A Unified Approach to Shot Change Detection and Camera Motion Characterization

Patrick Bouthemy, Marc Gelgon, Fabrice Ganansia

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A unified approach to shot change detection and camera motion characterization

Patrick Bouthemy, Marc Gelgon and Fabrice Ganansia

N° 3304

THÈME 3

Rapport de recherche

ISSN 0249-6399

INRIA
RENNES

A unified approach to shot change detection and camera motion characterization

Patrick Bouthemy, Marc Gelgon and Fabrice Ganansia

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Temis

Rapport de recherche n 3304 — — 17 pages

Abstract: This paper describes an original approach which jointly addresses two fundamental issues of video partitioning which represent the early important stage of any content-based video indexing system. These two issues are the detection of shot changes, and the labeling of the shot configuration related to the camera movement in terms of static shot, panning, traveling, zooming,... They are both derived from the computation, at each time instant, of the dominant motion in the image represented by a 2D affine model, and from the variation of the size of its associated support. The successive steps of the method rely on statistical techniques ensuring robustness and efficiency. In particular, it can cope with scenes containing moving objects. Results on a real documentary video are reported and validate the proposed approach.

Key-words: Shot change detection, Camera motion analysis, Video indexing

(Résumé : tsvp)

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Une approche unifiée pour la détection de changement de plan et la caractérisation du mouvement de caméra

Résumé : Nous décrivons une approche traitant conjointement deux problèmes fondamentaux du partitionnement de vidéo en plan, la détection des changements de plan et l' étiquetage d'une configuration de plan relié à un mouvement de caméra en termes de plan fixe, traveling, zoom... Il s'agit d' une première étape importante pour tout système d'indexation de vidéo par le contenu. La méthode proposée exploite, pour les deux aspects du problème, l'estimation du mouvement dominant dans l'image, représenté par un modèle affine 2D, et de l' évolution temporelle du support associé. Les étapes successives de cette méthode reposent sur des techniques statistiques, qui en assurent la robustesse et l'efficacité.

Mots-clé : Découpage en plan, Mouvement de caméra, Indexation vidéo

1 Introduction and related work

Archiving image and video information represents an important, either established, more recent, or emerging task in several important application fields. Areas that will benefit from advances on this subject include audio-visual archives (news, films, documentaries, etc...), road traffic surveillance, remote sensing and meteorology (satellite images), or medical imaging (hospital medical records),... However, it remains hard to easily identify information pertinent with respect to a given query, or to efficiently browse large video files, making the exploitation of such databases highly cumbersome. The most commonly used approach consists in assigning key words to each stored video, and doing retrieval only on these words.

The need to index and retrieve image sequences by their content and not just by information external to them is becoming quite obvious. It is thus crucial to be able to define content-based indexing techniques. A survey on objectives of image indexing has been proposed in [9], reviewing the major cues for still image indexing, such as color, shape and spatial organization. Another review, more concerned with indexing of video, has been presented in [2]. Several research groups have investigated such issues for a few years, leading to the construction of prototypes accounting for first advances in that direction, [10, 19, 24]. Nevertheless, numerous problems are still open due to the fact that image interpretation and dynamic scene analysis are complex topics in computer vision, that the range of scene contents and of possible queries is vast, etc...

As far as video indexing and video editing are concerned, the primary requirements are the structuration of the video into elementary shots, and the recognition of typical forms of video shooting like static shot, traveling, zooming, panning, [8, 21, 22]. Then, further analysis of the video content relies on such a temporal partition, according to a hierarchical approach or not, [2, 16, 22, 23]. This video partitioning step enables to provide content-based, fast, adaptive and efficient browsing of a video. The use of key-frames to summarize the content of video shots and to facilitate access to the content is also of major interest. Tools for efficient video visualization are useful if, for instance, the user cannot express in a well-defined manner his query. As long as effective solutions to the complex problem of elaborated video content analysis (in terms of semantic content) are not available, visualization will remain one of the main means of accessing information. Nevertheless, work towards compact representation of shot content has been proposed in [3, 12, 14]. Indexing of events in a video, such as appearance, deposit or removal of an object has been explored in [7].

Substantial efforts have been devoted to the detection of cuts and of transitions related to special effects (fade-in, fade-out, dissolve,...) in order to achieve video partitioning. The later kind of changes is obviously more difficult to handle. A survey on this topic was proposed in [5]. Different solutions have been designed to detect cuts, based on, merely pixel- or block-based temporal image difference, variation of correlation measurements, or more efficiently, on difference of histograms, [1, 2, 16, 21, 25]. To detect transition effects, the use of a two-level thresholding technique applied to histogram differences has been proposed [25], or more elaborately, the modeling of the temporal intensity change law during the transition interval [1]. In most approaches, several tests must be operated to detect the various possible types of transitions, leading to sensible tuning of multiple and related parameters. In [25], these thresholds are determined from a preliminary pass through the video content itself. Since video databases are often available in compressed format, direct processing of the MPEG bit-stream is also a field of interest. In [18], the image intensity histograms are computed using the DC component of the DCT related to I-frames.

Some of these solutions deliver quite satisfactory results, but false alarms may still occur in case of important camera motion or in the presence of mobile objects leading to a undesirable over-segmentation of the video stream. It is often crucial that all shot changes are correctly detected and located. For instance, if the spatio-temporal content of the shot is to be analyzed, motion-based segmentation and tracking phases could severely be perturbed by missing or misplaced shot changes.

The recognition of parts of the video in which camera is static, or traveling, or panning, has been achieved using rather dedicated methods. Usually, they rely on the exploitation of motion vectors issued from block-matching techniques, or on the search for specific distributions of motion vectors or of a few global representative motion parameters, [25]. An original alternative constructs so-called "X-ray" images (related to the xt- or yt-plane in the image sequence after projection of the intensities along given lines), and looks for particular patterns in them using a Hough transform, [15, 21]. The MPEG-1 bit stream may also be directly exploited for camera motion characterization [18], using motion vectors related to P- and B-frames. One of the main shortcomings of these approaches is that they cannot cope with scenes including moving objects. Indeed, most methods are not resilient to the presence of mobile objects of significant size. In [20], this issue is overcome by computing so-called optical flow streams, built from the dominant optical flow over some extent in time. The algorithm depends however on many thresholds, and assumes a constant camera motion type during the extent in time over which optical flow streams are built.

We present in this paper an original approach for video partitioning and camera motion characterization. A preliminary version was presented in [6]. The principle of the proposed method is shown in Figure 1. It is based on the estimation of a 2D affine motion model between each pair of successive frames accounting for the global dominant image motion. Studying the temporal evolution of the associated estimation support enables the detection of both cuts and progressive transitions, with the same scheme and the same parameter values. Then, testing the significance of each of the components of the estimated global affine motion model provides a qualitative description of the dominant motion at each instant (assumed due to camera motion). The main features of the method are the following: 1) it addresses these two issues in an unified way; 2) it is able to handle scenes containing moving objects; 3) it only exploits 2D parametric motion models (affine models); 4) it is based on several statistical techniques ensuring robustness and efficiency. Section 2 outlines the motion estimation method based on a robust multiresolution scheme which allows us to compute the dominant motion between two successive images (step 1). In Section 3, we describe how we can determine the partitioning of the video into elementary shots (step 2) from the temporal variation of the size of the estimation support derived in step 1. Section 4 deals with the qualitative interpretation of the camera motion in each delimited shot (step 3); it exploits the motion information recovered at step 1 and relies on likelihood ratio tests. Section 5 contains results obtained on a real documentary video, using both the original and the MPEG-1 compressed/decompressed sequences, and an experimental comparison with a histogram comparison technique. Section 6 provides concluding remarks.

2 Dominant motion estimation

In order to retrieve the required motion information, we do not compute any dense velocity field. We only make use of the spatio-temporal derivatives of the intensity function. Since several motions may be present, we only seek for the estimation of the dominant one. We represent the corresponding motion field between two successive images by a global 2D parametric model. This will be sufficient to achieve the video partitioning as shown in the next section whatever this dominant motion may represent in the scene. If we aim at characterizing the type of shots in terms of camera traveling, zooming, panning, etc..., we need to further assume that this dominant motion corresponds to the apparent motion induced by the 3D camera movement. To estimate the dominant motion without prior motion segmentation, we have developed a technique based on robust statistics. We will only outline it hereafter; we refer the reader to [17] for more details.

This method (called RMR for Robust MultiResolution) takes advantage of a multiresolution framework and an incremental scheme. It minimizes a M-estimator criterion to ensure the goal of robustness to outliers formed by the points corresponding to secondary motions or to areas where the classical image motion equation [13] used is not valid. Any 2D polynomial motion model can be considered. We have chosen the affine motion model $\vec{w}_\Theta$ defined at point $p = (x, y)$, considering

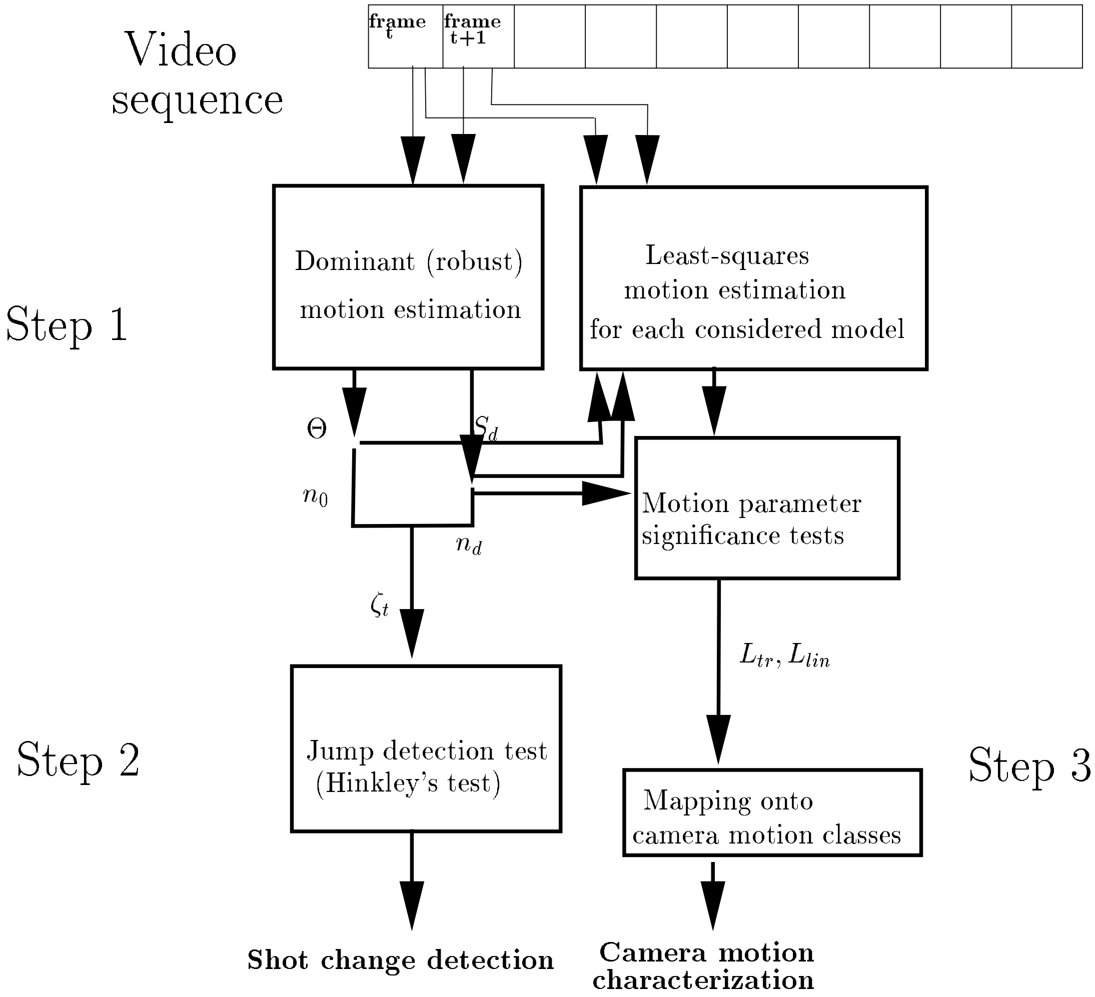

Video sequence
frame t
frame t+1
Step 1
Dominant (robust) motion estimation
Least-squares motion estimation for each considered model
Θ
S_d
n_0
n_d
Motion parameter significance tests
ζ_t
L_{tr}, L_{lin}
Step 2
Jump detection test (Hinkley's test)
Mapping onto camera motion classes
Step 3
Shot change detection
Camera motion characterization

Figure 1: *Flow chart of the method. Θ denotes the robustly estimated motion parameter, $\mathcal{S}_d$ the estimation support of this motion, and n_d the size of $\mathcal{S}_d$. n_0 designates the predicted size of the overlap between images $I(t)$ and $I(t+1)$, and ζ_t is defined as the ratio n_d/n_0.*

a reference point (x_g, y_g) by:

$$\vec{w}_\Theta(p) = \begin{pmatrix} a_1 + a_2(x - x_g) + a_3(y - y_g) \\ a_4 + a_5(x - x_g) + a_6(y - y_g) \end{pmatrix} \tag{1}$$

This model is a good trade-off between complexity and representativity. In practice, the image center is taken as reference point.

The parameter vector $\Theta = (a_1, a_2, a_3, a_4, a_5, a_6)$ is estimated between images $I(t)$ and $I(t+1)$ as follows :

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \sum_{p_i \in I} \rho\left(\mathrm{DFD}_\Theta(p_i)\right) \tag{2}$$

with $\mathrm{DFD}_\Theta(p_i) = I(p_i + \vec{w}_\Theta(p_i), t+1) - I(p_i, t)$, and $\rho(x)$ is a hard-redescending M-estimator. Here, we consider Tukey's biweight function. This function $\rho(x)$ and its derivative $\psi(x)$ depend on parameter C and are respectively defined as :

$$\rho(x, C) = \begin{cases} \frac{x^6}{6} - \frac{C^2 x^4}{2} + \frac{C^4 x^2}{2} & \text{if} |x| < C, \\ \frac{C^6}{6} & \text{otherwise.} \end{cases} \tag{3}$$

$$\psi(x, C) = \begin{cases} x(x^2 - C^2)^2 & \text{if} |x| < C, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

At each instant, a Gaussian pyramid of the image is built. ∇I will denote the spatial intensity gradient at the lowest resolution. We first search for the following estimate :

$$\widehat{\Theta^0} = \underset{\Theta^0}{\operatorname{argmin}} \sum_{p_i \in I} \rho\left(r_i\right) \tag{5}$$

$$\text{where } r_i = I(p_i, t+1) - I(p_i, t) + \nabla I(p_i, t).\vec{w}_{\Theta^0}(p_i) \tag{6}$$

r_i is a first-order differential version of the $DFD_\Theta(p_i)$. Let us recall that $r_i = 0$ is the usual image motion equation, introduced in [13], and adapted to the consideration of a parameterized motion field. Then, increments for the estimation of Θ are computed within a given resolution level, and from a coarse resolution level to a finer one, until the finest resolution level is reached. At step k where we aim at estimating the increment $\Delta\Theta^k$, given the current estimate $\widehat{\Theta}_k$ of the motion parameter vector, we can write $\Theta = \widehat{\Theta^k} + \Delta\Theta^k$. The estimated increment $\widehat{\Delta\Theta^k}$ is calculated as:

$$\widehat{\Delta\Theta^k} = \underset{\Theta^k}{\operatorname{argmin}} \sum_{p_i \in I} \rho\left(r_i\right) \tag{7}$$

where the residual r_i is now computed as :

$$r_i = I(p_i + \vec{w}_{\widehat{\Theta^k}}(p_i), t+1) - I(p_i, t) + \nabla I(p_i + \vec{w}_{\widehat{\Theta^k}}(p_i), t+1).\vec{w}_{\Delta\Theta^k}(p_i) \tag{8}$$

The estimated increment is then used to update the estimate $\widehat{\Theta^k}$:

$$\widehat{\Theta^{k+1}} = \widehat{\Theta^k} + \Delta\widehat{\Theta^k} \tag{9}$$

Increments are computed and cumulated until a convergence criterion is met or a given number of iterations is reached. The estimated motion parameter vector is projected onto the level of higher resolution, and serves as an initial value to compute some more increments and thereby to refine $\widehat{\Theta}$. This is performed down to the finest resolution level in the pyramid.

This multiresolution incremental robust estimator allows us to get an accurate computation of the dominant motion between two images, even if other motions are present.

3 Detection of shot changes

As explained in [17], the minimization problems defined in relations (5) and (7) is solved using a IRLS (Iteratively Re-weighted Least Squares) technique. The initial minimization problem (5) or (7) is in fact substituted for by the equivalent problem :

$$\widehat{\Delta\Theta} = \underset{\Delta\Theta}{\mathrm{argmin}} \sum_{p_i} \frac{1}{2}\, w_i\, r_i^2 \quad \text{with} \quad w_i = \frac{\psi(r_i)}{r_i}. \tag{10}$$

ψ is the derivative of the ρ function, and r_i is given by relation (6) or (8).

Once the dominant motion estimation step is completed, the final value of the weight w_i indicates if a point p_i is likely or not to belong to the part of the image undergoing this dominant motion. In the former case, w_i is close or equal to 1, in the later case, w_i is equal or close to 0 (outliers). We define the support of the dominant motion $\mathcal{S}_d$ as the set of points p_i satisfying $w_i \geq \nu$, where ν is a predefined threshold, (typically 0.2). The pertinent information here is in fact the size of this support.

Indeed, within a given shot, the size n_d of the support $\mathcal{S}_d$ is supposed to remain nearly constant. On the other hand, if a cut occurs between image $I(t)$ and $I(t+1)$, these images are completely uncorrelated. If we try to estimate a 2D affine motion model between $I(t)$ and $I(t+1)$, then it turns out that no coherent estimation can be derived. Thus, the corresponding support $\mathcal{S}_d$ "tends to vanish", and n_d is suddenly close to 0. In case of a progressive transition (a dissolve for instance), we have observed a less pronounced decrease of the size of the derived support, but the value n_d is still usable to detect such gradual changes.

Assuming the global dominant motion between t and $t{+}1$ close to the estimated global dominant motion between $t-1$ and t, a simple geometric projection from t to $t+1$ exploiting the estimated motion model parameters $\widehat{\Theta_{t-1}}$ enables us to determine the size n_o of the part of $I(t)$ that is likely to overlap $I(t+1)$ (i.e. the areas in $I(t)$ that disappear from are deduced). This size n_0 forms an upper bound for the size n_d of the dominant motion support estimated between t and $t+1$. We consider now the normalized ratio n_d/n_0. Let us denote ζ_t this ratio at time t. Since we have considered n_0 instead of the full image size, ζ_t is correctly normalized in $[0,1]$.

The point now is to define an appropriate criterion to validate significant jumps of the variable ζ_t among meaningless small variations. We resort to a cumulative sum test, Hinkley's test, [4], known to be robust (by taking into account "all the past" of the observed quantity), efficient, and inducing a very low computational load. The other attractive features of this test are two-fold. First, it can straightforwardly and accurately provide the jump instant. Secondly, due to its formulation (cumulative sum test), it can simultaneously handle both very abrupt and important changes like cuts, and gradual smaller ones like progressive transition without adapting the involved thresholds. Originally, it was designed to detect jump in mean of an observed signal, [4].

Two tests are performed in parallel to look for downwards or upwards jumps, respectively defined by:

$$\begin{aligned} S_k &= \sum_{t=0}^{k} \left(\zeta_t - m_0 + \frac{\delta_{min}}{2} \right) \quad (k \geq 0) \\ M_k &= \max_{0 \leq i \leq k} S_i; \text{ detection if } M_k - S_k > \alpha \end{aligned} \tag{11}$$

$$\begin{aligned} T_k &= \sum_{t=0}^{k} \left(\zeta_t - m_0 - \frac{\delta_{min}}{2} \right) \quad (k \geq 0) \\ N_k &= \min_{0 \leq i \leq k} T_i; \text{ detection if } T_k - N_k > \alpha \end{aligned} \tag{12}$$

in which the mean m_0 before the jump is estimated on-line. δ_{min} denotes the jump minimal magnitude that we want to detect, and α is a predefined threshold. The starting idea and the principle of Hinkley's test are illustrated in Figure 2. m_0 is re-initialized after each jump detection. If a jump is validated, with a short delay by construction, the jump location, i.e. shot change instant, is given by the last instant k' when $M_{k'} - S_{k'} = 0$, or, $T_{k'} - N_{k'} = 0$.

4 Camera motion characterization

After steps 1 and 2, we have determined the successive shots of the sequence, and we have estimated the parameters $\widehat{\Theta}_t$ of the dominant motion at each instant t. We can now exploit this information to characterize the type of the camera movement at each time t (step 3). To this end, we have used and adapted the qualitative interpretation method we presented in [11]. We express the parameter vector $\Theta = (a_1, .., a_6)$ in another basis of elementary motion sub-fields :

$$\Phi = (a_1, a_4, div, rot, hyp_1, hyp_2) \text{ with :}$$

$$\begin{array}{rlrl} div = \frac{1}{2}(a_2 + a_6) & & rot = \frac{1}{2}(a_5 - a_3) \\ hyp_1 = \frac{1}{2}(a_2 - a_6) & & hyp_2 = \frac{1}{2}(a_3 + a_5). \end{array}$$

These last four terms, divergence, curl, hyperbolic terms, are more convenient for an easy physically meaningful interpretation of the dominant motion.

If the dominant motion is a pure panning (resp., camera tilt), then only parameter a_1, (resp., a_4) is supposed to be non zero. If it is a zooming or forward traveling, div is supposed to be the only non zero linear parameter. If it is a lateral traveling, and if the scene cannot be assimilated to a fronto-parallel plane, all the parameters are supposed to be non zero. Of course, if the camera is static, all the parameters are equal to 0. In practice, due to noise, estimation errors, and the use of an approximate motion model, these quantities cannot be strictly equal to 0 if it should be the case. The aim is then to be able to decide whether these estimated values are significant or not.

We resort to a statistical approach based on likelihood ratio tests. As shown in [11], this is the most efficient way to tackle this problem, compared to direct thresholding of the parameters values, or the use of statistical information criteria such as Akaike (AIC), or Rissanen (MDL) ones. The delicate and unstable threshold selection is turned into a better-controlled issue of setting a threshold on a likelihood ratio. Also, the model error related to the inadequacy of the model to explain the data, can be taken into account, along with measurement noise. We will test in turn each component of the motion parameter vector Φ.

Two competing hypotheses will be considered. The first one, denoted H_0, assumes that the considered component of Φ is significant. The second one, denoted H_1, considers that on the contrary it is equal to 0, the five other parameters being let free. Let us note $\widehat{\Phi}_{m_0}$ and $\widehat{\Phi}_{m_1}$ the motion model parameter vectors associated to respectively hypothesis H_0 and hypothesis H_1. A remarkable property of such a test is that it is independent from the value of the descriptors that remain free. For each hypothesis, we can define the associated likelihood function f, where the considered random variables are the quantities r_i defined in relation (8). They are assumed to be independent, and to all follow a zero-mean Gaussian law. The variance $\sigma^2_{m_l}$ of r_i is a posteriori estimated as follows :

$$\widehat{\sigma_{m_l}}^2 = \frac{1}{n_d} \sum_{p_i \in \mathcal{S}_d} r_i(\widehat{\Phi}_{m_l})^2, \qquad l = 0, 1 \tag{13}$$

The expressions of the two likelihood functions f for the optimized values of motion parameters $\widehat{\Phi}_{m_l}$ $(l = 0, 1)$ are given by :

$$\begin{aligned} f(\widehat{\Phi}_{m_l}) &= \prod_{p_i \in \mathcal{S}_d} \left(\frac{1}{\sqrt{2\pi \widehat{\sigma_{m_l}}^2}} \ \exp \left(\frac{-\sum_{p_i \in \mathcal{S}_d} r_i^2}{2\widehat{\sigma_{m_l}}^2} \right) \right) \\ &= \left(\frac{1}{\sqrt{2\pi \widehat{\sigma_{m_l}}^2}} \right)^{n_d} \ \exp \left(-\frac{n_d}{2} \right), \qquad l = 0, 1 \end{aligned} \tag{14}$$

To test the significance of a given motion parameter vector component, the two motion parameter vectors corresponding to the two hypotheses are to be first estimated.

$\Phi_{m_0} = (a_1, a_4, div, rot, hyp_1, hyp_2)$ is the full affine model, and Φ_{m_1} is the affine model, for which the component to be tested is constrained to 0. If we take the example of the analysis of the divergence term, we have for hypothesis H_0, $\Phi_{m_0} = (a_1, a_4, div, rot, hyp_1, hyp_2)$, and for hypothesis H_1, $\Phi_{m_1} = (a_1, a_4, 0, rot, hyp_1, hyp_2)$. Both parameter vectors are estimated on the dominant motion support $\mathcal{S}_d$ determined at step 1. They are estimated in a multiresolution and incremental way similar to the one described in Section 2 and using the same image motion constraint. The estimation of both parameter vectors benefits from the knowledge of the dominant estimation support $\mathcal{S}_d$, which has been computed by the shot detection change step. The computational load is thus much reduced, since a usual least-square estimation technique can be used instead of an IRLS technique.

To decide which hypothesis is selected, the following likelihood log-ratio test is performed :

$$ln(\frac{f(\widehat{\Phi}_{m_1})}{f(\widehat{\Phi}_{m_0})}) \underset{H_0}{\overset{H_1}{\leq}} \lambda$$

If the ratio is lower than the threshold λ, the component at hand is declared to be significant, otherwise it is considered to be null. In other terms, the test comes down to comparing the adequacy of the constrained and unconstrained motion models to the data, by means of an appropriate comparison of the resulting residual variances.

If the component is judged significant, we can use the sign of the parameter to infer more information. As far as the divergence term is concerned, this will allow us to differentiate between zoom-in or zoom-out or equivalently, forward or backward traveling (with a perspective image projection model).

These tests are performed in turn on the different components of Φ while taking into account the physically possible configurations of null parameters. Then, we can identify the type of the camera movement. Since the translational parameters of the affine motion model depend on the choice of the reference point, the information they carry may be meaningless depending on the situation at hand. We denote respectively L_{tr} and L_{lin} the binary label decision vectors respectively associated to the significance of the translational and the linear dominant motion parameters. Considering the only physically possible motions of the camera, $L = (L_{tr}, L_{lin})$ can be mapped onto six camera motion classes as follows :

L_{lin}	L_{tr}	Camera motion
$(0,0,0,0)$	$(0,0)$	Static camera
$(0,0,0,0)$	$\neq (0,0)$	Pan, tilt, or sideways camera traveling if the scene background is parallel to the image plane
$(div, 0, 0, 0)$		Zoom (in/out) or forward/backward traveling
$(0, rot, 0, 0)$		Rotation around the optical axis
$(div, rot, 0, 0)$		Zoom (in/out) or forward/backward traveling and rotation around the optical axis
(div, rot, hyp_1, hyp_2)		Sideways camera traveling, (if the scene background is not approximately a plane parallel to the image) or complex camera motion.

The setting of λ is carried out as follows [11]. It is non critical for the translational components, because cancelling one of two these components causes a dramatic rise in the variance $\widehat{\sigma_{m_l}}$ of the residual r_i, if this component is significant. Values within the interval $\lambda \in [1, 3.5]$ have been found satisfactory. Cancelling a significant linear component of the model causes a less marked increase in this same variance. Hence, a reasonable interval for setting λ is $[1, 2]$.

Once this last processing step achieved, we can also determine another kind of shot change which might not be detected in step 2. It corresponds to camera maneuver, like a panning followed by a zooming. It can be achieved by simply detecting changes over time of the camera motion type. This partitioning is only based on qualitative motion.

In fact, steps 2 and 3 can be performed in parallel. As a result, the video is partitioned into shots, which are themselves divided into sub-shots of homogeneous camera motion.

5 Results

This approach has been validated by experiments with several kinds of video sequences. We report here results obtained on a part of a real documentary video presenting our Institute, which we will call the Irisa sequence. It contains usual features related to film shooting and editing. Besides, this video comprises shots involving both moving elements in the scene and moving camera. From time t_1 to t_{21} (shot 1), there is a special video effect: a general view of the site is continuously magnified over a map of Brittany. From t_{22} to t_{110} (shot 2), we can observe a global aerial panning, progressively stopped at the end of the shot, over the campus site. A dissolve transition appears from t_{111} to t_{115}. From t_{116} to t_{140} (shot 3), the camera is static. At time t_{141}, a first cut occurs. From t_{142} to t_{187} (shot 4), the camera motion is first a pan, and then becomes a pure forward traveling. A second cut occurs at time t_{188}. From t_{189} to t_{268} (shot 5), the camera is panning an indoor scene containing elements at different depth positions. There is a second transition effect from t_{269} to t_{278}. Shot 6 (t_{279}-t_{300}) in the video corresponds to a backward traveling of the camera, but a moving person is occupying a significant area in the image. From t_{189} onwards, important depth variations in the scene can be noticed.

The performance of the method was first tested on the original image sequence. Computational time for a pair of frames (of size 256x256 pixels) is about $1.5s$ on a workstation Sun-UltraSparc, down to about $0.9s$ if camera motion characterization is not selected. The evolution of the normalized motion support ratio ζ_t along the sequence is shown on Figure 3(a)(b). The on-line measured mean value m_0 of ζ_t during the shot, and the detected jumps in ζ_t are also indicated. Cuts and progressive transitions are correctly identified, and the bounds of the two progressive transitions (start and end time instants) and of the video effect are accurately determined. One can notice the smaller, but longer decrease in ζ_t during progressive transitions or the video effect, relatively to cuts. For all experiments, the value of α is kept constant and equal to 0.1. Figure 4 illustrates the use of the estimation support for shot change detection purposes. The maps of the weights w_i are shown in three different cases : Fig 4a,d shows an example of a pair of successive frames within a shot. Since almost all the pixels conform to the estimated dominant motion, we have $\zeta_{21} \approx 1$. Fig 4b,e corresponds to the case of a pair of frame across a cut, in which case, almost all points are outliers ($\zeta_{141} \approx 0$). Fig 4c,f shows what happens within a progressive transition shot ($\zeta_{144} \approx 0.5$). Finally, Fig 4d,g illustrates the case of a pair of frames within a shot including a large mobile person ($\zeta_{299} \approx 0.7$).

The proposed method has been compared with a standard histogram comparison method. Denoting $H_t(n)$ the histogram value at time t for grey level n, and DH_t the measured histogram difference between two successive images, the distance used for this test is :

$$DH_t = \sum_{n=1}^{N} \frac{|H_t(n) - H_{t+1}(n)|^2}{H_{t+1}(n)} \tag{15}$$

where N is the number of grey values.

A comparison of the variable ζ_t processed with our method with the one (DH_t) delivered by the histogram comparison technique is shown in Figure 3. Cuts can be easily detected and correctly located with both methods. Progressive transitions are also correctly detected. It can be seen, though, that the second dissolve (frames 269 to 278) causes the emergence of two neighbouring peaks in DH_t values In such a case, a simple thresholding on DH_t would erroneously detect two transitions. Also, the beginning of the first fading effect is not as accurately determined with the histogram difference procedure. Besides, the video effect (frames 0 to 21) causes a relative variation in DH_t which is far below what is obtained with ζ_t for our method. In general, the relative variations of DH_t due to a progressive transition are rather weak. Thus, delicate thresholding is involved. In the presence of mobile objects, a high threshold value is required, which in turn may not guarantee a correct detection of all transitions.

The result of camera motion analysis is shown in Figure 5. The threshold λ was set to 1.5. Decisions are in accordance with the description of the video given in the text. Taking into account that camera motion across cuts or during transitions are obviously not significant, it may be noticed that hyperbolic terms are only stated as significant during the last shot, where camera motion is complex and the background scene is far from parallel to the image plane, which is quite coherent with the real camera motion described above.

In order to evaluate the ability of the method to process typical video data stored after compression, the method was tested on the reconstructed Irisa sequence once processed by a MPEG encoder, considering a 1.5 Mbit/s bit rate and a I/P frame distance of 3. The algorithm parameters α and λ remained unchanged. Partitioning into shots lead to identical results as with the original sequence, in terms of number of shots and transition time instants. The effect of the compression-decompression process on ζ_t is shown on Figure 6. Perturbations mainly occur at transitions, because the location of I-frames in MPEG is independent from the structuration into shots, thus causing particularly strong frame prediction errors in the coding scheme. Camera motion characterization was different for 7 frames out of 298. In 6 of these cases, confusion actually consists in mislocating from at most 2 frames the transition between two successive types of camera motions within a shot. The other difference is a pan wrongly labeled as "complex motion" in a difficult situation where the scene is partly seen through a window. A MPEG-2 encoder was also employed, with a bit rate of 10 Mbit/s. The results are identical to those of the original sequence.

An interface module was designed to visualize a summary of the video structure inferred from the video partitioning performed by the method (Figure 7). Each shot is represented by a key-frame (we simply chose the median frame in the shot), the number of the shot, its bounds in terms of frame number. The user can then select a particular shot of interest and view a mosaic-type image constructed by combining frames within the selected shot again by exploiting the estimated dominant motion at each instant. For most types of sequences, such mosaic images are appropriate for a global representation of scene contents. Beside it, the list of camera motion types identified during that shot are annotated. For examples shown on Figure 7a,b, one can relate the shape of mosaic image to the sequence of camera motions displayed beside it.

6 Conclusion

We have described in this paper an original, unified and efficient approach to video partitioning in the context of content-based video indexing. It involves the detection of shot changes and characterization of the camera motion. All the information required to achieve these two goals results from the direct estimation of a 2D affine motion model accounting for the 2D dominant motion within each successive image pair of the sequence. This method can handle situations where moving objects are present in the scene observed with a mobile camera. The solution has been defined in a well-formalized way involving general motion models, statistical techniques, and no restriction on the kind of scene and camera movements. This last point, along with having several shot detection tests to cope with the various possible shot transitions, stands among the usual shortcomings of many other techniques proposed so far. In the method proposed here, a single test and parameter value detects both cuts and progressive transitions. Computational time is low. The technique was also validated on MPEG1 and MPEG2 sequences. We are now

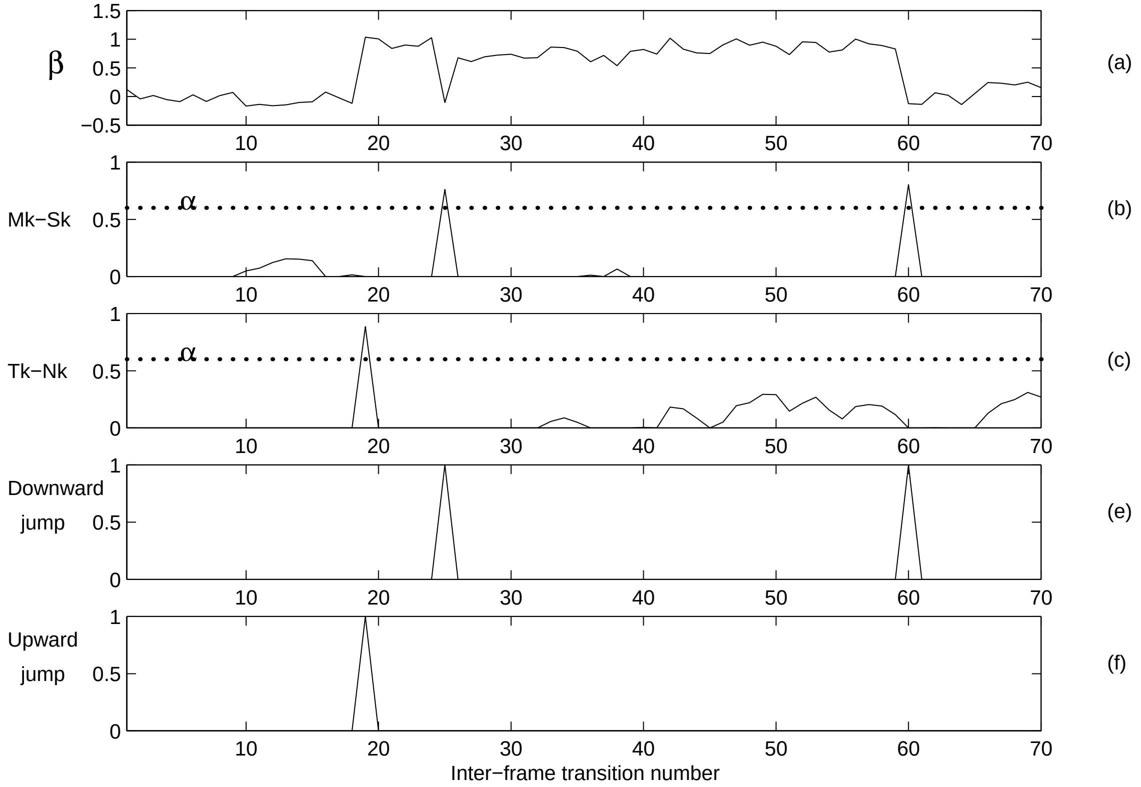


Figure 2: *Principle of Hinkley's test. All variables are plotted against the frame number in the sequence. (a) The variable in which jumps are to be detected, denoted β in this example, is shown in Fig 2a. When it suddenly quits the interval defined by the two bounds $m_0 - \delta_{min}/2$ and $m_0 + \delta_{min}/2$, it causes a rise in either $M_k - S_k$ (Fig 2b) or $T_k - N_k$ (Fig 2c), depending whether the jump is directed upwards or downwards. When $M_k - S_k$ (respectively $T_k - N_k$) exceeds the threshold α, a downward (see Fig 2e) (resp. upward (see Fig 2f)) jump is detected. The jump is then located at the frame number just after the last one for which either M_k or N_k (depending on the case at hand) exceeds 0.*

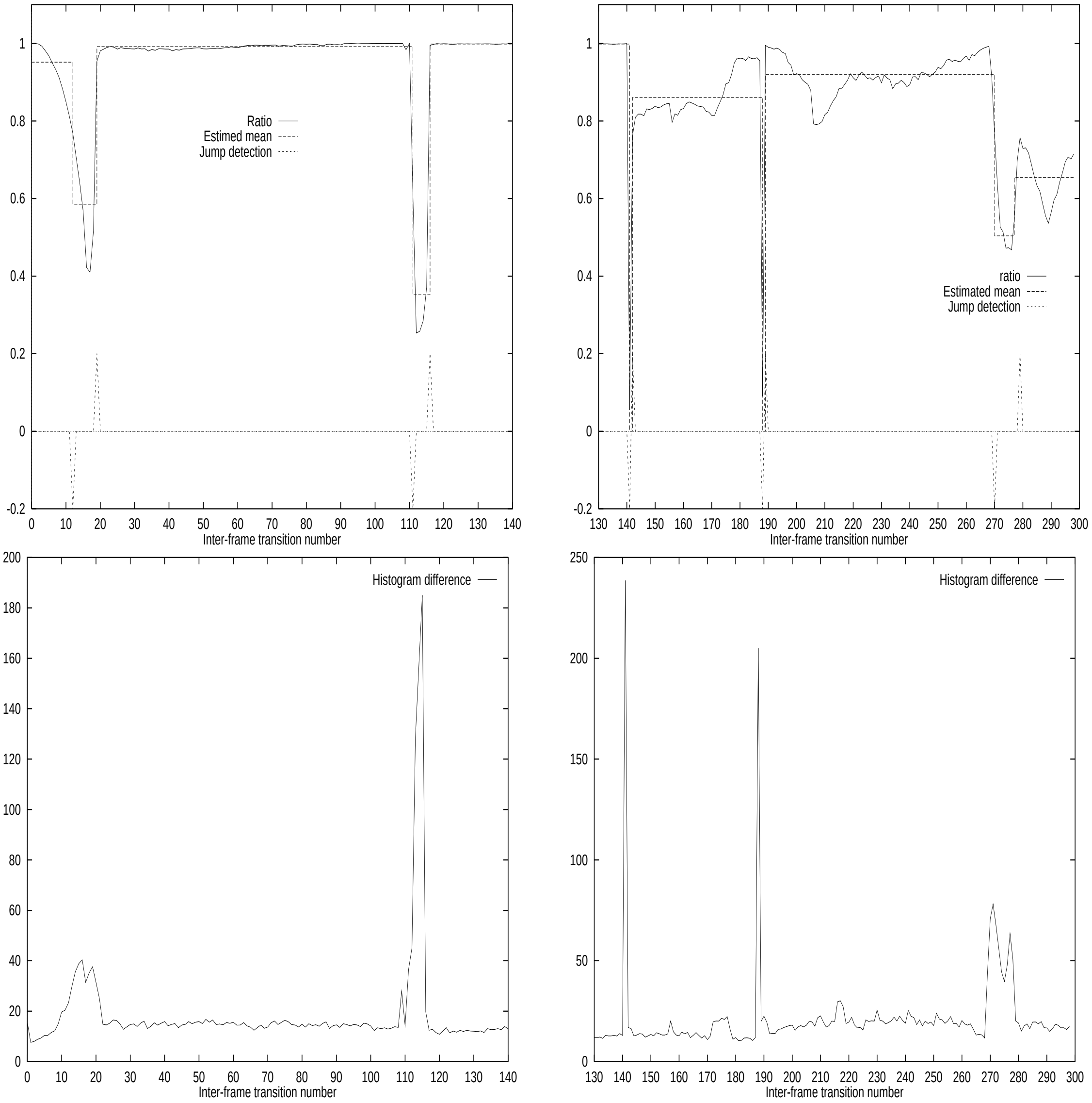


Figure 3: *IRISA sequence: results of video partitioning from t_1 to t_{140} and from t_{130} to t_{300}. Figures 3a and 3b correspond to the use of Hinkley's test. The ζ_t variable is plotted in continuous line, the on-line estimated mean m_0 in dashed line. The validated jump instants (beginning and end of shot transitions) are indicated at the bottom of each figure. $\alpha = 0.1$, $\delta_{min} = 0.2$. Figure 3c and d correspond to the use of histogram differences. The variable DH_t is plotted in continuous line. (c),(d) is to be compared with (a)(b).*

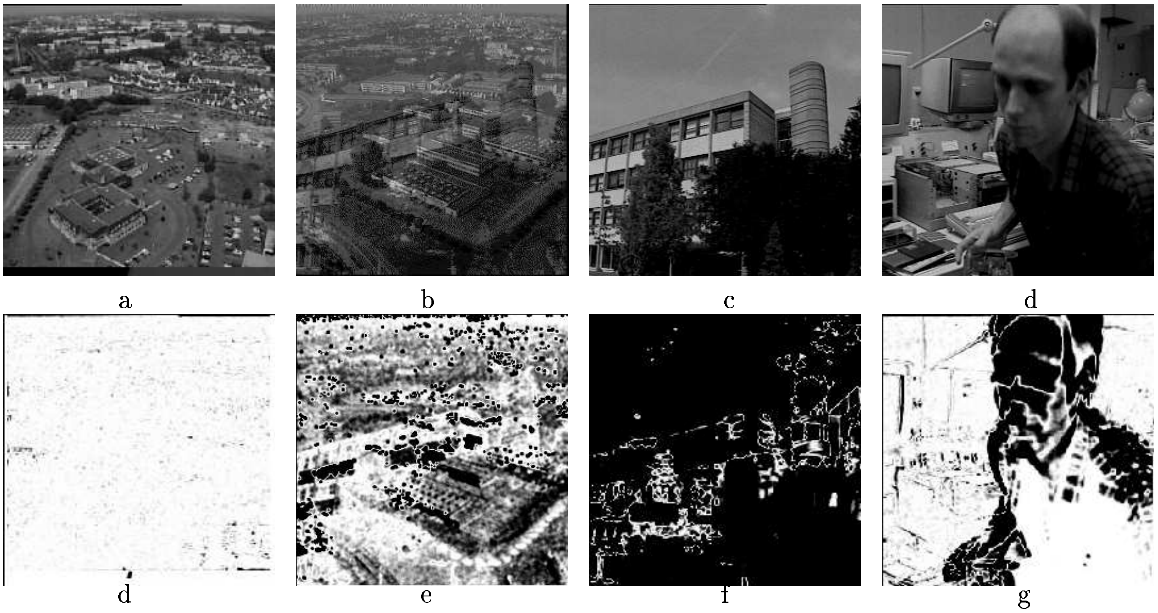


Figure 4: *IRISA sequence: Examples of estimation supports for four types of situations within a pair of successive frames (the estimation support is displayed in white): a,b,c,d show the first of the two processed frames, respectively, frames 21 (within a shot) , 114 (within a "dissolve" transition), 141(shot cut), and 299 (within a shot, with a large mobile object), and d,e,f,g show the respective corresponding estimation supports.*

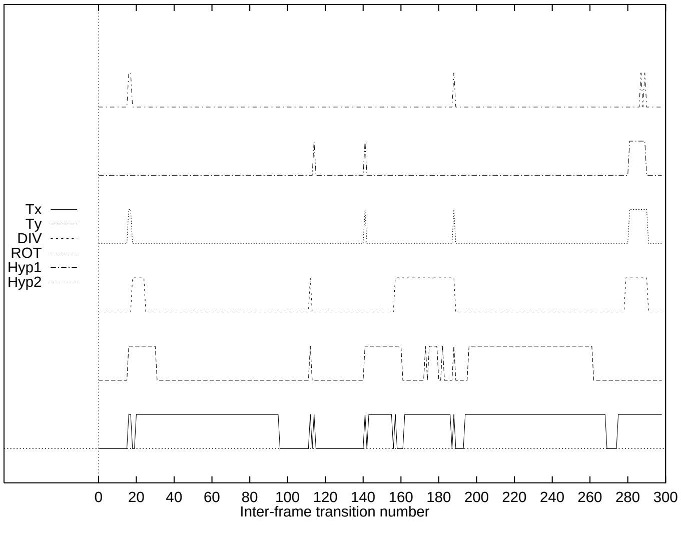


Figure 5: *IRISA sequence: output of the likelihood ratio tests. From bottom to top, outputs of the likelihood test deciding upon the significance or nor of respectively parameters a_1, a_4, div, rot, hyp_1 and hyp_2 along the image sequence.* $\lambda = 1.5$.

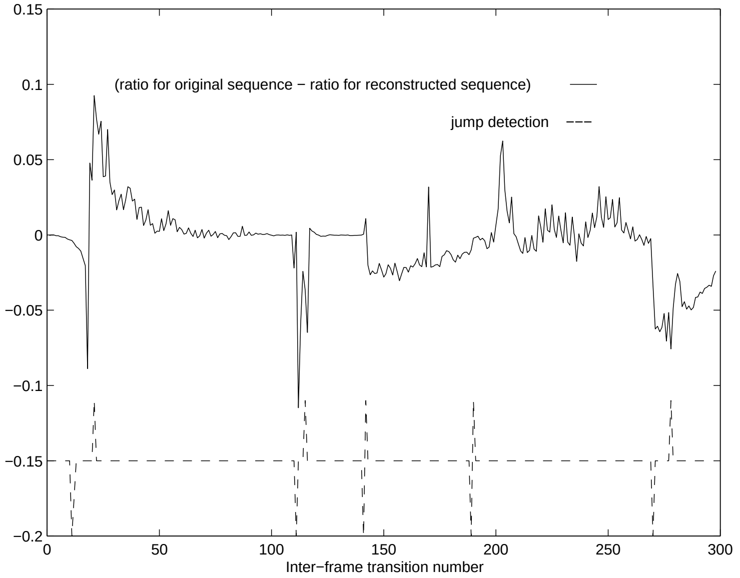


Figure 6: *Comparison of ζ_t for the original and MPEG1-reconstructed sequences. The difference between the values of ζ_t for the two sequences is drawn, as well as the detected jumps in the signal.*

investigating other issues related to content-based video indexing which is of increasing interest in numerous application domains, particularly concerning the analysis of each shot according to its spatio-temporal content [12].

References

[1] P. Aigrain and P. Joly. – The automatic real-time analysis of film editing and transition effects and its applications. – *Computer & Graphics*, 18(1):93–103, 1994.

[2] P. Aigrain, H.J. Zhang, and D. Petkovic. – Content-based representation and retrieval of visual media : a state-of-the-art review. – *Multimedia Tools and Applications*, 3(3):179–202, November 1996.

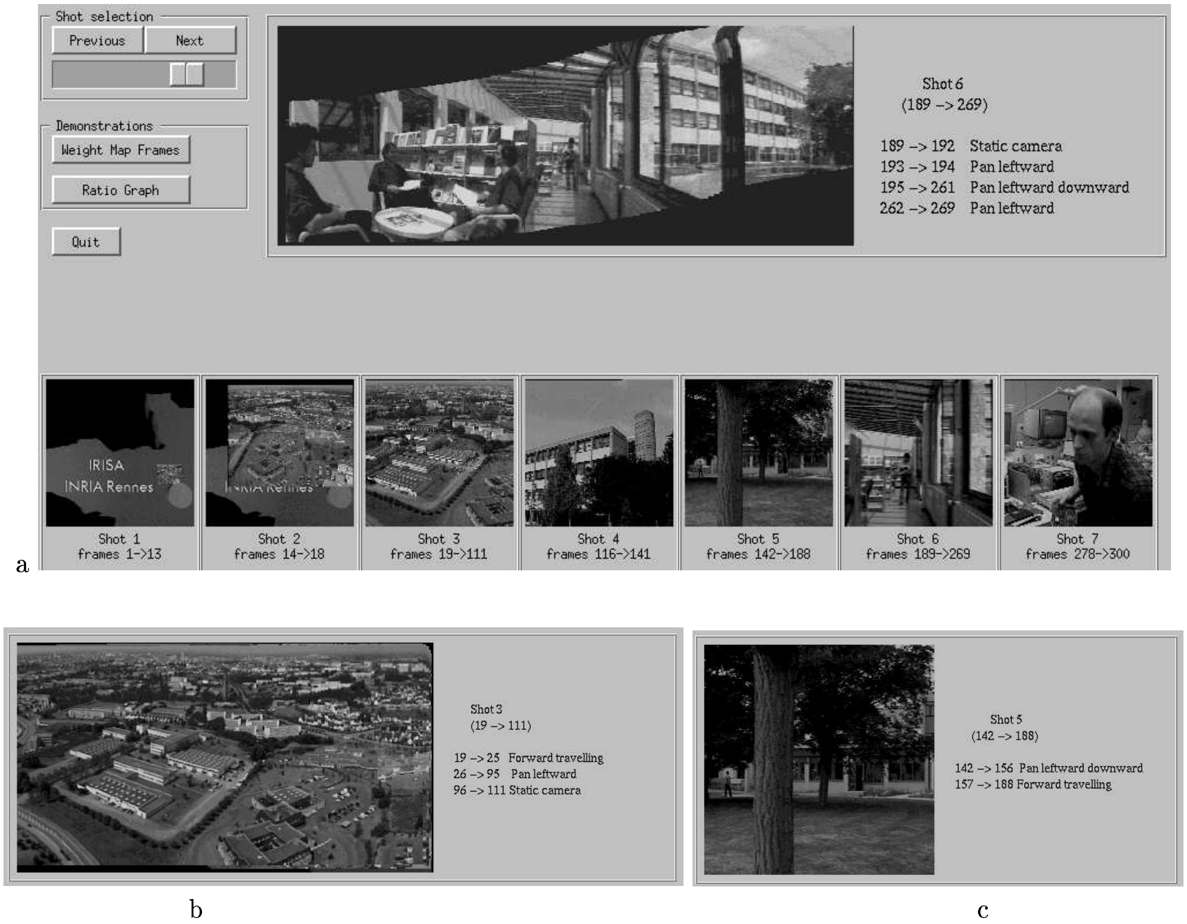
[3] S. Ayer and H.S Sawhney. – Compact representations of videos through dominant and multiple motion estimation. – *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(8):814–830, August 1996.

[4] M. Basseville. – Detecting changes in signals and systems - a survey. – *Automatica*, 24(3):309–326, 1988.

[5] J.S. Boreczky and L.A. Rowe. – Comparison of video shot boundary detection techniques. – In *In I.K. Sethi and R.C. Jain, editors, Proceedings of IS-T/SPIE Conference on Storage and Retrieval for Image and Video Databases IV, Vol. SPIE 2670*, pages 170–179, 1996.

[6] P. Bouthemy and F. Ganansia. – Video partitioning and camera motion characterization for content-based video indexing. – In *Proc. of 3rd IEEE Int. Conf. on Image Processing*, volume I, pages 905–909, Lausanne, Sept 1996.

[7] J.D. Courtney. – Automatic video indexing via object motion analysis. – *Pattern Recognition*, 30(4):607–625, April 1997.



```
    Shot 6
  (189 -> 269)

189 -> 192   Static camera
193 -> 194   Pan leftward
195 -> 261   Pan leftward downward
262 -> 269   Pan leftward
```

```
    Shot 3
  (19 -> 111)

19 -> 25   Forward travelling
26 -> 95    Pan leftward
96 -> 111 Static camera
```

```
      Shot 5
   (142 -> 188)

142 -> 156  Pan leftward downward
157 -> 188 Forward travelling
```

Figure 7: *IRISA sequence: visualization of the video summary (a). A key-frame per detected shot is displayed (bottom row), and, for a user-selected shot, more information about its contents is supplied (mosaic image and sequence of camera motion types). Examples shown correspond to (a) shot 6, (b) shot 3 and (c) shot 5.*

[8] G. Davenport, T.A. Smith, and N. Pincever. – Cinematic primitives for multimedia. – *IEEE Computer Graphics and Applications*, pages 67–73, July 1991.

[9] M. De Marsico, L. Cinque, and S Levialdi. – Indexing pictorial documents by their content : a survey of current techniques. – *Image and Vision Computing*, (15):119–141, 1997.

[10] M. Flickner et al. – Query by image and video content : the QBIC system. – *IEEE Computer*, pages 23–32, Sept. 1995.

[11] E. François and P. Bouthemy. – Derivation of qualitative information in motion analysis. – *Image and Vision Computing*, 8(4):279–287, Nov. 1990.

[12] M. Gelgon and P. Bouthemy. – A hierarchical motion-based segmentation and tracking technique for video storyboard-like representation and content-based indexing. – In *WIAMIS'97 Workshop on Image Analysis for Multimedia and Interactive Services*, pages 93–98, Louvain-la-Neuve, Belgium, June 1997.

[13] B. Horn and B. Schunck. – Determining optical flow. – *Artificial Intelligence*, 17:185–203, 1981.

[14] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. – Efficient representations of video sequences and their applications. – *Signal Processing : Image Communication*, (8):327–351, 1996.

[15] P. Joly and H.K. Kim. – Efficient automatic analysis of camera work and microsegmentation of video using spatiotemporal images. – *Signal Processing : Image Communication*, (8):295–307, 1996.

[16] A. Nagasaka and Y. Tanaka. – Automatic video indexing and full-video search for objects appearances. – *Visual Database Systems II*, pages 113–127, 1992. – E. Knuth and L.M. Wegner (eds.), Elsevier Science Publ.

[17] J.M Odobez and P. Bouthemy. – Robust multiresolution estimation of parametric motion models. – *Jal of Visual Communication and Image Representation*, 6(4):348–365, December 1995.

[18] N.V. Patel and I.K. Sethi. – Video shot detection and characterization for video databases. – *Pattern Recognition*, 30(4):607–625, April 1997.

[19] A. Pentland, R.W. Picard, and S. Sclaroff. – Photobook : Content-based manipulation of image databases. – Technical Report 255, MIT Media Lab, Nov. 1993.

[20] G. Sudhir and J.C.M. Lee. – Video annotation by motion interpretation using optical flow streams. – *Jal of Visual Communication and Image Representation*, (4):354–368, Dec. 1996.

[21] Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakata. – Videomap and videospaceicon: Tools for anatomizing video content. – *Proc. Conf. INTERCHI'93*, pages 131–136, April 1993.

[22] H. Ueda, T. Miyatake, and S. Yoshizawa. – Impact: An iteractive natural-motion-picture dedicated multimedia authoring system. – *Proc. Conf. ACM CHI'91*, pages 343–350, 1991.

[23] M.M Yeung and B. Liu. – Efficient matching and clustering of video shots. – In *Proc of Second IEEE Int. Conf. of Image Processing*, pages 338–341, Washington, October 1995.

[24] H.J. Zhang. – Swim : A prototype environment for visual media retrieval. – *in Recent Developments in Computer Vision*, pages 531–540, 1996. – S.Z. Li, D.P. Mital, E.K. Teoh, H. Wang (Eds.),LNCS 1035, Springer.

[25] H.J Zhang, A. Kankanhalli, and S.W. Smoliar. – Automatic partitioning of full-motion video. – *Multimedia Systems*, 1:10–28, 1993.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399