Reconstruction et augmentation simultanées de scènes planes par morceaux

Gilles Simon, Marie-Odile Berger

Reconstruction et augmentation simultanées de scènes planes par morceaux

Simultaneous reconstruction and augmentation of piecewise planar surfaces

G. Simon[1] M.-O. Berger[2]

[1] LORIA - Nancy Université
[2] LORIA - INRIA Lorraine

LORIA - Campus Scientifique - BP 239 - 54506 Vandœuvre-lès-Nancy Cedex
gsimon@loria.fr

Résumé

Nous proposons une méthode de localisation et construction de carte simultanées, adaptée aux objectifs et contraintes de la RA. Cette méthode repose sur le suivi inter-images de surfaces planes de la scène, dont les intersections avec un plan de référence sont détectées automatiquement par filtrage particulaire. Ces intersections sont utilisées pour lever l'ambiguïté du positionnement mono-plan et connaître les équations des plans intersectant le plan de référence. Un système de RA semi-automatique est proposé, qui ne réclame que très peu d'intervention de la part de l'opérateur : celui-ci doit simplement "viser" avec la caméra les plans à intégrer et valider ou non les solutions proposées par le système (par simple pression sur un bouton). Différentes stratégies de filtrage particulaire sont comparées sur une séquence synthétique, et une utilisation complète du système est présentée sur une scène réelle. Enfin, nous montrons des résultats préliminaires d'extension de cette méthode à un système de type SLAM entièrement automatique.

Mots Clef

SLAM, réalité augmentée, reconstruction et positionnement multi-plan, vision par ordinateur.

Abstract

A simultaneous localization and mapping (SLAM) method is presented, which is particularly suited to AR applications. The method is based on inter-image tracking of planar surfaces, whose intersections with a reference plane are automatically extracted using a particle filter. These intersections are used to clean up the ambiguity of the SFM problem using a single plane, and to get the equations of the intersecting planes with the reference plane. A semi-automatic system is proposed, which requires minimum intervention from the user, who only has to aim the planes to be integrated with the camera , and validate or cancel the solutions generated by the system. Different strategies of particle filtering are compared on synthetical data, and a complete use of the interactive system is presented on a real scene. Finally, preliminary results are shown that illustrate how the method can be extended to a fully automatic SLAM system.

Keywords

SLAM, augmented reality, multi-planar reconstruction and camera tracking, computer vision

1 Introduction

Le problème de localisation et construction de carte simultanées, appelé SLAM (*Simultaneous Localization And Mapping*) dans la littérature, est un problème actuellement très étudié aussi bien dans la communauté robotique que dans la communauté vision. Si des résultats remarquables ont été obtenus à l'aide d'une simple caméra comme dispositif d'acquisition, les systèmes proposés sont encore mal adaptés à la réalité augmentée. La carte reconstruite est généralement un ensemble de points 3D [3, 10], ou au mieux de patchs plans [8, 12], qui n'offre pas de référentiel intuitif pour le placement des objets virtuels, et ne permet pas de représenter les échanges lumineux entre les surfaces réelles et virtuelles de la scène.

Nous proposons une méthode de localisation et construction de carte simultanées, adaptée aux objectifs et contraintes de la RA en environnement urbain ou d'intérieur. Cette méthode repose sur le suivi inter-images de surfaces planes de la scène (typiquement des murs), dont les intersections avec un plan de référence (le sol) sont détectées automatiquement par filtrage particulaire. Ces intersections sont utilisées pour lever l'ambiguïté du positionnement mono-plan et connaître les équations des plans intersectant le plan de référence.

Un système semi-automatique est proposé, qui ne réclame que très peu d'intervention de la part de l'opérateur : celui-ci doit simplement "viser" avec la caméra les plans à intégrer et valider ou non les solutions proposées par le sys-

tème (par simple pression sur un bouton). Cette solution remplace avantageusement les traditionnelles méthodes de détourages de zone et désignations de primitives à la souris [14], difficiles à mettre en œuvre quand l'opérateur se déplace, et incompatibles avec les applications de RA fonctionnant sur un PDA ou un téléphone portable [16].

Un système de détection de plans entièrement automatique (mais projectif uniquement) est aussi présenté, qui repose sur le regroupement par plans de patchs rectangulaires, selon les intersection avec un plan de référence des plans auxquels sont supposés appartenir les patchs.

La section 2 présente la méthode de SFM (*Structure From Motion*) causale que nous proposons pour calculer la structure d'une scène bi-plan observée dans un flux d'images, en même temps que le mouvement de la caméra ayant donné lieu aux observations. Les deux mises en œuvres de cette méthode sont ensuite proposées : le système semi-automatique en section 3, et le système automatique en section 4. Enfin, des résultats obtenus sur des données réelles avec chacun des deux systèmes sont montrés en section 5.

2 Estimation causale de la structure et du mouvement

Lorsque deux plans sont observés dans deux images, il existe une solution unique au problème de SFM [17, 8]. Dans [8], les paramètres intrinsèques de la caméra sont supposés connus, et un filtre de Kalman étendu est utilisé pour estimer de manière causale le mouvement de la caméra, les équations des plans associés à des patchs planaires et les paramètres du modèle affine utilisé pour tenir compte des changements d'illumination observés sur les patchs. Dans [17], les paramètres intrinsèques de la caméra sont estimés en même temps que son mouvement et la structure de la scène : l'épipole est estimé en calculant les vecteurs propres d'une homologie (cf. section 2.1), et un calcul linéaire permet d'obtenir les équations des plans en même temps que le mouvement de la caméra. Comme deux solutions sont obtenues (une pour chaque plan), et que les deux solutions sont théoriquement valables, l'une ou l'autre est indifféremment choisie, et utilisée comme estimée initiale d'un ajustement de faisceau global.

La méthode que nous proposons est à mi-chemin entre ces deux approches. Les paramètres intrinsèques de la caméra sont supposés connus, et nous estimons la structure de la scène et le mouvement de la caméra en utilisant un algorithme à deux étapes :

1. la projection dans le plan rétinien de l'intersection des deux plans observés est estimée dynamiquement par filtrage particulaire,
2. le résultat de la première étape est utilisé pour calculer de manière directe le mouvement de la caméra et l'équation des deux plans dans l'espace métrique euclidien.

L'intérêt de ce découpage est multiple :

- comme dans [8], le filtrage dynamique opéré en étape 1 permet de tenir compte de la cohérence temporelle des observations. L'utilisation d'un filtre particulaire au lieu d'un filtre de Kalman évite le problème de l'initialisation. D'autre part, nous estimons un vecteur d'état à deux paramètres seulement (les paramètres d'une droite 2D), au lieu de 18 dans [8] (sans compter les 2 paramètres par plan du modèle affine de changement d'illumination) ;
- le résultat de l'étape 1 peut être évalué aisément par l'utilisateur (visuellement) ou par le système (en utilisant le gradient de l'image) puisqu'il doit correspondre à une intersection physiquement observable dans l'image. Une mesure de vraisemblance liée au gradient de l'image peut d'ailleurs être fusionnée très facilement avec la contrainte géométrique utilisée dans le filtrage (cf. section 2.3) ;
- l'étape 2 bénéficie de la rapidité d'une résolution linéaire, avec une meilleure fiabilité que dans [17] puisqu'une partie de la solution est connue au préalable ;

Par ailleurs, l'étape 1 peut être utilisée pour obtenir immédiatement les équations de nouveaux plans apparaissant dans l'image, le point de vue de la caméra étant obtenu à l'aide de plans détectés auparavant (cf. section 3) ; Cette étape étant purement projective, elle peut aussi être employée pour détecter automatiquement des régions planes dans une séquence d'images non calibrées (section 4).

2.1 Notations et résultats préliminaires

En coordonnées projectives (homogènes), un point image (p_x, p_y) est représenté par le vecteur colonne 3×1 $\mathbf{p} = (p_x, p_y, 1)^t$. Une droite définie par l'équation $\mathbf{l}^t.\mathbf{p} = 0$ est aussi représentée en coordonnées projectives par le vecteur $\mathbf{l}$. La droite passant par les points $\mathbf{p}_1$ et $\mathbf{p}_2$ est donnée par le produit vectoriel $\mathbf{p}_1 \times \mathbf{p}_2$ et le point d'intersection de deux droites $\mathbf{l}_1$ et $\mathbf{l}_2$ est donné par le produit vectoriel $\mathbf{l}_1 \times \mathbf{l}_2$ (points et droites sont duaux dans l'espace projectif).

Un concept important en géométrie projective est l'homographie plane $\mathbf{H}$, une matrice 3×3 régulière, qui relie deux images rétiniennes non calibrées d'un plan 3D. Plus formellement, si **p** est la projection dans une vue d'un point du plan et **p'** sa projection dans une deuxième vue, alors les deux projections sont liées par la transformation linéaire projective :

$$\mathbf{p}' \sim \mathbf{H}\mathbf{p},$$

où $\sim$ dénote l'égalité à un facteur près. Une équation similaire relie une pair de droites du plan entre deux vues :

$$\mathbf{l}' \sim \mathbf{H}^{-t}\mathbf{l},$$

où $\mathbf{H}^{-t}$ est l'inverse de la transposée de $\mathbf{H}$.

Soit **p** la projection dans une vue d'un point appartenant à l'intersection de deux plans, et soient $\mathbf{H}_1$ et $\mathbf{H}_2$ les homographies induites par les deux plans entre cette vue et une deuxième vue. Comme **p** appartient à l'intersection des deux plans, $\mathbf{H_1 p} \sim \mathbf{H_2 p}$, ce qui se traduit par **p** vecteur propre de $\mathbf{S} = \mathbf{H_2^{-1} H_1}$, où $\mathbf{S}$ est une homologie plane [6]. Géométriquement, une homologie est une transformation

projective admettant une droite de points fixes (dans notre cas, la projection de l'intersection des deux plans), et un point fixe n'appartenant pas à cette droite (l'épipole dans la première vue). En utilisant la dualité point/droite dans l'espace projectif, on montre facilement que la transformation $\mathbf{T} = \mathbf{S}^t$ admet la droite recherchée comme droite fixe, ainsi qu'un faisceau de droites fixes s'intersectant en l'épipole de la première vue.
En théorie, le problème peut être résolu algébriquement [9] en calculant le vecteur propre associé à la valeur propre simple de $\mathbf{T}$ (le faisceau de droites fixes étant porté par les deux vecteurs propres associés à la valeur propre double de $\mathbf{T}$). En pratique, l'extraction des vecteurs propres de $\mathbf{T}$ est particulièrement instable, cette matrice n'étant pas symétrique. Nous obtenons une bien meilleure stabilité en résolvant le problème géométriquement, et en tenant compte de la cohérence temporelle des mesures à l'aide d'un filtre particulaire.

2.2 Filtrage particulaire

Les techniques de filtrage particulaire sont des méthodes de Monte Carlo pour l'estimation récursive du vecteur d'état d'un système stochastique Markovien [1]. Leur but est d'approximer la densité de probabilité *a posteriori* $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ du vecteur d'état $\mathbf{x}_k$ à l'instant k conditionnellement aux mesures $\mathbf{z}_{1:k} = \mathbf{z}_1, ...\mathbf{z}_k$, par une distribution ponctuelle

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i), \sum_{i=1}^{N} w_k^i = 1, \qquad (1)$$

exprimant la sélection d'une valeur (ou particule) $\mathbf{x}_k^i$ avec la probabilité (ou poids) w_k^i. Une estimée $\hat{\mathbf{x}}_k$ du vecteur d'état est alors donnée par le premier mode de la distribution (1), par sa moyenne $\sum_{i=1}^{N} w_k^i \mathbf{x}_k^i$, ou par d'autres fonctions de $\mathbf{x}_k$.
L'algorithme de filtrage particulaire générique comprend les étapes suivantes à l'instant k (cf. Algorithme 1) :

1. les N particules $\mathbf{x}_k^i$ sont échantillonnées selon une *densité d'importance* $q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$,
2. les poids w_k^i sont mis à jour (puis normalisés) en tenant compte de la *vraisemblance* de la particule $p(\mathbf{z}_k|\mathbf{x}_k^i)$ par rapport à une mesure z_k, de la loi de dynamique $p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)$ et de la densité d'importance :

$$w_k^i \sim w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)}, \qquad (2)$$

3. pour éviter les problèmes de dégénérescence (quand après quelques itérations toutes les particules, sauf une, ont un poids négligeable), on mesure la taille "effective" N_{eff} de l'échantillonnage :

$$N_{eff} \approx \frac{1}{\sum_{i=1}^{N} (w_k^i)^2}, \qquad (3)$$

Algorithm 1 Filtrage particulaire générique.

fonction $[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N]$ = PF$[\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^N, \mathbf{z}_k]$
1: **pour** i=1 :N **faire**
2: Tirage de $\mathbf{x}_k^i$ suivant $q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$
3: Calcul du poids w_k^i (équation (2))
4: **fin pour**
5: Normalisation des poids assurant que $\sum_{i=1}^{N} w_k^i = 1$
6: Calcul de N_{eff} (équation (3))
7: **si** $N_{eff} \leq N_t$ **alors**
8: $[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N]$ = RESAMPLE$[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N]$
9: **fin si**

4. si N_{eff} est inférieur à un certain seuil N_t, alors l'ensemble des particules est rééchantillonné : les particules de plus faibles poids sont éliminées et les autres particules sont dupliquées en nombre proportionnel à la valeur de leur poids (la fonction RESAMPLE est décrite en détails dans [1]).

Le choix de la densité d'importance est une étape cruciale dans la conception d'un filtre particulaire. La densité la plus communément employée est simplement la densité de probabilité $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. L'équation (2) se simplifie alors en :

$$w_k^i \sim w_{k-1}^i p(\mathbf{z}_k|\mathbf{x}_k^i). \qquad (4)$$

Le filtre est initialisé par une séquence indépendante identiquement distribuée. Différentes variantes de l'algorithme 1 peuvent être utilisées en fonction du contexte. En particulier, la stratégie SIR (Sampling Importance Resampling) consiste à utiliser $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ comme fonction d'importance, et à rééchantillonner systématiquement les particules à chaque itération (les particules ont alors toutes le même poids $1/N$ à la fin de chaque itération).

2.3 Implémentation

Le filtre que nous utilisons possède les caractéristiques suivantes (des valeurs typiques des constantes introduites dans cette section sont données en table 1) :
– les particules sont les coordonnées homogènes des droites $\mathbf{l}_k^i$ du plan rétinien. À l'initialisation, ces droites sont uniformément réparties dans l'ellipse $\mathcal{E} : \left(\frac{x}{l}\right)^2 + \left(\frac{y}{h}\right)^2 = 1$, $l \times h$ étant les dimensions de l'image (cf. Fig.1, Image 1) ;
– la loi de dynamique $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ est la loi normale centrée en $\mathbf{x}_{k-1}$, de covariance $\mathbf{V}$. La densité d'importance est la loi de dynamique ;
– la mesure de vraisemblance $\mathbf{z}_k^h$ est liée à la contrainte imposée par les homographies $\mathbf{H}_1$ et $\mathbf{H}_2$. Lorsque le contexte le permet, cette mesure peut être fusionnée avec une mesure $\mathbf{z}_k^g$ qui tient compte des gradients d'intensité de l'image k. Sous l'hypothèse d'indépendance des deux sources de mesures, la mesure de vraisemblance à l'instant k s'écrit :

$$p(z_k^h, z_k^g|\mathbf{x}_k) = p(z_k^h|\mathbf{x}_k)p(z_k^g|\mathbf{x}_k),$$

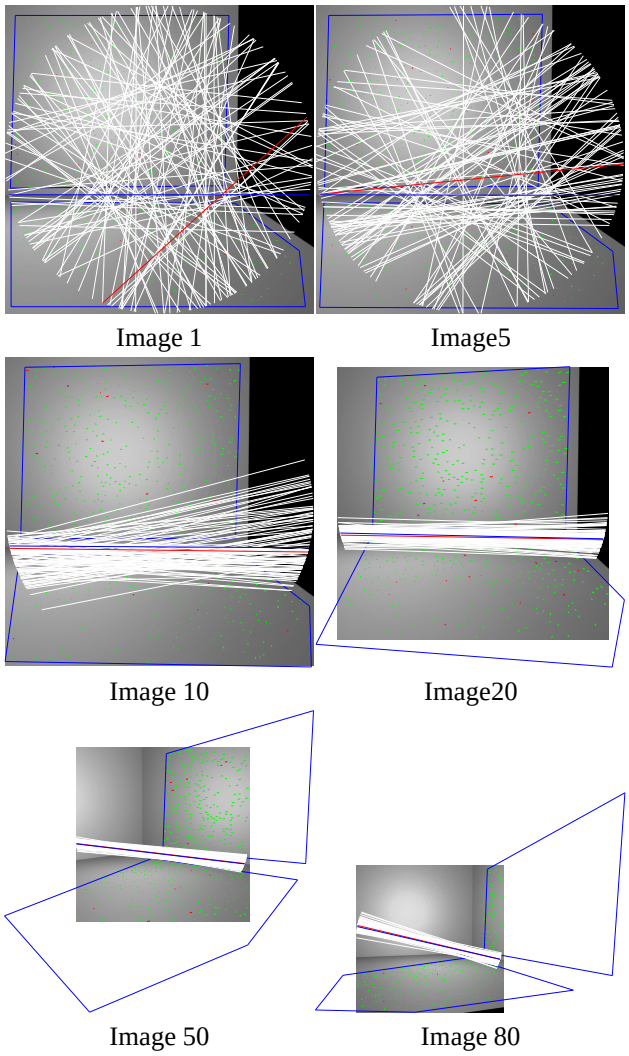

FIG. 1 – Filtrage particulaire sur une séquence synthétique de 80 images ($N = N_t = 100$). La droite attendue est dessinée en bleu, le premier mode de la distribution (1) en rouge (les extrêmités des segments verts, pour les inliers, et rouges, pour les outliers, sont les points appariés utilisés pour le calcul des homographies).

où les deux densités d'observation élémentaires sont précisées ci-dessous.

Mesure liée aux homographies. On cherche à faire converger l'estimée $\hat{\mathbf{x}}_k$ du vecteur d'état vers l'intersection des deux plans dans l'image 0 (on obtient donc l'estimée de cette intersection dans l'image k en calculant $\mathbf{H}_1^{-t}\hat{\mathbf{x}}_k$ ou $\mathbf{H}_2^{-t}\hat{\mathbf{x}}_k$, $\mathbf{H}_1$ et $\mathbf{H}_2$ désignant les homographies calculées entre l'image 0 et l'image k). La contrainte géométrique est alors que $\hat{\mathbf{x}}_k$ doit être fixe par $\mathbf{T} = \mathbf{H}_1^t\mathbf{H}_2^{-t}$. Cependant, cette contrainte est aussi vérifiée par toutes les droites passant par l'épipole de la vue k dans la vue 0 (cf. section 2.1). Pour palier ce problème, nous exploitons le fait que la droite recherchée est une droite de points fixes, à la différence des autres droites vérifiant la contrainte, dont les points sont transformés en d'autres points de la même droite. Ainsi, nous mesurons la fixité non pas de $\mathbf{l}_k^i$ mais d'un échantillon de points sur $\mathbf{l}_k^i$. En pratique, deux points suffisent : nous calculons les intersections $\mathbf{p}_1$ et $\mathbf{p}_2$ de $\mathbf{l}_k^i$ avec l'ellipse $\mathcal{E}$, puis mesurons la vraisemblance :

$$p(z_k^h|\mathbf{x}_k) = \exp\left(-\frac{D^2}{2\sigma_h^2}\right), D = \frac{1}{2}\sqrt{\sum_{j=1}^{2}||z(\mathbf{p}_j) - z(\mathbf{S}\mathbf{p}_j)||^2},$$

où $z([a,b,c]^t) = [a/c, b/c]^t$ pour $c \neq 0$, et $||.||$ est la norme euclidienne.

Mesure liée au gradient de l'image. Lorsque les deux plans suivis sont suffisamment contrastés et qu'il n'y a pas trop d'objets occultants à l'endroit de leur intersection, il peut être intéressant de tenir compte de la distance des particules aux gradients de l'image. Cela permet à la fois d'accroître la vitesse de convergence du filtre et d'obtenir une meilleure précision de l'estimée. Cette mesure repose sur les traitements suivants de l'image :

– filtrage par MDIF,
– seuillage par hystérésis,
– extraction des droites $\{\mathbf{m}_k^i\}_{1\leq i\leq M}$ (M étant imposé) par transformée de Hough[1].

Cela permet de calculer la mesure de vraisemblance suivante :

$$p(z_k^g|\mathbf{x}_k) = \exp\left(-\frac{D^2}{2\sigma_g^2}\right), D = \frac{1}{2}\min_{i=1}^{M}\sqrt{\sum_{j=1}^{2}(\mathbf{m}_k^i|\mathbf{H}_1\mathbf{p}_j)^2},$$

où $(.|.)$ dénote le produit scalaire de deux vecteurs, et $\mathbf{m}_k^i$ est exprimée sous la forme $[\cos(\theta)\ \sin(\theta)\ -\rho]^t$. Dans cette mesure, $\mathbf{H}_1$ peut être remplacé par $\mathbf{H}_2$, et on peut aussi utiliser une distance symétrique intégrant $\mathbf{H}_1$ et $\mathbf{H}_2$ (nous n'avons pas observé de différence dans nos expérimentations). Cette mesure bénéficie de la robustesse de la transformée de Hough, et tolère donc une occultation partielle de la droite d'intersection. Cependant, il est généralement préférable de donner moins d'importance à ce critère qu'à celui lié aux homographies, même lorsque l'intersection est bien visible : la mesure liée aux homographies permet aux particules de s'approcher de la solution attendue, et celle liée au gradient de l'image d'affiner la convergence. Pour que le deuxième critère ait moins d'influence que le premier, il suffit de prendre $\sigma_g > \sigma_h$ (cf. section 5).

Constantes de filtrage. Le choix des constantes N et N_t repose sur des tests synthétiques. Nous utilisons une séquence de 80 images de deux plans, l'un horizontal et l'autre vertical (cf. Fig. 1). La caméra suit une trajectoire circulaire, en même temps qu'une rotation autour de son axe vertical. Des points 3D sont projetés dans les images, et les projections sont perturbées par un bruit gaussien, d'écart-type 0.3. Les appariement inter-images obtenus sont utilisés pour calculer les homographies $\mathbf{H}_1$ et $\mathbf{H}_2$

[1] Les temps de calcul de la transformée de Hough sont considérablement réduits si on incrémente uniquement les accumulateurs (ρ, θ) pour lesquels θ est compatible avec l'orientation du gradient donnée par MDIF (avec une tolérance de ± 20 deg par exemple).

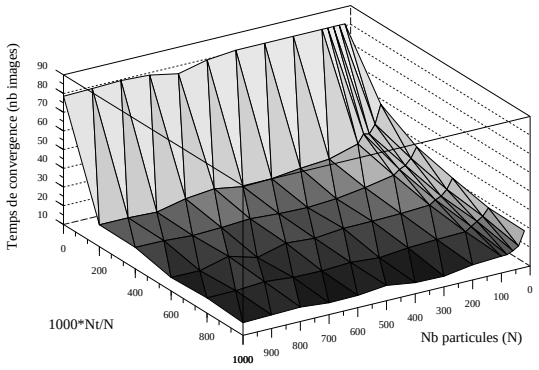

FIG. 2 – Convergence du filtre particulaire sur une séquence synthétique, en fonction des valeurs de N et N_t.

par RANSAC. On calcule le premier mode $\hat{\mathbf{x}}_k$ de la distribution (1) depuis la deuxième image de la séquence jusqu'à l'image k_c où la convergence est atteinte (distance entre la droite calculée et la droite théorique, mesurée sur les intersections avec l'ellipse $\mathcal{E}$, inférieure à 1.5 pixels). Cette opération est répétée 100 fois pour chaque valeur de N et de N_t. N varie de 20 à 1000 avec un pas de 20 entre 20 et 100, puis un pas de 100 entre 100 et 1000. N_t varie de 0 à 100% de N, avec un pas de 20%.

La figure 2 reporte la valeur moyenne de k_c obtenue pour chaque couple de valeurs (N, N_t). Lorsque cette valeur vaut 80, le filtre n'a pas convergé : cela se produit quand $N_t = 0$ (problème de dégénérescence évoqué plus haut). En dehors du cas $N_t = 0$, le filtre converge toujours, même lorsque très peu de particules sont utilisées (par exemple, pour $N = 20$ et $N_t = 12$, la convergence est atteinte en 34 images). On observe toutefois une convergence plus rapide pour les grandes valeurs de N, même si au-delà de 300 particules, le gain devient faible. On constate aussi que la convergence est plus rapide lorsque N_t se rapproche de N. Le cas $N_t = N$ illustre la stratégie SIR puisque N_{eff} est toujours inférieur à N, et donc dans ce cas à N_t. Ces résultats nous incitent donc à utiliser la stratégie SIR, avec un nombre de particules compris entre 300 et 1000 (en pratique, nous utilisons $N = 1000$ car le temps utilisé pour mettre à jour le poids d'une particule est négligeable devant le temps utilisé par les autres étapes de l'algorithme – suivi des plans, transformée de Hough, ...).

2.4 Reconstruction métrique de la scène

Nous montrons dans cette section comment résoudre le problème de SFM à partir des homographies $\mathbf{H}_1$ et $\mathbf{H}_2$ induites par l'observation de deux plans π_1 et π_2 et la connaissance dans la première vue de la droite d'intersection $\mathbf{l}$ entre ces deux plans. On suppose que des points appartenant à chacun des deux plans sont connus dans la première image (au moins quatre points $\mathbf{P}_k^i$ par plan π_k). En pratique, ces points sont les sommets des polygones utilisés pour le suivi des régions planes.

Soit $\mathbf{R}$ la rotation et $\mathbf{t}$ la translation de la caméra entre les deux vues, et soit $\mathbf{n}_k$ la normale et d_k la distance à l'origine du plan π_k, exprimées dans le repère de la première caméra. L'homographie $\hat{\mathbf{H}}_k$ induite par le plan π_k est [4] :

$$\hat{\mathbf{H}}_k = \mathbf{K}\left(\mathbf{R} + \frac{\mathbf{t}\mathbf{n}_k^t}{d_k}\right)\mathbf{K}^{-1}. \quad (5)$$

Inversement, si $\mathbf{H}_k$ et $\mathbf{K}$ sont connues, l'extraction de $\mathbf{R}$, $\mathbf{t}$, $\mathbf{n}_k$ et d_k à partir de $\mathbf{A} = \mathbf{K}^{-1}\mathbf{H}_k\mathbf{K}$ admet [4] :

1. 8 solutions dont 2 physiquement correctes[2] dans le cas général,
2. 4 solutions dont 1 physiquement correcte si $\mathbf{A}$ possède une valeur singulière de multiplicité 2,
3. une solution partiellement indéterminée si $\mathbf{A}$ possède une valeur singulière de multiplicité 3.

Le cas 2 correspond à un mouvement de translation normal au plan ; le cas 3 à une rotation pure ou à $\mathbf{t} = -2\mathbf{R}\mathbf{n}$ (les deux vues sont de part et d'autre et à la même distance d'un plan transparent). Dans le troisième cas, le mouvement de la caméra peut être calculé, mais pas l'équation du plan.

L'algorithme mis en œuvre est le suivant : la matrice $\mathbf{K}^{-1}\mathbf{H_1}\mathbf{K}$ est décomposée en valeurs singulières. Soit m la multiplicité maximale des valeurs singulières calculées.

– Si $m = 3$, le mouvement est calculé mais la structure est indéterminée.
– Si $m = 2$, la structure de π_1 et le mouvement de la caméra sont déterminés de manière unique. L'équation de π_2 est obtenue grâce à la connaissance de la droite $\mathbf{l}$: soit $\mathbf{L}$ la droite d'intersection de π_1 avec le plan défini par $\mathbf{l}$ et le centre optique de la première caméra. Cette droite est contenue dans une famille de plans à un paramètre $\pi(\mu)$, où μ est l'angle entre π_1 et $\pi(\mu)$ [13] (Fig. 3). Si l'angle ν entre π_1 et π_2 est connu, nous obtenons directement $\pi_2 = \pi(\nu)$. Dans le cas contraire, nous utilisons la connaissance de l'homographie $\mathbf{H}_2$: on prend $\pi_2 = \pi(\nu)$, où ν minimise l'erreur de transfert $err_t(\hat{\mathbf{H}}_2(\mu)), \mu \in]0, \pi[$, $\hat{\mathbf{H}}_2(\mu)$ étant l'homographie induite par $\pi(\mu)$ selon l'équation (5), et $err_t(\mathbf{H})$ l'erreur de transfert des points de $\mathbf{P}_2$ par $\mathbf{H}$:

$$err_t(\mathbf{H}) = \sum_i ||z(\mathbf{H}\mathbf{P}_2^i) - z(\mathbf{H}_2\mathbf{P}_2^i)||^2. \quad (6)$$

– Si $m = 1$, deux solutions sont obtenues pour π_1 et le mouvement de la caméra. Si l'angle ν entre les deux plans est connu, on choisit la solution qui obtient l'erreur de transfert $err_t(\hat{\mathbf{H}}_2(\nu))$ la plus petite. Dans le cas contraire, on choisit la solution qui obtient le plus petit minimum de l'erreur de transfert $err_t(\hat{\mathbf{H}}_2(\mu)), \mu \in]0, \pi[$.

[2] Les solutions physiquement correctes sont obtenues à partir des coordonnées image d'un seul point du plan visible dans les deux images : n'importe quel point de $\mathbf{P}_k$ vérifiant la condition de visibilité convient.

Pour $m = 2$ et $m = 1$, les mêmes opérations peuvent être répétées en inversant le rôle de π_1 et π_2. Deux solutions théoriquement correctes sont alors obtenues, le critère 6 pouvant être utilisé pour les départager.

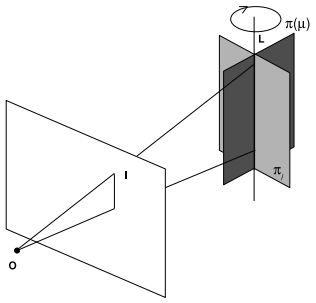

FIG. 3 – L'intersection **L** entre π_1 et le plan défini par **l** et **O** est contenue dans une famille de plans à un paramètre.

3 Système interactif

Nous présentons dans cette section un système de RA interactif, reposant entièrement sur la méthode de SFM présentée en section 2. La version actuelle du système est prévue pour des scènes composées d'un sol horizontal et de murs verticaux, cette hypothèse étant presque toujours vérifiée en environnement intérieur, et souvent vérifiée, au moins localement, en milieu urbain. Si le système est assez proche dans le principe de celui présenté dans [14], un gain important est obtenu en confort d'utilisation. Dans [14], les bases des plans et les régions à suivre étaient indiquées à l'aide de la souris, et les sommets d'un motif rectangulaire devaient être désignés pour initialiser le point de vue de la caméra. Ces opérations nécessitaient (i) de tenir une souris à la main (ce qui n'est pas possible avec un dispositif de RA de type PDA ou téléphone portable puisque la main de l'opérateur est déjà utilisée pour tenir l'appareil portable et viser la scène) et (ii) de ne pas bouger la caméra au moment de la désignation des différents éléments (ce qui est difficile à obtenir lorsque la caméra est tenue à la main ou posée sur la tête de l'utilisateur). Le nouveau mode d'interaction que nous proposons est mieux adapté aux dispositifs et applications de RA actuels. Le point de vue initial ainsi qu'une modélisation 3D de la scène observée sont obtenus en visant certains éléments de la scène (la caméra pouvant bouger à tout moment), puis en validant ou annulant les solutions proposées par le système.

Les différents états du système et les transitions entre ces états sont indiqués en figure 4. Les flèches dans les quatre directions représentent les quatre différentes impulsions que l'utilisateur doit communiquer au système pour changer d'état ou progresser dans un même état. Dans notre implémentation, elles correspondent effectivement à un appui sur l'une des touches directionnelles du clavier, mais cela peut se traduire différemment suivant le contexte d'utilisation (boutons d'une télécommande, mouvements d'un joystick, etc.). L'utilisation des flèches est intuitive : la flèche de gauche est utilisée pour annuler un résultat proposé par le système, la flèche de droite pour le valider ; la flèche du bas permet de définir la région du sol (en bas dans l'image), la flèche du haut la région du mur (en haut dans l'image). Le système peut se trouver dans trois états principaux, que nous détaillons ci-dessous.

Définition des polygones. Deux fenêtres carrées centrées horizontalement, de la taille de la demi-hauteur de l'image, sont affichées en permanence dans le champ de vision de l'opérateur : l'une dans la partie haute de l'image, l'autre dans la partie basse (cf. Fig. 6). Lorsque l'utilisateur appuie sur la flèche du haut, le carré du haut est "gelé" comme faisant partie d'un mur (l'utilisateur doit donc au préalable orienter la caméra de manière à ce que ce soit le cas), puis immédiatement suivi à l'aide de points de Harris détectés à l'intérieur du carré[3]. À chaque pression sur la flèche du haut, le carré du haut est fusionné au polygone en cours de suivi (on prend l'enveloppe convexe des sommets du polygone et du carré). L'utilisateur peut aussi laisser le doigt appuyé sur la flèche du haut tout en bougeant la caméra de manière à couvrir rapidement une large zone du mur. Le principe est le même pour le carré du bas, qui permet de définir le sol. Dans la suite, $\mathbf{H}_1$ et $\mathbf{H}_2$ désignent les homographies induites respectivement par le sol et par le mur.

Reconstruction initiale de la scène. Lorsqu'un polygone existe à la fois pour le mur et pour le sol, on passe à l'étape de filtrage de l'intersection des deux plans. L'intersection **l** estimée est affichée, et l'utilisateur peut la valider en appuyant sur la flèche de droite. Si la convergence est trop lente ou que le filtre diverge, l'utilisateur peut réinitialiser le filtrage (nouveau tirage uniforme des particules, remise à l'identité de $\mathbf{H}_1$ et $\mathbf{H}_2$) en appuyant sur la flèche de gauche. Dès que l'intersection est connue, le positionnement biplan commence : l'homographie $\mathbf{H}_1$ est décomposée, et si une solution au moins existe pour la structure du plan, la base canonique $(O, \vec{i}, \vec{j}, \vec{k})$ du repère du monde est affichée en utilisant la ou les solutions trouvées (O est placé au milieu de la partie de **l** située à l'intérieur de l'ellipse $\mathcal{E}$, $\vec{i}$ est sur **l** et $\vec{k}$ est la normale au plan calculé, cf. Fig. 6.(a)). L'utilisateur appuie sur la flèche de droite si la solution affichée ou l'une des deux solutions lui paraît correcte, le système retenant la solution qui minimise le critère (6) dans le second cas. Comme précédemment, l'utilisateur peut à tout moment réinitialiser les homographies $\mathbf{H}_1$ et $\mathbf{H}_2$ en appuyant sur la flèche de gauche. Remarques :

– le filtre n'est pas mis à jour dans le cas où les homographies $\mathbf{H}_1$ et $\mathbf{H}_2$ sont proches. Cela se produit lorsque la caméra est fixe ou subit un mouvement de rotation pure, ou lorsque les deux polygones appartiennent au même plan[4]. Une telle situation est facilement détectable : elle

[3] Les points de Harris sont appariés automatiquement entre images consécutives du flux d'images, ce qui permet de calculer les homographies inter-images par RANSAC – voir [15] pour les détails.

[4] Le dernier cas ne doit pas arriver si l'utilisateur maîtrise le fonctionnement du système, mais il peut se produire dans le cadre de la détection

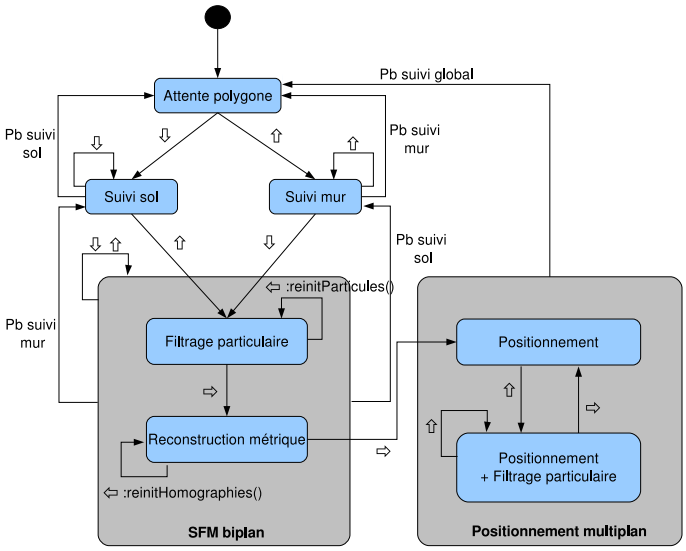

FIG. 4 – Diagramme d'états du système interactif.

correspond au cas où les appariements à l'intérieur du polygone $\mathbf{P}_1$ sont compatibles avec l'homographie $\mathbf{H}_2$, et réciproquement ;

- l'affichage des deux solutions calculées plutôt que de la meilleure des deux permet d'éviter des "sautillements" entre les deux solutions. L'utilisateur peut ainsi plus facilement ajuster sa position en fonction de la tendance de la solution qu'il juge la meilleure. De même, nous ne décomposons pas l'homographie $\mathbf{H}_2$ pour ne pas ajouter à la complexité de l'affichage ;
- une hauteur approximative de caméra est donnée en début de session (la taille de l'opérateur dans le cas où la caméra posée sur la tête), qui est utilisée pour fixer l'échelle de la scène.

Augmentation et extension de la scène. Une fois la reconstruction validée, on se trouve en mode classique de positionnement multi-plan et augmentation simultanée de la scène [15]. D'autres murs peuvent être définis en les visant avec la caméra à travers le carré du haut, puis en calculant leur intersection avec le sol par filtrage particulaire (les objets virtuels peuvent continuer à être observés pendant cette opération). Lorsqu'une intersection de plans est validée, les polygones ayant permis de l'obtenir sont rognés de manière à ne pas dépasser des plans ainsi délimités.

4 Détection automatique des plans

Nous voyons à présent comment étendre cette méthode à la détection automatique et causale de régions planes texturées, dans un flux d'images non calibrées. Nous supposons qu'une partie du sol est visible dans le tiers inférieur de l'image, et une partie des plans à détecter dans la moitié supérieure (Fig. 5). Dans cet article, nous présentons uniquement des résultats de segmentation 2D, mais l'algorithme proposé devrait pouvoir être utilisé pour obtenir un système de SLAM complet, comprenant la reconstruction métrique de la scène (nous discutons de ce point en conclusion).

Travaux antérieurs. La détection automatique de plans dans une séquence d'images est un problème difficile, dont la résolution passe souvent par un réglage délicat de paramètres peu discriminants. Ainsi dans [11], des couples point - segment appariés entre deux images sont utilisés pour calculer des homographies, et compter le nombre de primitives de l'image compatibles avec ces homographies. Les primitives compatibles avec l'homographie obtenant le plus large consensus sont considérées comme appartenant à un même plan, et le processus est répété sur les primitives restantes. Ce type d'approche est malheureusement confronté au problème d'une distribution hétéroscédastique des erreurs de transfert par une homographie plane. Il est donc difficile de trouver une mesure de compatibilité à l'homographie qui ne conduise pas à un nombre élevé de faux positifs (seuil trop permissif) ou de faux négatifs (seuil trop restrictif). Ce problème est particulièrement sensible sur les régions frontières entre les plans, mais aussi sur les points les plus éloignés de la caméra, présentant une faible disparité.

L'approche que nous proposons est tout autre : elle consiste à introduire des patchs supposés plans dans la moitié supérieure du flux d'images, à suivre ces patchs par ESM (*Efficient Second-order Minimization* [2]), et à les regrouper en fonction de leur intersection avec le sol. Nous détaillons ci-dessous les différentes étapes de cette méthode.

Gestion des patchs. Initialement, $\min(P_{\max}, s \times t)$ patchs rectangulaires sont introduits sur une trame de taille $s \times t$ (Fig. 5), où $P_{\max}$ est le nombre maximal de patchs utilisés (des valeurs typiques des constantes introduites sont données en table 2). Lorsqu'un patch est éliminé (son suivi par ESM a échoué), un nouveau patch est introduit sur la trame, à un endroit non déjà recouvert (complètement ou partiellement) par un patch existant. Parmi les rectangles libres, on choisit celui qui contient le plus grand nombre de points de Harris (cela évite de choisir des patchs dans des zones uniformes ou peu discriminantes). Des patchs sont aussi introduits quand un rectangle de la trame est libre et que $P_{\max}$ n'est pas atteint. À chaque patch p sont associées deux homographies, calculées entre l'image où le patch a été introduit et l'image courante : une homographie liée au patch, donnée par ESM, et une homographie liée au sol, obtenue par composition des homographies inter-images calculées dans le tiers inférieur de l'image. Ces homographies sont obtenues par RANSAC, ce qui permet d'être robuste à la présence d'éléments de la scène n'appartenant pas au sol dans la zone considérée.

Clustering. La méthode de filtrage particulaire présentée en section 2.3 peut être appliquée à chacun des patchs, produisant P droites $\mathbf{l}'_p$, supposées être l'intersection dans l'image courante du sol avec les plans auxquels appartiennent les patchs. Les patchs sont alors regroupés par

automatique de plans présentée en section 4.

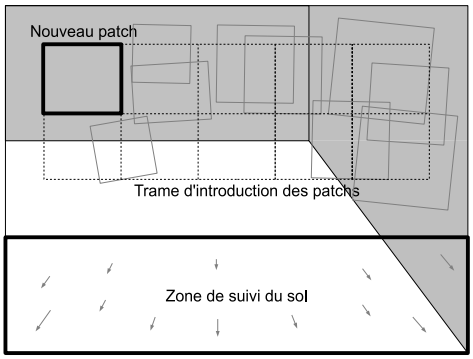

FIG. 5 – Zones utilisées pour la segmentation automatique.

clustering de l'ensemble $\{\mathbf{l'}_p\}_{1\leq p\leq P}$. On considère uniquement les patchs dont la droite associée est stable (selon la distance utilisée pour le clustering) depuis au moins C images. Un patch est intégré à un cluster existant si la distance entre la droite qui lui est associée et la dernière droite intégrée au cluster est inférieure au rayon du clustering R (les distances sont calculées sur les points d'intersection de la droite avec l'ellipse $\mathcal{E}$). Les droites restantes forment de nouveaux clusters, des clusters de taille maximale étant obtenus en utilisant l'algorithme QT [7]. Un cluster est donc constitué d'un ensemble de patchs et de la droite correspondant au dernier patch intégré, cette droite étant propagée d'image en image en utilisant l'homographie inter-images associée au sol. Le fait d'utiliser la droite associée au dernier patch intégré permet de corriger la dérive éventuelle à chaque nouvelle intégration de patch. Un plan est détecté quand un cluster atteint un nombre minimal de patchs T. Ce critère de consensus permet d'écarter naturellement les patchs mal suivis, non plans ou pour lesquels le filtrage diverge. Suivant les besoins de l'application, on peut aussi calculer l'enveloppe convexe des patchs d'un même clusters, ou encore découper les patchs (ou l'enveloppe convexe) au niveau de la droite d'intersection du cluster.

5 Résultats expérimentaux

Cette section présente des résultats obtenus avec le système interactif et la méthode de segmentation automatique. La table 1 donne les valeurs des constantes utilisées pour le filtrage particulaire dans les deux cas.

Cons	Valeur	Description
N	1000	Nombre de particules
N_t	1000	Seuil de rééchantillonage
σ_h	3.33	Écart-type de la mesure liée aux homographies
σ_g	10.	Écart-type de la mesure liée au gradient de l'image
$\mathbf{V}$	$([0.01, 0.01, 5]\mathbf{I})^2$	Matrice de covariance de la loi de dynamique
M	100	Nb de droites retenues dans la transformée de Hough

TAB. 1 – Constantes du filtrage particulaire.

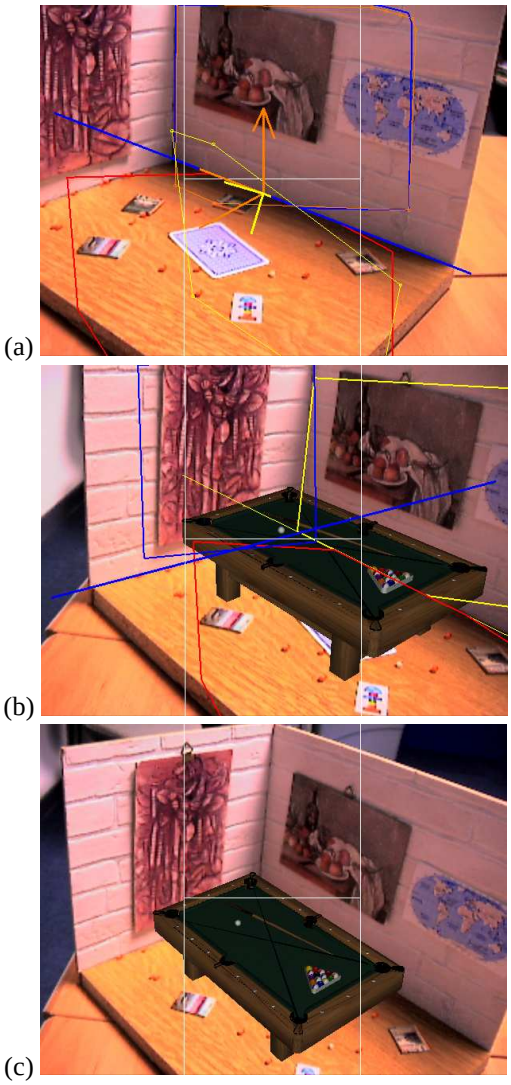

FIG. 6 – Système interactif. (a) Initialisation du point de vue de la caméra : l'intersection des deux plans est utilisée pour lever l'ambiguïté du positionnement mono-plan. (b) La scène est augmentée en même temps qu'un nouveau plan est reconstruit. (c) Augmentation obtenue depuis un autre point de vue.

5.1 Système interactif

L'utilisation du système interactif est illustrée sur une séquence de 1200 images, disponible sur le serveur web du Loria [5]. La scène utilisée est un décor d'intérieur miniature, constitué d'un sol et de deux murs à angle droit (Fig. 6). Les surfaces ne sont pas parfaitement planes, des éléments décoratifs texturés ajoutant de légers reliefs à certains endroits. La caméra est utilisée en résolution 320×240. Ses paramètres intrinsèques sont obtenus par la méthode de Faugeras-Toscani [5] en utilisant une mire de calibration. Le fréquence de calcul est de 10Hz (ralentie à 4Hz quand la transformée de Hough est calculée, les deux mesures de

[5] `http ://www.loria.fr/~gsimon/rfia08/easyslam.avi`, les vidéos sont encodées en MPEG-4 V2.

vraisemblance étant prises en compte dans le filtrage particulaire) sur une machine usuelle achetée il y a quatre ans (Dell Precision 360, Pentium 4 - 3GHz).
Deux polygones sont d'abord définis, sur le sol et sur le mur de droite. La droite d'intersection des deux plans est alors calculée (Fig. 6.(a)) en environ deux secondes (21 images). Les solutions de reconstruction métrique sont alors affichées, puis validées par l'opérateur au bout de 6 secondes. À cet instant, un billard virtuel est ajouté dans la scène, à une distance de 1 (pour le centre du billard) du mur (l'échelle est donnée par la hauteur de la caméra exprimée en décimètres en début de session et à peu près constante jusqu'à la validation de la reconstruction métrique). Un nouveau polygone est ensuite défini sur le mur de gauche, et sa droite d'intersection avec le sol (Fig. 6.(b)) est obtenue en environ 10 secondes (103 images). La caméra subit ensuite des mouvements variés, faisant apparaître la scène sous différents angles, et à différentes distances de la caméra (voir par exemple l'image en figure 6.(c)). Durant toute cette séquence, le billard présente une perspective cohérente avec le reste de la scène, et semble bien ancré au sol (aucun glissement ni de tremblement ne sont observés[6]).
Deux autres vidéos sont disponibles sur notre serveur web : la première[7] atteste de la robustesse du filtrage particulaire (utilisant la transformée de Hough) aux occultations partielles de la droite d'intersection ; la deuxième[8] montre la convergence du filtre en utilisant uniquement la mesure liée aux homographies : on voit que dans ce cas, le filtre met beaucoup plus de temps à converger.

5.2 Détection automatique des plans

Des résultats de détection automatique de plans sont montrés sur une séquence de 700 images d'un couloir du Loria (Fig. 7). La texture naturelle de la scène est utilisée pour le suivi des patchs et du sol (la moquette au sol, le parquet au mur, les cadres des portes et fenêtres, quelques posters, ...). Aucun élément n'a été ajouté à l'environnement tel qu'il se présentait avant l'expérimentation. L'utilisateur se déplace, caméra à la main, en visant différents endroits du couloir : il commence par regarder deux murs contigus, formant un angle de 150 degrés, situés sur le côté droit du couloir (Fig. 7.(a)), tout en avançant vers l'autre extrémité du couloir (le premier mur disparaît rapidement, le second est observé durant un assez long moment). L'utilisateur regarde ensuite vers le bout du couloir, qui est occulté par un escalier montant, puis se tourne vers le mur de gauche (Fig. 7.(c)). La table 2 donne les valeurs des constantes de l'algorithme utilisées pour cette séquence. Les images ont été acquises à l'aide d'un camescope numérique de résolution 720×576, puis traitées hors ligne à la fréquence d'environ 0.2 Hz (le niveau de précision requis pour le suivi des patchs ne permet pas d'exécuter ESM en temps réel sur la machine décrite précédemment).

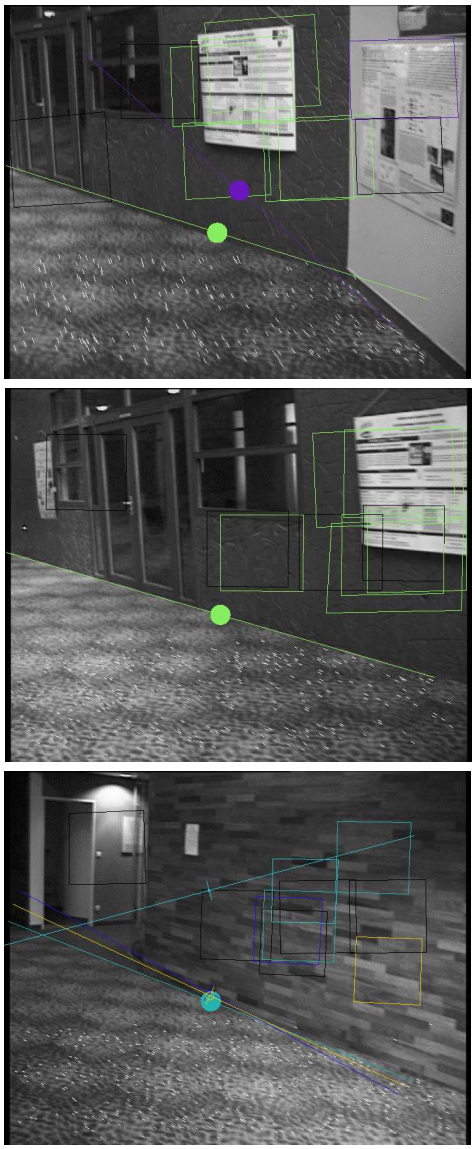

FIG. 7 – Clusters atteignant le seuil d'intégration sur la séquence du couloir.

Le résultat est visible sur une vidéo déposée sur notre serveur web[9]. Une couleur identique est utilisée pour les patchs et la droite d'intersection d'un même cluster (les patchs associés à une droite d'intersection non stabilisée apparaissent en noir). Un cercle est dessiné au milieu de la droite d'intersection, dont le rayon est proportionnel au nombre de patchs intégrés au cluster. Le cercle se remplit de la couleur du cluster et cesse de s'agrandir quand le seuil d'intégration T est atteint. La figure 8 montre l'évolution de la taille des clusters sur la séquence : 218 clusters ont été créés au cours de la séquence, dont la plupart n'obtiennent jamais plus d'un ou deux patchs. Trois clusters dépassent largement du niveau maximal des autres clusters,

[6] un décalage d'*affichage* entre la vidéo et les objets graphiques est parfois perceptible, mais il s'agit uniquement d'un problème de synchronisation entre le système temps réel et le logiciel utilisé pour capturer la vidéo.
[7] `http ://www.loria.fr/gsimon/rfia08/convocc.avi`
[8] `http ://www.loria.fr/gsimon/rfia08/convsg.avi`
[9] `http ://www.loria.fr/gsimon/rfia08/autoslam.avi`

Const.	Valeur	Description
$P_{\max}$	10	Nombre maximal de patchs introduits
$s \times t$	2×5	Résolution de la trame d'introduction des patchs
C	10	Seuil de stabilté des droites d'intersection (nb images)
R	15	Rayon du clustering (pixels)
T	10	Seuil d'intégration d'un cluster (nb patchs)

TAB. 2 – Constantes de la segmentation automatique.

qui correspondent aux trois murs présents dans la scène (Fig. 7(a)-(c)). Ce graphique montre que la méthode est peu sensible au choix du seuil T. Un autre point intéressant est que le mur de gauche est détecté, alors que des mouvements de caméra rapides, faisant échouer le suivi du sol et/ou de presque tous les patchs, ont lieu à plusieurs reprises durant la séquence. Cela est dû au fait que de nouveaux patchs sont introduits dès que le suivi d'autres patchs échoue, ces patchs pouvant former de nouveaux clusters indépendamment de ce qu'il s'est produit avant leur introduction. Le système peut donc fonctionner indéfiniment, puisqu'il "s'auto-réinitialise" à la volée. En revanche, on s'aperçoit en regardant la séquence, que la droite d'intersection du cluster correspondant au deuxième mur de droite finit par s'écarter de l'intersection effectivement observée dans l'image : cela est dû au fait que ce cluster est filmé pendant une longue période, sans que de nouveaux patchs soient introduits au cluster.

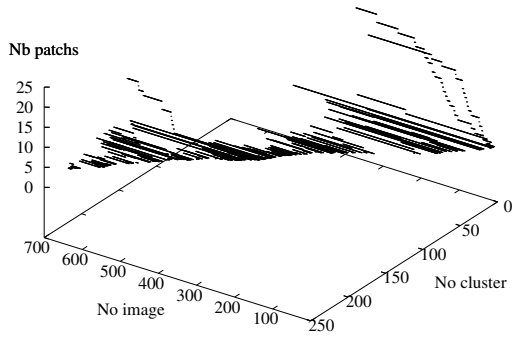

FIG. 8 – Évolution de la taille des clusters au cours de la séquence.

6 Conclusion

Dans cet article, nous avons présenté un système de RA permettant à un utilisateur de définir et augmenter simultanément une scène plane par morceaux. Les opérations à effectuer sont simples et intuitives, et reposent sur un mode d'interaction adapté aux dispositifs de RA actuels. La partie projective de ce système a été étendue pour permettre une détection automatique des plans dans un flux d'images non calibrées. La compréhension de la scène par l'opérateur, mise à profit dans le système interactif, est alors remplacée par la recherche d'un consensus entre des patchs rectangulaires introduits arbitrairement dans le partie supérieure de l'image.

Le système interactif peut être amélioré sur plusieurs points. En particulier, la méthode de SFM gagnerait sans doute en précision si un faible nombre d'images, au lieu d'une seule actuellement, était utilisé pour affiner la solution obtenue localement par un ajustement de faisceau global. D'autre part, on peut envisager de faire croître automatiquement les plans définis initialement par l'utilisateur, en utilisant la connaissance des frontières des plans pour limiter leur extension.

L'intégration de l'algorithme de détection de plans à un système de SLAM entièrement automatique (pour une utilisation en robotique par exemple) nous paraît largement envisageable, au regard des résultats déjà obtenus avec le système interactif. Le point crucial concernera la manière dont sera validée la solution de SFM, en l'absence d'expertise humaine : une attente de consensus ou un ajustement global entre plusieurs plans seront sans doute nécessaires.

Références

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2) :174–188, February 2002.

[2] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *EEE/RSJ International Conference on Intelligent Robots Systems, Sendai, Japan*, oct 2004.

[3] Andrew J. Davison and David W. Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on PAMI*, 24 :865–880, 2002.

[4] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. Rapport de recherche 856, INRIA, 1988.

[5] O. D. Faugeras and G. Toscani. Camera Calibration for 3D Machine Vision. In *Proceedings of International Workshop on Machine Vision and Machine Intelligence*, Tokyo, 1987.

[6] L. Van Gool, M. Proesmans, and A. Zisserman. Grouping and Invariants using Planar Homologies. In *Workshop on Geometrical Modeling and Invariants for Computer Vision*, 1995.

[7] L.J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data : Identification and analysis of coexpressed genes. *Genome Research*, 9(11) :1106–1115, 1999.

[8] H. Jin, P. Favaro, and S. Soatto. A semi-direct approach to structure from motion. *The Visual Computer*, 19(6) :377–394, 2003.

[9] Björn Johansson. View synthesis and 3d reconstruction of piecewise planar scenes using intersection lines between the planes. In *ICCV*, pages 54–59, 1999.

[10] M. Lourakis and A. Argyros. Efficient, causal Camera Tracking in Unprepared environments. *Computer Vision and Image Understanding*, 2005.

[11] M. Lourakis, A. Argyros, and S. Orphanoudakis. Detecting planes in an uncalibrated image pair. In *In Proc. of BMVC'02, volume 2, pages 587–596*, 2002.

[12] N. Molton, A. Davison, and I. Reid. Locally planar patch features for real-time structure from motion. In *British Machine Vision Conference*, 2004.

[13] C. Schmid and A. Zisserman. Automatic line matching across views. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 666–671, 1997.

[14] Gilles Simon and Marie-Odile Berger. Reconstructing while registering : a novel approach for markerless augmented reality. In *International Symposium on Mixed and Augmented Reality, Darmstadt (Germany)*, September 2002.

[15] Flavio Vigueras, Marie-Odile Berger, and Gilles Simon. Iterative multi-planar camera calibration : Improving stability using model selection. In Eurographics Association, editor, *Vision, Video and Graphics (VVG)'03, Bath, UK*, Jul 2003.

[16] Daniel Wagner and Dieter Schmalstieg. Handheld augmented reality displays. In *VR '06 : Proceedings of the IEEE Virtual Reality Conference (VR 2006)*, page 67, Washington, DC, USA, 2006. IEEE Computer Society.

[17] G. Xu, J. Terai, and H. Shum. A Linear Algorithm for Camera Self-Calibration, Motion and Structure Recovery for Multi-Planar Scenes from Two Perspective Images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina (USA)*, 2000.