Integrated Video Object Segmentation and Shape Coding

JANEZ ZALETELJ
Department of Telecommunications, Faculty of Electrical Engineering
University of Ljubljana
Trzaska 25, SI-1000 Ljubljana
SLOVENIA

Abstract: To enable content-based access and manipulation of video content, modern multimedia communications require per-object based access to video data. Two additional building blocks of video transmission systems needs to be defined, which enable access to video objects at the receiver end. These are video segmentation, which decomposes video frame into a set of layers, where each layer includes segmentation mask of one semantic video object, and shape coding system, which enables transmission of the segmentation information to the receiver. This paper presents a novel, integrated approach to the fore mentioned problem, where the parameters of object segmentation module and shape coding module are dynamically adjusted depending on bandwidth constraints. Background differencing with morphological filtering is used for extracting shape information, and B-splines are used for shape approximation. An efficient predictive scheme for adaptive arithmetic encoding of spline control points is used for shape coding.

Key-Words: video coding, shape coding, video object segmentation

1 Introduction

Multimedia applications like digital libraries or studio postproduction require access to video data based on object descriptions, where objects are described by their texture, motion and shape. In the framework of object-based video coding [1], individual video objects are represented by layers, which allow independent coding and manipulation of each object. In this context shape information is essential and its efficient encoding needs to be investigated [6].

MPEG-4 standard provides the first standardized representation of an object's shape within a video bitstream. In MPEG-4 it is assumed that each video object is provided with its corresponding shape information, given as a binary pixel map. Pixel map is generally the same size as the bounding box of the object, and the value of each pixel indicates if it belongs to video object or background. The problem of efficient encoding of video object shape is becoming increasingly important, since the shape information occupies a great part of the total bit stream.

Two main classes of binary shape coding techniques exist. A bitmap-based coder encodes for each pixel of the object's bounding box whether it belongs to the object or not, while a contour-based coder encodes the outline of the object, and to retrieve the bitmap of the object shape, the contour is filled with the object label. Bitmap based compression for shape coding was chosen for MPEG-4, because it was shown that it offers good compression efficiency with reduced computational complexity. However, further work on contour-based shape coding has shown that by utilizing temporal decorrelation it is possible to substantially increase coding efficiency.

In MPEG-4, the problem of image analysis was avoided by using presegmented video sequences, and also the literature on shape coding clearly separates the two problems of extracting and coding of shape information. However, in a real object-based video coding system, both problems could be treated simultaneously, increasing the flexibility of the system. By dynamic adjustment of parameters of object segmentation module based on shape bandwidth requirements, it is possible to control the number and quality of extracted segmentation masks. Using such a system setting, shape coding module can adapt to different network bandwidth conditions, always allocating bandwidth on encoding most important objects.

In this paper, an integrated approach to the shape extraction/coding problem is presented. Under a common framework, shape extraction, approximation and coding modules are integrated. Given a total bandwidth available for coding shape parameters, shape extraction and coding parameters are dynamically adjusted. For shape extraction, we use an algorithm based on image differencing. Shape approximation module is responsible for finding a B-spline approximation of object's contour, which minimizes the contour distortion. In the intra-frame mode of operation, starting contour parameterization using given number of spline segments is determined based on multi-scale analysis of curvature function of the boundary. This results in an initial B-spline approximation where low-

curvature parts of the boundary are modelled by larger B-spline segments than high-curvature parts. B-spline control point optimization is performed next to minimize the distortion of the approximation. In the inter-frame mode, current frame shape approximation is based on the motion-compensated approximation from the previous frame, which is re-optimized, and based on the distortion constraint, segments are added or removed if necessary.

A predictive encoding of B-spline control points is employed to minimize temporal redundancy of shape information. Sequence of symbols is further compressed using adaptive arithmetic encoding.

1.1 Previous Work

Video object segmentation has been an active research topic for a long time, but no universal solution has been found yet. Automatic video segmentation methods can be classified into three approaches: 1) intra-frame segmentation with inter-frame tracking [9], 2) motion field clustering [10], and 3) frame differencing. Automatic methods often fail to produce semantically meaningful objects since these objects may contain multiple motions and local deformations. For a class of simple videophone sequences with a fixed camera, a frame differencing approach was found successful in extracting semantically meaningful video objects.

Investigations in the fields of image processing, object recognition, and computer vision have led to the development of many techniques for both lossless and lossy representation of shape information. For lossless encoding of the object boundaries, a chain code technique [2] was developed. A chain code follows the contour and encodes the direction in which the next boundary pixel is located. Lossy polygon approximation [11] was developed for object-based analysis-synthesis video coders. Subjective evaluations of such coders showed that the error of 1.4 pel is acceptable to allow proper representation of objects in low-bit-rate applications.

The second class of lossy contour coding techniques uses simple parametric functions to approximate the contour. Amongst them, B-splines [5], which are piecewise polynomial functions, are most popular, since they give a smooth analytic representation of the contour. They possess nice properties such as local control, continuity and differentiability and the restored shape looks more natural. In most cases a planar shape is approximated by a quadratic or cubic B-spline model. Several techniques are aimed at finding an optimal B-spline approximation in the rate-distortion sense. Lu and Milios [4] try to find the optimal spline model by optimizing positions of the control points, Katsaggelos et al. [6] use dynamic programming to find the shortest path in the directed acyclic graph, and Lagendijk [7] achieves optimal solution by iteratively pruning (merging) initial accurate approximation with large number of spline segments.

In this work we concentrate on:

- segmentation and shape approximation modules which are dependent on the available bitrate for encoding,
- finding a close-to-optimal spline segment distribution for approximation object boundary,
- minimization of approximation distortion,
- increasing coding efficiency by utilizing temporal decorrelation of shape parameters.

2 Video Object Segmentation Module

To extract a video object shape, background differencing is used. A pixel is said to belong to a video object, if its squared difference from the reference background pixel divided by a variance of background image values exceeds a given threshold *T*.

Background image is continually updated, and smoothing and morphological filtering of difference image is used to produce binary object segmentation masks. Number of iterations of morphological filtering and size of structuring element is adjusted dynamically based on rate constraints. Boundary of each object is extracted as an 8-connected contour.

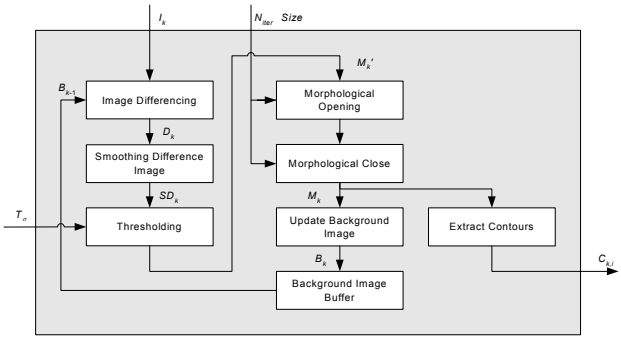

Fig.1: Video object segmentation module, which uses frame background differencing and morphological filtering.

3 Shape Approximation Module

Shape approximation module is responsible for finding a polygon/B-spline approximation of object's contour, which is optimal in the rate-distortion sense. Given total available bit budget for one object, it finds maximum number of spline segments for contour approximation.

The main problems of video object shape approximation by B-splines, which need to be addressed are:

- selection of a suitable number of spline segments (and consequently control points) for shape approximation so that rate and distortion constraints are satisfied,
- calculation of a distribution of spline segments (contour parameterization) which yields minimal distortion at a given number of segments,
- optimization of control point positions to minimize the distortion.

In the intra-frame mode, shape approximation module determines starting contour parameterization using given number of spline segments. This process is based on multi-scale analysis [3] of curvature function of the contour.

In the inter-frame coding mode, current frame shape approximation is based on the motion-compensated approximation from the previous frame, which is re-optimized, and based on the rate constraint, segments are added or removed if necessary.

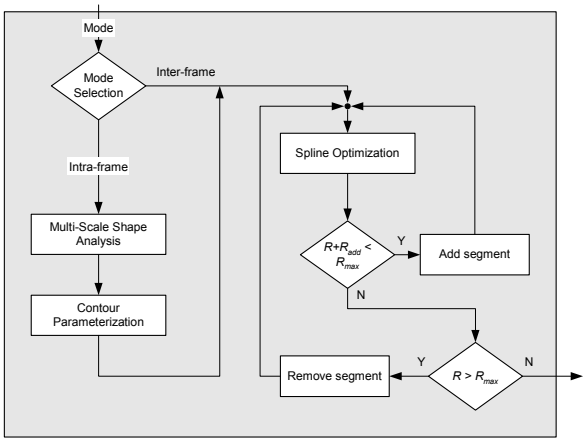

Fig.2: Shape approximation module.

3.1 Multi-Scale Shape Analysis

In order to use the available bit rate for contour encoding in an optimal way, our idea is to perform shape analysis first. The role of the analysis is twofold: first, we can identify and smooth the insignificant shape details in order to control the bit rate, and second, we can allocate spline segments according to the (smoothed) curvature of the original contour.

The curvature $\kappa(u)$ of a plane curve is defined as the derivative of the tangent angle to the curve [3]. Having two parameterized coordinate functions of the contour, $x(u)$ and $y(u)$, it can be written as

$$\kappa(u) = \frac{\dot{x}(u)\,\ddot{y}(u) - \ddot{x}(u)\,\dot{y}(u)}{(\dot{x}^2(u) + \dot{y}^2(u))^{3/2}}. \qquad (1)$$

Computing curvature reliably for a discrete contour is not a trivial problem. The computation of the partial derivatives $\dot{x}(u)$ and $\ddot{x}(u)$ is an ill-posed problem, influenced by spatial quantization noise. A general framework for regularized differentiation and smoothing of the signals is provided by linear scale-space theory.

The (smoothed) derivatives are computed by convolving the coordinate functions with the derivatives of the Gaussian, since we may commute the differential and convolution operators:

$$\dot{X}(u,\sigma) = \frac{\partial}{\partial u}(x * g) = x(u) * \dot{g}(u,\sigma) \qquad (2)$$

$$\ddot{X}(u,\sigma) = x(u) * \ddot{g}(u,\sigma) \qquad (3)$$

By substituting derivatives of $x(u)$ and $y(u)$ by their smoothed versions in Equation 1, we can calculate the approximation of the curvature function $\kappa(u,\sigma)$ at a given scale of smoothing. The result of this procedure is elimination of insignificant shape details and quantization noise from the curvature function.

It is evident that the number of curvature extremes decreases at coarser scales. We use this feature of smoothing process to control the bit rate of the spline approximation. Since one cubic spline segment is defined by a third order polynomial, it has a limited degree of freedom in terms of its curvature.

By tracing the curvature extremes across scales, we are able to select only the desired number of most prominent ones. Other extremes (shape details), which end below the selected scale, are considered noise, and are not approximated by the final B-spline.

3.2 Contour Parameterization

The analysis of the segment distributions has shown that the majority of knots (points where two segments meet) are located close to maximums or minimums of curvature. This observation leads to the idea of making the knot assignments dependent on the curvature of the original contour right from the beginning, by initially positioning a knot at the location of each curvature extreme. Then, for each initial segment, its length and the integral of $|\kappa(t)|$ is determined [7]. Based on these two measures, some of the segments are nominated for merging or splitting. The splitting and merging procedure uses only curvature information, and the following two additional constraints:

- each resulting segment may have no more than one curvature zero crossing,
- the quotient of the lengths of the adjoining segments should be smaller than a given threshold.

3.3 Spline Optimization

Because the initial parameterization is based on curvature heuristics, the resulting initial control points are not optimal. To decrease the distortion of the approximation, we employ a numerical optimization

procedure, which adjusts the positions of control points. Since the approximating spline is given by a set of n control points, the error norm which we want to minimize (Equation 8) is a function of $2n$ coordinates $(x_1..x_n, y_1..y_n)$ of control points.

If the objective function for minimization is defined simply as a quadratic error norm between given contour points and the spline, optimization process may lead to undesirable results, such as self-intersecting B-spline. In [4] a penalty term was added to the quadratic error to ensure that each segment of the curve stays close to the contour. However, we have found that the selection of the parameters of this term is difficult and that self-intersections (loops) often occur. To solve this problem the proposed optimization scheme uses different objective function, defined as a sum of quadratic perpendicular distances from regularly sampled B-spline to the polygon defined by given contour points (Figure 3).

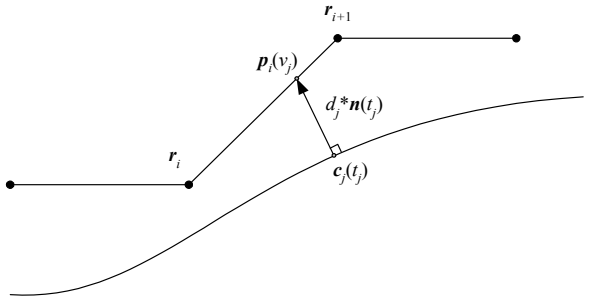

Fig. 3: Definition of perpendicular distance from spline to contour polygon.

The contour of the video object is given by a ordered set of m ($m > n+1$) two-dimensional points $\boldsymbol{r} = (\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_m) = ((r_{1,x}, r_{1,y}), (r_{2,x}, r_{2,y}), ... (r_{m,x}, r_{m,y}))$. The i-th segment of the polygon defined by contour points is given by

$$\boldsymbol{p}_i(v) = (p_{i,x}(v), p_{i,y}(v)) = \boldsymbol{r}_i + (\boldsymbol{r}_{i+1} - \boldsymbol{r}_i) \cdot v\,, \qquad (4)$$

where the parameter v may take values from the interval [0,1].

We will derive the expression for the distance of the spline point $\boldsymbol{C}_j$ from the polygon. An i-th segment of the approximating spline is denoted by $\boldsymbol{c}_i(t) = (c_{i,x}(t), c_{i,y}(t))$, and the point on the spline at parameter t is denoted by $\boldsymbol{C}(t)$. Vector of coefficients of the x coordinate function $c_{i,x}(t)$ is denoted by $\boldsymbol{P}_x = (P_{0,x}, P_{1,x}, ..., P_{n-1,x})$, and vector of coefficients of the $c_{i,y}(t)$ is denoted by $\boldsymbol{P}_y = (P_{0,y}, P_{1,y}, ..., P_{n-1,y})$. The normal vector $\boldsymbol{n}$ at the point $\boldsymbol{C}_j$ is given by

$$\boldsymbol{n} = (n_x, n_y) = \frac{\left(\frac{dc_y(t)}{dt}, -\frac{dc_x(t)}{dt}\right)}{\left\|\left(\frac{dc_y(t)}{dt}, -\frac{dc_x(t)}{dt}\right)\right\|}. \qquad (5)$$

Equation which defines the distance d_j of the spline point from the polygon is

$$\boldsymbol{p}_i(v_j) = \boldsymbol{c}(t_j) + d_j \cdot \boldsymbol{n}(t_j) \qquad (6)$$

From this equation it is possible to calculate the distance d_j

$$d_j = \frac{1}{n_x}\left[r_{i,x} - c_{j,x} + \frac{c_{j,y} - r_{i,y} + \frac{n_y}{n_x}(r_{i,x} - c_{j,x})}{\frac{r_{i+1,y} - r_{i,y}}{r_{i+1,x} - r_{i,x}} - \frac{n_y}{n_x}} \right] \qquad (7)$$

Now the approximation error to be minimized is given by a sum of quadratic distances d_j of the spline points from the polygon and is a function of $2n$ control point coordinates:

$$E(P_{0,x},...,P_{n-1,x},P_{0,y},...,P_{n-1,y}) = \frac{1}{m}\sum_{j=1}^{m}\left(\frac{d_j}{d_{\max}}\right) \qquad (8)$$

4 Shape Coding Module

Design of control point coding algorithm is following these goals:

- to exploit inter-frame temporal correlation of control point positions for increasing coding efficiency,
- to enable flexibility in adding and removing spline control points,
- use context-based encoding, where the context determines the meaning of each symbol,
- use of adaptive arithmetic encoding of symbols, which further reduces bit rate.

Two coding modes are used. In the first frame of the sequence, where no previous information is available, INTRA-frame mode is used. In the following frames, using shape information from previous frame, INTER-frame mode is used to reduce temporal redundancy. If a new object appears in the scene or temporal correspondence between objects could not be established, INTRA coding mode is used.

The shape approximation algorithm provides information on the total number of objects inside current frame, and lists of their control points. Temporal correspondence between objects is also provided, so that the coding algorithm uses INTER mode for objects, which are present in the previous frame, and INTRA mode for new objects.

A majority of vertex-based shape coding algorithms is using predefined variable-length codes for encoding relative vertex positions. VLC tables are optimized for a fixed probability distribution, where longer codes are assigned to larger relative distances. This approach has a major drawback because of its inflexibility. Coding

efficiency is lost if the actual distribution does not closely follow the predefined one.

We use adaptive arithmetic encoding to avoid these problems. Different probability distributions are used for different sets of symbols, such as escape codes, relative octants, relative distances etc. Probability distributions are continuously adapted in both coder and decoder, which mean that they provide good approximation to the actual probability distributions of the video object. This allows us to substantially increase coding efficiency compared to fixed VLC coding schemes.

A current coding context defines, which probability distribution is used at the moment. Coding context defines, what type of information is being encoded. We use the following coding contexts: absolute address, octant code, differential octant code, major intra component, minor intra component, major inter component, minor inter component, and escape code. Each coding context has an associated histogram, which defines cumulative probability distribution of symbols. Histograms are maintained in both encoder and decoder, and they must be updated in the same way to ensure successful decoding.

4.1 Coding shape in INTRA mode

In the INTRA-frame coding mode, successive control points are encoded differentially. INTRA mode utilizes redundancy between directions of two neighboring vertices to increase coding efficiency.

Steps for encoding a control point in intra mode are the following:

- calculate relative address: $\boldsymbol{r}_i = (r_{i,x}, r_{i,y}) = (P_{i,x} - P_{i\text{-}1,x}, P_{i,y} - P_{i\text{-}1,y})$ for $i>0$,
- calculate octant number o_i of relative address $\boldsymbol{r}_i$ for $i>0$: based on angle between $\boldsymbol{r}_i$ and x coordinate axis, 8 octants are defined.
- calculate differential octant code $d_i = o_i - o_{i\text{-}1}$ for $i>1$,
- calculate major component of $\boldsymbol{r}_i$: for octants $o_i =$ 0, 3, 4, or 7, $|r_{i,x}|$ is larger or equal to $|r_{i,y}|$, and major component is computed as $M_i = |r_{i,x}| - 1$, otherwise major component is computed as $M_i = |r_{i,y}| - 1$,
- calculate minor component of $\boldsymbol{r}_i$: for o_i = 0,3,4,7, compute $m_i = |r_{i,y}|$, otherwise $m_i = |r_{i,x}|$.

The initial control point $\boldsymbol{P}_0$ is encoded by its absolute address:

- $P_{i,x}$ is encoded using histogram $H_{abs}(P_{i,x})$
- $P_{i,y}$ is encoded using histogram $H_{abs}(P_{i,y})$

Then the relative address of the second control point $\boldsymbol{P}_1$ is encoded:

- o_1 is encoded using histogram $H_{oct}(o_i)$,
- M_1 is encoded using histogram $H_{major,intra}(M_i)$,
- m_1 is encoded using histogram $H_{minor,intra}(m_i)$.

The following control points $\boldsymbol{P}_i$ are encoded using:

- d_i is encoded using histogram $H_{doct}(d_i)$,
- M_i is encoded using histogram $H_{major,intra}(M_i)$,
- m_i is encoded using histogram $H_{minor,intra}(m_i)$.

A special care is taken in case that a symbol is larger than the total number of symbols in the current context histogram. For example, a major component M_i can be larger than total number of symbols in histogram $H_{major,intra}$. In that case, N_{symb} is subtracted from M_i, and additionally an escape sequence is sent indicating that at the decoder, M_i should be increased for a value of N_{symb}.

4.2 Coding shape in INTER mode

In the INTER-frame coding mode, corresponding control points in two successive frames are encoded differentially using the same techniques as in INTRA mode. Figure 4 illustrates how temporal correspondence between control points in two successive frames is established (Children test sequence).

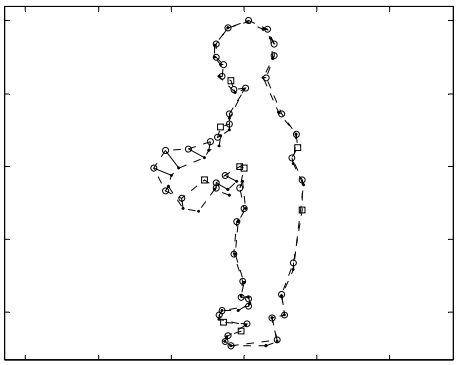

Fig.4: Example of temporal matching of control points. Circles represent control points which are motion compensated, solid lines represent their motion vectors, and squares represent control points, for which the correspondence could not be established, and are encoded in intra mode.

For motion compensated control points, their relative addresses (motion vectors) are encoded with respect to corresponding control point $\boldsymbol{P}_j(t\text{-}1)$ in previous frame: $\boldsymbol{r}_i(t) = (P_{i,x}(t)\text{-}P_{j,x}(t\text{-}1), P_{i,y}(t)\text{-}P_{j,y}(t\text{-}1))$. Differential octant code and major and minor components are coded, using histograms H_{doct}, $H_{major,inter}$, $H_{minor,inter}$.

An escape sequence is transmitted when a control point without correspondence is encountered, and then it is encoded by its intra relative address.

An additional set of escape codes is used to increase efficiency in special cases. If an intra relative address (motion vector) is zero, this information is encoded by an escape code. If a sequence of control points has zero motion vectors, then the whole sequence can be very efficiently encoded by a single escape sequence.

5 Results and conclusion

In order to demonstrate the performance of the proposed shape coding module, we have compared rate-distortion efficiency with two inter-frame coding methods, baseline method [13] and polynomial vertex-based method [12]. The evaluation of coding efficiency was performed on 100 frames of the Children test sequence for one video object plane. Distortion was measured using measure D_n, which is defined as the number of erroneously represented shape pixels of the coded shape divided by the total number of pixels belonging to the original shape.

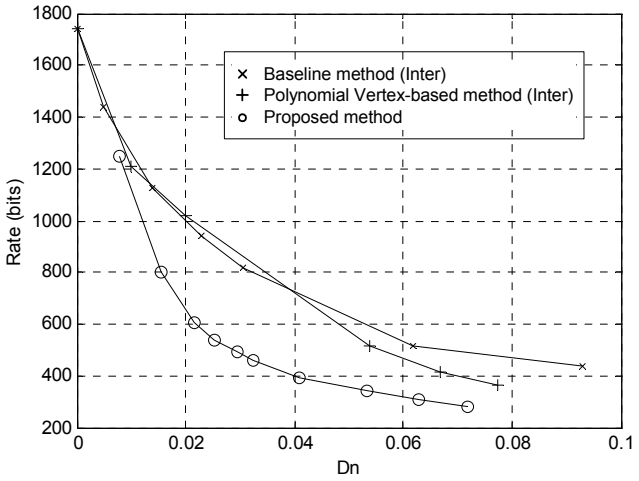

Fig. 5: Rate-distortion curves for one object of the Children test sequence.

The results given in Figure 5 indicate that the proposed coding method greatly outperforms both baseline and polynomial coding in wide range of rate and distortion. This is possible due to use of elaborate adaptive arithmetic control point encoding scheme, and also due to efficient minimization of distortion by spline optimization.

However, the main advantage of the proposed system setting is the integration of segmentation/coding modules, which enables larger flexibility of the system to the changing network bandwidth conditions by adaptively adjusting parameters of both segmentation and coding modules.

References:

[1] ISO/IEC JTC1/SC29/WG11 N2202 "Information Technology - Coding Of Audio-Visual Objects: Visual", Tokyo, March 1998

[2] S.H.Cho, R.C.Kim, S.S.Oh, S.U.Lee, "A Coding Technique for the Contours in Smoothly Perfect Eight-Connectivity Based on Two-Stage Motion Compensation", *IEEE Trans. CSVT*, Vol.9, pp. 59-69, Feb. 1999

[3] F. Mokhtarian, A.K. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves", *IEEE Trans. PAMI*, vol.14, no.8, pp.789-805.

[4] F.Lu, E.Milios, "Optimal spline fitting to planar shape", *Signal Processing*, Vol. 37, pp. 129-140, 1994

[5] G. Farin, "*Curves and surfaces for computer aided geometric design, a practical guide*", Academic Press, Inc., New York, 1994.

[6] A. Katsaggelos, L.P. Kondi, F.W. Meier, J. Ostermann, G.M.Schuster, "MPEG-4 and Rate-Distortion-Based Shape-Coding Techniques", *Proceedings of IEEE*, vol. 86, pp. 1126-1153, 1998

[7] J.Zaletelj, R.Pecci, F.Spaan, A.Hanjalic, R.L.Lagendijk, Rate Distortion Optimal Contour Compression Using Cubic B-Splines, *Proceedings of Eusipco-98*, Rhodes, Greece, 3, 1998, 1497-1500

[8] L.Torres, M.Kunt, "*Video Coding: The Second Generation Approach*", Kluwer Academic Publishers, Boston, 1996

[9] F.Moscheni, S.Bhattacharjee, M.Kunt, "Spatiotemporal Segmentation Based on Region Merging", *IEEE Trans.PAMI*, Vol.20, No.9, pp. 897-915, Sept. 1998

[10] A.M.Tekalp, "*Digital Video Processing*", Prentice-Hall, 1995

[11] J.Chung, J.Lee, J.Moon, J.Kim, "A new vertex-based binary shape coder for high coding efficiency", *Signal Processing: Image Communication*, Vol 15, pp. 665-684, 2000

[12] K.J. O'Connell, "Object/adaptive vertex/based shape coding method", *IEEE Trans. Circ. Syst. Video Technol.*, Vol 7, pp. 251-255, 1997

[13] S.Lee, D.Cho, Y.Cho, S.Son, E.Jang, J.Shin, "Binary shape coding using 1/D distance values from baseline", *Proc. ICIP*, 1997, Vol.1, pp. 508-511, 1997