Video summarization and semantics editing tools

Li-Qun Xu^a, Jian Zhu, and Fred Stentiford

Future Content Group, BTexaCT Research
Adastral Park, Martlesham Heath, Ipswich IP5 3RE, UK

ABSTRACT

This paper describes a video summarization and semantics editing tool that is suited for content-based video indexing and retrieval with appropriate human operator assistance. The whole system has been designed with a clear focus on the extraction and exploitation of motion information inherent in the dynamic video scene. The dominant motion information has been used explicitly for *shot boundary detection*, *camera motion characterization*, *visual content variations description*, and for *key frame extraction*. Various contributions have been made to ensure that the system works robustly with complex scenes and across different media types. A window-based graphical user interface has been designed to make the task very easy for interactive analysis and editing of semantic events and episode where appropriate.

Keywords: Video content analysis, dominant motion estimation, M-estimator, shot change detection, user interface, semantics editing

1. INTRODUCTION

With the convergence of computing, mobile, consumer electronics and Internet technologies, there is now large amount of audio-visual and textual data that is either accessible on-line from archives administered by various service / content providers, Internet portals or made available in home storage mediums for personal entertainment and education purposes. The needs for efficient retrieval of desirable information when and where as required from increasingly large data stores have become ever so imperative. This necessitates effective means by which the content of the data can be automatically characterized, structured, indexed and retrieved, alleviating and ultimately doing away with the slow, labor-intensive manual annotation and search tasks. Though content-based visual information retrieval has been a subject attracting enormous interest from both academic and business communities over the last decade,¹⁻³ automation and understanding of the whole process remains very complex and a challenging problem in the new millennium.⁴ This is due to the existing variety types of media and diverse specific requirements, and the difference between current computational feature-based semantics analysis approach and the paradigm of human cognitive perception and understanding of high-level events, episodes or stories. As an intermediate step, however, automating large part of this tedious process chain at the low and middle-level feature analysis stages to facilitate human operators would be a desirable and achievable goal.

We describe here our recent efforts in this respect on video information analysis and summarization through key frames representation and the design of user interface to facilitate semantics annotation. Some related research activities can be found in Refs. 5-7, motivated by different application problems and using various visual-content variations modeling features. We, however, focus on the use of generic motion information, mainly induced by the *intentional* movements of the camera, to automatically capture the scene changes and potentially the semantics behind these changes. In the current research we limit our scope to the low-level visual content variations modeling tasks, while leaving the high-level task of grouping the scenes into meaningful episodes or stories to a flexible browsing and semantics editing tool to perform. It should be pointed out that the use of motion information, has gained more popularity recently for multimedia content characterization and retrieval, spurred by the emergence of MPEG-7 content description Standards.⁸ The motion extracted from the video scene can take various forms as distributed or parametric model, either local or global or object-based

Correspondence author: Li-Qun Xu, Email: li-qun.xu@bt.com. Other author information: Jian Zhu: currently with School of Engineering, The University of Warwick, Coventry, CV4 7AL, UK; Email: zhuj@eng.warwick.ac.uk; Fred Stentiford: Email: fred.stentiford@bt.com.

representation. Some representative work and application issues of using motion as primary cues in video and media databases analysis can be found in Refs. 9-15.

The paper is organized as follows. In Section 2 we present the proposed system block diagram, briefly outlining the functions of each module and its underlying techniques. This is followed in Section 3 by a detailed description of the dominant motion estimation procedure and the techniques for shot change detection based on the motion-induced visual content variations feature of the video scene. In Section 4 we discuss the key frame extraction technique based on qualitative camera motion and a well-defined visual content variation measurement. In Section 5, evaluation of aspects of the system performance is given using an MPEG-7 test video set, which is a culture documentary containing various dynamic scenes of different tempo and variable length with almost all known special scene change editing effects. The paper concludes in Section 6 with a summary and discussion of future research directions.

2. SYSTEM ARCHITECTURE

Figure 1 shows the schematic diagram of the video summarization and semantics editing system that we have developed. The working mechanism of the system is briefly as follows. The luminance part of the input *YUV* raw video stream first undergoes geodesic reconstruction¹⁶ to reduce noise disturbance in the video data. This is followed by the process of dominant motion estimation, which computes a 2D affine motion model parameters for successive pair of frames and as by-product a motion-based visual content variations measurement ζ_i for each frame, the so-called *effective supporting factor*. The key property of this variable is that it relates closely to the content change of a dynamic scene. When two consecutive frames are within the same shot, or the visual content does not change appreciably, it remains more or less stable, when the two frames belong to two different shots or go through gradual changes, it presents substantial downwards or upwards accumulative jumps. Based on these observations, a further module for statistical hypothesis testing can be employed to detect shot change boundaries. In view of certain unexpected visual scene changes, such as large foreground moving objects, severe perspective distortions, sudden changes in illumination and various fancy editing effects, a shot-based sanity checks module has been adopted to validate the obtained results, reducing especially the positive false alarms.

Figure 1. A schematic diagram of the video summarization and semantics editing system

On the other hand, from the 2D affine model, a qualitative camera motion description can be acquired again through statistical hypothesis tests. This information, together with the detected shots and a histogram-based visual content variations metric, defined in *YUV* color space for the *supporting pixels* of the dominant motion, provides a very flexible and sensible way to define the key frame(s) to summarize the visual content for each shot. This set of elementary indexing information is then fed into a video browsing and semantic editing graphical user interface to perform various interactive visualizations and editing functions. A few of these are, for example, playing back any shot sequence corresponding to a key

frame, grouping consecutive visually similar shots into a semantic event or episode to compute new key frames representation or other image descriptors if needed. These results are ready to form a (pseudo-)semantics based indexing file for content-based information retrieval. In the subsequent sections we will explain the main techniques employed for each functional module.

3. SHOT CHANGE DETECTION

We base our shot change detection on the analysis of the apparent dominant motion present in the dynamic scene. As the dominant motion is primarily induced by the camera movement when a shot is being captured, the characteristics of this motion (could be zero), if estimated reliably, will provide essential computational visual / semantic cues to where the shot boundaries might lie. It also allows for new means for evaluation of the visual content changes of the scene to help capture the interesting moment of the scene.

3.1. Dominant motion estimation

In the field of digital video processing and its variety applications,¹⁷ there are various methods that have been proposed to estimate the global dominant motion. For example, one can use the fast and accurate block matching method¹⁸ to estimate block motion vectors for consecutive frames, and then employs the low cost 2D parametric global motion estimation strategy¹¹ to approximate these vectors, which results in using only 4 parameters to roughly characterize the motion of the scene. On the other hand, one can obtain very high accuracy of motion description at the cost of more computational complexity.¹⁴

In the work reported here we adopt the motion model and approach studied in Ref.19, which is a simplified 2D affine motion model depicted as follows:

$$\begin{cases} \Delta x = X - x = a_1 + a_2(x - x_r) + a_3(y - y_r) \\ \Delta y = Y - y = a_4 + a_5(x - x_r) + a_6(y - y_r) \end{cases} \quad (1)$$

where (x_r, y_r) is the reference point, usually the center of the frame. This model indicates that a pixel (x, y) , in the reference frame (n-1), moves to the position (X, Y) in the current frame (n). This simplified model works generally quite well provided that the 3-D scene depth is moderately small relative to the distance from the camera. It is a good tradeoff between computational complexity and accuracy. We use a vector of seven elements $\theta = [a_1, a_2, a_3, a_4, a_5, a_6, \xi]^T$ to denote the parameter set of the model, with the last element ξ accounting for the average luminance difference between the two frames.

In order to speed up the computation and yet discount the possible outliers that are moving pixel patches caused by small man made or natural moving objects, a multi-resolution, robust estimation algorithm is adopted. Specifically, the robust M-estimator has been studied. Depending on the accuracy required the algorithm could stop at any coarser level of the up to five-level image pyramid. An iteratively re-weighted least square solution is used to realize the M-estimator according to,

$$\begin{aligned} \theta_{k+1} &= \theta_k + \Delta\theta \\ \Delta\theta &= [\Delta a_1, \Delta a_2, \Delta a_3, \Delta a_4, \Delta a_5, \Delta a_6, \Delta \xi]^T \end{aligned} \quad (2)$$

at iteration k , which is meant to minimize the weighted square error of pixels over the whole frame,

$$E(\Delta\theta) = \sum \rho(r_{k+1}) = \sum \left(\frac{1}{2} w \cdot r_{k+1}^2 \right) = \sum [w \cdot (A' \cdot \Delta\theta + b)^2], \quad (3)$$

where ρ is the Tukey's biweight function, and w , derived from ρ , is the associated weight for each pixel,

$$\begin{aligned} A &= [g_x, g_x(x - x_r), g_x(y - y_r), g_y, g_y(x - x_r), g_y(y - y_r), 1]^T \\ b &= I(X_k, Y_k, n) - I(x, y, n-1) + \xi_k \end{aligned} \quad (4)$$

g_x and g_y are the spatial gradients of pixel at point (X_k, Y_k) . Due to the fractional pixel position $I(X_k, Y_k, n)$ is calculated using the familiar *bilinear interpolation* operation.

This iterative process continues until the motion parameters are accurate enough according to a predefined criterion. Normally, a few iterations are sufficient.

It is particularly noted that after the estimation of the 2D affine motion model, we also obtained a weight value w associated with each pixel to indicate its degree of conformance with the motion described by the model. As compared with previous approach⁹ we have introduced the following improvement to use this feature:

- The *weight* value w associated with each pixel is compared against a threshold T_w (set at 0.2 in our experiments), all the pixels whose weights above T_w are termed as *supporting pixels* in compliance with the dominant motion. We add together all the *weights* of these supporting pixels, instead of simply the number of pixels, and define a variable called the *effective supporting factor* ζ_i after appropriate normalization. The normalization is done against the number of *valid* pixels in the reference frame whose mapped counterparts lie inside of the current frame. It is noted that ζ_i changes significantly during shot transition and it varies consistently wherever the visual content of the scene undergoes certain changes. These facts will be exploited fully in the following sections.

3.2 Shot change detection

We employ a statistical hypothesis test – the Hinkley test²⁰ – to detect the significant changes of ζ_i , or the downward / upward jumps that may indicate the shot boundaries. We have introduced the following improvements relative to the way the Hinkley test is carried out in Ref. 9.

- We use *dynamic tolerance* (δ) and *threshold* (α) values instead of the fixed absolute values to detect downward and upward jumps. This has proved to be more effective than fixed ones in detecting transitional scenes (dissolves, various wipes) where the build up of changes is not so prominent.

The tolerance range is set at $\delta = 5\%$ around the mean value of ζ_i . Then, the accumulations of the difference are tested, respectively, in two cases when ζ_i is going above the upper tolerance and below the lower tolerance range.

For *upward jump* detection, we accumulate the difference when ζ_i is above the upper tolerance:

$$\begin{aligned} U_0 &= 0 \\ U &= U_k + [\zeta_{k+1} - m_0 \cdot (1 + \delta)], \quad (k \geq 0) \\ U_{k+1} &= \begin{cases} U & \text{if } U > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

where m_0 is the mean value of ζ_i during down move status.

For *downward jump* detection, we accumulate the difference when ζ_i is below the lower tolerance:

$$\begin{aligned} D_0 &= 0 \\ D &= D_k + [m_1 \cdot (1 - \delta) - \zeta_{k+1}], \quad (k \geq 0) \\ D_{k+1} &= \begin{cases} D & \text{if } D > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (6)$$

where m_1 is the mean value of ζ_i during up move status.

The accumulation U_k or D_k is compared with a threshold α , and the decision is made as follows:

An upward jump is detected if $U_k > m_0 \cdot \alpha$,

A downward jump is detected if $D_k > m_1 \cdot \alpha$

We have set the threshold to be 10% of the mean value of ζ_t i.e., $\alpha=0.1$.

Figures 2-5 describe two representative scene changes for dissolve and zoom wipe and their respective trends of six testing parameters used in the Hinkley test for shot change detection.

Figure 2. An example of the dissolve visual effects during a scene transition, selected seven sample frames from 3965 to 3995 of a test video sequence.

Figure 3. An example of the Hinkley test for the dissolve scene change. The top two curves represent, respectively, the effective supporting factor ζ_t and percentage tolerance range δ . The four curves in the bottom part overlap at value zero at different places, representing, respectively, the changes in $D_k, m_1\alpha$ and $U_k, m_0\alpha$ for downward and upward jump detection (Section 3.2).

Figure 4. An example of the zoom wipe visual effects during a scene transition, selected eight sample frames from 680 to 730 of a test video sequence.

Figure 5. An example of the Hinkley test for the zoom wipe scene change. The top two curves represent, respectively, the effective supporting factor ζ_t and percentage tolerance range δ . The four curves in the bottom part overlap at value zero at different places, representing, respectively, the changes in $D_k, m_1\alpha$ and $U_k, m_0\alpha$ for downward and upward jump detection (Section 3.2).

3.3 Shot change validation - the sanity checks

In order to ensure that the gradual shot changes just detected are valid, reducing especially various false positive alarms, we have introduced three types of sanity checks. These are carried out on a shot by shot basis, if necessary, with negligible computation cost.

- **Limit the shot changing period**

Normally, the number of frames over which a gradual shot change has taken place is limited. This means that after a detected downward jump in ζ_t , there should soon be followed by an upward jump.

If no upward jump can be detected within a pre-set number of frames, T_g say, then either the detected downward jump is a false alarm, or an upward jump has been missed out. In this case, we reduce the *threshold* of the Hinkley test to $\alpha=0$, and rerun the upward jump detection procedure. If still there is no upward jump being detected within the period T_g , the downward jump is regarded as a false alarm and discarded.

- **Merge the shots separated by false shot change detection**

After an upward gradual shot change has been detected, we check the validity of the shot change by considering the two frames from across the shot boundaries, using the last frame from previous shot as reference frame and the current (starting) frame from the current shot. The dominant motion θ_g between them is then computed, together with the effective supporting factor ζ_t and the average intensity I_a over the supporting area.

If the shot change has been detected correctly, the scenes of the two separated shots are likely to be very different, *i.e.* the two frames are not correlated, thus the value of ζ_t should be very small. We can compare ζ_t with a threshold ε to make the following decision:

If $\zeta_t > \varepsilon$, the detected shot change is a false alarm, reject it. Otherwise, the detected shot change is true, keep it.

However, by observation in our experiments, we found that the *luminance* values of pixels in the two frames affect the outcome significantly. The value of ζ_t for true shot change is likely to be very big if the frames are dim (or I_a is small), and relatively small if the frames are bright (or I_a is big). An explanation of the phenomenon is that there tends to be more false matches of spatial distribution patterns between the two frames if they both are dark, although the contents may be totally different. So the new idea is to take both the ζ_t and I_a into account simultaneously. We define a visual-content consistency measure as $V_s = \zeta_t \cdot I_a$, then a decision is made,

If $V_s > \nu$, the detected shot change (both the upward and downward jumps) is a false alarm, reject it. Otherwise, the detected shot change is true, keep it.

The threshold ν is set at 25 in our experiments.

- **Reject the short shot falsely separated during shot transition period**

If an identified shot before a detected gradual shot change is very short (shorter than a pre-set number of frames), it may be garbage during the shot transition period. We calculate ζ_t and I_a from selected two frames in the same way as above, but they serve the opposite purpose:

If $V_s > \nu$, the detected shot is garbage, reject it. Otherwise, it is true, keep it.

The threshold ν is set at 17 on this occasion in our experiments.

These measures have proved to be very useful in our experiments across various video sequences of different media types. The shot change detection has been more accurate and robust, and the false positive alarms have been largely reduced as will be shown later in the experiments Section 5.

4. KEY FRAME EXTRACTION

Along with the shot boundary detection we have also obtained results for qualitative camera motion characterization from the computed dominant motion. This is done by employing a hypothesis test regarding the significance of individual or combination of some of the six affine motion parameters $(a_1, a_2, a_3, a_4, a_5, a_6)$. This result is a qualitative description for each shot detected with one or more camera motion types such as

Static,

Pan to (left, right, top, bottom, top-left, top-right, bottom-left, and bottom-right),

Zoom in /out,

Rotate clockwise / anti-clockwise, and

Complex (none of the above).

As previously shown in Figure 1 that our key frame extraction has been carried out on a shot by shot basis. Within each shot one or more representative frames are determined based on a combination of two factors: the camera motion types described above, and the visual content variation $V_c(i, i+1)$ between consecutive frames i and $i+1$, as defined below:

$$V_c(i, i+1) = \sum_{k=1}^{N_b} (|H_k^Y(i) - H_k^Y(i+1)| + |H_k^U(i) - H_k^U(i+1)| + |H_k^V(i) - H_k^V(i+1)|) \quad (7)$$

where we have used the difference in color histograms in the YUV color space to measure the visual content changes between successive frames. Note that the color histogram in this case is computed *within the effective supporting area* only rather than the usual entire image frame. This has effectively discounted the adverse influence of small moving objects and other disturbances in the scene, which are normally not as relevant for key frames extraction as camera motion which constantly introduces new visual materials.

So we accumulate $V_c(i, i+1)$ from the beginning of each shot as $A_c(i)$. If a particular shot lasting n frames is associated with the *zoom in/out* type of camera motion, a decision is made to use the first and last frame of the shot as key frames, and a third one where $A_c(i) = 0.5 \cdot A_c(n-1)$ if $A_c(i)$ is above a threshold. For a static camera shot, the middle frame of the shot is chosen as the key frame. For other types of camera shots, the first frame as well as the frames where $A_c(i)$ is above a threshold are chosen as key frames.

Up to now, we have all the useful low-level information regarding a video program available, including shots, camera motion types, visual content variation measure etc, it would not be a difficult task to control the total number of key frames representations if one wishes to do so. A method similar to the strategy proposed by Hanjalic et al ⁶ can be employed.

5. EXPERIMENTAL RESULTS

Experiments on the developed system have been carried out using video sequences from a variety type of visual media. Especially, a culture documentary video sequence originally supplied by Lancaster University²¹ has been under detailed analysis. We illustrate the system operation and present some of the shot change detection results on this integrated test video.

The video clip lasts for about 4 minutes. Original MPEG-1 compressed sequence was first decoded off-line. In the analysis stage, the system operates on the raw YUV sequence, with functional parameters (requirements, thresholds) choosing from a selection menu (shown in Figure 6 right) or by default. A number of intermediate results can be recorded for evaluation purpose as well as the desired elementary indexing information. Another optional window (Figure 6 left) displays the progress information - the two consecutive frames, the supporting area of dominant motion, the displaced frame difference. There are total 56 shots in the video segments tested. We have manually obtained this together with the types of shot changes and precise boundaries (the Ground Truth).

Table 1 shows the shot change detected together with the ground truth. There are 17 cuts, 33 dissolve changes of various length from as short as 15 frames to as long as 55 frames. There is also one zoom wipe and 4 other special editing effects. We can see that the percentage of the shot changes *missed* is very low, despite that the scenes encountered are so diverse and the same set of parameters or thresholds have been used for the whole video clip.

Table 2 shows the *false alarms* occurred in shot change detection. These are places where no real scene content updates have taken place rather the large imaging properties varied due to, for instance, the camera flash, abrupt motion activities, floating synthetic objects etc. Nevertheless, we can see that the introduction of *sanity check* reduces the false alarm rate

dramatically. Correspondingly, Table 3 shows the influence of the false alarms on the actual shot separations. The system ends up detecting 72 shots, and this is reduced to 61 shots with sanity check. Against the ground truth of 56 shots, the system detects automatically over 80% shots correctly. This demonstrates further the success of the Sanity check procedure.

Figure 6. A snapshot of the system operation in the analysis stage: left) display window viewing the execution progress; right) Operational parameters selection menu.

Table 1. Shot change detected for the Lancaster culture documentary

Type of shot changes	Ground truth	Detected		Missed	
		Number	%	Number	%
Cut	17	17	100%	0	0%
Dissolve	33	30	91%	3	9%
Zoom wipe	1	1		0	
Others	4	4		0	
Total	55	52	95%	3	5%

Table 2 . False alarms occurred in shot change detection for the Lancaster culture documentary

Shot change	Detected	True	False	False alarm
Without sanity check	71	52	19	27%
With sanity check	60	52	8	13%

Table 3. Shots separated for the Lancaster culture documentary

Shot	Separated	True		False	
		Number	%	Number	%
Without sanity check	72	38	53%	34	47%
With sanity check	61	45	74%	16	26%
Against Ground Truth	56	80.4%		28.6%	

6. DISCUSSION

Referring to the system diagram described in Figure 1, the elementary indexing information thus obtained in the analysis stage is now fed into a video browsing, summarization and semantics editing user interface to demonstrate the system functions. Figure 7 shows a screen dump of the user interface in action for a test video sequence. It facilitates fast browsing of the video content through a succession of key frames representations. The main functions include the playback facility to display a chosen shot or several shots represented by key frames, together with its camera motion types description, shot boundaries, and shot length (Figure 7 right). It can interactively merge adjacent shots into a semantic event or episode and re-compute appropriate features or key frame descriptions, it can also modify a detected shot boundary and split a shot into two if any errors occurred in the automatic analysis stage.

Our future work includes the enhancement of the functionality and computational efficiency of the current system and the investigation into certain high-level semantics analysis issues in specific application domains. The combination of motion-induced dynamical visual content features studied so far and various spatially distributed or object features will be related to particular semantic events and requirements. The tracking and indexing of dynamic events in the visual scene is also a natural follow up, given that the dominant motion descriptor is now available and can thus be easily incorporated to detect moving objects in the scene.

Figure 7. The interactive graphical user interface for video summarization by key frames, shots editing and semantic events grouping.

REFERENCES

1. B. Furht, S.W. Smoliar, and H.J. Zhang, *Video and Image Processing in Multimedia Systems*, Kluwer Academic Publishers, Boston, 1996.
2. A. D. Bimbo, *Visual Information Retrieval*, Morgan Kaufmann Publishers, San Francisco, CA, 1999.
3. EU SMASH project Web site, <http://www-it.et.tudelft.nl/pda/smash>.

4. M.R. Naphade and T.S. Huang, "Multimedia understanding: challenges in the new millennium," , " *Proc. of IEEE International Conference on Image Processing*, Vancouver, September 2000.
5. M.-M. Yeung and B.-L. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content," *IEEE Trans on Circuits and Systems for Video Technology*, vol. 7, no. 5, pp. 771-785, October 1997.
6. A. Hanjalic, R.L. Lagendijk, J. Biemond, "A new method for key frame based video content representation," in *Image Databases and Multimedia Search*, Eds. A.W.M. Smeulders, R. Jain, World Scientific Singapore, 1977.
7. F. Dufaux, "Key frame selection to represent a video," *Proc. of IEEE International Conference on Image Processing*, Vancouver, September 2000.
8. MPEG-7 working documents available at URL: http://www.cselt.it/mpeg/public/mpeg-7_visual_xm.zip. October 2000.
9. P. Bouthemy, M. Gelgon, and F. Ganansia, "A unified approach to shot change detection and camera motion characterization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1030-1044, October 1999.
10. B. Adams, C. Dorai, and S. Venkatesh, "Novel approach to determining tempo and dramatic story sections in motion pictures," *Proc. of IEEE International Conference on Image Processing*, Vancouver, September 2000.
11. A. Smolic, M. Hoeynck, and J.-R. Ohm, "Low-complexity global motion estimation from P-frame motion vectors for MPEG-7 applications," *Proc. of IEEE International Conference on Image Processing*, Vancouver, September 2000.
12. J.D. Courtney, "Automatic video indexing via object motion analysis," *Pattern Recognition*, vol. 30, no.4, pp. 607-625, April 1997.
13. A. Ekin, R. Mehrotra, and A.M. Tekalp, "Parametric description of object motion using EMUs," *Proc. of IEEE International Conference on Image Processing*, Vancouver, September 2000.
14. A. Smolic, T. Sikora, and J.-R. Ohm, "Long-term global motion estimation and its application for sprite coding, content description, and segmentation," *IEEE Trans on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1227-42, December 1999.
15. J.R. Kender and B.L. Yeo, "Video scene segmentation via continuous video coherence," *Proc. of International Conference on Computer Vision and Pattern Recognition (CVPR'98)*, Santa Barbara, pp. 367-373, June 1998.
16. L. Vincent, "Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms," *IEEE Trans. On Image Processing*, vol. 2, pp. 176-201, 1993.
17. A.M. Tekalp, *Digital Video Processing*, Prentice Hall, NJ, 1995.
18. F. Moschetti, M. Kunt, and F. Calvano, "A nested multilevel redundancy exploitation for fast block matching," *Proc. of IEEE International Conference on Image Processing*, Vancouver, September 2000.
19. J.M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348-365, December 1995.
20. M. Basseville, "Detecting changes in signals and systems - A survey," *Automatica*, vol. 24, no.3, pp. 309-326, 1988.
21. E. Hartley, "MPEG7 content test sequence," Culture Documentary, Produced by Lancaster University Television, March 1998.