A Moving Object Segmentation Technique Using Dynamic Programming

Zhengping Wu, Chun Chen
Dept. Computer Science, College of Information Science and Engineering, Zhejiang University
Hangzhou, 310027, P. R. China
zhengpingw@telekbird.com.cm, chenc@cs.zju.edu.cn

Abstract

This paper presents a technique for automatic moving object segmentation based on dynamic programming. This technique consists of three phases: motion region identification, motion edge extraction and contour linkage. New algorithms for contour linkage are proposed for the present technique. The algorithms can track the object's motion edges in the motion regions as well as detect the existing edges of the still object in the scene. Its computational efficiency is realized by the use of localized edge extraction in motion region and the state independence of dynamic programming. Simulation results demonstrate that the proposed technique can efficiently segment video streams with good visual effect as well as spatial accuracy and temporal coherency in real time.

Keywords: Moving object segmentation, Dynamic programming, Motion edge, Contour linkage

1. Introduction

Moving object segmentation aims to partition an image sequence into moving objects and to track the evolution of the moving objects along the time axis. Many applications related to video compression and transmission, and pattern recognition rely on moving object segmentation. Moving object segmentation techniques are also important tools for content-based video coding and manipulation, and interactive multimedia applications. Moving object segmentation usually divides the contents of a video frame into semantic regions that can be dealt as objects. These semantically segmented objects can be coded so that object-based manipulation of video content can be realized in interactive multimedia applications. For example, in the context of the MPEG-4 [6,7] standard, a video is considered to consist of independently moving objects and is encoded object by object. In the MPEG-7 [9,10], segmented results based on the frame-to-frame motion information or abrupt shape change can be utilized for a high-level description.

This paper describes a technique for automatic moving object segmentation based on dynamic programming. This technique consists of three phases: motion region identification, motion edge extraction and contour linkage. The flow diagram of this technique is shown below.

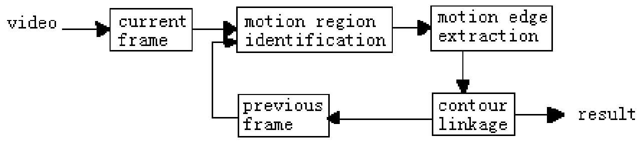


Fig. 1 Flow diagram of the proposed technique

2. Related works

A number of techniques and algorithms have been proposed for moving object segmentation in the past, each of which has its particular features and applications.

Arch and Kaup proposed a moving object segmentation technique using a statistical approach in [3,4]. They model the characteristics of pixel difference for background between two consecutive frames as a gaussian distribution. A change detection mask (CDM) is yielded by thresholding the frame difference image. The CDM is a binary image in which pixel differences exceeding the threshold value are declared as being intensity-changed or, otherwise, as being intensity-unchanged. Since this technique relies on frame intensity difference, unsatisfactory segmentation results are obtained in slow object motion or lack of object motion.

Meier and Ngan proposed an automatic segmentation technique for moving objects using a binary image model to track a moving object [5]. The binary model is derived from an edge image and is updated every frame to keep the changes in location and shape. The detection of a moving object is made on the basis of binary model matching between two consecutive frames using Hausdoff distance. However, in order to obtain reliable segmentation results, the initial segmentation result should be precise enough to localize the boundary of a moving object so that the object boundary is well identified in the subsequent frames. The advantage of this technique lies in its capable of tracking an object that stops moving for a certain period of time.

However, the segmentation results depend on the success of the initial segmentation at the first frame.

In the structure of our technique, the frame difference and the edge image are combined to extract the motion edge. Then the contour linkage algorithm is used to form a complete contour. The key feature of the proposed moving object segmentation technique is the dynamic-programming-based contour linkage algorithm. The algorithm allows the proposed technique to track the object's motion edges in the motion regions as well as to detect the existing edges of the still object in the scene. Another feature of the proposed technique is its computational efficiency resulting from the use of localized edge extraction in motion region and the state independence of dynamic programming. That is to say that the optimal decision for each of the remaining states derived from current state does not depend on the previous states or decisions.

This paper is organized as follows. In section 3, the video object segmentation technique is described. Promising simulation results are reported in Section 4. Conclusion and future work are followed in Section 5.

3. Moving object segmentation based on dynamic programming

3.1. Motion region identification

In video streams, motion region can be identified by the difference between consecutive frames, which is calculated by comparing each pixel in one frame and its correspondent in the other. Given that the luminance in video stream is not very variable, the difference of the consecutive frames indicates the object's changes of location and shape. In practice, because of the infection of random noises, the pixel pair in region without motion also has a non-zero value. To distinguish the noises and real motion, a threshold has to be introduced. That is to say, if the difference is greater than the given threshold, we consider that the difference is caused by real motion. Let $f(x,y,t_i)$ and $f(x,y,t_j)$ be two frames captured at time t_i and t_j, the difference can be expressed as:

$$d_{ij}(x,y) = \begin{cases} 1 & if \ | \ f(x,y,t_i) - f(x,y,t_j) > T \ | \\ 0 & else \end{cases} \qquad (1)$$

T denotes the threshold, 0 denotes the pixel without real motion, and 1 denotes the pixel with real motion. If these two frames are consecutive frames, then j = i –1.

After identifying the motion region by frame difference, we need to remove all the irrelevant points and blocks. A region-labeling algorithm is exploited to finish this task [1,2]. After the scan of the binary image denoting motion region, all connected blocks are labeled, and the algorithm eliminates the blocks in which the number of points is smaller than a threshold.

The frame difference of consecutive frames identifies the motion region efficiently. The motion region along with its neighbor region always covers up the accurate edge of moving object (motion edge). So the dilate operation has to be performed on the motion region after identification.

3.2. Motion edge extraction

After calculating motion region by difference of frames, we will extract the motion edge of the current frame fn according to the Laplacian filter.

The Laplacian is a scalar second-derivative operator for functions of two dimensions. It is defined as:

$$\nabla^2 f(x,y) = \frac{\partial^2}{\partial x^2} f(x,y) + \frac{\partial^2}{\partial y^2} f(x,y) \qquad (2)$$

Since it is a second derivative, the Laplacian will produce an abrupt zero-crossing at an edge. Compared with the wide range of large gradient value resulted from the first derivatives, the abrupt zero-crossing of the Laplacian can be used to locate the precise edge. The Laplacian is a linear, shift-invariant operator, and its transfer function is zero at the origin of frenquency space. Thus, a Laplacian-filtered image will have zero mean gray level.

If a noise-free image has sharp edges, the Laplacian can find them. The binary image resulted from thresholding a Laplacian-filtered image at zero gray level will produce closed, connected contours when interior points are eliminated. The presence of noise, however, imposes a requirement for lowpass filtering before the use of the Laplacian.

A Gaussian lowpass filter is a good choice for this pre-smoothing. Since convolution is associative, we can combine the Laplacian and Gaussian impulse responses into a single Laplacian of Gaussian kernel:

$$-\nabla^2 \frac{1}{2ps^2} e^{-\frac{x^2+y^2}{2s^2}} = \frac{1}{ps^4}[1 - \frac{x^2+y^2}{2s^2}]e^{-\frac{x^2+y^2}{2s^2}} \qquad (3)$$

This impulse response is separable in x and y and thus can be implemented efficiently. The parameter s controls the amount of smoothing.

Thus, we can get rid of the noises and extract the accurate motion edge through the abrupt zero-crossing in the motion region.

3.3. Dynamic-programming-based contour linkage

3.3.1. Important theorems of dynamic programming

The deterministic decision-making process consists of five parts: stage, decision, stage transform rule, cost and goal, five parts. The basic model for the definition of deterministic decision-making process (deterministic dynamic programming) is described below.

Definition: the deterministic decision-making process (deterministic dynamic programming) is a quintuple {X，U，T_k（x_k，u_k），v_k（x_k，u_k），V_{kN}（x_k，p_{kN}）}（{X,U,T,v,V} in brief).

1) X is the stage space. The k-th stage set is X_k, its element is x_k, viz $x_k \in X_k$.

2) U is the decision space. The k-th permissible decision set is $D_k(x_k)$, its element is $u_k(x_k)$, viz $u_k(x_k) \in D_k(x_k) \subset U$.

The permissible decision from x_0 is $p_{0N}(x_0)=\{u_0(x_0),u_1(x_1),\ldots,u_{N-1}(x_{N-1})\}$, and its permissible decision set is P(x), viz $p_{0n}(x_0) \in P_{0N}(x_0)$.

The permissible decision of k-N sub-process is $p_{kN}(x_k)=\{u_k(x_k),\ldots,u_{N-1}(x_{N-1})\}$, and its permissible decision set is $P_{kN}(x_k)$, viz $p_{kN}(x_k) \in P_{kN}(x_k)$.

3) T is the stage transform rule: $x_{k+1}=T_k(x_k,u_k)$.

4) $V_k(x_k,u_k)$ is the effect (cost) of decision u_k that is applied on the k-th stage x_k, called cost function. It is defined on the $X \times D$ space.

5) $V_{0N}(x_0,p_{0N}(x_0))$ is the whole effect (whole cost) of the process when taken decision p from x_0, called goal function. It is a quantitative function defined on the whole process (including its all sub-process). The goal function can be expressed in the form below, and it is strictly monotone.

$V_{0N}(x_0,p_{0N}(x_0))=V_{0N}(x_0,x_1,\ldots,x_N)$
$=V_{0N}(x_0,u_0,u_1,\ldots,u_{N-1})$

$V_{kN}(x_k,p_{kN}(x_k))=U_k(x_k,u_k,V_{k+1}(*x_{k+1},p_{k+1}(*x_{k+1})))$

Function U should be strictly monotone with respect to variable $V_{k+1,N}$.

The goal function is the sum of the costs in all stages:

$$V_{oN}(x_0, p_{oN}(x_0)) = \sum_{j=0}^{N-1} v_j(x_j, u_j).$$

From the above conditions and rules, we can get the optimum theorem in dynamic programming, which serves as the theoretical base of dynamic programming.

Theorem 3.3.1.1:

If $V_{oN}(x_0, p_{oN}(x_0)) = \sum_{j=0}^{N-1} v_j(x_j, u_j)$, then the permissible decision p_{oN}^* is the optimal decision of {X,U,T,v,V} if and only if: for every k, when 0<k<N and $x_0 \in X_0$,

$$V_{oN}(x_0, p_{0N}^*(x_0)) = \underset{p_{0k}(x_0)}{opt}\{V_{0k}(x_0, p_{0k}(x_0)) + \underset{p_{kN}(x_k^*)}{opt}[V_{kN}(x_k^*, p_{kN}(x_k^*))]\}.$$

Theorem 3.3.1.2 (optimum theorem):

If $V_{0N}(x_0, p_{0N}(x_0)) = \sum_{j=0}^{N-1} v_j(x_j, u_j)$, and permissible decision $p_{0N}^*(x_0)$ is the optimal decision of {X,U,T,v,V}, then for every k: when 0<k<N, and $x_k^* = T_{k-1}(x_{k-1}^*, u_{k-1}^*)$, the sub-decision $p_{kN}^*(x_k^*)$ from x_k^* derived from its k—N sub-process must be the optimal decision.

Theorem 3.3.1.3:

If $V_{oN}(x_0, p_{oN}(x_0)) = \sum_{j=0}^{N-1} v_j(x_j, u_j)$, then $\{f_k(x_{k0})\}$ and $\{u_k^*(x_k)\}$ are the optimal goal function and the optimal decision sequence if and only if they meet the following requirements:

$$\begin{cases} f_k(x_k) = \min\limits_{u_k(x_k)}\{v_k(x_k, u_k) + f_{k+1}[T_k(x_k, u_k)]\} \\ \qquad = v_k(x_k, u_k^*) + f_{k+1}(T_k(x_k, u_k^*)) \\ k = n-1, n-2, \cdots, 0 \\ f_n(x_n) = 0 \end{cases}$$

3．3．2 Contour linkage in motion region using dynamic programming

After extracting accurate edges by Laplacian edge detector, we have to link the edges to form continuous contour in motion region. Firstly, the image smoothed by Gaussian smoothing filter will have some blur boundaries. So there will be some edges extracted by Laplacian edge detector have widths greater than one pixel. We use the thinning algorithm [3] to thin these edges. Secondly, we have to find out the start point and the end point of the contour segment contained in the motion region. The width and height of the motion region's circum-rectangle are width and height respectively. If width is great or equal to height, then the edge points on the left and right limit serve as the start and end point respectively (if such points are more than one, choose the upper-left and bottom-right ones). Otherwise, the edge points on the upper and bottom limit serve as the start and end point (if such points are more than one, choose the leftmost and rightmost ones).

The contour of the moving object must be continuous, so we use the equation below to calculate the cost of path from one edge point to another. Di,j denotes the distance between edge point i and edge point j.

$$D_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (4)$$

Wi,j denotes the cost of the path from edge point i to edge point j.

Di,j	1	2	3	4	5	6	7	8	>8
Wi,j	0	2	8	10	200	400	800	1200	2000

Table 1 Cost of continuity

Because of the continuity of the contour, Wi,j is set to 200 when the distance is 5.

At the same time, the contour segment in a motion region is unique. Thus, the direction of the contour segment starting from the start point should be toward the end point. We use the equation below to calculate the cost of each candidate point's direction, in which Li denotes the distance between the current candidate point (x_i, y_i) and the end point (x_l, y_l).

$$L_i = \sqrt{(x_i - x_l)^2 + (y_i - y_l)^2} \quad (5)$$

Ci denotes the ratio of Li and the minimum distance between the scanning line containing x_i and the end point, and Ui denotes the direction cost of point i.

Ci	1	1.1	1.2	1.3	1.4	1.5	>1.5
Ui	0	2	4	6	8	200	1000

Table 2 Cost of direction

Because of the uniqueness of the contour segment's direction, Ui is set to 200 when the ratio is 1.5.

From the cost equations above, we can define the cost of the whole contour segment below.

$$V = \sum_{i=1}^{l-1} W_{x_{i+1},x_i} + \sum_{i=1}^{l-1} U_{x_i} \quad (6)$$

Xi is the candidate point in the i-th scanning line.

According to the basic model described in 3.3.1, the optimal goal function is:

$$f_k(x_k) = \underset{(u_k,\cdots,u_{n-1})}{opt} V_{kn}(x_k, u_k, u_{k+1}, \cdots, u_{n-1}) = \underset{p_{kn}(x_k)\in P_{kn}(x_k)}{opt} V_{kn}(x_k, p_{kn}(x_k)) \quad (7)$$

K denotes stage (viz the k-th scanning line), k=0,1,..., n-1; x_k denotes the stage variable in stage k. If given the state of a certain stage in the process, the process after that stage is not influenced by the states of previous stages, only influenced by the current stage's state. This feature is called state independence, and it is an important feature in dynamic programming. $u_k(x_k)$ denotes the decision taken at state x_k in stage k; $p_{kN}(x_k)$ is the permissible decision in k-N sub-process, in which $p_{kN}(x_k)=\{u_k(x_k),\ldots,u_{N-1}(x_{N-1})\}$; $v_k(x_k,u_k)$ is the effect (cost) of decision u taken at state x in stage k; $f_k(x_k)$ is an performance indicator of the whole process, which is a deterministic quantitative function defined on the whole process (the goal function).

According to the deduction of theorem 3.3.1.3, we get

$$f_k(x_k) = \min_{u_k(x_k)}\{v_k(x_k,u_k) + f_{k+1}[T_k(x_k,u_k)]\}$$

. $T(x_k,u_k)$ denotes the stage transform rule from stage k to stage k+1.

At last, the mathematical model of the contour linkage problem in motion region can be expressed as:

$$\begin{cases} f_k(x_k) = \min\limits_{u_k(x_k)}\{v_k(x_k,u_k) + f_{k+1}[T_k(x_k,u_k)]\} \\ v_k(x_k,u_k) = D_{x_k,u_k(x_k)} + C_{x_k} \\ k = n-1, n-2, \cdots, 0 \\ f_n(x_n) = 0 \end{cases} \quad (8)$$

In the process of seeking the optimal goal function of this mathematical model, we can get all the contour points in motion region by going through the stages. Because no guarantee of the continuity of the motion edge is confirmed before, there will be some short lines inserted between the motion edges by the seeking process, and the resulted contour segment is continuous in the motion region.

3．3．3 Contour linkage outside motion region using dynamic programming

Because the moving object is not always a rigid-object, and the motion is occurred only on part of the object, the contour extracted from motion region is not always complete. That is to say, the contour is not continuous outside the motion region. So we have to take advantage of the image information to find out the contour segments between the end points of the contour in the motion region. These contour segments are between pixel-pairs, that consist of one end point in a contour segment extracted from one motion region and the most proximate end point in the contour segment extracted from another motion region. Considering the difference in color and brightness between edge points and other points, we use the equation below to signify this difference.

$$t_i = \left|[(r_i - r_{i-1}) + (g_i - g_{i-1}) + (b_i - b_{i-1})]\right| / 3 \quad (9)$$

The subscript i denotes the i-th point in a scanning line.

To make the cost of each stage comparable, we define the cost by the equations below.

$$T_{\max} = \max_{1\le i\le width}(t_i) \quad (10)$$

$$C_i = 1 - (t_i / T_{\max} \times 100\%) \quad (11)$$

width is the length of the scanning line. It is obvious that the smaller the Ci, the more likely is the point an edge point.

We know that the contour of the moving object must be continuous, so we use the equation below to calculate the cost of path from one edge point to another. Di,j denotes the distance between edge point i and edge point j.

$$D_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (12)$$

Wi,j denotes the cost of the path from edge point i to edge point j.

Di,j	1	2	3	4	5	6	7	8	>8
Wi,j	0	2	8	10	200	400	800	1200	2000

Table 3 Cost of continuity

Because of the continuity of the contour, Wi,j is set to 200 when the distance is 5.

From the cost equations above, we can define the cost of the whole path:

$$V = \sum_{i=1}^{n} C_{x_i} + \sum_{i=2}^{n} W_{x_i, x_{i-1}} \quad (13)$$

Xi is the candidate point in the i-th scanning line.

According to the deduction of theorem 3.3.1.3, the optimal goal function is also:

$$f_k(x_k) = \min_{u_k(x_k)} \{ v_k(x_k, u_k) + f_{k+1}[T_k(x_k, u_k)] \} \quad (14)$$

K denotes stage (viz the k-th scanning line), k=0,1,..., n-1; x_k denotes the stage variable in stage k; $u_k(x_k)$ denotes the decision taken at state x_k in stage k; $v_k(x_k,u_k)$ is the cost of decision u taken at state x in stage k; $f_k(x_k)$ is the performance indicator of the whole process; $T(x_k,u_k)$ denotes the stage transform rule from stage k to stage k+1. The final mathematical model of the contour linkage problem outside motion region is:

$$\begin{cases} f_k(x_k) = \min_{u_k(x_k)} \{ v_k(x_k, u_k) + f_{k+1}[T_k(x_k, u_k)] \} & (15) \\ v_k(x_k, u_k) = C_{x_k} + D_{x_k, u_k(x_k)} \\ k = n-1, n-2, \cdots, 0 \\ f_n(x_n) = 0 \end{cases}$$

In the process of getting the optimal goal function of this mathematical model, we obtain all the contour points outside the motion region by going through the stages. Thus, we get a complete contour of the moving object.

4. Experiments

Currently, there is no commonly accepted measure to evaluate the system performance for automatic segmentation. The spatial accuracy and temporal smoothness are two important measures of the quality of segmentation results. The spatial accuracy of an estimated binary video object mask at frame t is defined as [8]:

$$A_t(I_t^{ref}, I_t^{est}) = 1 - \frac{\sum_{(x,y)} I_t^{ref}(x,y) \oplus I_t^{est}(x,y)}{\sum_{(x,y)} I_t^{ref}(x,y)} \quad (16),$$

where I_t^{ref} and I_t^{est} are the reference and the estimated binary object masks at frame t, respectively, and $\oplus$ is the binary "XOR" operation. The temporal coherency $C(t)$ is defined as:

$$C(t) = A_t(I_t, I_{t-1}) = 1 - \frac{\sum_{(x,y)} I_t(x,y) \oplus I_{t-1}(x,y)}{\sum_{(x,y)} I_t(x,y)} \quad (17),$$

where I_t and I_{t-1} are binary object masks at frame t and t-1, respectively. Temporal coherency $C^{est}(t)$ of the estimated binary mask I_t^{est} should be compared to temporal coherency $C^{ref}(t)$ of the reference mask I_t^{ref}. Any significant deviation from the reference indicates a bad temporal coherency.

The proposed technique is applied to "Hall Monitor" sequence, and compares the result with those of the techniques described in [3] and [5]. Fig .2 shows the spatial accuracy, in which the reference binary object mask is obtained manually. Fig. 3 shows the temporal coherency, and fig.4 shows the segmented results.

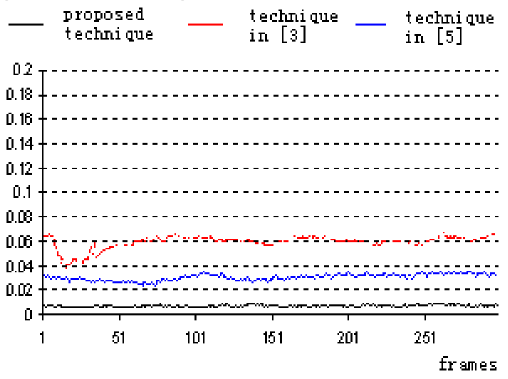


Fig.2 Spatial accuracy

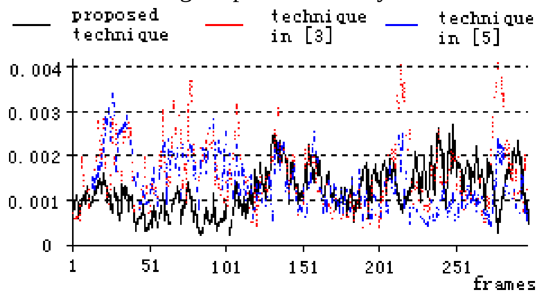
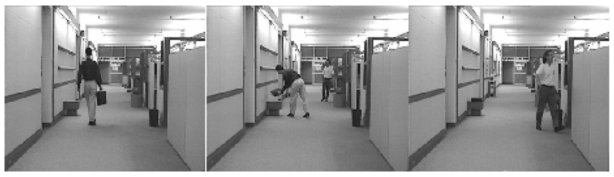
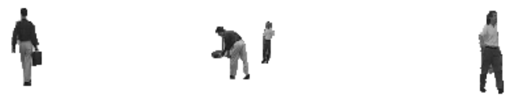


Fig. 3 Temporal coherency



(a) (b) (c)



(d) (e) (f)

Fig. 4 Original frames and segmented frames

We see from these results that our dynamic-programming-based segmentation technique gives good visual effect as well as spatial accuracy and temporal coherency.

The proposed technique is implemented on a 500MHz Pentium III PC, and the system also runs on video streams with a frame size of 320*240 pixels captured by a USB video camera. The process of capturing, moving object segmentation and result showing can reach the speed of 15Hz, which meets the requirement of real-time process.

5. Conclusion and future work

Despite the fact that the human visual system can easily distinguish moving objects, automatic moving object segmentation is known to be one of the most challenging issues in the field of video processing. An automatic segmentation technique of moving object is presented which incorporates motion edge extraction with contour linkage based on dynamic programming. The motion edge extraction allows this technique to utilize the motion information, thus provide more reliability than do image segmentation techniques. We separate the contour linkage problem into two modules. For the contour segment in motion region a linkage algorithm which emphasizes on the continuity is proposed. For the contour segment outside motion region a linkage algorithm which emphasizes on color, intensity and gradient is used. The proposed technique exhibits better segmentation performance than some existing techniques do.

Two problems still exist. One is that when the motion edge accounts for only a small part of the whole contour, the technique will reduce to an image segmentation technique. The other is that when more than one object appear in the scene, the proposed technique will merge them. So future work will emphasize on incorporation of user interaction to circumvent these ill-posed problems in image understanding and computer vision.

6. References

[1] Intel image processing library, http://developer.intel.com/vtune/perflibst/ipl.

[2] Identifying connected components, http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/OWENS/LECT3/node2.html.

[3] T. Arch, A. Kaup. Statistical model-based change detection in moving video. *Signal Processing*, 1993, 31, pp. 165-180.

[4] T. Arch, A.Kaup, R. Mester. Change detection in image sequences using Gibbs random fields: a Bayesian approach. *Proc. of International Workshop on Intelligent Signal Processing and Communication Systems*, Sendai, Japan, October 1993, pp. 56-61.

[5] T. Meier, K. N. Ngan. Automatic segmentation of movings for video object plane generation. *IEEE Transactions on Circuits and Systems for Video Technology*, 1998, 8(5), pp. 525-538.

[6] ISO/IEC JTC1/SC29/WG11. Overview of the MPEG-4 standard. MPEG98/N2323, Dublin, July 1998.

[7] ISO/IEC JTC1/SC29/WG11. MPEG-4 visual final committee draft. Dublin, July 1998.

[8] M. Wollborn, R. Mech. Refined procedure for objective evaluation of video object generation algorithms. ISO/IEC JTC1/SC29/WG11 M3448, March 1998.

[9] ISO/IEC JTC1/SC29/WG11. MPEG-7 requirements document. N4035, Singapore, March 2001.

[10] ISO/IEC JTC1/SC29/WG11. MPEG-7 applications document. N3934, Pisa, January 2001.