Extraction d'objets en mouvement par pyramide locale

Jérémy Huart, Guillaume Foret, Pascal Bertolino

Extraction d'objets en mouvement par pyramide locale

Jérémy Huart[1] Guillaume Foret[1] Pascal Bertolino[1]

[1] Laboratoire des Images et des Signaux

BP 46, 38402 Saint Martin d'Hères, France

{Jeremy.Huart, Pascal.Bertolino}@lis.inpg.fr, foret@ensea.fr

Résumé

Cet article présente une méthode (cf. figure 1) qui combine une extraction approximative des objets en mouvement et un raffinement de la segmentation de leurs contours. L'extraction des objets en mouvement est obtenue par une compensation de mouvement global classique. Pour obtenir des contours précis, une segmentation spatiale est effectuée à l'aide de la méthode originale de la pyramide de graphe locale, qui focalise le traitement de segmentation soit sur la zone de l'objet soit sur ses contours.

Mots clefs

Segmentation, pyramide locale, racines, compensation de mouvement.

1 Introduction

De nombreux travaux ont été (et sont encore) conduits sur le suivi d'objets dans les séquences vidéo. Dans de nombreuses approches, le suivi se déroule correctement si l'initialisation de l'entité à suivre a été minutieusement réalisée et si le masque de l'entité est localisé avec précision sur son vrai contour.

Cette précision peut parfois être obtenue manuellement. Cependant, l'interaction manuelle précise n'est guère appréciée par l'utilisateur car elle est laborieuse et coûteuse en temps. Par ailleurs, les objets d'intérêt ont souvent un mouvement qui est différent de celui de la caméra et peuvent donc être extraits approximativement de façon automatique [1].

Cet article présente tout d'abord le principe de la pyramide locale pour la segmentation spatiale dans des régions d'intérêt, puis la façon dont est réalisée la localisation automatique de ces régions d'intérêt. Enfin, des résultats illustrent le potentiel de cette approche.

2 La pyramide locale

2.1 Principe de la pyramide de graphe et de la pyramide locale

La pyramide de graphe [2] est un outil puissant qui fournit une segmentation en multirésolution en un seul traitement.

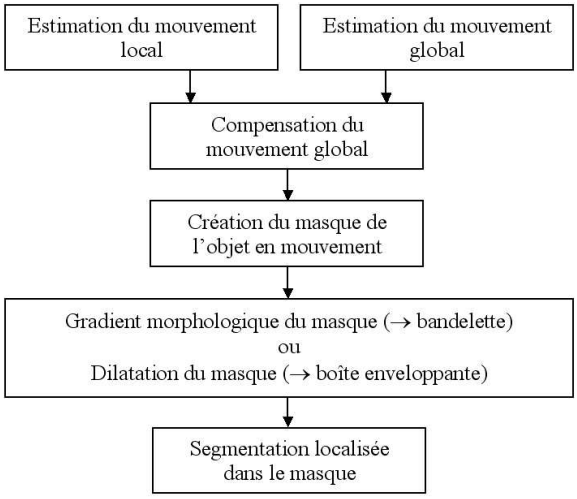


Figure 1 – *Schéma général de l'approche proposée*

Il est aussi utilisé pour effectuer la segmentation et le suivi d'objets dans les séquences vidéo [3].

Le principe de cette méthode est d'initialiser un graphe d'adjacence, où chaque sommet correspond à un pixel de l'image. Utilisant un algorithme local (c.à.d au niveau pixel) effectué sur l'image entière, les sommets adjacents similaires peuvent fusionner, entraînant une décroissance du nombre de sommets, chacun représentant alors un regroupement de pixels (régions).

Les régions i et j sont similaires si la distance entre leur couleur moyenne, dans l'espace YUV, est plus faible qu'un seuil donné : $d(YUV(R_i), YUV(R_j)) < T$.

Ce traitement est effectué itérativement sur des graphes successifs jusqu'à ce que plus aucune fusion ne soit possible.

Habituellement, le graphe d'adjacence est initialisé avec autant de sommets que de pixels dans l'image pour effectuer une segmentation de l'image entière (figure 2). Dans la pyramide locale que nous proposons, seul un sous-ensemble des pixels de l'image est associé à des sommets, tandis que le reste des pixels est associé arbitrairement à un,

voire à quelques sommets (figure 3). Dans ce cas, ces sommets sont considérés comme des *racines* (régions qui appartiendront au résultat final) tel que le fond par exemple.
La focalisation sur une zone particulière de l'image, nécessitant une interaction humaine, peut être réalisée en surlignant soit l'objet entier (à l'aide d'une boîte enveloppante) soit uniquement son contour (à l'aide d'une bandelette).

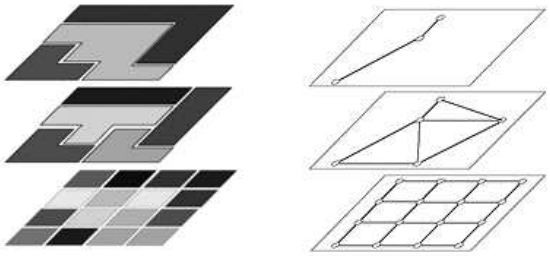


Figure 2 – *Exemple de pyramide de graphe construite sur une image 4 × 4 : les partitions et leur graphe*

Figure 3 – *Exemple d'une initialisation de pyramide locale*

2.2 Pyramide locale et boîte enveloppante

Dans cette première approche, une initialisation par boîte enveloppante classique est utilisée. Cette dernière est un contour fermé situé à l'extérieur de l'objet à segmenter. Ce contour peut être un rectangle, une ellipse ou un simple masque approximatif qui contient l'objet (figure 4.a). Tous les pixels se trouvant à l'extérieur de cette boîte sont considérés comme appartenant à un même objet racine (un seul sommet), tandis que les pixels contenus dans la boîte enveloppante représentent une zone indéfinie. Ces derniers forment la base d'une pyramide locale et sont segmentés afin qu'une partie d'entre eux (pixels similaires au fond) soit fusionnée avec la racine et que les autres fusionnent entre eux pour créer une, voire plusieurs régions.
Ce traitement permet non seulement d'extraire l'objet mais il en propose également une partition (figures 8.b et 9.b).

2.3 Pyramide locale et bandelette

Cette approche nécessite une connaissance approximative de la localisation du contour de l'objet : le vrai contour est sensé être sous la bandelette (figure 4.b) qui sert de base à la pyramide locale. La bandelette définit implicitement trois zones : l'extérieur, l'intérieur et la bandelette elle-même. Les pixels de l'extérieur appartiennent au fond (le premier sommet racine) tandis que l'intérieur appartient à l'objet d'intérêt (un second sommet racine). Tous les pixels formant la bandelette représentent la zone indéfinie. Ces

(a) Avec des boîtes enveloppantes

(b) Avec des bandelettes

Figure 4 – *Exemples d'initialisation de régions d'intérêt*

derniers sont segmentés afin qu'ils fusionnent avec l'un ou l'autre des sommets racines. Quelques-uns peuvent éventuellement fusionner entre eux sans jamais fusionner avec une des racines.

3 Localisation automatique des régions d'intérêt

Afin d'effectuer une segmentation locale automatique, la boîte enveloppante ou la bandelette doivent être positionnées automatiquement dans l'image. Ceci est réalisé à l'aide d'une analyse de mouvement entre deux images qui contiennent les objets d'intérêt à extraire. Les objets en mouvement sont supposés avoir un mouvement différent de celui induit par la caméra (mouvement global).

3.1 Estimation du mouvement local

L'estimation du mouvement local entre deux images est effectuée à l'aide d'un algorithme rapide du *block-matching* : Le *Block Sum Pyramid Algorithm* (BSPA) [4], fondé sur l'erreur de correspondance partielle et le *Successive Elimination Algorithm* (SEA) proposé par Li et Salari [5]. Ce traitement permet une estimation locale du mouvement entre deux images consécutives $I1$ et $I2$: un vecteur mouvement est classiquement assigné à chaque bloc carré de taille $M \times M$ de l'image. Pour nos expériences, des blocs de 8×8 pixels sont utilisés.

3.2 Estimation du mouvement global

Dans cette partie, nous calculons un modèle paramétrique du mouvement global. Ce modèle est obtenu en deux phases [6] : d'abord sur l'image entière puis plus précisément sur l'image entière sans les objets en mouvement. Un modèle de mouvement rigide à 4 paramètres est calculé entre $I1$ et $I2$ à l'aide de la transformation de Helmert qui inclut une translation (en x et y), une rotation et un facteur de zoom comme suit :

$$\begin{pmatrix} x_i'' \\ y_i'' \end{pmatrix} = \begin{pmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} a_3 \\ a_4 \end{pmatrix} \quad (1)$$

Les couples $(x_i'',\ y_i'')$ et $(x_i,\ y_i)$ représentent respective-

ment la position centrale du bloc i dans l'image prédite et dans l'image courante. a_1, a_2, a_3 et a_4 sont les valeurs des paramètres à déterminer.

D'autre part, la relation entre chaque bloc i et sa projection par block-matching peut être décrite comme :

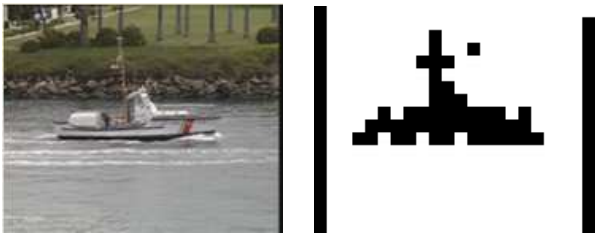
$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = \begin{pmatrix} a_{1_i} & -a_{2_i} \\ a_{2_i} & a_{1_i} \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} a_{3_i} \\ a_{4_i} \end{pmatrix} \quad (2)$$

les paramètres a_1, a_2, a_3 et a_4 doivent minimiser sur l'ensemble des N blocs, l'erreur quadratique Φ entre les positions $(x'_i,\ y'_i)$ estimées par le *block-matching* et les positions $(a_1x_i - a_2y_i + a_3, a_2x_i + a_1y_i + a_4)$ prédites par le modèle lui-même. Cette fonction de coût est définie par :

$$\Phi = \sum_{i=1}^{N} [(a_1x_i - a_2y_i + a_3 - x'_i)^2 + \quad (3)$$
$$(a_2x_i + a_1y_i + a_4 - y'_i)^2]$$

La minimisation du critère (3) est obtenue par les moindres carrés en utilisant la décomposition en valeurs singulières (SVD). Le code est disponible dans [7].

La distance Euclidienne seuillée entre les deux prédictions $(x'_i,\ y'_i)$ et $(x''_i,\ y''_i)$ nous permet de distinguer les blocs qui ne sont pas animés du mouvement global. L'estimation du mouvement global peut être réitérée (et donc raffinée) en ne prenant pas en compte ces derniers blocs. La distance Euclidienne est alors calculée une nouvelle fois pour obtenir un masque binaire temporel (figure 5) qui localise les blocs en mouvement des régions d'intérêt.

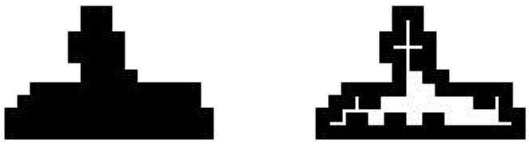


(a) Image originale (b) masque binaire temporel

Figure 5 – *Exemple de masque binaire temporel extrait par compensation de mouvement global*

3.3 Extraction automatique des régions d'intérêt

La boîte enveloppante est obtenue par dilatation des blocs du premier plan déterminés par l'analyse de mouvement (figure 6.a). La bandelette est le résultat de la soustraction de la dilatation et de l'érosion des blocs du premier plan (figure 6.b). Les blocs du premier plan situés à la périphérie de l'image sont rejetés afin d'éviter les problèmes d'occlusions et de désocclusions dus au mouvement de la caméra (figure 5.b).
Un seuillage sur une taille minimum ou un filtrage morphologique peuvent être utilisés pour éviter que trop de régions d'intérêt apparaissent dans le cas de vidéos bruitées. Néanmoins, la représentation par graphe permet autant de régions d'intérêt que possible, et ceci quel que soit le nombre d'objets partageant le même arrière plan.



(a) Boîte enveloppante (b) Bandelette

Figure 6 – *Régions d'intérêt obtenues automatiquement*

4 Résultats

Les expériences ont été réalisées sur des vidéos au format CIF (352×288 pixels). Avec un processeur Pentium IV cadencé à 2 GHz, le temps de traitement se situe entre une à deux secondes. L'analyse de mouvement est très rapide et la plus grande partie du temps de traitement est due à la segmentation. Pour les mêmes régions d'intérêt, la bandelette est environ deux fois plus rapide que la méthode de la boîte enveloppante puisque le nombre de pixels utilisé dans la construction de la pyramide est plus faible. Notons que le temps de traitement est proportionnel au nombre de pixels des zones indéfinies.

Les figures 7.b, 7.d, 8.a et 9.c montrent les résultats de la méthode appliquée sur des séquences vidéo classiques avec différents mouvements de caméra, différents fonds et des objets non rigides. Excepté pour la figure 9.c, aucun post-traitement n'a été appliqué. Le seuil de similarité T (section 2.1) doit encore être ajusté par l'utilisateur, même si dans la plupart des cas, une valeur par défaut donne de bons résultats. Un seuil adaptatif pourrait être calculé surtout dans le cas de la bandelette où l'information au niveau pixel est très concentrée et disponible.
Les figures 7.d et 8.a comparent (d'après le même masque temporel de la région d'intérêt) les résultats obtenus avec les deux approches. La bandelette permet une localisation plus précise des contours mais suivant le critère de similarité utilisé, des régions peuvent avoir des difficultés à fusionner avec l'une des deux racines. La figure 7.d indique en noir les régions qui n'ont pas pu fusionner ni avec l'objet ni avec le fond.

La boîte enveloppante fournit des résultats binaires (objet ou fond) mais avec une précision moindre (ceci étant dû à la segmentation qui est effectuée sur une large zone).

Quelle que soit l'approche choisie, la qualité du résultat dépend de la précision du masque de la région d'intérêt et de l'homogénéité de l'objet par rapport au fond : même si les objets et le fond sont peu contrastés, la méthode fonctionne bien si l'objet ou le fond (voire les deux) sont homogènes suivant le critère de similarité.

(a) Image originale

(b) Objet extrait (Régions non classées en noir)

(c) Image originale

(d) Objet extrait (Régions non classées en noir)

Figure 7 – *Extraction d'objet avec l'approche de la bandelette*

(a) Objet extrait

(b) Détails des régions qui forment l'objet

Figure 8 – *Extraction d'objet avec l'approche de la boîte enveloppante*

(a) Image originale

(b) Details des régions

(c) Après la suppression de régions

Figure 9 – *Extraction d'objet présent sur un fond texturé avec l'approche de la boîte enveloppante*

5 Conclusion

La qualité des masques obtenus a été testée avec succès dans une application de suivi d'objets [3]. Pour ce type d'applications, il est important que l'initialisation du masque soit faite automatiquement, c'est-à-dire en tenant compte des données numériques.
La localisation automatique de la boîte enveloppante ou de la bandelette peut être améliorée, par analyse temporelle de plusieurs images pour la création du masque binaire temporel et par lissage de la boîte enveloppante et de la bandelette.
Il serait intéressant d'utiliser ces méthodes pour générer des dictionnaires d'objets clé décrivant le contenu d'une vidéo ou pour les normes de compression.

Références

[1] A.M. Tekalp. *Digital Video Processing*. Prentice Hall, Inc, 1996.

[2] A. Montanvert, P. Meer, et A. Rosenfeld. Hierarchical image analysis using irregular tessellations. Dans *IEEE Trans. on PAMI*, volume 13(4), pages 307–316, April 1991.

[3] G. Foret et P. Bertolino. Label prediction and local segmentation for accurate video object tracking. Dans *Visual Communications and Image Processing, VCIP'03*, Lugano, Switzerland, 2003.

[4] C.H. Lee et L.H. Chen. A fast motion algorithm based on the block sum pyramid. *IEEE Transactions on Image Processing*, 6(11), November 1997.

[5] W. Li et E. Salari. Successive elimination algorithm for motion estimation. Dans *IEEE Transactions on Image Processing*, volume 4(1), pages 105–107, 1995.

[6] S. Liu, Z. Yan, J. W. Kim, et C. C. Jay Kuo. Global/local motion-compensated frame interpolation for low bitrate video. *Image and Video Communications and Processing*, pages 223–234, 2000.

[7] Cambridge university Press. Numerical recipes in c : The art of scientific computing. Website. `http://gpiserver.dcom.upv.es/Numerical_Recipes/bookc.html`.