Codage interpolatif de séquences d'images utilisant un suivi temporel de segmentation spatio-temporelle

Laurent Bonnaud et Claude Labit
IRISA/INRIA-Rennes, Campus de Beaulieu, 35042 Rennes Cédex,
E-mail: <nom>@irisa.fr

Résumé

Nous présentons un nouvel algorithme d'interpolation temporelle utilisant une segmentation en régions polygonales ayant un mouvement affine. Le but de ce travail est d'améliorer le codage interpolatif existant dans la norme MPEG (B-*frames*). Dans une première partie, nous décrivons brièvement l'algorithme de suivi temporel qui fournit la segmentation et les paramètres des mouvements affines. Dans une seconde partie, nous regardons comment utiliser cette segmentation dans un but d'interpolation, et présentons un schéma de codage avec en particulier le traitement des zones d'occlusion. Nous montrons la prédiction ainsi obtenue et nous comparons sa qualité avec d'autres schémas (*block-matching*, prédiction causale) à la fois visuellement et en termes d'EQM.

Abstract

This paper presents a new temporal interpolation algorithm based on segmentation of images into polygonal regions undergoing affine motion. The goal of this work is to improve upon the block-based interpolation used in MPEG (B-frames). In the first part, we briefly describe the region-based framework and the temporal linking algorithm that jointly provide the segmentation and motion parameters. In the second part, we present various applications of the proposed algorithm to temporal interpolative prediction. We examine one of these schemes in detail, including the special processing of occlusion areas. Results are illustrated by predicted images and using the MSE criterion we compare their quality with other schemes (block-matching, causal prédiction).

1 Introduction

En compression de séquences d'images, la prédiction par compensation de mouvement peut s'effectuer à partir d'une image précédente (P-*frames* de MPEG) ou à partir d'une image précédente et d'une image suivante (B-*frames* de MPEG). Dans le standard MPEG, le même vecteur déplacement est appliqué à tous les pixels d'un bloc 16×16. Ce découpage fixe ne s'adapte ni aux zones de mouvement complexe, ni aux contours d'occlusion et cause donc des effets de blocs visuellement gênants dans les images reconstruites à bas débit. De plus, le codage des vecteurs de mouvement est redondant puisque le mouvement d'une grande région pourrait être décrit avec seulement quelques paramètres de mouvement (par exemple dans le cas d'un zoom ou d'un panoramique). Pour atteindre de plus bas débits, la compensation de mouvement orientée régions a été explorée [4, 7]. Cependant, ces études utilisent seulement l'image précédente, donc la prédiction des zones découvertes est impossible. Notre interpolateur est capable de prédire ces zones découvertes grâce à l'utilisation d'une image suivante, tout en étant capable d'assurer une meilleure prédiction des autres zones.

2 Suivi temporel de segmentation

Dans cette étude, les images sont segmentées en régions homogènes au sens du mouvement. Celles-ci ont une forme polygonale quelconque, et constituent une partition de l'image. Le critère d'homogénéité repose sur un modèle de mouvement affine simplifié à 4 paramètres ou sur un modèle affine complet à 6 paramètres. On note $\Theta^{\pm}_{\mathcal{R},t_i \to t_j}$ un descripteur de mouvement de l'image I_{t_i} vers l'image I_{t_j} pour la région $\mathcal{R}$ avec un exposant $^+$ si $t_i < t_j$ (direction des t croissants) ou un exposant $^-$ si $t_j < t_i$ (direction des t décroissants). Le déplacement de chaque point $p \in \mathcal{R}_{t_i}$ de l'image I_{t_i} vers l'image I_{t_j} est noté $\vec{d}^{\pm}_{t_i \to t_j}(p)$ (voir la figure 1).

Dans le cas affine simplifié, $\Theta^{\pm}_{\mathcal{R},t_i \to t_j} = [t_x, t_y, k, \theta]$ et

$$\vec{d}^{\pm}_{t_i \to t_j}(p) = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{bmatrix} k & -\theta \\ \theta & k \end{bmatrix} \begin{pmatrix} x_p - x_r \\ y_p - y_r \end{pmatrix}$$

où r est le point de référence du mouvement. Ce point est choisi comme le centre de gravité de la région $\mathcal{R}_{t_i}$ tant qu'elle ne subit pas d'occlusion.

Dans le cas affine, $\Theta^{\pm}_{\mathcal{R},t_i \to t_j} = [t_x, t_y, a, b, c, d]$ et

$$\vec{d}^{\pm}_{t_i \to t_j}(p) = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{pmatrix} x_p - x_r \\ y_p - y_r \end{pmatrix}.$$

Quand un descripteur Θ^+ doit être calculé à partir d'un descripteur Θ^- (et réciproquement), on calcule la transformation réciproque en inversant la matrice 2×2.

Comme l'estimation des paramètres de mouvement nécessite une bonne initialisation (et peut donc être biaisée dans le cas contraire), un filtrage long-terme des paramètres de mouvement reposant sur un filtrage récursif de Kalman a été implémenté.

L'algorithme de suivi temporel se décompose en 3 parties ; pour l'image I_t :

Prédiction : les paramètres de mouvement $\Theta^+_{\mathcal{R},t-\delta t \to t}$ des régions sont prédits, indépendamment les uns des autres, grâce au filtre de Kalman selon un modèle à accélération constante [6] et sont utilisés pour prédire la segmentation.

Ajustement : la segmentation spatio-temporelle prédite est ajustée sur les contours spatiaux dans l'image I_t, grâce à un modèle de contours actifs (*snakes*) : l'énergie minimisée est

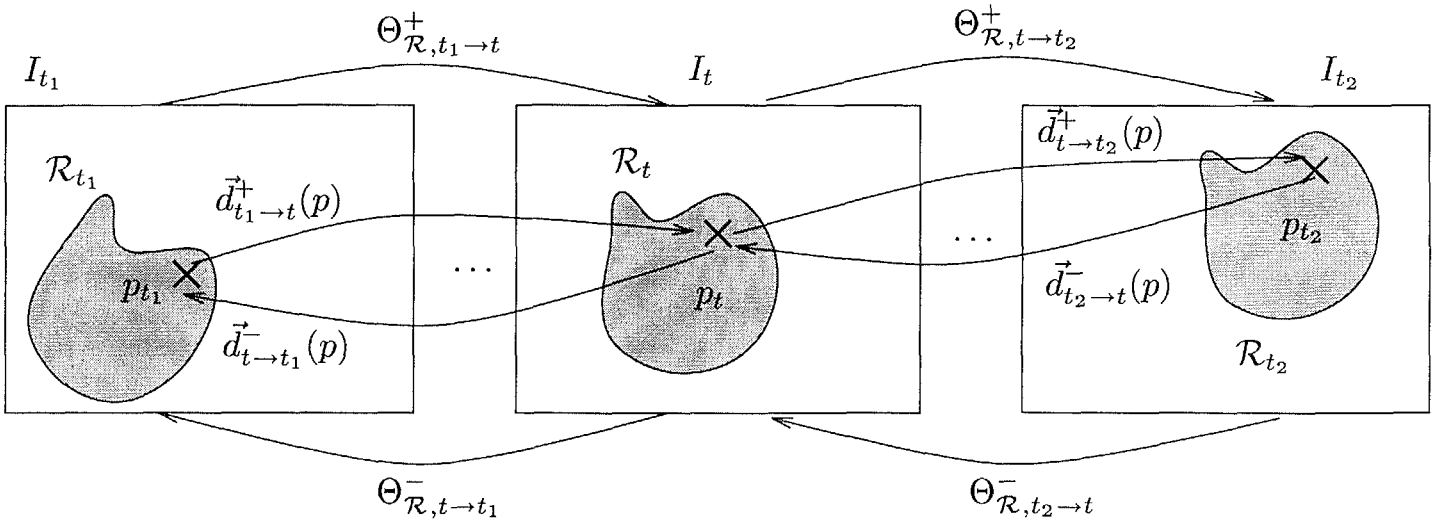

FIG. 1 - *Les deux sens de description du mouvement.*

la somme de $-\|\vec{\nabla} I_t\|$ le long de la frontière polygonale de la région. Le déplacement du snake est contraint à être affine et la minimisation faite par une descente de gradient sur les paramètres de mouvement [1].

Estimation : les paramètres de mouvement $\Theta^{-}_{\mathcal{R},t\to t-\delta t}$ des régions sont estimés par mise en correspondance : l'erreur quadratique moyenne (EQM) de prédiction est minimisée par une descente de gradient. Comme les vecteurs de mouvement ne sont pas à coordonnées entières, l'EQM est calculée grâce à une interpolation bicubique [5]. La minimisation par descente de gradient nécessite les dérivées spatiales de l'image ; par cohérence, celles-ci sont calculées avec cette même interpolation bicubique [5].

La figure 3 montre un exemple de segmentation obtenue. Pour plus de détails, on pourra se reporter à [3].

3 Compensation de mouvement bidirectionnelle basée régions

L'algorithme de segmentation traite toutes les images entre deux I- ou P-*frames* appelées images de référence (I_{t_1} et I_{t_2}). Il fournit aussi des descripteurs de mouvement de la forme $\Theta^{-}_{\mathcal{R},t+\delta t\to t}$ entre 2 images successives. Ensuite, la segmentation et les paramètres de mouvement sont utilisés pour interpoler les B-*frames* intermédiaires. Afin de prédire les B-*frames* $(I_t)_{t_1<t<t_2}$, le décodeur a besoin de la segmentation des 3 images et pour chaque région $\mathcal{R}$ des 2 descripteurs de mouvement de la B-*frames* vers les 2 images de référence, soit $\Theta^{-}_{\mathcal{R},t\to t_1}$ et $\Theta^{+}_{\mathcal{R},t\to t_2}$. Le codeur transmet la segmentation de I_{t_1} et de I_{t_2} au décodeur, ainsi que $\Theta^{-}_{\mathcal{R},t_2\to t_1}$ (utilisé si I_{t_2} est une P-*frame*). Ce descripteur est calculé grâce à l'estimateur de mouvement, initialisé avec la transformation affine qui est la composée des transformations affines $(\Theta^{-}_{\mathcal{R},t+\delta t\to t})_{t_1\le t<t_2}$.

Il y a un compromis entre la quantité d'information transmise au décodeur pour les mouvements et la segmentation et la qualité de la B-*frame* prédite. Voici plusieurs possibilités :

Compensation de mouvement bidirectionnelle : Le codeur transmet la segmentation de I_t, ainsi que les mouvements $\Theta^{-}_{\mathcal{R},t\to t_1}$ et $\Theta^{+}_{\mathcal{R},t\to t_2}$.

Prédiction bidirectionnelle de segmentation : Le codeur transmet seulement $\Theta^{-}_{\mathcal{R},t\to t_1}$ et $\Theta^{+}_{\mathcal{R},t\to t_2}$. Le décodeur reconstruit la segmentation de I_t en appliquant $\Theta^{+}_{\mathcal{R},t_1\to t}$ aux frontières de $\mathcal{R}_{t_1}$ et $\Theta^{-}_{\mathcal{R},t_2\to t}$ aux frontières de $\mathcal{R}_{t_2}$.

Interpolation pure : Le codeur ne transmet ni segmentation ni mouvements. Le décodeur interpole $\Theta^{-}_{\mathcal{R},t\to t_1}$ et $\Theta^{+}_{\mathcal{R},t\to t_2}$ à partir de $\Theta^{-}_{\mathcal{R},t_2\to t_1}$ et des descripteurs de mouvement préalablement transmis, selon un modèle de mouvement adéquat (vitesse ou accélération constante par exemple). Il reconstruit la segmentation de I_t comme dans le cas précédent.

Seul le premier schéma a été implémenté et testé, en supposant une transmission sans pertes de la segmentation et des descripteurs de mouvement.

Quant à la prédiction par interpolation de la B-*frame* $\hat{I}_t$, elle s'effectue ainsi : pour chaque point $p_t \in \mathcal{R}_t$, on a

$$p_{t_1} = p_t + \vec{d}^{-}_{t\to t_1}(p_t), p_{t_2} = p_t + \vec{d}^{+}_{t\to t_2}(p_t)$$
$$\hat{I}_t(p_t) = \alpha_p I_{t_1}(p_{t_1}) + \beta_p I_{t_2}(p_{t_2})$$

Les vecteurs de mouvement étant non-entiers, il est nécessaire d'interpoler spatialement dans I_{t_1} et I_{t_2}. Pour cela, nous utilisons l'interpolation bicubique [5].

Afin de prendre en compte les occlusions, nous distinguons les zones "normales" des zones recouvertes ou découvertes par le mouvement d'une autre région. Cette distinction est faite pixel par pixel selon les règles du tableau suivant.

type de zone and (α_p, β_p)	$p_{t_2} \in \mathcal{R}_{t_2}$	$p_{t_2} \notin \mathcal{R}_{t_2}$
$p_{t_1} \in \mathcal{R}_{t_1}$	zone "normale" (α, β)	zone recouverte $(1, 0)$
$p_{t_1} \notin \mathcal{R}_{t_1}$	zone découverte $(0, 1)$	prédiction spatiale $(0, 0)$

Dans les zones normales, l'interpolation temporelle est une combinaison linéaire des points correspondants dans les images de référence avec des coefficients fixes (α, β) :

- si les 2 mouvements sont de qualité équivalente et s'il

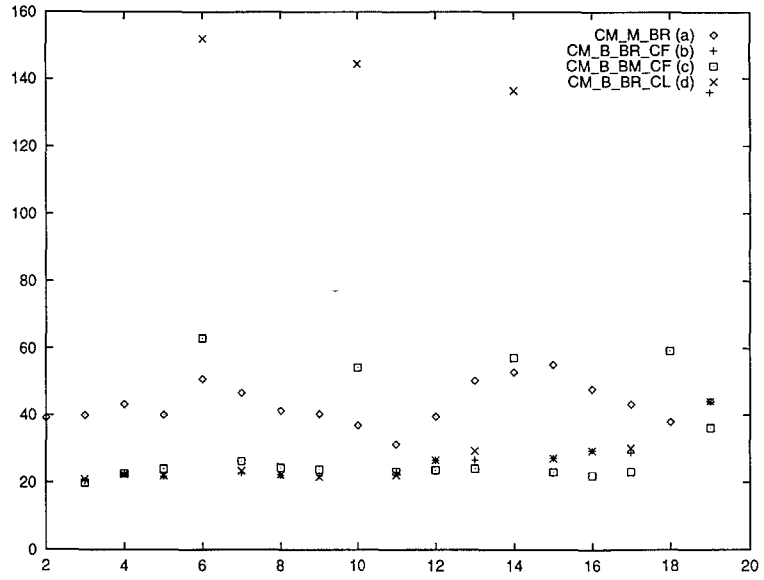

FIG. 2 - *Les images numéro 2, 6, 10, 14, 18 et 19 sont des P-frames ; les autres sont des B-frames. EQM de prédiction obtenues par plusieurs algorithmes : CM=Compensation de Mouvement ; M=Monodirectionnelle ; B=Bidirectionnelle ; BM=Block-Matching ; CF=Coefficients Fixes ; CL=Coefficients Linéaires.*

n'y a pas de variations d'illumination, on peut choisir les coefficients les plus simples, soit $(0.5, 0.5)$.

- s'il y a des variations d'illumination, on peut choisir un modèle d'éclairement linéaire en temps : $(\frac{t_2-t}{t_2-t_1}, \frac{t-t_1}{t_2-t_1})$.
- Les coefficients (α, β) peuvent aussi être estimés en même temps que les descripteurs de mouvement [8]. Dans ce cas, il faut aussi les transmettre.

Les pixels recouverts ou découverts sont prédits avec la même intensité que dans l'image de référence où ils ont leur correspondant.

Le cas singulier $(\alpha_p, \beta_p) = (0, 0)$ est très rare et se produit lorsque p n'a pas de correspondant dans les 2 images de référence. C'est par exemple le cas au bord des régions quand les frontières sont mal ajustées, ou si les images de référence sont trop éloignées. Une prédiction temporelle est donc impossible. Elle est remplacée par une prédiction spatiale utilisant un filtre médian avec fenêtre croissante. Pour assurer une certaine cohérence temporelle entre des interpolations spatiales successives, seuls des pixels appartenant à la même région $\mathcal{R}_t$ que p sont pris en compte dans le filtre médian.

4 Résultats expérimentaux et conclusions

Nous avons testé notre algorithme de prédiction bidirectionnelle sur une séquence d'images synthétiques, avec 3 images interpolées entre 2 I- ou P-*frames*. Nous avons essayé les coefficients fixes $(\alpha_p, \beta_p) = (0.5, 0.5)$ et les coefficients linéaires en temps, et nous avons comparé avec une prédiction basée régions monodirectionnelle et une prédiction bidirectionnelle par *block-matching*.

La figure 2 montre une comparaison fondée sur l'EQM définie par $\text{EQM}(\mathcal{R}) = \frac{1}{\#\mathcal{R}} \sum_{p \in \mathcal{R}} [I_t(p) - \hat{I}_t(p)]^2$. Les images interpolées (CM-B-BR-CF) ont une EQM nettement plus faible que les images prédites en utilisant uniquement l'image précédente (CM-M-BR) : 20–30 au lieu de 40–50. L'énergie (l'innovation apparaissant dans les zones découvertes) est concentrée dans la P-*frame* suivante dont l'EQM est par conséquent supérieure. Avec une EQM si basse (et surtout une bonne qualité visuelle, voir paragraphe suivant), un codeur hybride pourrait envisager de ne transmettre les erreurs de prédiction que pour les P-*frames*, réalisant ainsi un gain de codage. Pour les B-*frames*, notre prédiction bidirectionnelle est aussi bonne que celle par *block-matching*, alors que pour les P-*frames*, notre prédiction monodirectionnelle est moins bonne. Cela s'explique par le fait que notre algorithme détecte les zones découvertes et n'essaye pas de les prédire (elles restent noires). Le même type de prédiction spatiale que précédemment pourrait être utilisé, mais le problème est plus difficile car les zones découvertes sont plus grandes (une distance de $4\delta t$ sépare une P-*frame* de l'image à partir de laquelle elle est prédite). Dans notre séquence où les mouvements sont correctement estimés et où il n'y a pas de variations d'illumination, les coefficients fixes donnent des résultats similaires aux coefficients linéaires.

Une comparaison visuelle peut être faite dans la figure 3 qui montre les images prédites par divers algorithmes. Notre algorithme a un défaut dans la façon dont il traite les contours. Les images ont souvent des contours flous à cause du préfiltrage qui est fait avant l'échantillonnage dans les images réelles et à cause de l'interpolation dans une séquence d'images synthétiques. Notre segmentation ne prend pas en compte ce fait et les frontières des régions coupent l'image abruptement. Cela explique la présence de contours parasites

dans l'image interpolée (par exemple le fond a un contour qui reste d'une position antérieure du grand rectangle sombre). Pour corriger ceci, nous érodons le masque de segmentation de chaque région dans les images de référence avant de tester si p_{t_1} ou p_{t_2} lui appartient. Ainsi, la zone de transition entre les niveaux de gris des régions de part et d'autre d'une frontière n'est pas utilisée dans l'interpolation. Cela cause plus de cas singuliers où $(\alpha_p, \beta_p) = (0, 0)$, mais tant que l'érosion n'est pas trop importante, la prédiction spatiale peut reconstituer ces pixels. Le résultat ne présente pas les mêmes effets de bloc que le *block-matching*.

Nous avons donc décrit un nouvel algorithme d'interpolation capable de prédire les zones découvertes. De plus, les images prédites ont une EQM nettement plus faible qu'avec une prédiction monodirectionnelle et sont de meilleure qualité visuelle qu'avec le *block-matching*. Des travaux sont actuellement en cours pour tester cette interpolation sur des séquences d'images réelles.

Références

[1] Bascle (B.) et Deriche (R.). – Features extraction using parametric snakes. *In : Proceedings of 11th IAPR Int. Conf. on Pattern Recognition (ICPR'92)*, pp. 659–662. – The Hague, The Netherlands, août 1992.

[2] Bergeron (C.) et Dubois (E.). – Parametric block estimation of motion and application to temporal interpolation of video sequences. *In : Proc. IEEE Int. Conf. Pattern Recognition*, pp. 140–146.

[3] Bonnaud (L.). – *Étude d'algorithmes de suivi temporel de segmentation basée mouvement pour la compression de séquences d'images.* – Rapport technique n2253, France, INRIA, janvier 1994. `ftp.inria.fr: INRIA/ tech-reports/ RR/ RR-2253.ps.gz`.

[4] Garcia-Garduño (V.), Labit (C.) et Bonnaud (L.). – Temporal linking of motion-based segmentation for object-oriented image sequence coding. *In : Proceedings of EUSIPCO 94.* – University of Edinburgh, Scotland, UK, septembre 1994.

[5] Keys (R.G.). – Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, n6, décembre 1981, pp. 1153–1160.

[6] Meyer (F.). – *Suivi de régions et analyse des trajectoires dans une séquence d'images.* – Thèse de PhD, IRISA–Université de Rennes 1, juin 1993.

[7] Musmann (H.), Hötter (M.) et Ostermann (J.). – Object-oriented analysis-synthesis coding of moving images. *Signal Processing, Image Com.*, vol. 1, n2, 1989, pp. 117–138.

[8] Nicolas (H.), Konrad (J.) et Labit (C.). – Joint estimation of motion and illumination variations for coding of image sequences. *In : Proc. Scandinavian Conf. Image Analysis.*

[9] Tziritas (G.) et Labit (C.). – *Motion analysis for image sequence coding — Motion-compensated image interpolation*, chap. 7, pp. 269–285. – Elsevier, 1994.

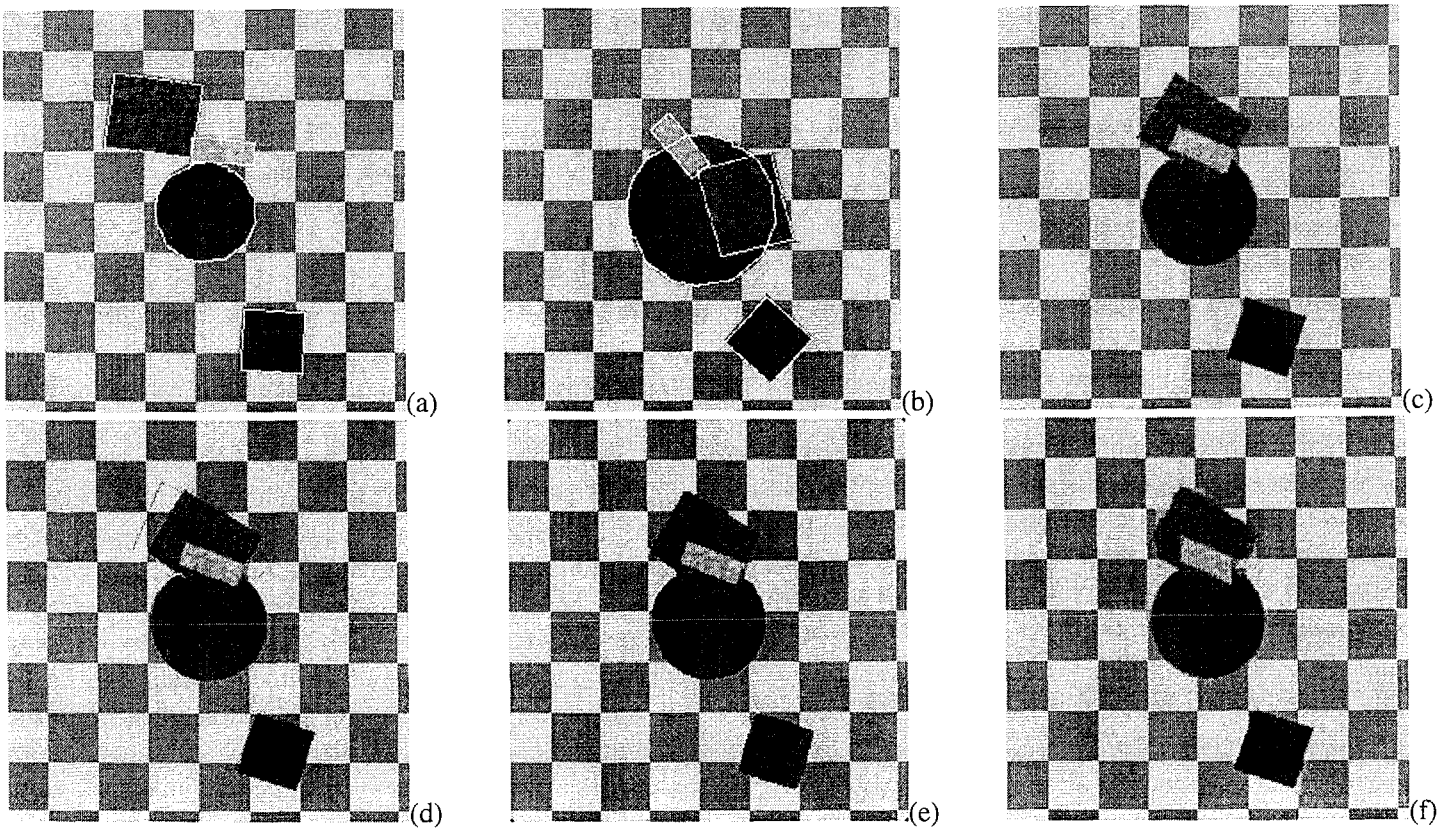

FIG. 3 - *(a) Initialisation de la segmentation dans la 1ère image de la séquence. (b) Segmentation de la dernière image après traitement de toute la séquence (19 images). (c) Image numéro 8. (d) Interpolation basée régions. (e) Image interpolée avec correction des contours. (f) Image interpolée par block-matching.*