mais qui possèdent chacune des défauts plus ou moins graves. L'objectif du stage à venir pourrait donc être, d'une part de travailler sur l'implémentation de T. Colleu pour en améliorer chaque étape, et d'autre part de tester d'autres méthodes, comme celles abordées ici, pour effectuer le recalage.

Remerciements à Luce Morin pour ses conseils avisés quant à la structure de ce document.

Références

[1] Thomas Colleu. Recalage initial vidéo, GPS et modèle numérique de terrain pour des applications de réalité virtuelle augmentée dans

des environnement urbains. *Rapport de stage, université de Rennes, 2007*

[2] Thomas Colleu, Gael Sourimant et Luce Morin. Une méthode d'initialisation automatique pour le recalage de données SIG et vidéo. *COmpression et REpresentation des Signaux Audiovisuels (CORESA 2007)*

[3] Gael Sourimant. Reconstruction de scènes urbaines à l aide de fusion de données de type GPS, SIG et Vidéo. *PhD Thesis, université de Rennes, 2007*

[4] E. Marchand and F. Chaumette. Virtual visual servoing : a framework for real-time augmented reality. *In EUROGRAPHICS'02 Conference Proceeding, volume 21(3) of Computer Graphics Forum, pages 289–298, Saarebrcken, Germany, 2002*

[5] L. Ding and A. Goshtasby. The Canny edge detector. *Pattern Recognition, 34(3) :721–725, 2001*

[6] Canny, J. A computational approach to edge detection. *IEEE Computer Society, 1986*

[7] Analysis Josep Llados. The Hough Transform As A Tool For Image.

[8] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Geometric context from a single image. *InICCV, volume1, pages 654–661.IEEE, 2005*

[9] 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*

[10] Thomas Werner and Andrew Zisserman. New Techniques for Automated Architectural Reconstruction from Photographs. *Robotics Research Group, Department on Engineering Science, University of Oxford, 2002*

[11] Konrad Schindler and Joachim Bauer. A Model-based Methode For Building Reconstruction. *Inffeldgasse 16, 8010 Graz, Austria, 2003*

[12] Paul E. Debevec and 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, p. 11–20, 1996*

[13] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. *Press, ISBN : 0521540518, seconde édition, Cambridge University, 2004*

[14] C. Baillard and C. Schmid and A. Zisserman and A. Fitzgibbon. Automatic line matching and 3D reconstruction of buildings from multiple views. *In Proc. ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery, IAPRS Vol.32, Part 3-2W5, 1999*

[15] A. Bartoli. Piecewise planar segmentation for automatic scene modeling. *Conference on Computer Vision and Pattern Recognition, Hawaii, pages 283–289, 2001*

[16] G. Simon, M.-O. Berger. Pose estimation for planar structures. *IEEE CG-A, 22(6) :46–53, 2002*

[17] Daniel DeMenthon and Larry S. Davis. Model-Based Object Pose in 25 Lines of Code. *European Conference on Computer Vision, p. 335–343, 1992*

[18] E. Marchand and F. Chaumette. A new formulation for non-linear camera calibration using virtual visual servoing. *Technical Report 4096, INRIA, 2001*

[19] Martin Armstrong and Andrew Zisserman and Richard I. Hartley. Self-Calibration from Image Triplets. *ECCV (1), p. 3–16, 1996*

[20] Q. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *IJCV, 22(3) :261–89, 1997*

[21] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. *CMU-CS-91-132, Carnegie Mellon University, 1991*

[22] T. Auer. Hybrid Tracking of Augmented Reality. *PhD thesis, technical university of Graz, 2000*

[23] C. Rother. A new approach for vanishing point detection in architectural environments. *In BMVC2000, pages 382–391, 2000*