

Fast method of segmentation and indexing MPEG1-2 flow

Lionel Brunel, Pierre Mathieu

Laboratoire I3S UMR 6070 CNRS Universit de Nice - Sophia Antipolis
2000, route des lucioles BP 121, 06903 Sophia-Antipolis Cedex FRANCE

ABSTRACT

Multimedia data accessibility depends on a precise indexing, involving a computational cost. This paper proposes a new fast method of segmentation and indexing in order to fill out in an automatic way several MPEG7 fields (e.g. camera and objects movement). In order to accelerate segmentation process, we exploit most of the information contained in MPEG1-2 flow; the decompression is restricted to entropic decoding and inverse quantization, the estimation of the camera movement is obtained from MPEG1-2 motion prediction. Segmentation in homogeneous color zones is obtained by a “split and merge” algorithm improved by a B-Splines active contour segmentation regularization.

Keywords: MPEG1-2 flow indexation, camera movement approximation, objects in movement segmentation, MPEG7

1. INTRODUCTION

With the increase of the quantity of multimedia data, MPEG7¹ gives a standardized and efficient form of indexing. Our purpose is to present an automatic algorithm for indexing with a high time constraint, meanwhile, we lose in precision. We use MPEG1-2² flow (e.g. format of TV satellites, DVD) and exploit most of the analysis carried out during compression. Motion prediction enables to calculate the camera movement; DCT coefficients of error images bring information on the prediction accuracy. DCT coefficients are given by entropic decoding and inverse quantization. We calculate neither the inverse DCT nor the image reconstruction at the pixel level.

The outline of this paper is as follows: Sect. 2 introduces what we use thereafter: firstly, a summary of the MPEG1-2 standard; secondly, the principle of spatiotemporal segmentation method developed in Ref. 3. Section 3 presents methods implemented for the camera movement estimation and the objects segmentation. Finally, Sect. 4 shows experimental results.

2. NECESSARY KNOWLEDGE

2.1. MPEG1-2 Standard (Video Part)

MPEG1-2 exploits the strong temporal correlation between successive images in a film. It cuts the film into Groups Of Pictures (*GOP*) beginning with an *Intra* (*I*) followed by *Predicted* (*P*) and *Bidirectional* predicted (*B*) (Fig. 1).

* *I* type images: JPEG like coding which uses DCT, quantization and entropic coding.

* *P* type images: for each macroblock, the coder searches for corresponding zone in the previous image (*I* or *P*). This leads to field of displacement vector (\vec{F}) which can be assimilated to a motion estimator. If the prediction is precise, the error macroblock has a low average and standard deviation.

* *B* type images: two predictions are given, one forward (\vec{F}), from previous *P* or *I* (like for *P* type macroblocks) and the other, backward (\vec{B}), from following *P* or *I*.

Further author information:

L.B.: E-mail: lbrunel@i3s.unice.fr, Telephone: +33 (0)4.92.94.27.87

P.M.: E-mail : mathieu@i3s.unice.fr

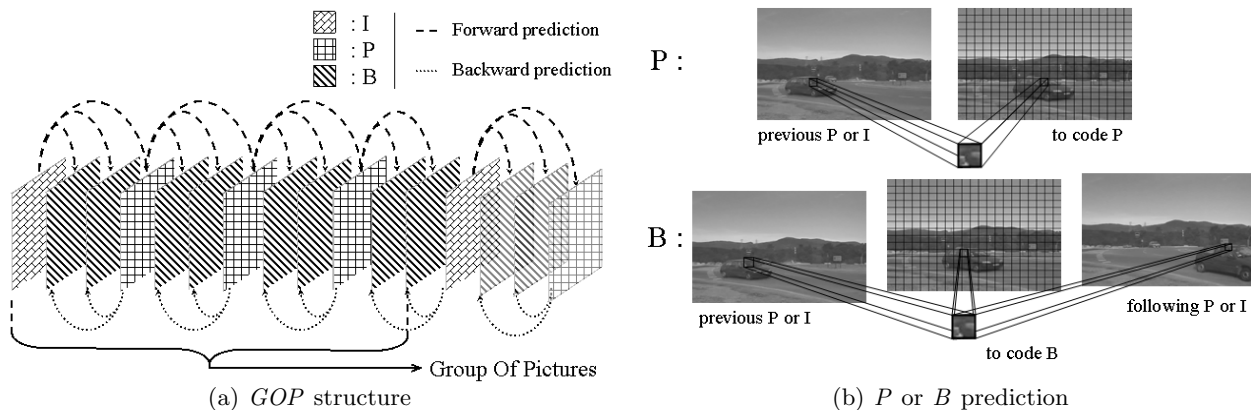


Figure 1. MPEG1-2 standard

2.2. Active Contour Segmentation⁴

The purpose of the segmentation is to cut an image into pixels areas of similar characteristics. The active contour segmentation is obtained by a criterion minimization of three terms: two terms of zone characterization (interior and exterior) and a regularization constraint on the border. The areas based terms are defined by a field integral:

$$J_{\tau}(\Omega) = \int_{\Omega} k(x, \Omega) dx, \quad (1)$$

with $k(x, \Omega)$ the Ω region descriptor. For the contour based terms, the contour integrals is used:

$$J_{\Gamma}(\Gamma) = \int_{\Gamma} k(x) ds, \quad (2)$$

with $k(x)$ being the boundary descriptor Γ of the Ω area and ds an element of surface which is related to x .

We can define an image as the meeting of three parts which are all distinct from each other: $\Omega_{in} \cap \Omega_{out} = \emptyset$ and $\Omega_{in} \cup \Omega_{out} \cup \Gamma = \Omega$ with Ω_{in} the interior of the segmented zone, Ω_{out} the outside, and finally Γ the boundary.

Therefore, the area partition is obtained by the criterion minimization:

$$J(\Omega_{in}, \Omega_{out}, \Gamma) = \int_{\Omega_{in}} k_{in}(x, \Omega_{in}) dx + \int_{\Omega_{out}} k_{out}(x, \Omega_{out}) dx + \int_{\Gamma} k(x) ds. \quad (3)$$

In order to conclude this minimization, with use of the Green-Riemann theorem, it is possible to transform the fields integrals into contour integrals. To solve the partial differential equations (PDE), Euler-Lagrange equations are applied to those contours.

As shown in Ref. 5, after calculation, in the case of two areas (interior and exterior), the equation of following evolution (force in the normal direction \mathbf{N}) is:

$$\frac{\partial \Gamma^{(i,j)}}{\partial \tau} = \left(k_{in}^{(i,j)} + k_{out}^{(i,j)} + \lambda \kappa^{(i,j)} \right) \mathbf{N}, \quad (4)$$

with κ the contour curvature and $\frac{\partial \Gamma^{(i,j)}}{\partial \tau}$ the force $F^{(i,j)}$.

According to Epstein and Cage, if we are only interested in the geometry of the deformation, the tangential part of the contour equation is useful (statement seen in Ref. 6).

Curvature calculation at a M point is long (where $M(l) = (x(l), y(l))^t$ the point followed over the contour):

$$\kappa(l) = \frac{\ddot{x}(l)\dot{y}(l) - \dot{x}(l)\ddot{y}(l)}{\left[\dot{x}(l)^2 + \dot{y}(l)^2\right]^{3/2}}. \quad (5)$$

Now, the boundary modeled by cubic B-Spline curve,³ on which the force $F^{(i,j)}$ is applied, can be modified. Ref. 3 use:

$$k_{out}^{(i,j)} = \left(S_n^{(i,j)} - S_{n-1}^{(i,j)}\right)^2, \quad k_{in} = 0.037, \quad \lambda = 0.005, \quad (6)$$

with $S_n^{(i,j)}$ the value of pixel (i, j) in image n . In addition, B-Spline method allows to obtain directly the contour curvature avoiding an heavy computational load.

3. IMPLEMENTED ALGORITHM

The broad outline of the implemented algorithm is firstly the apparent camera movement estimation; secondly, images are split into homogeneous luminance and chrominances zones. Results are refined by an active contour segmentation. The next step is merging: the zone movement is then used.

After objects extraction and monitoring, the objects movements are estimated. With all these data, some MPEG7 fields are filled out.

3.1. Standardized Forward Vector (*SFV*) Estimation

In order to estimate movement, it is necessary to obtain the apparent movement between two given images, it is the *SFV* (the movement between the macroblock in the image and the previous one⁷). Unfortunately, *Intra* images do not contain motion vectors.

Considering a time continuous movement, we interpolate the movement vector given in flow in order to obtain the movement with the previous image: \vec{N}_l (the *SFV* for the image l , attached with a block (i, j)).

* Intra Case (only for *Intra* images): $\vec{N}_l = \frac{\vec{F}_{l+1} - \vec{B}_{l-1}}{2}$;

* Predict Case (for all Predict macroblock): $\vec{N}_l = \frac{\vec{F}_l}{nb_F + 1}$;

* Bidirectional Case: $\vec{N}_l = \frac{\vec{F}_l}{nb_F + 1} - \frac{\vec{B}_l}{nb_B + 1}$;

nb_F (respectively nb_B): number of images between this image and the previous one (resp. next one) used for prediction (*P* or *I*).

3.2. Apparent Camera Movement (*ACM*) Determination

Possible camera movements are:

- Fig. 2(a): T_X (resp. T_Y or T_Z) represents the translation following the X -axis (resp. Y or Z);
- Fig. 2(b): R_X (resp. R_Y or R_Z) represents the rotation following the X -axis (resp. Y or Z);
- R_{zoom} the *Zoom*.

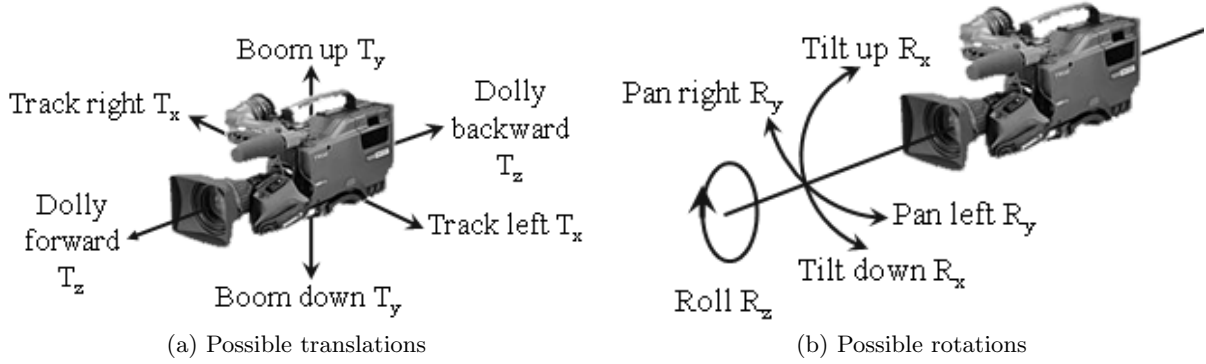


Figure 2. Possible camera movements

For the calculation of the *ACM* defined by the movement vector \vec{N}_i , we need some assumptions. First of all, we assume that surface background is larger than the surface of the objects in movement; otherwise, we estimate the movement of the greatest object and no more the camera movement (it would be possible to solve this problem by using a history of several seconds).

Then, we do not use the complete *ACM* system (nine unknowns) given in Ref. 1:

$$\left\{ \begin{array}{l} U_x(x, y) = -\frac{f}{Z} \cdot (T_x - x \cdot T_z) + \frac{x \cdot y}{f} \cdot R_x - f \cdot \left(1 + \frac{x^2}{f^2}\right) \cdot R_y + y \cdot R_z \\ \quad + f \cdot \tan^{-1}\left(\frac{x}{f}\right) \cdot \left(1 + \frac{x^2}{f^2}\right) \cdot R_{zoom} \\ U_y(x, y) = -\frac{f}{Z} \cdot (T_y - y \cdot T_z) - \frac{x \cdot y}{f} \cdot R_y + f \cdot \left(1 + \frac{y^2}{f^2}\right) \cdot R_x - x \cdot R_z \\ \quad + f \cdot \tan^{-1}\left(\frac{y}{f}\right) \cdot \left(1 + \frac{y^2}{f^2}\right) \cdot R_{zoom} \end{array} \right. \quad (7)$$

Indeed, between two images, it is not possible to differentiate T_x and R_y , T_y and R_x , T_z and R_{zoom} , so the simplified system is obtain with only six unknowns:

$$\left\{ \begin{array}{l} u_x = -\frac{f}{Z}(T_x - xT_z) + y \cdot R_z \\ u_y = -\frac{f}{Z}(T_y - yT_z) - x \cdot R_z \end{array} \right. , \quad (8)$$

with (x, y) , position on the retina; (u_x, u_y) coordinates of \vec{N}_i ; f , focal distance; Z , distance from the object to the retina.

Assuming a case of a pinhole camera in an orthographic model, we approximate $\frac{f}{Z}$ by a constant, and obtain a system with four unknowns (T_x , T_y , T_z , R_z). With abuse of notation, we will note in the continuation T_x by *Pan*, T_y by *Tilt*, T_z by *Zoom* and R_z by *Rotate*.

The estimation of considered movement $(\widehat{u}_x, \widehat{u}_y)$ is given by minimizing the following criterion:

$$J = \sum_{i,j} \left\{ [u_x - \widehat{u}_x]_{(x_i, y_j)}^2 + [u_y - \widehat{u}_y]_{(x_i, y_j)}^2 \right\} . \quad (9)$$

By solving the system:

$$\frac{\delta J}{\delta L} = 0 \quad L \in \{T_x, T_y, T_z, R_z\},$$

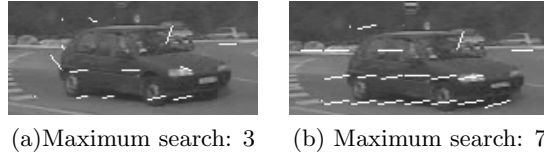


Figure 3. Coded with MPEG Software Simulation Group 1994 coder

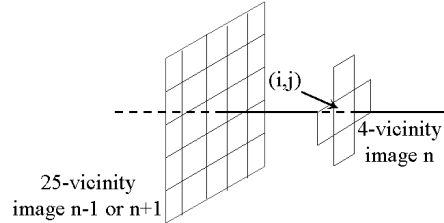


Figure 4. 54-connectivity

we obtain an estimation of the four unknowns. This approach gives a quicker result compared to Ref. 7 and Ref. 8.

To improve the camera movement precision, incoherent vectors are rejected, which leads to keep the background vectors. By assuming that the motion error on the background follows a centered normal distribution, and by eliminating the vectors outside the confidence interval fixed at 90%, we iterate estimation.

The choice of the maximum amplitude search of the similar block in the previous (or following) image P or I is a fixed parameter of the coder and the motion vector is estimated only in order to maximize the similarity and not a real movement calculation.⁹ Therefore, this maximum value is not sent in flow; it can not exceed 7 pixels between two consecutive images in *IBBP* case (Fig. 1(a)). The quality of motion estimation depends on the arbitrary choice made by the coder. The larger this value is, the longer the search becomes. Fortunately, such an important movement of the main object is not common.

In Fig. 3, with a fixed camera sequence, the car executes a translation of six pixels between two images. In 3(b), motion vectors correspond nearly to the real movement of the object, which can not be obtained in 3(a). To solve this problem, error image weight and Intra-macroblocks number in a not *Intra*-image allow prediction relevance check.

3.3. By Colors Interest Zones Detection

Our aim is to segment each image in zones of uniform color (luminance and chrominances). We have the average values by block for luminance and macroblock for the chrominances, and we wish to obtain major zones. It not necessary during decompression process to determinate the inverse DCT transform. For an intra I , these values are obtained, from a multiplicative constant, by coefficients DC of DCT blocks. The motion vectors, the averages of the previous P or I blocks (resp. previous and following) and average values of error image blocks, allow the reconstruction of the average values of a P block (resp. a B block). The same method is applied on chrominances macroblocks.¹⁰

3.3.1. 3D distance criterion

We introduce a color distance value between two blocks (in the same image or in two successive images) involving the knowledge of their belonging to the same object. The standardization of each component by image mean, allows the estimation robustness. In 3D, a bloc (i, j) and an other block are close if they are in a 4-connectivity in the current image but also if they are in a 25-connectivity (centered in (i, j)) in the previous or following image. It is thus necessary for each block to look at 54 neighbors (Fig. 4).

For a block (i, j) , in image n , its distance with a close block is defined by:

$$dist_{i,j}^n(f) = \frac{\sum_{l \in L} \left[\lambda_l \cdot \left| \frac{f_n^l}{mean_n^l} - \frac{f_m^l}{mean_m^l} \right| \right]}{\sum_{l \in L} \lambda_l}, \quad (10)$$

$L = \{lum, Cr, Cb\}$
 $m \in \{(n-1), n, (n+1)\}$

f_n^l and f_m^l , $mean_n^l$ and $mean_m^l$, stand respectively for the average value of two neighbor blocks in image n and m , the average value in image p and a weight value. Note: for the separation of two blocks in the same macroblock of an image, λ_{Cr} and λ_{Cb} are pointless.

For each block (i, j) of an image n , we determinate the value of $dist_{i,j}^n(f)$. The belonging to a same object of two blocks is evaluated according a variable threshold. The determination of the threshold is explained in the next paragraph.

3.3.2. Iterative merging

The thresholding technique is commonly used in video analysis to reduce the discretization effect and the noise. As shown in Ref. 11, the computation of the entropy power of the information source, as defined by Shannon, gives an adaptative thresholding. Assuming of an additive Gaussian noise, and an exponential distribution $dist_{i,j}^n(f)$, the optimal threshold is proportional to the standard deviation.

To make the method more robust, we take a rather small threshold, and we carry out several iterations by increasing the threshold at each turn of algorithm loop. The distance is calculated not between two colors of block but between the mean color of each zone. As the first iteration, each zone is reduced to one block.

To refine adaptive merging, the standard deviation of each zone is used¹² i.e., if the merging increases significantly the standard deviation, merging is canceled.

With the contour model seen in the Eq. (4) with variable defined in (6), boundaries are modified involving a fine tuning.

3.4. Merging 3D Color Zones Using Movement

To fill out MPEG7 moving objects fields, color zones with a coherent movement (Ref. 12 belonging to the same object) must be merged. The difference between the MPEG1-2 motion vector and the *ACM* vector provides results that can only be used in the background. According to Sect. 3.2, objects can move more than 7 pixels and the overlapping zones reduce dramatically the number of usable vectors. Each macroblock can be classified in two categories. In the first, are included macroblocks whose vectors belong to the confidence interval (high probability to be a part of the background). The second includes macroblocks whose vectors are rejected as well as the ones coded without motion prediction in a not *I* image.

For objects in movement, after iterative merging step, a same color zone can be followed during successive images. This apparent movement is extracted and zones having the same movement are joined together.

3.5. Objects Movement Estimation

Concerning objects, the number of MPEG vectors is too low and not often pertinent. One of the reasons is that the object often moves more than seven pixels, but also that the object contains flat tints. As MPEG1-2 calculates the Block-Matching only with an aim of reducing the quantity of transmitted data, it will thus give a motion vector which does not correspond to the real movement of the object. Another method, more robust, should be found. Therefore, the apparent object movement given by the translational object displacement is used between two images, then the object displacement T_X and T_Y are estimated. The T_Z estimation requires several images.

3.6. Indexing

We obtain, in real time, the apparent camera movement estimation, the extraction of moving objects in the sequence, the estimation of their movements in the scene (reduces, till now, to T_X and T_Y).

It is now possible to monitor an object from an image to the next one; to obtain the image numbers where it appears and disappears; to estimate the movement of its including rectangular minimum box and its percentage occupancy.

According to this information, some MPEG7 fields are filled out.

4. EXPERIMENTAL RESULTS

4.1. ACM Examples

We compare our method with an elaborated method seen in Ref. 13 with six parameters, block 5×5 , a precision prediction of a quarter of pixel (MPEG1-2 has a precision of half pixel) and a search at 15 pixels. On the truncated sequence Stefan Edberg (COST 211*) (images from 17 to 160), the cost of our method (ACM and segmentation) is **18 seconds** (on Intel PIII 500 Mhz), while the others takes **ten hours**. The two methods are not optimized and the Block-Matching takes a lot of time. With the same sequence coded by MPEG Software Simulation Group (1994) coder at two search sizes (3 and 7 pixels). Results are compared with the reference method.

On Fig. 5(a), with a search size of seven pixels, we see that our method gives good results (± 0.75 pixel in comparison with the other method), excepted between images 85 to 110. With the search size at three pixels, results are not pertinent in three spots (where the *Pan* is higher than three pixels). The use of the motion vectors of MPEG1-2 will never give a movement of *Pan* or *Tilt* higher than ± 7 or ± 3 respectively. We avoid taking into account the erroneously estimated vectors. The graph of the DC means of error macroblocks (Fig. 5(b)-bottom) shows a significant increase where the real movement is larger than the estimated movement and on these intervals, the PSNR decreases (Fig. 5(b)-up). We use the DC mean values criterion to validate the camera movement estimation or to detect when a more precise algorithm is needed.

Figure 6 is an other example, with, between each frame, a *Pan* near -1 pixel, *Tilt* near 1 pixel, *Rotation* near -0.01 radian and *Zoom* near -0.03 . The *Pan* and the *Tilt* are close to the result of reference (± 0.3 pixel). Research being made only with half pixel, the result is acceptable. For the *Zoom*, the result is close to ± 0.00037 ; for an 400×400 image, this brings a shift of ± 0.13 pixel on the edges of the image. In the same way, for *Rotation*, the result is close to ± 0.001167 , that is to say a maximum shift of ± 0.23 pixel on the edges.

4.2. Segmentation Examples

In images extracted from sequence Hall (COST 211): first image (Fig. 7(a)) represents the ruptures between blocks of uniform luminance and chrominances zones; second image (Fig. 7(b)) stands for the merging obtained by the motion vectors of each zone.

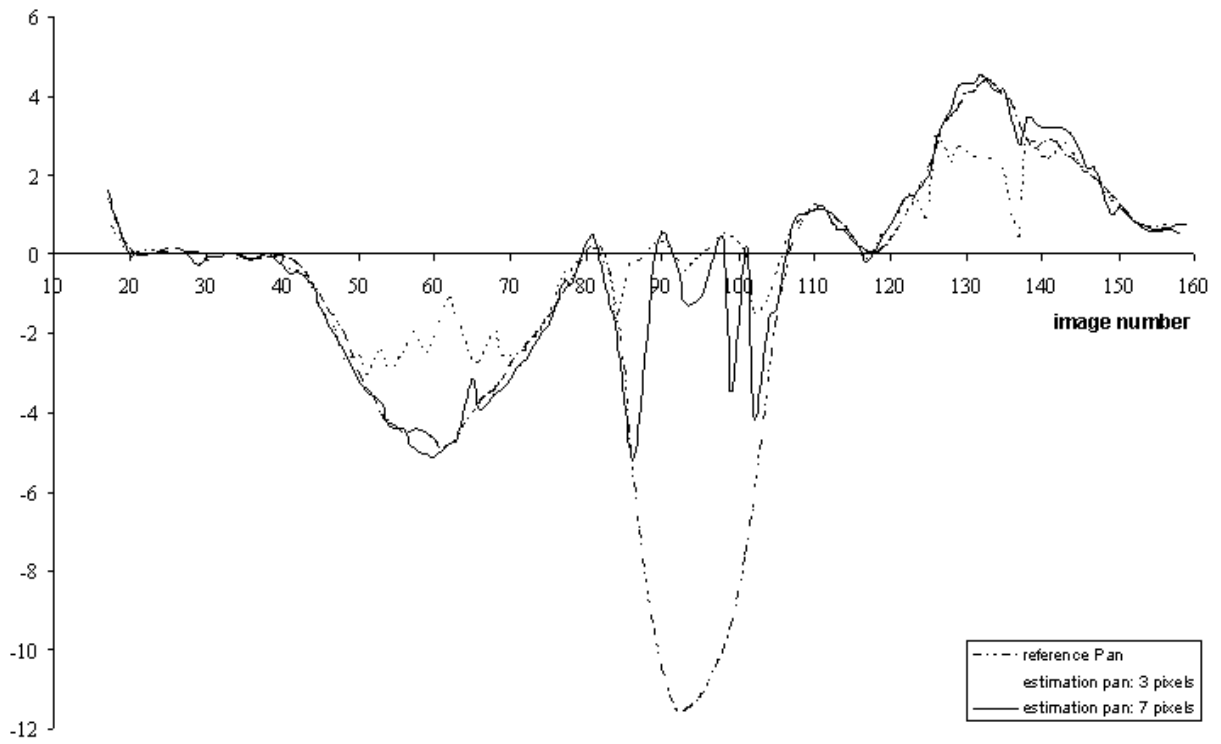
Other examples of segmentation (Fig. 8): the following images are extracted from sequences Stefan Edberg, Coast Guard and Hall (COST 211 sequences).

5. CONCLUSION

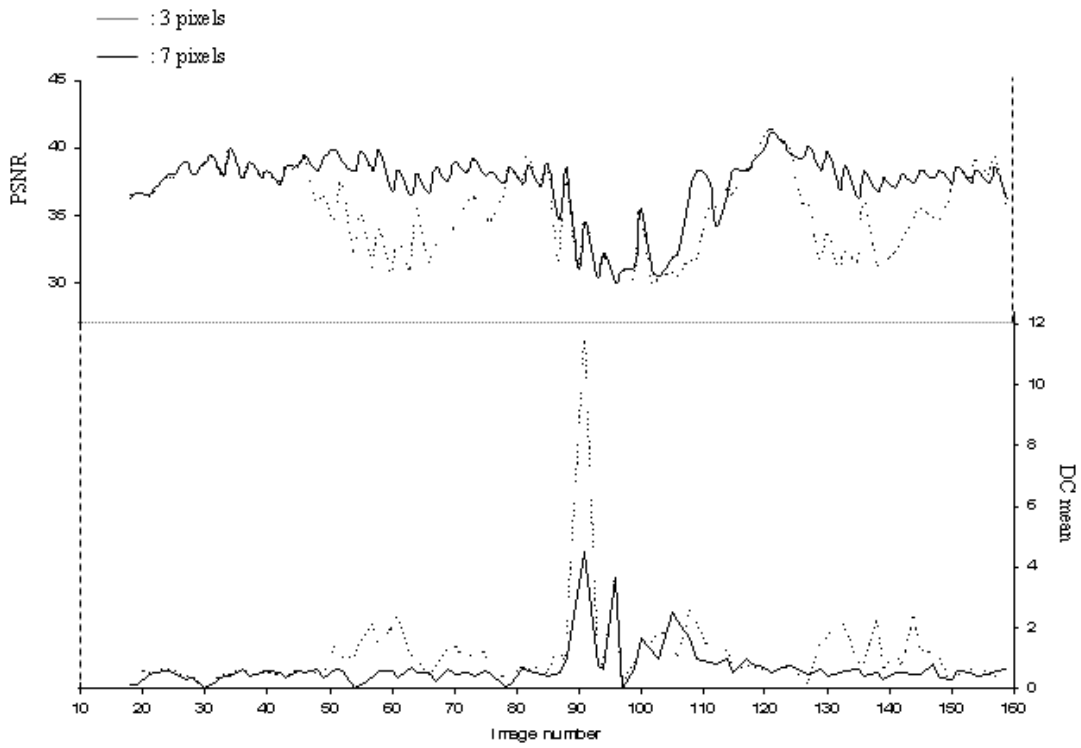
The proposed method avoids resorting to complete MPEG1-2 flow decompression. It allows to take the greatest advantage on the informations provided by the flow as motion vectors for images B and P , or DCT coefficients. Remaining on the concept of block and macroblock allows time cost decrease (near real time) with a lesser precision loss. The camera movement estimation is near to the correct movement with a precision of ± 0.75 pixel for translations (Fig. 5(a)).

In the future, in order to still refine the result, it will be possible to pass at the pixel level only in one small neighborhood around the object.

*The European COST 211 Group - Research on Redundancy Reduction Techniques and Content Analysis for Multimedia Services



(a) *Pan* comparison between the reference method and the proposed method with two search-sizes



(b) up: PSNR comparison with two search sizes and the original;
 bottom: error macroblocks DC mean values given in B images.

Figure 5. Comparison between the two methods of *ACM*

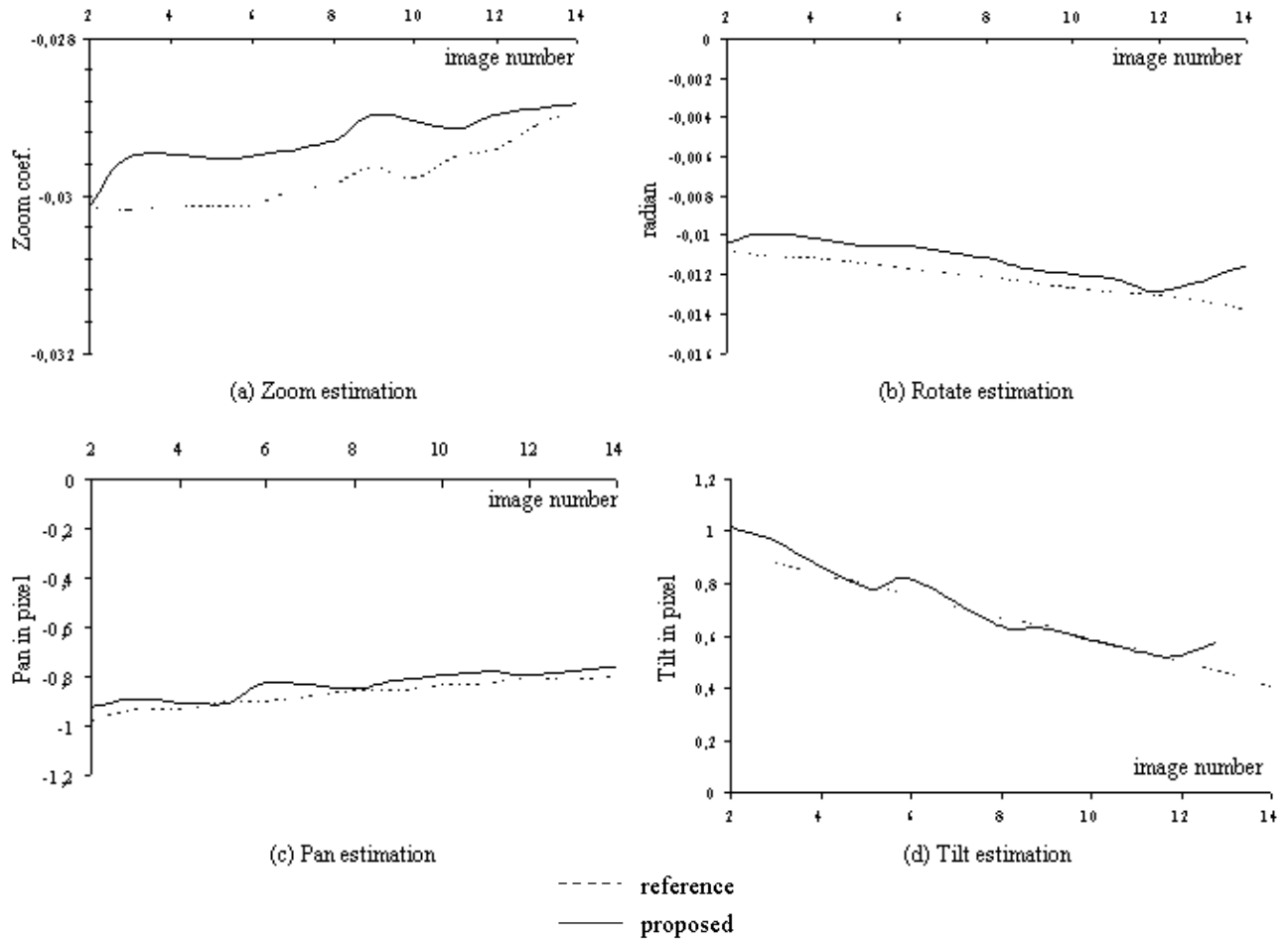


Figure 6. Comparison between the two methods of ACM

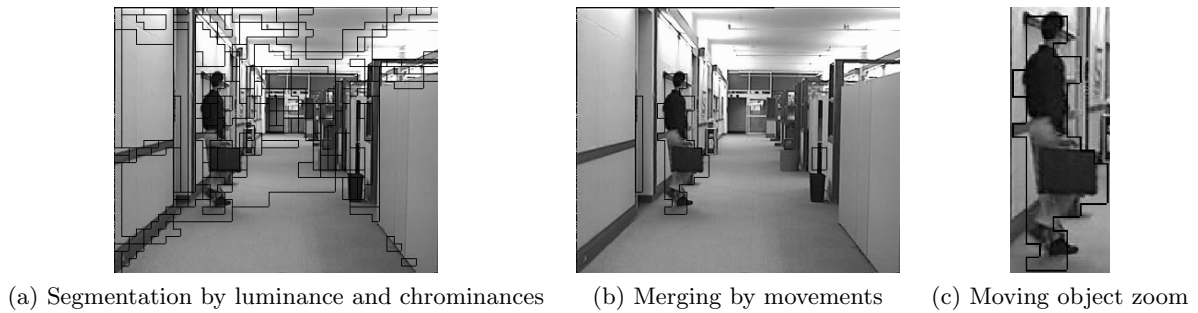


Figure 7. Segmentation of sequence Hall (COST 211)

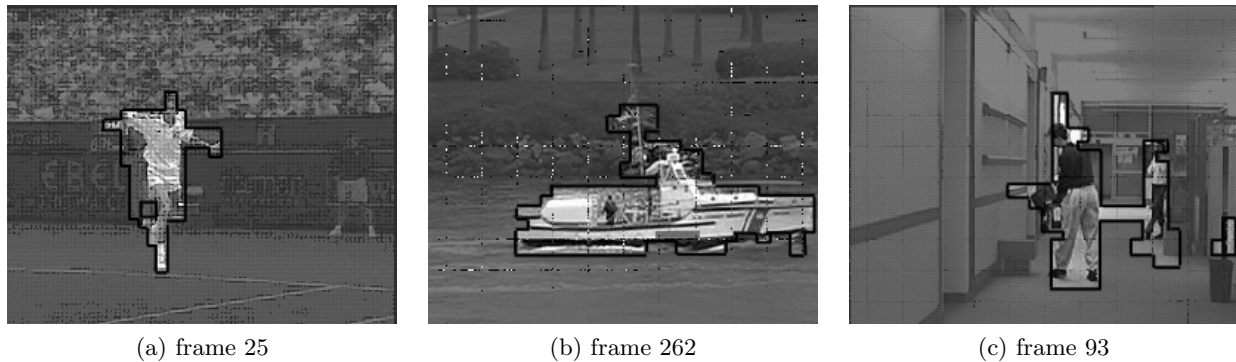


Figure 8. Segmentation examples

ACKNOWLEDGMENTS

Frdric Prcioso for his explication of B-Splines; Stphane Tramini and Karine Sanche for their help during the English article redactions.

REFERENCES

1. JTC1 / SC29 / WG11 / M5578, *MPEG-7, Visual Part of Experimentation Model Version 3.1*, ISO/IEC International Organisation for Standardisation, Maui, December 1999.
2. JTC1 / SC 29 / WG 11, *Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s - Part 2: Video*, ISO/IEC International Standard 11172-2, Geneva, Switzerland, 1993.
3. F. Precioso, and M. Barlaud, "Regular spatial B-Spline active contour for fast video segmentation", in *IEEE International Conference on Image Processing*, Rochester, USA, September 2002.
4. S. Jehan-Besson, M. Barlaud, and G. Aubert, "Video object segmentation using Eulerian region-based active contours", in *IEEE International Conference on Computer Vision*, Vancouver, Canada, July 2001.
5. S. Zhu, and A. Yuille, "Region Competition : Unifying Snakes, Region Growing, and Bayes/MDL for Multi band Image Segmentation", *IEEE Transaction on Pattern Analysis and Machine Intelligence* **Vol. 18 n9**, pp 884-900, September 1996.
6. G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*, Cambridge University Press, New York, USA, January 2001.
7. R. Wang, and T. Huang, "Fast camera motion analysis in MPEG domain", in *IEEE International Conference on Image Processing*, Cobe, Japan, October 1999.
8. M. V. Srivasan, S. Venkatesh, and R. Hosie, "Qualitative estimation of camera motion parameters from video sequences", *Pattern Recognition* **30**, pp 593-606, 1997.
9. K. Duc Vo, I. Nishihara, T. Yoshida, and Y. Sakai, "Precise estimation of motion vectors and its application to MPEG video retrieval", in *IEEE International Conference on Image Processing*, Cobe, Japan, October 1999.
10. B. Yeo, and B. Liu, "Rapid Scene Analysis on Compressed Video", *IEEE Transactions on Circuits and Systems for Video Technology* **5 (6)**, pp 533-544, 1995.
11. F. Luthon, and M. Livin, "Entropy power for thresholding technique in image processing", in *EUropean Signal Processing COnference*, Toulouse, France, September 2002.
12. S.-S. Wong, "Color segmentation and figure-ground segregation of natural images", in *IEEE International Conference on Image Processing*, Vancouver, Canada, October 2000.
13. S Fuh, and P. Maragos, "Affine models for image matching and motion detection", in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Canada, may 1991.