Suivi temps-réel de séquences vidéo dans un panoramique pour le codage par objets

M. Douze[1] Ph. Puech[1] V. Charvillat[1] J. Conter[1]

[1] IRIT (Institut de Recherche en Informatique de Toulouse)

UMR CNRS 5505, ENSEEIHT
2, rue Camichel, 31071 Toulouse, FRANCE

{douze, puech, charvi, conter}@enseeiht.fr

Résumé

Nous présentons un système de suivi en temps réel qui utilise une image panoramique de la scène comme référence. Ce panoramique est acquis à l'aide d'une caméra spécialisée que nous avons développé. Le suivi est robuste, dans le sens où il n'est pas perturbé par des objets de petite taille qui se déplacent par rapport au fond. On peut donc segmenter ces objets, ce qui permettra de coder par objets le flux vidéo.

Mots clefs

suivi, temps-réel, panoramique, codage vidéo, multi-source

1 Introduction

Nous présentons dans cet article une technique de suivi de trames vidéo au sein d'une image panoramique préalablement acquise par un capteur spécifique. Dans des conditions expérimentales très favorables, nous pouvons ainsi résoudre en temps réel les problèmes délicats des nouvelles approches géométriques du codage de la vidéo, proposées par exemple dans le contexte normatif MPEG-4 :
– compensation du mouvement dominant,
– segmentation spatio-temporelle d'objets,
– codage vidéo par objets.
L'approche retenue est un suivi basé sur :
– l'apparence de l'environnement,
– un modèle de mouvement homographique.
Le papier se décompose en trois étapes majeures. Dans un premier temps, les conditions expérimentales de prise de vue panoramique sont détaillées (section 2), la modélisation et la résolution du suivi sont résumées dans les sections 3 et 4, les expériences et les applications à la vidéo terminent l'exposé.

2 La prise de vue panoramique

Réaliser un panorama à partir d'une mosaïque de vues présente plusieurs inconvénients :
– Les images élémentaires sont prises à des instants différents et sont donc sensibles aux variations de l'environnement (luminosité notamment).
– Les images élémentaires doivent être cadrées avec soin
– certaines aberrations géométriques sont difficilement compensables.
La caméra panoramique remédie partiellement à ces difficultés.

2.1 Caméra Panoramique à balayage

Un balayage continu permet d'obtenir en une seule prise un panorama sans aucun raccord. Le rebouclage du panorama sur lui-même ne pose aucun problème géométrique. Les aberrations géométriques liées à l'optique n'apparaissent que dans le sens vertical. La courbure de la perspective est régulière et connue ; elle est due à la variation continue de l'azimuth de prise de vue. Elle peut être aisément corrigée sur des vues partielles.

2.2 Le prototype

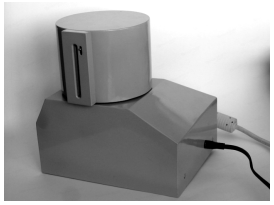

Figure 1 – *Notre prototype de caméra panoramique*

Un capteur CCD trichrome linéaire (type barrette) monté verticalement et muni d'une optique grand angle est mis en rotation autour d'un axe vertical coplanaire de l'axe optique horizontal. La course totale de 380° assure une prise de vue continue et complète de la scène. La résolution verticale est celle du CCD soit 2545 pixels couvrant un champ de 60°. La résolution horizontale est fonction du balayage mécanique. Le temps de prise de vue est donc lié à cette résolution, et au mode de transfert utilisé. L'obtention via un port parallèle de 3500 pixels horizontaux prend 70 se-

condes. La nature du capteur CCD à triple barrette RVB nécessite un réalignement très simple des 3 plans couleurs.

2.3 La géométrie

L'image obtenue par la caméra panoramique est en projection cylindrique. Or, il nous faut des images issues d'une projection perspective classique pour pouvoir faire une mise en correspondance à base d'homographie. Donc, il faut « redresser » le panoramique pour simuler ce type de projection. On convertit le panoramique en 4 images utilisables qui couvrent chacune 90° du champ de vue (figure 2).

Figure 2 – *Panoramique pris avec la caméra, et les 4 images redressées.*

3 L'algorithme de suivi

Ensuite, on filme la scène avec une caméra vidéo classique, et on veut recaler la séquence par rapport au panoramique. Ceci est possible à plusieurs conditions :

– la scène est immobile (cette condition est relaxée à la section 5),
– le centre optique de la caméra qui filme la séquence est immobile, et coïncide avec le centre optique de la caméra panoramique.

3.1 Exposé du problème

Après le « redressement » du panoramique, on peut mettre en correspondance le panoramique avec une trame de la séquence video à l'aide d'une homographie. Un point de la scène qui se projette en (x, y) sur le panoramique I a sur la trame I_t de la séquence vidéo les coordonnées :

$$H\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} \frac{h_1 x + h_2 y + h_3}{h_7 x + h_8 y + 1} \\ \frac{h_4 x + h_5 y + h_6}{h_7 x + h_8 y + 1} \end{bmatrix}$$

La transformation homographique H est paramétrée par les coefficients h_i qu'il faut estimer pour toutes les images de la séquence.
Pour résoudre ce problème, nous allons d'abord examiner un problème de suivi classique, et ensuite adapter sa solution au suivi de panoramiques.

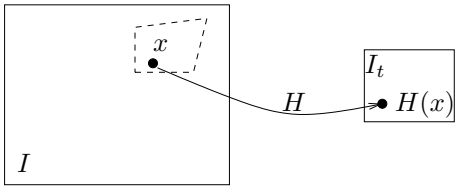

Figure 3 – *La mise en correspondance d'une image panoramique I avec une image I_t de la séquence vidéo.*

3.2 Le suivi à base de points avec un modèle homographique

Les données du problème sont une séquence d'images $(I_t)_{t\in\mathbb{N}}$, et une zone d'intérêt J_0 sur I_0. Cette zone d'intérêt, en forme de quadrilatère, est contenue dans l'image de l'objet de la scène qu'on veut localiser dans les trames suivantes.

Modélisation. On suppose qu'on peut décrire le mouvement de l'image de l'objet par une homographie. C'est-à-dire que, à la trame t, on passe des points de J_0 à ceux de I_t par une homographie H_t dont les composantes sont $\mu_t \in \mathbb{R}^8$ (figure 4). Ce modèle est correct notamment dans le cas où l'objet suivi est une surface plane, la caméra et l'objet étant en mouvement.

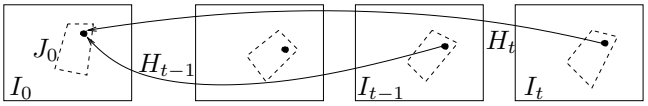

Figure 4 – *Mise en correspondance dans une séquence d'images.*

On suppose aussi que l'image d'un point donné de la scène ne change pas de couleur (ou, dans notre cas, de niveau de gris). On peut alors poser :

$$\forall x \in J_0, I_t(H_t(x)) = I_0(x)$$

où $I(x)$ est le niveau de gris de x sur l'image I.
On suppose par ailleurs que, entre les prises de vue, les mouvements des objets et de la caméra sont *petits*. Ceci induit que la différence $\Delta\mu_t = \mu_t - \mu_{t-1}$ est faible.
Le suivi consiste donc à trouver la suite de $(\mu_t)_{t\in\mathbb{N}}$ telle que, par exemple,

$$\mathcal{O}'_t(\mu_t) = \sum_{x\in J_0} (I(H_t(x)) - I(x))^2$$

soit minimal.

Résolution. Comme μ_0 est connu et μ_{t+1} est peu différent de μ_t, on va faire un suivi incrémental. À chaque trame t, on cherche $\Delta\mu_t$, en deux phases :

1. Phase d'apprentissage : On utilise l'image I_0. L'algorithme choisit n points de référence sur la zone d'intérêt : $K_0 = (x_1, ..., x_n)$. Il enregistre leurs niveaux de gris $G = (I_0(x))_{x\in K_0}$.

2. Phase de suivi : On traite l'image I_t en utilisant le résultat μ_{t-1} de l'image précédente. On mesure $G_t = (I_t(H_{t-1}(x)))_{x\in K_0}$. On cherche à retrouver $\Delta\mu_t$ à partir de $\Delta G_t = G_t - G$. Typiquement, ΔG_t est faible parce que $G = (I_{t-1}(H_{t-1}(x)))_{x\in K_0}$, et I_{t-1} est peu différent de I_t. Cette phase s'exécute en temps réel.

Dans ce contexte, on va optimiser $\mathcal{O}_t(\mu_t) = \sum_{x\in K_0}(I(H_t(x)) - I(x))^2$ en supposant que la solution sera peu différente de celle pour $\mathcal{O}'_t(\mu_t)$.

Choix des points de référence. Une estimation homographique fiable nécessite plusieurs dizaines de points.
On utilise un mélange de points d'intérêt de Harris et de points répartis (points *réguliers*) arbitrairement sur J_0 (figure 5).

Figure 5 – *Trame d'une séquence vidéo, et ses points de référence. Le cadre représente J_0, les points d'intérêt sont en noir, les points réguliers en gris.*

Estimer $\Delta\mu$. Il y a plusieurs méthodes pour estimer $\Delta\mu_t$ pendant la phase de suivi. Nous en avons implanté trois.
Dans la méthode *non-linéaire* [6], on estime $\Delta\mu_t$ en minimisant directement le critère $\mathcal{O}_t(\mu)$. Ce problème peut alors être résolu avec un algorithme itératif d'optimisation non-linéaire [7].
Cette méthode ne peut qu'améliorer la valeur du critère par rapport à l'estimation initiale, mais elle tombe très facilement dans des minima locaux. On ne peut rien précalculer dans la phase d'apprentissage.
Hager [1] propose une méthode plus rapide. Elle consiste à faire un développement limité de l'expression de $\mathcal{O}_t(\mu)$ en $\Delta\mu_t$. On obtient :

$$\mathcal{O}_t(\mu_t) \approx \| M_t(\mu_{t-1})\Delta\mu_t + \Delta G_t \|^2$$

où M_t est une matrice $8 \times n$ qui dépend des dérivées de l'homographie et du gradient de l'image. L'expression atteint son minimum pour $\Delta\mu_t = -M_t(\mu_{t-1})^+\Delta G_t$ (où $M^+ = (M^T M)^{-1}M^T$ est la pseudo-inverse de M).
Dans ce cas, on peut précalculer la valeur du gradient de l'image pour tous les pixels de I_0. On s'y ramène alors par un changement de repère. Hager propose une version accélérée de la méthode pour un modèle affine, mais on ne peut pas l'adapter au modèle homographique.
Le paramètre de cette méthode est la taille du voisinage utilisé pour calculer le gradient de I_0.

Dans la méthode de Jurie et Dhome [2], on commence par un apprentissage coûteux sur I_0. Pendant cet apprentissage, on fait plusieurs expériences qui consistent à perturber les points de K_0 par des homographies tirées aléatoirement : $\mu' = \mu_0 + \Delta\mu$. Ces auteurs postulent l'existence d'une relation linéaire approchée entre les différences ΔG de niveaux de gris observées et les valeurs des perturbations appliquées :

$$\Delta\mu \approx A\Delta G$$

On estime cette matrice A en faisant beaucoup d'expériences.
Pendant la phase de suivi, mesure ΔG_t et on calcule la perturbation de μ_0 qui aurait causé la même différence de niveaux de gris à l'aide de A. Ensuite, par une habile composition d'homographies, on peut se ramener au cas de I_t, et en déduire la perturbation $\Delta\mu_t$.
Les paramètres de cette méthode sont l'intensité de la perturbation appliquée aux coins de la zone J_0. Si celle-si est forte, on a un suivi peu précis (il « sautille ») mais robuste (il supporte de forts mouvements de la cible), et vice-versa.

Discussion. La méthode de Hager a les caractéristiques suivantes, qui sont encore plus saillantes pour la méthode non-linéaire :
– la phase d'apprentissage est courte,
– la phase de suivi est relativement coûteuse,
– il faut une bonne estimation initiale,
– le résultat est précis.
À l'opposé, la méthode de Jurie et Dhome est plus rapide lors de la phase de suivi mais la durée de la phase d'apprentissage est non-négligeable. Elle est paramétrable, puisqu'on peut, pendant la phase d'apprentissage, choisir l'ordre de grandeur des perturbations $\Delta\mu$.
L'algorithme qui donne finalement les meilleurs résultats enchaîne, pendant la phase de suivi, plusieurs étapes d'estimation de Jurie et Dhome, de plus en plus précises, puis une étape de Hager et une étape non-linéaire.

3.3 Adaptation au cas des panoramiques

Lors de la transposition, le panoramique I joue le rôle de I_0, et une méthode *ad hoc* (éventuellement interactive) fournit la position μ_1 de la première trame. Le problème est le choix de la zone d'intérêt J_0, parce qu'on n'est pas sûr de voir un même objet pendant toute la séquence vidéo.

Les tuiles. Nous utilisons donc *plusieurs* zones d'intérêt (des « tuiles ») qui couvrent l'image panoramique $I = \bigcup_{k=1}^{r} J_k$. On choisit des tuiles, éventuellement entrelacées, assez petites pour qu'on en voie au moins une sur chaque image de la séquence, mais suffisament grandes pour que l'estimation reste robuste. Pour chaque tuile J_k, on choisit un ensemble de points de référence K_k, et on fait une phase d'apprentissage (figure 6).
À l'étape t de la phase de suivi, on utilise l'estimation précédente μ_{t-1} pour trouver quelles sont les r_t tuiles qu'on voit en entier dans l'image I_{t-1}. Pour chaque tuile k concernée, on calcule un μ^k par une ou plusieurs des

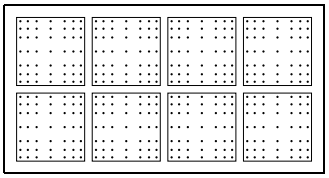

Figure 6 – *Points de référence réguliers sur des tuiles d'un panoramique*

méthodes décrites précédemment. Il faut ensuite combiner les μ^k pour trouver la valeur de μ_t.

Combinaison des résultats. La combinaison des résultats est assez simple, pour chaque tuile k, on calcule l'image par $f(\mu^k, .)$ des 4 coins du quadrilatère, ce qui fournit 4 paires de points correspondants. Chaque paire de points fournit deux équations. On a donc un sysème linéaire surdéterminé de $8r_t$ équations pour estimer les composantes de μ_t.
On peut résoudre ce système aux moindres carrés, mais une méthode plus robuste est exposée au 5.

4 Expériences

Après une validation sur des images de synthèsen, nous traitons en temps réel un flux vidéo DV sur un PowerPC G4. Il y a normalement 2 ou 3 tuiles tuiles par image, donc on a le temps d'appliquer plusieurs méthodes d'optimisation, voire de calculer $\mathcal{O}(\mu)$ pour choisir entre plusieurs résultats.

5 La robustesse

Lorsqu'on combine les résultats des différentes tuiles, il se peut que pour certaines tuiles, le suivi ait divergé. Il faut donc éviter de prendre en compte les équations qu'elles génèrent dans le système linéaire. Pour cela, nous utilisons la méthode FastLTTS [8] robuste pour éliminer les équations en faible quantité qui donnent des résultats différents des autres.
La détection des tuiles pour lesquelles le suivi échoue fournit des indices sur l'endroit où se trouvent les objets mouvants, qu'on va détecter par la suite.

6 Le codage vidéo

Nous voulons utiliser cette méthode pour faire du *streaming* d'objets vidéos indépendants (VOP). On transmet d'abord le panoramique, puis, dans deux flux séparé, les objets mouvants devant le fond, et leurs positions [4].
Pour isoler les objets qui changent entre les deux images, nous utilisons une simple différence pixel à pixel seuillée. Il faut cependant tenir compte des disparités de luminosité, de saturation et de bruit entre les signaux fournis par deux CCD très différents. Nous appliquons donc une correction chromatique calibrée lors de la phase d'apprentissage. Nous éliminons aussi les zones de différence trop petites pour êtres considérées comme des objets, à l aide d'une ouverture morphologique (figure 7).

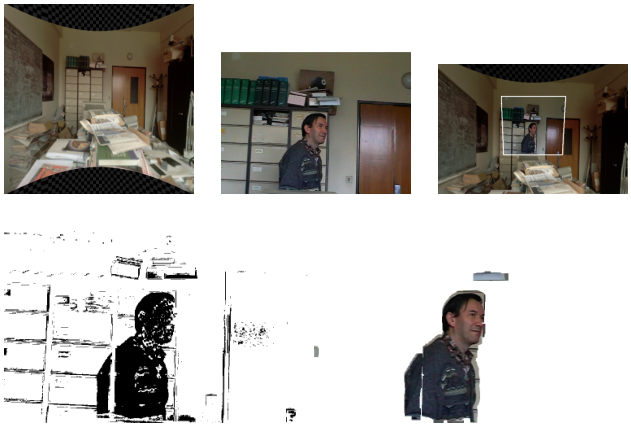

Figure 7 – *Panoramique, image de la séquence, image recalée, différence seuillée des deux, et « objet » extrait.*

7 Conclusion

Notre méthode de suivi est originale à plusieurs égards :
– elle utilise une image panoramique comme image de référence,
– la solution ne repose pas sur une technique, mais sur une hiérarchie de modèles de plus en plus précis,
– elle permet d'envisager un encodeur fond/forme temps réel pour MPEG-4.

Cependant, elle nécessite d'être soigneusement paramétrée.

Références

[1] Gregory D. Hager et Peter N. Belhumeur, *Efficient Region Tracking With Parametric Models of Geometry and Illumination*, IEEE pami, Vol. 20, oct 1999, p. 1025-1039

[2] F. Jurie, M. Dhome, *Un algorithme de "template matching" simple et efficace*, actes du congrès RFIA, Angers, 2002.

[3] M. Irani, P. Anadan, J. Bergen, R. Kumar, S. Hsu, *Efficient Representation of Video Sequences and Their Applications*, Signal Processing : Image Communication, spec. issue, Vol. 8, No. 4, May 1996.

[4] ISO, *Overview of the MPEG-4 Standard*, Melbourne, October 1999.

[5] *FLYSPEC : A multi-user Video Camera System with Hybrid Human and Automatic Control*, ACM Multimedia 2002, December 2002.

[6] M. Black et A. Jepson, *Eigentracking : Robuste matching and tracking of articulated objects using a view-based represesentation*, IEEE transactions on pattern analysis and Machine intelligence, 20(10) :1025-1039, octobre 1998.

[7] NetLib mirror site `http://www.enseeiht.fr`, ODRPack software.

[8] M. Douze, B. Thiesse et V. Charvillat, *Des mosaïques plus robustes, plus universelles*, conférence RFIA 2002, Angers, janvier 2002.