Automatic Key Video Object Plane Selection Using the Shape Information in the MPEG-4 Compressed Domain

Berna Erol and Faouzi Kossentini, *Senior Member, IEEE*

***Abstract*—Object-based video representation, such as the one suggested by the MPEG-4 standard, offers a framework that is better suited for object-based video indexing and retrieval. In such a framework, the concept of a "key frame" is replaced by that of a "key video object plane." In this paper, we propose a method for key video object plane selection using the shape information in the MPEG-4 compressed domain. The shape of the video object (VO) is approximated using the shape coding modes of I, P, and B video object planes (VOPs) without decoding the shape information in the MPEG-4 bit stream. Two popular shape distance measures, the Hamming and Hausdorff distance measures, are modified to measure the similarities between the approximated shapes of the video objects. Although they feature different computational and implementation complexity tradeoffs, the corresponding algorithms achieve essentially the same performance levels in selecting key video object planes that represent efficiently the salient content of the video objects.**

***Index Terms*—Key frame selection, key video object plane selection, object-based video retrieval, video databases, video summarization.**

I. Introduction

STORAGE and coded representation of digital video have been a subject of research for the last two decades, resulting in many efficient video coding algorithms and several video coding standards, such as MPEG-1 [1], MPEG-2 [2], and H.263 [3]. These advancements made it possible to have large digital video databases, such as the ones on the Internet or the ones associated with surveillance applications. In order to access this data, efficient indexing and retrieval of digital video are required. Research in video indexing and retrieval is still in its infancy, and recent research efforts have led to the emergence of the MPEG-7 standard [4] as well as the development of several systems such as VideoQ [5], Jacob [6], and NeTra [7].

In a typical frame-based digital video indexing and retrieval system, key frames are used to represent the salient content of a video sequence. Besides visual summarization, representation using key frames allows some of the still image features (such as shape, texture, and color) to be used for video retrieval. Many algorithms have been proposed for key frame selection. Some of these algorithms are applied to uncompressed video, and they involve comparing color and motion histograms, computing pixel differences, and performing edge tracking [8], [9]. Other algorithms involve operations in the compressed domain (e.g., MPEG-1/2) and take into account the texture coding modes (intra, inter, etc.), the motion vectors, and the significant changes in DC coefficients to detect shot boundaries and to select key frames [10]–[13].

While key frames provide a summary of the video content, they cannot provide an accurate description of the individual objects within a video scene. Access to individual objects in a video sequence is essential for content-based video indexing and retrieval systems that support object-based video queries, such as querying a video object with a given shape and color, and moving in a given direction at a given speed. The most recent MPEG video coding standard, MPEG-4, offers an object-based representation of video, where individual video objects (VOs) are coded into separate bit streams [14]. In the MPEG-4 terminology, temporal instances of video objects are referred to as video object planes (VOPs). Similar to key frames, key VOPs can be used for visual summarization of the video object content in an object-based framework.

Unlike in key frame selection, very little work has been reported on key VOP selection. Gunsel *et al.*proposed that the motion of the video object and its uncompressed shape data be used for temporal segmentation of video objects and key VOP selection [15]. However, such an algorithm is very computationally intensive, often making key VOP selection unpractical. Ferman *et al.*suggested an algorithm that uses the texture coding modes in the MPEG-4 compressed domain to extract the key VOPs [16]. Their proposed algorithm employs the percentage of intra coded macroblocks as a measure for significant change in the content. Although the algorithm is simple, the difficult problem of threshold selection has not been addressed. Moreover, the accuracy of using the percentage of intra coded macroblocks is too low for the effective selection of key VOPs.

In this paper, we propose a key VOP selection method that is based on the shape content of video objects. Significant changes in the shape of video objects are detected by comparing the shape content of the VOPs to each other by using the Hamming and Hausdorff distance measures. The shape of a video object is approximated using the shape coding modes that can be determined directly (i.e., without decoding the shape data) from the MPEG-4 bit stream. The Hamming and Hausdorff distance measures have been modified so that approximations of the video object shapes can be employed. The organization of

Manuscript received June 8, 1999; revised March 7, 2000. This work was supported by the Natural Science and Engineering Research Council of Canada. The associate editor coordinating the review of this paper and approving it for publication was Prof. Yao Wang.

The authors are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, B.C. V6T 1Z4, Canada (e-mail: bernae@ece.ubc.ca; faouzi@ece.ubc.ca).

Publisher Item Identifier S 1520-9210(00)04239-5.

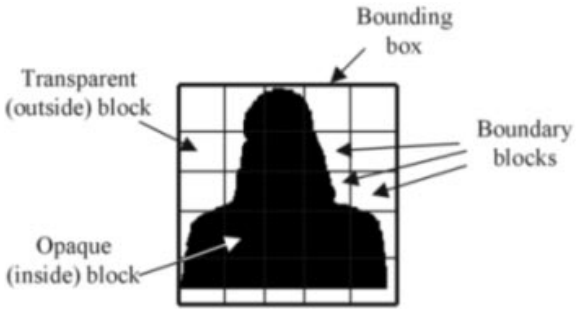


Fig. 1. Shape representation in MPEG-4.

the rest of the paper is as follows. In Section II, we provide an overview of the MPEG-4 shape representation. In Section III, we present the proposed key VOP selection method including the Hamming and Hausdorff distance based algorithms. Experimental results that illustrate the performance of the proposed algorithms, as well as their complexity-performance tradeoffs, and our conclusions, are presented in Section IV and Section V, respectively.

II. Background

The MPEG-4 video coding standard provides an object-based representation of video by allowing the coding of arbitrarily shaped video objects [14], [17]. In the MPEG-4 framework, the texture and the shape of each temporal instance of a video object, i.e., each VOP, are coded separately. Similar to MPEG-1/2, MPEG-4 supports intra coded (I), temporally predicted (P), and bi-directionally predicted (B) VOPs. PVOPs are predicted from the temporally previous I or PVOPs. BVOPs are predicted from the temporally previous and/or future I or PVOPs.

Texture coding of VOPs is very similar to the coding of frames in MPEG-2 [2]: Each VOP is divided (using shape information) into macroblocks, and the luminance and chrominance blocks of each macroblock are coded using DCT, quantization, and variable length coding. Additionally, motion compensation and predictive coding of texture are employed for P and BVOPs.

The shape of a VOP is described by a binary alpha plane, which indicates whether or not a pixel belongs to a VOP. The borders of a binary alpha plane are determined by the VOP bounding box, which is the tightest rectangle around the video object. A binary alpha plane is divided into 16×16 blocks, as illustrated in Fig. 1. The shape data associated with each of these 16×16 blocks are transmitted in the bit stream, along with the texture information that corresponds to the same area. Three shape coding modes are possible for IVOPs: 1) transparent, where all pixels in a 16×16 block fall outside the object; 2) opaque, where all pixels in a 16×16 block are located inside the object; and 3) intra, where the pixels in a 16×16 block are at the boundary of the object. In intra shape coding, the pixels inside the boundary blocks are raster order scanned and the corresponding binary shape data is context-based arithmetic coded [14]. Lossy shape coding is achieved by subsampling of the binary alpha plane by a factor of 2 or 4 prior to arithmetic encoding.

TABLE I
Shape Coding Modes in MPEG-4

Coding mode	Coding type	Used in
0	MVDs=0 & no inter update	P and BVOPs
1	MVDs!=0 & no inter update	P and BVOPs
2	Transparent	I, P and BVOPs
3	Opaque	I, P, and BVOPs
4	Intra coded	I, P, and BVOPs
5	MVDs=0 & inter coded	P and BVOPs
6	MVDs!=0 & inter coded	P and BVOPs

Seven shape coding modes are supported for the P and BVOPs, as presented in Table I. The transparent, opaque, and intra coding modes are the same as in the IVOP case. The four additional inter shape coding modes involve the transmission of motion vectors and additional update (difference) information. In P and BVOPs, the intra shape coding mode is employed only if the current boundary block cannot be efficiently predicted. In inter shape coding, the boundary block is first predicted from the temporally previous or future VOP (depending on the VOP type) and then the difference between the current and the predicted shape blocks is context-based arithmetic coded. The shape motion vectors are also coded predictively using the motion vectors of the surrounding texture and shape blocks.

III. Key VOP Selection in the MPEG-4 Compressed Domain

Typically, key VOPs should be selected such that they reflect significant changes in the shape, color, and texture content of a video object. Using the shape content of a video object for key VOP selection has many advantages over using the color and/or texture content. First, the texture and color of a video object remain generally consistent during a video object's lifespan. This fact is used in many spatio-temporal segmentation algorithms for video object segmentation [18], [19]. The shape of a video object, however, may vary significantly due to the object's movement, structure (e.g., articulated, elastic), occlusion, etc. Therefore, a significant change in the content of a video object is more likely to be detected if the object's shape is employed as a measure. Second, using the shape of a video object instead of its color or texture is potentially more computationally efficient. The MPEG-4 bit stream structure is designed such that it is not possible to decode the texture information without having to decode the shape information [20]. On the other hand, the shape information can be extracted from the bit stream without having to decode the texture information when a resynchronization marker is placed before each macroblock. In such a case, extracting the shape information from the bit stream requires very few operations.

Because of the above reasons, our proposed key VOP selection method is based on the shape content of video objects. We approximate the shape of the VOP from the shape coding modes in the MPEG-4 bit stream, without requiring the decoding of the shape information. Besides saving computations, this approximation makes the proposed algorithms less dependent on the segmentation errors and how lossy the shape information is coded.

In this section, we propose a key VOP selection method that is based on the significant changes in the approximated

shape content of video objects. That is, the first VOP of a video object is selected as a key VOP, and a new key VOP is declared whenever a significant change occurs in the shape of the video object. In order to detect the significant changes in the shape content, a shape similarity measure is required. We employ the Hamming and Hausdorff distance measures to estimate the difference between two shapes. The Hamming distance measures the point-by-point difference between two shapes, whereas the Hausdorff distance measures the largest distance between the contours of two shapes. Both of these distance measures are commonly used in shape retrieval, and they both have different implementation and computational complexity requirements. We modify these distance measures so that they are computed based on the shape approximations derived from the shape coding modes in the MPEG-4 compressed domain.

In our shape approximation, each 16×16 shape block is represented with one value, depending on the location of the shape block, i.e., inside, outside, or at the border of the video object. Recall that three coding modes are possible for IVOPs: transparent, opaque, and intra. The coding mode of a shape block directly indicates the shape block type, i.e., whether a block is inside, outside, or a boundary block. This property makes IVOPs ideal key VOP candidates for our key VOP selection algorithms. Nevertheless, there may be MPEG-4 bit streams that do not have periodic IVOPs or the temporal distance between consecutive IVOPs may be very large. In such cases, it may be necessary to consider P and BVOPs as key VOP candidates as well. However, in P and BVOPs the shape blocks are coded predictively. Therefore, it may not be possible to determine whether a shape block is an inside, outside, or boundary block, without fully decoding and reconstructing the shape information. To address this problem, we next propose a method that allows the use of the same approximation (i.e., inside, outside, or boundary) of a shape block in P and BVOPs.

In MPEG-4, the shape of a P and BVOP is coded by using one of the seven possible coding modes that are summarized in Table I. If the coding mode of a shape block in a P or BVOP is opaque, transparent, or intra, then the shape block is inside, outside, or at the boundary of the video object, respectively. If the coding mode of a shape block is one of the four inter modes, then it is not possible to indicate in an accurate way that the shape block belongs to inside, outside, or the boundary of the video object, without decoding the shape information of the reference and predicted shape blocks, and reconstructing the predicted shape block. However, we can predict where the shape block is located by considering each possible combination of the shape coding modes of the reference and predicted shape blocks, as summarized in Table II. Our prediction rules are based on the following observations. If a shape motion vector and/or some update information is coded for a predicted shape block, then regardless of the shape coding mode of the reference block, the predicted shape block is very likely to be located at the boundary of the video object. If neither motion vectors nor update information is coded, then the shape block type will be expected to be exactly the same as that of the reference block. However, since the shape motion vector is coded predictively, this may not be always true. Therefore, this inter mode requires further analysis. The MPEG-4 variable length coding tables used for the shape coding modes have been constructed such that, if both the reference and the predicted block are opaque, then it is most efficient to transmit the predicted block in inter mode with no motion vector and update information. Hence, if the reference block is opaque and the predicted block is inter coded with no update information being sent to the decoder, then the current block is likely to be an opaque block. On the other hand, if both the reference and the predicted blocks are transparent, then it is most efficient to transmit the current block as transparent. Therefore, if the shape of a predicted block is inter coded with no motion vector and update information, and its reference block is transmitted as transparent, then it is very likely that the predicted block belongs to the boundary of the video object.

TABLE II
APPROXIMATION OF THE SHAPE CODING MODES FOR P AND BVOPS

Block coding mode in the reference VOP	Block coding mode in the predicted VOP	Approximated block coding mode
transparent, opaque, intra	transparent	transparent (outside)
transparent, opaque, intra	opaque	opaque (inside)
transparent, opaque, intra	intra	intra (boundary)
transparent, opaque, intra	MVDs!=0 & no inter update	intra (boundary)
transparent, opaque, intra	MVDs=0 & inter coded	intra (boundary)
transparent, opaque, intra	MVDs!=0 & inter coded	intra (boundary)
transparent	MVDs=0 & no inter update	intra (boundary)
opaque	MVDs=0 & no inter update	opaque (inside)
intra	MVDs=0 & no inter update	intra (boundary)

In order to reconstruct the approximated shape information of P and BVOPs, first, the approximation rules summarized in Table II are applied to each PVOP in a group of pictures so that all the inter modes of the PVOPs are mapped to one of the transparent (outside), opaque (inside), and intra (boundary) coding modes. Then, the same rules are applied to the BVOPs to approximate them from their reference I or PVOPs.

If a predicted block does not have a corresponding reference block in its reference VOP, then we apply the copy rule of MPEG-4 [14]. That is, if the number of lines (respectively, columns) is larger in the current VOP than in the reference VOP, the bottom line (respectively, rightmost column) is replicated as many times as needed in the reference VOP such that all blocks in the predicted VOP have corresponding blocks in the reference VOP.

A. Key VOP Selection Using the Modified Hamming Distance

The Hamming distance between two shapes is defined as the number of different pixels between the shapes. In our proposed method, the shape of the VOP is obtained by using the approximations described in the previous section, and the values "0," "1," and "2" are assigned to the outside, boundary, and inside shape blocks, respectively, as depicted in Fig. 2. Then, a modified version[1] of the Hamming distance between the two VOPs is computed as follows:

$$d = \sum_{n=0}^{N} \sum_{m=0}^{M} |\alpha_{m,n}^{1} - \alpha_{m,n}^{2}|$$

[1]For the Hamming distance measure, the distance between two pixels can be only 0 or 1. The distance between two pixels can here be 0, 1, or 2.

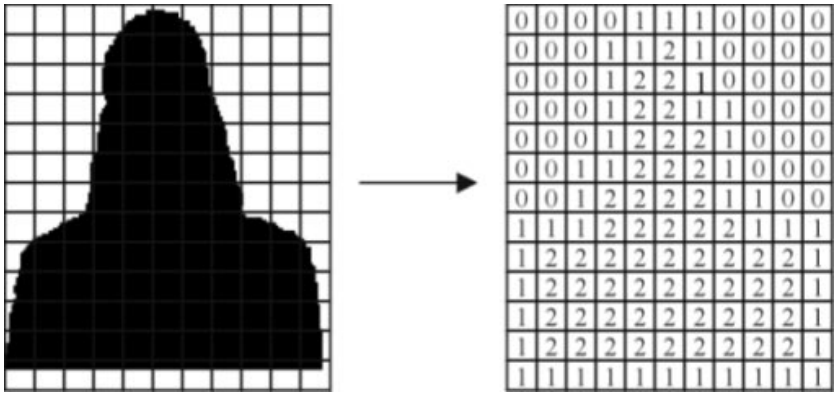


Fig. 2. Approximation of the shape of an IVOP by using the shape coding modes in MPEG-4. The "0," "1," and "2" values are assigned to the outside (transparent), boundary (intra), and inside (opaque) blocks, respectively.

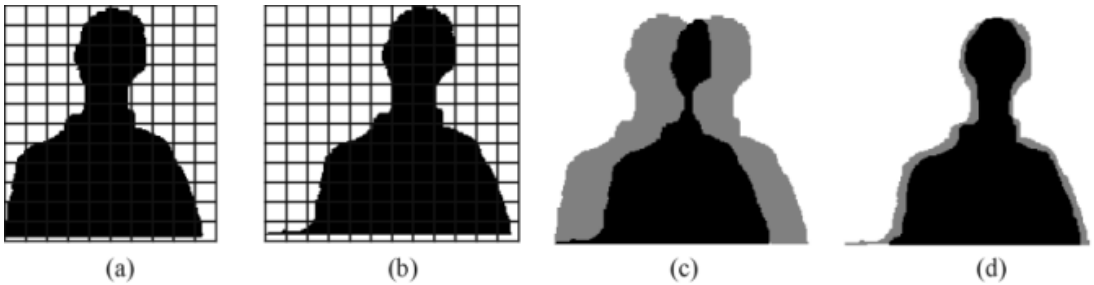


Fig. 3. (a) Shape of a key VOP, (b) shape of a key VOP candidate, (c) the large Hamming distance between the two VOPs (shown in gray) caused by the miss-alignment, and (d) the small Hamming distance between the two VOPs using mass center alignment.

where

$\alpha^1_{m,n}$ shape approximation value of the key VOP candidate in the mth row and nth column (in number of blocks) of the binary alpha plane;

$\alpha^2_{m,n}$ shape approximation value of the temporally closest key VOP corresponding to the same location;

M and N width and height, respectively, of the VOPs bounding box.

When the horizontal and/or vertical dimensions of the key VOP candidate are different from those of the key VOP, M and N are assigned to the larger dimensions, and the extended blocks are padded with "0." Because of the "0," "1," and "2" values assigned to outside, boundary, and inside blocks, respectively, the modified Hamming distance is larger when an outside block corresponds to a inside block than when an outside block corresponds to a boundary block, or vice versa.

A problem here is that a slight spatial shift between two very similar shapes may result in a large Hamming distance. Consider the two alpha planes presented in Fig. 3(a) and (b). Even though the shapes look almost the same, the Hamming distance between the two shapes is very large, as depicted in Fig. 3(c). The minimum Hamming distance between two shapes can be determined by computing the Hamming distance for every possible alignment of the two shapes. However, this would require a very large number of computations, making the algorithm impractical. Our experiments showed that aligning the mass centers of the two shapes provides a good approximation for the alignment corresponding to the smallest Hamming distance. This is depicted in Fig. 3(d). Since the actual shape of a VOP is not available without decoding the bit stream, the mass centers are found by using the shape approximations.

A new key VOP is selected when the distance between the approximated shape of a key VOP candidate and that of the key VOP is larger than a threshold. The threshold should be adaptive to 1) the activity level and 2) the size of the video object. First, the activity level of a video object needs to be considered because a threshold that is optimized for low activity video objects may result in an erroneous selection of every single key VOP candidate as a key VOP in highly active video objects. Even though it is desired to have more key VOPs for video objects that are more active, the threshold needs to be increased in order to avoid selecting an excessive number of key VOPs for such video objects. Second, the threshold should be selected so as to maintain size invariance. This can be achieved by scaling it with the area of the VOP bounding box. We compute the threshold for each key VOP candidate as follows:

$$T_1 = \lambda_1 \phi \min(M_1, M_2) \min(N_1, N_2)$$

where

λ_1 empirically determined parameter that is constant for all VOPs;

ϕ is determined by the activity level of the video object;

M_1 and N_1 width and height (in number of blocks) of the key VOP, respectively;

M_2 and N_2 width and height of the key VOP candidate, respectively.

In the cases where the heights and the widths of the current key VOP and key VOP candidate are different, the smaller dimen-

sions are used to determine the area of the VOP. This way, if the dimensions of the current key VOP and the key VOP candidate are significantly different, then the threshold is made small enough so that it is likely to be exceeded.

Since the parameter ϕ depends on the activity level of the video objects, and the video objects may not have uniform activity levels throughout their lifespans, we need to divide the video objects into temporal segments with uniform activity levels. The activity level of a video object can be predicted by monitoring the number of intra coded shape blocks in the PVOPs and BVOPs, and defining a new segment when a significant change is detected. The number of intra coded shape blocks, γ, can be obtained from the MPEG-4 bit stream without decoding the shape data. In order to provide size invariance, γ is scaled with the area of the VOP.

The gradient of γ is used to determine the significant variations in γ. We employ a five-point median filter in order to remove the spikes that correspond to sudden changes in γ with very short duration. This is followed by a three-point averaging filter to smooth the local changes. Then, the gradient is approximated by

$$\Delta\gamma \cong \gamma[n] - \gamma[n-1]$$

where $\gamma[n]$ and $\gamma[n-1]$ are the numbers of intra coded shape blocks of the current PVOP or BVOP and the temporally previous PVOP or BVOP, respectively. A large gradient value indicates a significant change in γ. Whenever the absolute value of the gradient of γ is above a threshold T_{ts}, a new temporal activity segment is defined. After thresholding, very small temporal segments are combined with the neighboring temporal segments to prevent having an excessive number of temporal segments.

B. Key VOP Selection Using the Hausdorff Distance

The Hausdorff distance measure can also be used to measure the similarity between two shapes. It is defined as the maxmin function between two sets of points as follows [21]:

$$h(A, B) = \max_{a \in A}\{\min_{b \in B}\{d(a, b)\}\}$$

where a and B are the points of the sets A and B, respectively, and $d(a, b)$ is the Euclidean distance between these points. More specifically, the Hausdorff distance between the sets of points A and B is the maximum distance of the points in set A to the nearest point in set B. The Hausdorff distance is not symmetric, i.e., $h(A, B)$ may not be equal to $h(B, A)$. Therefore, a more general definition of the Hausdorff distance is given by

$$H(A, B) = \max\{h(A, B), h(B, A)\}$$

where $h(A, B)$ and $h(B, A)$ are the Hausdorff distances from A to B, and from B to A, respectively.

Similar to the key VOP selection algorithm proposed in the previous section, the first VOP of a video object is declared as a key VOP, and whenever the Hausdorff distance between a key VOP candidate and its temporally closest key VOP is larger than an adaptive threshold, the key VOP candidate is selected as a new key VOP. As in the Hamming distance case, the contours of the key VOP and the key VOP candidate are aligned using their mass centers in order to make the Hausdorff distance invariant of spatial shifts.

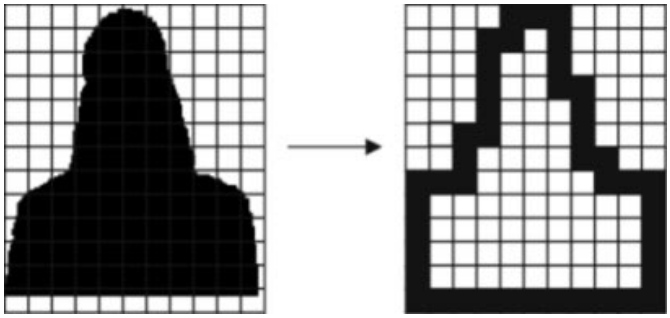


Fig. 4. Approximation of the shape contour of an IVOP by using the shape coding modes in MPEG-4. The intra coded (boundary) shape blocks in IVOPs are selected as the contour points.

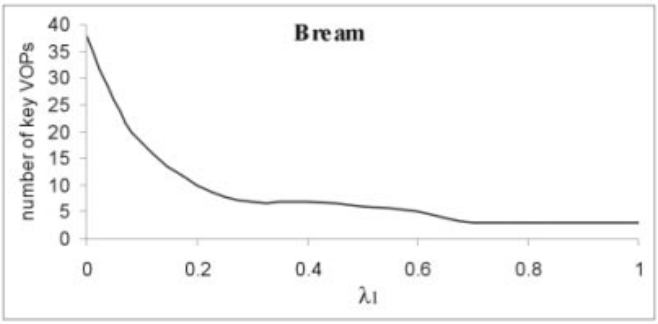


Fig. 5. Change in the number of key VOPs for different λ_1 values for the Bream video object.

TABLE III
SELECTION OF THE VALUES FOR THE PARAMETER ϕ DEPENDING ON THE ACTIVITY LEVEL OF VIDEO OBJECTS

ϕ	Activity level (average percentage of intra coded shape blocks)
1	0% to 29.9%
1.2	30% to 69.9%
1.5	70% to 100%

Finding the Hausdorff distance between the shape contours of the key VOP and the key VOP candidate involve a large number of Euclidean distance computations. Moreover, extracting the contours of the VOP requires the decoding of the shape data. In order to avoid these computations, we approximate the contour of a VOP shape from the shape coding modes by defining the boundary shape blocks as the contour points. This is depicted in Fig. 4. As a result, the number of contour points is significantly reduced, yielding a 16×16 times reduction in computations, besides not needing to decode the shape data.

Unlike the threshold used in the Hamming distance based algorithm, the threshold used in this algorithm does not depend on the activity level of the video object. Our experiments showed that high activity video objects that have large numbers of intra coded shape blocks do not necessarily have large Hausdorff distances between the key VOPs and the key VOP candidates.

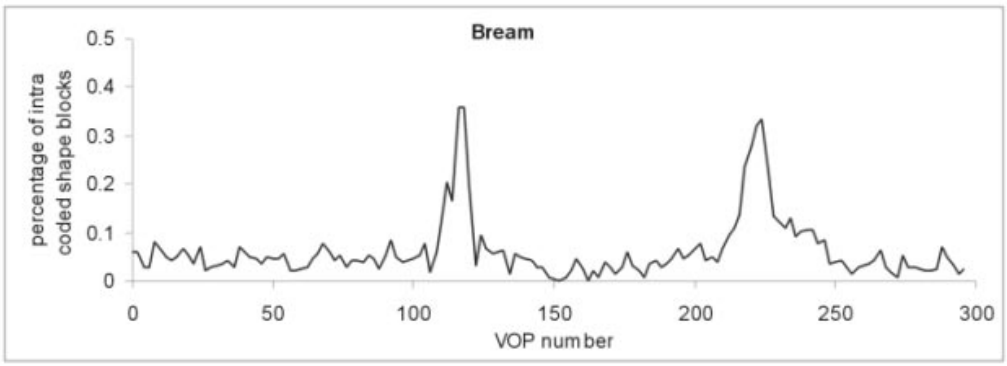


Fig. 6. Intra coded shape block activity for the Bream video object.

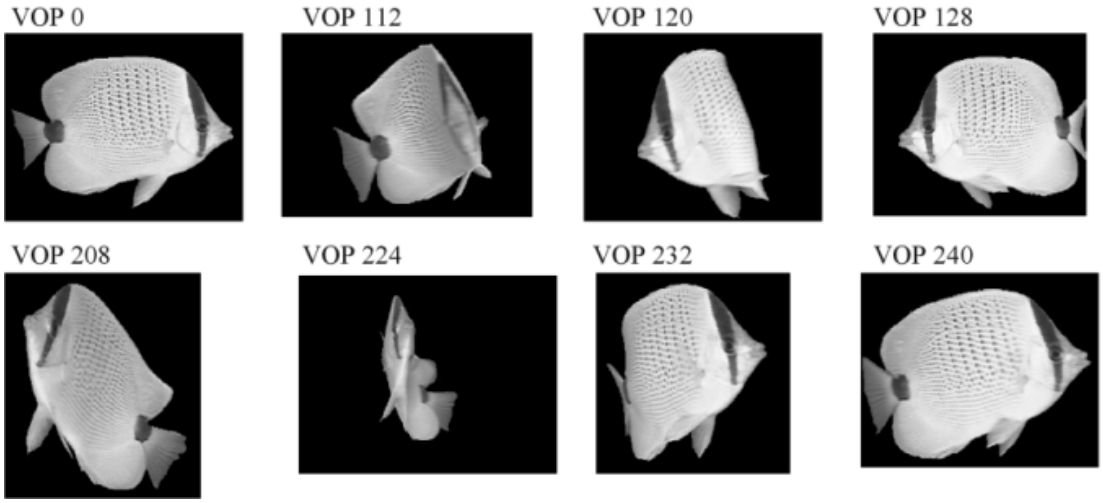


Fig. 7. Key VOPs selected using the Hamming distance based algorithm for the Bream video object.

This should be expected because, unlike the Hamming distance where all the points of the shape affect the distance between two VOPs, the Hausdorff distance is affected by only the two points that have the largest distance, one in the key VOP, and the other one in the key VOP candidate. The threshold, however, still depends on the size of the video object. Since the Hausdorff distance is based on the Euclidean distance, the threshold is scaled by the diagonal length of the VOP bounding box. The threshold is given by

$$T_2 = \lambda_2 \sqrt{\min(M_1, M_2)^2 + \min(N_1, N_2)^2}$$

where

- λ_2 predetermined scale-factor that is constant for all VOPs;
- M_1 and N_1 width and height (in number of blocks) of the key VOP, respectively;
- M_2 and N_2 width and height of the key VOP candidate, respectively.

If the widths and heights of the key VOP and the key VOP candidate are different, then the smaller dimensions are selected so as to make it more likely to declare a new key VOP if there is a significant size difference.

Fig. 8. Key VOPs selected using the Hamming distance based algorithm for the Weather video object.

IV. Experimental Results

The proposed key VOP selection algorithms are implemented in C++, and the Microsoft MPEG-4 decoder [22] is used for parsing and partial decoding of the MPEG-4 bit streams to obtain the shape coding modes. In this section, we present our key VOP selection results for three video objects: Hall Monitor, which is a surveillance video sequence, Bream, which is a sequence that shows a fish swimming and turning, and Weather, which is a sequence that shows an anchor woman presenting the weather forecast. These sequences cover a variety of video objects, from the highly active Hall Monitor to the low motion Weather. The Hall Monitor video sequence was segmented after production, whereas the Bream and Weather video sequences were segmented during production using chroma keying.

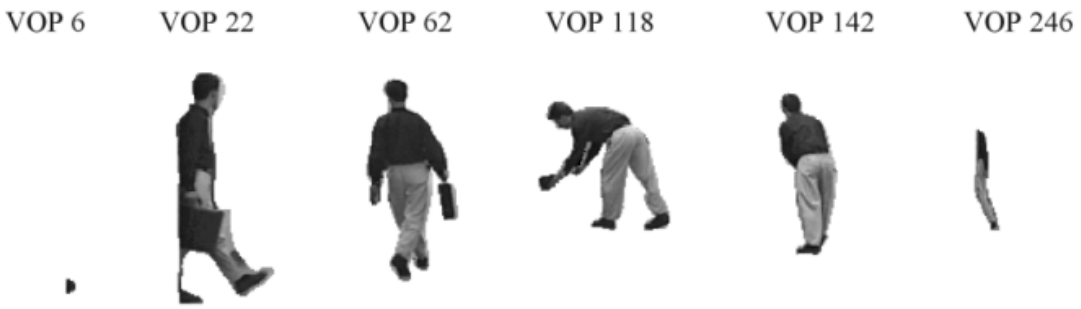


Fig. 9. Key VOPs selected for the Hall Monitor video object using the Hamming distance based algorithm and with employing a video object activity level (ϕ) dependent threshold.

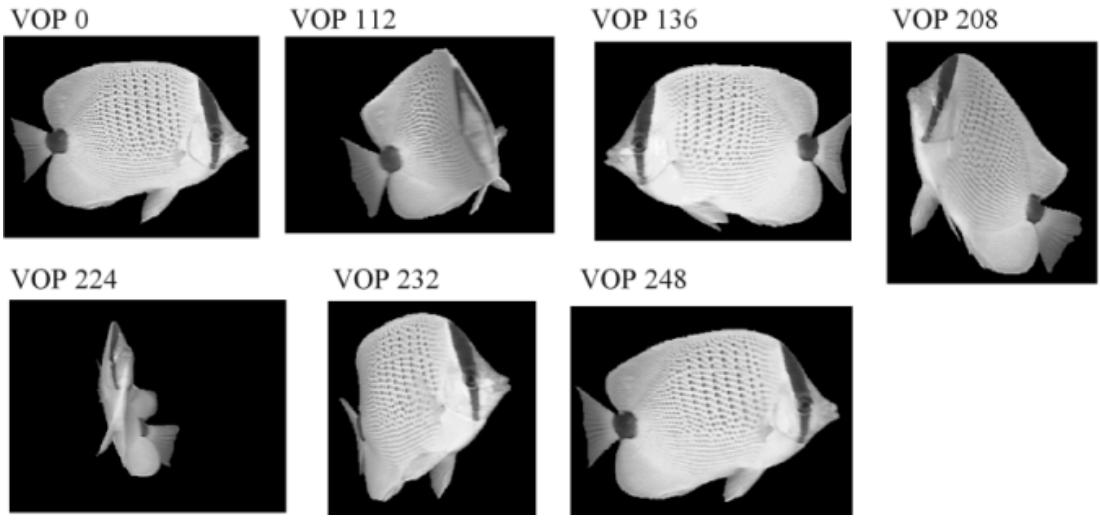


Fig. 10. Key VOPs selected using the Hausdorff distance based algorithm for the Bream video object.

The threshold used in the Hamming distance based key VOP selection algorithm depends on the parameters λ_1 and ϕ, as well as the dimensions of the video object. While the dimensions of the video object are extracted from the MPEG-4 bit stream, the values of the parameters λ_1 and ϕ are empirically determined. The parameter λ_1 indicates the percentage of the shape area that is allowed to be different before selecting a new key VOP. Selecting a lower value for λ_1 would result in a higher number of key VOPs and vice versa. The change in the number of key VOPs for different values of λ_1 is presented in Fig. 5 for the Bream video object. Our experiments show that setting the value of the parameter λ_1 to 0.25 and changing the value of the parameter ϕ from 1 to 1.5 depending on the activity level of the video object, as presented in Table III, result in key VOPs that represent efficiently the content of the video objects.

Since the parameter ϕ depends on the activity level of the video object segment, the video objects are divided into temporal segments with uniform activity levels prior to key VOP selection. The value of the temporal segmentation threshold T_{ts} is set to 0.01. Fig. 6 shows the change in the percentage of intra coded shape blocks for the Bream video object. The two major peaks of the graph correspond to the two highly active segments of the video object, that is, where the shape of the video object changes rapidly. The temporal segments and their corresponding activity levels for the Weather, Bream, and Hall Monitor video objects are shown in Table IV.

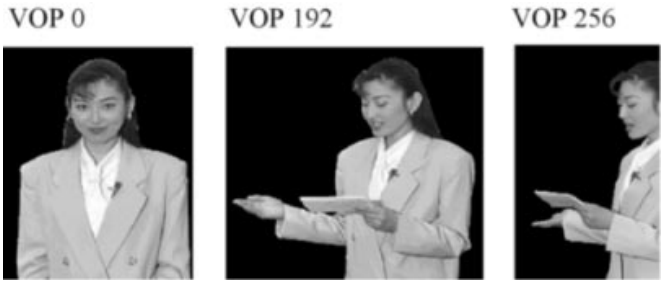


Fig. 11. Key VOPs selected using the Hausdorff distance based algorithm for the Weather video object.

Next, we present our key VOP selection results for video objects that are coded at 15 VOPs per second, following the IPPPIPPP structure, with lossless shape coding, and using a constant quantizer value of 10 for texture. In this set of experiments, only IVOPs are considered as key VOP candidates. We also demonstrate the performance of the proposed algorithms in the case where very lossy coding for shape, i.e., downsampling by four, is employed. Moreover, we present a set of experiments that demonstrate the effects of the use of video object activity level parameter ϕ. In the second set of experiments, we perform key VOP extraction from the IBBBPBBBPBBBP structured MPEG-4 bit streams, where I, P, and BVOPs are also considered as key VOP candidates. In the third set of experiments, we compare our algorithms to other available methods

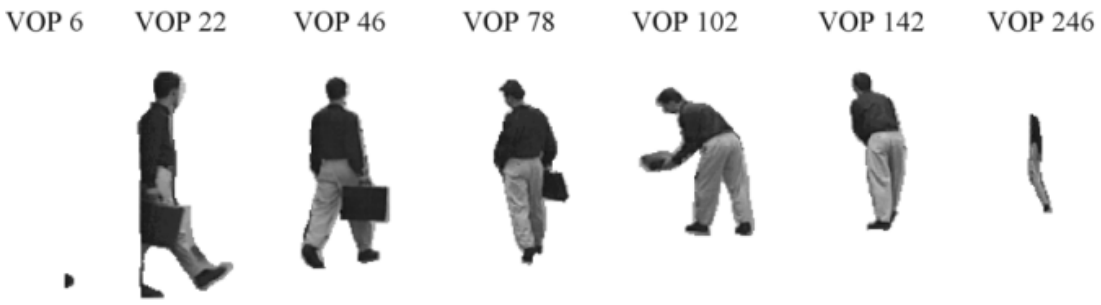


Fig. 12. Key VOPs selected using the Hausdorff distance based algorithm for the Hall Monitor video object.

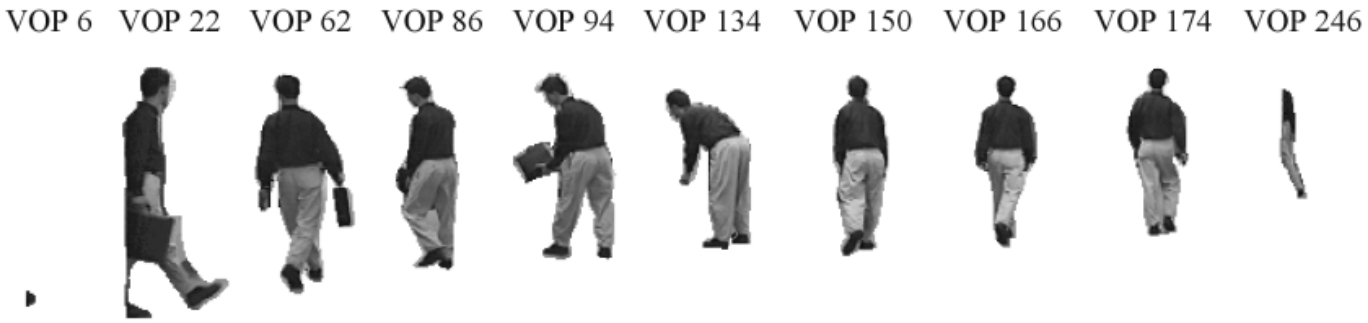


Fig. 13. Key VOPs selected for the Hall Monitor video object using the Hamming distance based algorithm and without employing a video object activity level (ϕ) dependent threshold.

for key VOP selection. Finally, we present a discussion on the computational complexity of our proposed algorithms.

A. Key VOP Selection Using IVOPs

Figs. 7–9 show the key VOPs selected for the IPPPIPPPI structured Bream, Weather, and Hall Monitor video object bit streams using the Hamming distance based algorithm and considering the IVOPs as key VOP candidates. The key VOPs extracted using the Hausdorff distance measure, by setting the value of the parameter λ_2 to 0.2, are presented in Figs. 10–12 for the Bream, Weather, and Hall Monitor video objects, respectively. As seen from the figures, both algorithms select key VOPs that provide a good summarization of the video objects.

The performance of our proposed algorithms has very little dependency on the coding rate of the shape information. In the next experiment, we employ the most lossy coding possible for the MPEG-4 shape information, where the intra coded shape blocks are downsampled by a factor of four. In this case, the selected key VOPs for the Bream video object using the Hamming distance based algorithm are 0, 112, 128, 208, 224, 232, and 240. The key selected VOPs using the Hausdorff distance based algorithm are 0, 112, 136, 200, 224, 232, and 240. These key VOPs are very similar to the ones shown in Fig. 7 and Fig. 10, for the Hamming and the Hausdorff distance based algorithms, respectively. Therefore, our proposed algorithms perform similarly when the video object shape is coded losslessly or in the most lossy mode possible.

We next demonstrate the effects of the video object activity level ϕ when determining the threshold for the Hamming distance based algorithm. Since the Bream and Weather video objects have moderate activity levels, as given in Table IV, the activity level parameter ϕ in most of the temporal segments is equal to 1. Therefore,it does not have any effect on the threshold computation. On the other hand, the activity level of the Hall Monitor video object is high in most of its temporal segments, as shown in Table IV. Consequently, the parameter ϕ affects the decision threshold when computing the Hamming distance. If the parameter ϕ is not employed when selecting key VOPs for the Hall Monitor video object, then the selected key VOPs are 6, 22, 62, 86, 94, 134, 150, 166, 174, and 246, as shown in Fig. 13. Because the Hall Monitor video object is highly active, using the threshold that is not scaled up with the parameter ϕ results in the selection of an excessive number of key VOPs. When these key VOPs are compared to the ones presented in Fig. 9, which were selected considering the activity level of the video object, it can

TABLE IV
TEMPORAL SEGMENTS FOR THE WEATHER, BREAM, AND HALL MONITOR VIDEO OBJECTS

Video object	Segment no	start VOP no	stop VOP no	Activity level (average percentage of intra coded shape blocks)
Weather	0	0	176	0.4%
	1	178	200	5.7%
	2	202	270	1.3%
	3	272	300	5.8%
Bream	0	0	102	4.6%
	1	104	124	15.4%
	2	126	206	3.6%
	3	208	228	18.0%
	4	230	242	11.0%
	5	224	300	3.6%
Hall Monitor	0	6	34	53.3%
	1	36	70	31.5%
	2	72	96	37.1%
	3	98	120	28.2%
	4	122	178	35.8%
	5	180	206	16.9%
	6	208	228	31.92%
	7	230	248	66.3%

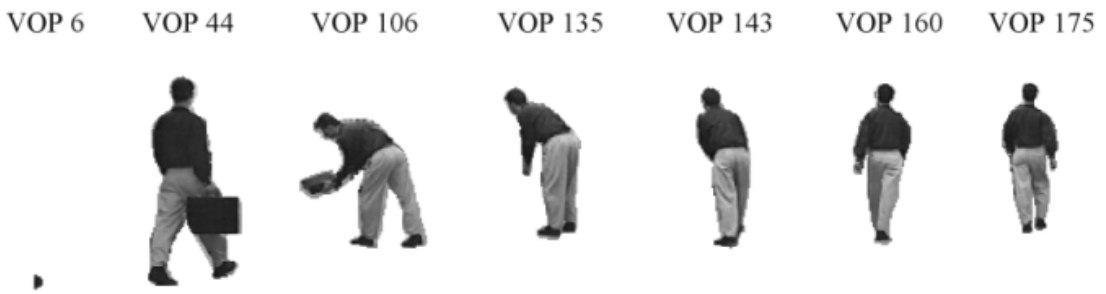


Fig. 14. Key VOP selection results for the Hall Monitor video object using the algorithm proposed in [16].

be seen that they do not improve much the summarization of the salient content of the video object. Therefore, employing the activity level of video objects for key VOP selection prevents the selection of an excessive number of key VOPs for highly active video objects, while yielding a sufficient number of key VOPs that represent efficiently the salient content of a video object.

B. Key VOP Selection Using I, P, and BVOPs

In our next experiment, we extract key VOPs from the IBBBPBBBPBBB structured Bream video object bit stream, where not only IVOPs, but also P and BVOPs are considered as key VOP candidates. The selected key VOPs in this case are 0, 112, 120, 128, 208, 224, 232, and 240 using the Hamming distance measure, and 0, 112, 120, 128, 216, 228, 236, and 252 using the Hausdorff distance measure. The key VOPs selected using the Hamming distance measure are identical to those presented in Fig. 7, where only IVOPs were key VOP candidates. The key VOPs selected using the Hausdorff distance measure are very similar to the VOPs presented in Fig. 10, although they are not exactly the same. This should be expected because, small prediction errors in P or BVOPs do not affect the Hamming distance significantly (since every block in a VOP is used for measuring the Hamming distance), while small prediction errors that may occur at the edge of P or BVOPs may affect the resulting Hausdorff distance (since the Hausdorff distance is measured between two points).

C. Comparisons with Other Methods

For comparison purposes, we have also implemented the pixel domain versions of the same algorithms, which are now applied to uncompressed (actual) shape data instead of the approximated shape data. Using the pixel domain algorithm that is based on the Hamming distance with the same values for the parameters λ_1 and ϕ, the selected key VOPs are 0, 240, 256, and 296 for the Weather, 0, 112, 128, 208, 224, 232, and 240 for the Bream, and 6, 22, 54, 102, 142, 190, 230, and 246 for the Hall Monitor video objects. The pixel domain version of the Hausdorff distance based algorithm yields the key VOPs 0, 192, and 248 for the Weather, 0, 112, 136, 208, 224, 232, and 248 for the Bream, and 6, 14, 22, 78, 110, 134, 182, and 246 for the Hall Monitor video objects. The key VOPs selected using the shape information in compressed domain are similar to the ones selected using the decompressed shape information.

[2]Hamming distance measure does not include the operations involved in computing the activity level.

TABLE V
THE ESTIMATED NUMBER OF OPERATIONS REQUIRED FOR THE HAMMING AND HAUSDORFF DISTANCE BASED KEY VOP SELECTION ALGORITHMS[2]

Key VOP selection algorithm	Number of operations	Example: N=8, M=8
Hamming distance based	NxM subtraction	64 subtraction
	NxM absolute value	64 absolute value
Hausdorff distance based	16x(N+M)x(N+M) square	4096 square
	8x(N+M)x(N+M) square-root	2048 square-root
	8x(N+M)x(N+M) addition	2048 addition

Therefore, processing in the compressed domain becomes very advantageous, since the same performance levels are achieved using 16×16 times less computations and without requiring the decompression of the shape data.

We also compare our key VOP selection algorithms with the compressed domain algorithm proposed by Ferman *et al.* [16]. Their key VOP selection algorithm is based on the texture coding modes of the PVOPs, and a key VOP is declared whenever the corresponding percentage of intra coded blocks exceeds a threshold. Using this algorithm, the key VOPs selected for the Hall Monitor video object are presented in Fig. 14 [16]. As can be seen from the figure, unlike our key VOP selection algorithms (see Figs. 9 and 12), the algorithm proposed in [16] selects redundant key VOPs (see VOPs number 143, 160, and 175), while also failing to represent some important content changes, more specifically the VOP number 246.

D. Computational Complexity

While the performance of the proposed Hamming distance and Hausdorff distance based algorithms are similar, their implementation and complexity tradeoffs differ significantly. The Hamming distance based algorithm requires $N \times M$ subtraction and absolute value operations, and $N \times M - 1$ additions per key VOP candidate. Dividing a video object into temporal segments with uniform activity levels requires approximately ten comparisons, one division, and two additions for the five-point median and three-point averaging filtering operations, and one comparison, one subtraction, and one addition for the derivative and clustering operations per key VOP candidate. On the other hand, the Hausdorff distance algorithm requires the computation of the distance from each contour block of the key VOP to each contour block of the key VOP candidate twice (once in each direction). If we approximate the number of contour blocks by $2 \times (N + M)$, then this algorithm would require $16 \times (N + M) \times (N + M)$ square, $8 \times (N + M) \times (N + M)$

square-root, and $8 \times (N+M) \times (N+M)$ addition operations for each key VOP candidate. Table V summarizes the estimated number of operations required for each algorithm and shows an example for typical N and M values. Even when the number of operations required to estimate the threshold for the Hamming distance based algorithm is considered, the Hausdorff distance based algorithm requires significantly more operations than that of the Hamming distance based algorithm. Nevertheless, there are many algorithms proposed for the efficient computation of the Hausdorff distance [21]. Moreover, even though the number of operations required is larger, the implementation of the Hausdorff distance based algorithm is simpler, since it does not require dividing the video objects into temporal segments with uniform activity levels.

V. Conclusions

In this paper, we have presented a method for key VOP selection using the Hamming and the Hausdorff distance measures. The corresponding algorithms employ shape approximations obtained from the shape coding modes of I, P, and BVOPs without decoding the shape information in the MPEG-4 bit stream. Using the shape approximations, the operations required to compute the Hamming and Hausdorff distances are reduced by approximately 16×16 times. Since the decompression of the shape data is not required, the bit stream processing time is also reduced significantly. Besides saving computations, using the shape approximations makes the proposed algorithms less dependent on the segmentation errors and how lossy the shape information is coded. The performances of the Hamming and Hausdorff distance based algorithms are similar, in the sense that they select key VOPs that represent efficiently the salient content of the video objects. Therefore, depending on the application and available processing resources, either one can be used for key VOP selection.

References

[1] ISO/IEC, "Information technology—Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbits/s: Video,", 11 172-2, 1993.

[2] ISO/IEC, "Information technology—Generic coding of moving pictures and associated audio information: Video,", 13 818-2, 1995.

[3] ITU2T, "Video coding for low bit rate communication,", Recommendation H.263, 1996.

[4] ISO/IEC, JTC1/SC29/WG11, "MPEG-7: Requirements document ver. 11.0,", N2723, March 1999.

[5] S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatiotemporal queries," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 602–615, Sept. 1998.

[6] E. Ardizzone and M. La Cascia, "Automatic video database indexing and retrieval," *J. Multimedia Tools Applicat.*, no. 1, pp. 29–56, Jan. 1997.

[7] Y. Deng and B. S. Manjunath, "NeTra-V: Toward an object based video representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 616–627, Sept. 1998.

[8] J. S. Boreczky and A. R. Lawrence, "Comparison of video shot boundary detection techniques," *Proc. SPIE*, vol. 2670, pp. 170–179, 1996.

[9] G. Lupatini, C. Saraceno, and R. Leonardi, "Scene break detection: A comparison," in *Proc. RIDE'98*, Feb. 1998, pp. 34–41.

[10] V. Kobla, D. Doermann, and K. I. D. Lin, "Archiving, indexing, and retrieval of video in the compressed domain," in *Proc. SPIE Conf. Multimedia Storage and Archiving Systems*, vol. 2916, Nov. 1996, pp. 78–89.

[11] V. Kobla and D. Doermann, "Indexing and retrieval of MPEG compressed video," *J. Electron. Imag.*, vol. 7, no. 2, pp. 294–307, April 1998.

[12] K. Tse, J. Wei, and S. Panchanathan, "A scene change detection algorithm for MPEG compressed video sequences," in *Canadian Conf. Electrical and Computer Eng. (CCECE '95)*, vol. 2, 1995, pp. 827–830.

[13] B. L. Yeo and B. Liu, "A unified approach to temporal segmentation of motion JPEG and MPEG compressed video," in *IEEE Conf, Multimedia Computing and Systems*, May 1995, pp. 81–88.

[14] MPEG-4 Video Group, JTC1/SC29/WG11, "Coding of audio–visual objects: Video,", Jan. 1999.

[15] B. Gunsel, A. M. Tekalp, and P. J. L. Van Beek, "Content-based access to video objects: Temporal segmentation, feature extraction and visual summarization," *IEEE Trans. Signal Processing*, vol. 46, pp. 261–280, April 1998.

[16] A. M. Ferman, B. Gunsel, and A. M. Tekalp, "Object-based indexing of MPEG-4 compressed video," in *Proc. IS&T/SPIE Symp. Electronic Imaging*, vol. 3024, Feb. 1997, pp. 953–963.

[17] ISO/IEC, JTC1/SC29/WG11, "Description of MPEG-4,", N2995, Oct. 1999.

[18] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 539–546, Sept. 1998.

[19] D. Zhong and S.-F. Chang, "Video object model and segmentation for content-based video indexing," in *IEEE Int. Conf. Circuits and Systems*, June 1997, pp. 1492–1496.

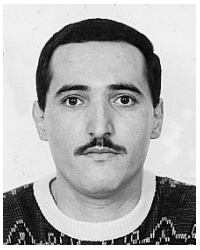
[20] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Commun. Mag.*, pp. 112–119, June 1998.

[21] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 850–863, Sept. 1993.

[22] Microsoft, JTC1/SC29/WG11, "MPEG-4 video encoder/decoder,", 2000.

Berna Erol received the B.S. degree in computer and control engineering in 1994 from the Istanbul Technical University, Istanbul, Turkey, and the M.A.Sc. degree in 1998 from the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, B.C., Canada, where she is currently a Ph.D. student and Research Assistant holding Killam and NSERC scholarships.

Upon completion of her undergraduate studies, she was a Scientific Engineer at the University of British Columbia for two years, where she developed software applications for DSPs. Her research interests include image and video compression, object-based video representations, and content-based visual information retrieval.



Faouzi Kossentini (S'89–M'95–SM'98) received the B.S., M.S., and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, in 1989, 1990, and 1994, respectively.

In 1995, was a Research Scientist at Nichols Research Corporation, Huntsville, AL. Since January 1996, he has been an Assistant Professor and then an Associate Professor with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, B.C., Canada, where he is involved in research in the areas of signal processing, communications, and multimedia. He has authored or coauthored more than 100 journal papers, conference papers, book chapters, and patents. He has been active as a voting member, and recently as Head of Delegation, of the Canadian Delegate to ISO/IEC/JTC1/SC29, which is responsible for the standardization of coded representation of audiovisual, multimedia, and hypermedia information. In particular, he has participated in the most current JBIG/JPEG and MPEG-4 standardization activities. He has also participated in the most current ITU-T low bit rate video coding standardization activities. Most notably, he is a coauthor of the current ITU-T H.263 Test Model.

Dr. Kossentini has served as a technical area coordinator and Member of the Technical Program Committee of ICIP'2000. He is currently an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the the IEEE TRANSACTIONS ON MULTIMEDIA.