

Shape Error Concealment Based on a Shape-Preserving Boundary Approximation

Evaggelia Tsiligianni, *Student Member, IEEE*, Lisimachos P. Kondi, *Senior Member, IEEE*, and Aggelos K. Katsaggelos, *Fellow, IEEE*

Abstract—In object-based video representation, video scenes are composed of several arbitrarily shaped video objects (VOs), defined by their texture, shape and motion. In error-prone communications, packet loss results in missing information at the decoder. The impact of transmission errors is minimized through error concealment. In this paper, we propose a spatial error concealment technique for recovering lost shape data. We consider a geometric shape representation consisting of the object boundary, which can be extracted from the α -plane. Missing macroblocks result in a broken boundary. A B-spline curve is constructed to replace a missing boundary segment, based on a T-spline representation of the received boundary. We use T-splines because they produce shape-preserving approximations and do not change the characteristics of the original boundary. The representation ensures a good estimation of the first derivatives at the points touching the missing segment. Applying smoothing conditions, we manage to construct a new spline that joins smoothly with the received boundary, leading to successful concealment results. Experimental results on object shapes with different concealment difficulty demonstrate the performance of the proposed method. Comparisons with prior proposed methods are also presented.

Index Terms—COM-ERC, error concealment, shape coding, T-splines.

I. INTRODUCTION

IN WIRELESS networks and the Internet, transmitted information is subject to losses. As retransmission of lost or damaged packets may incur delay, error resiliency methods have been developed to detect and correct transmission errors [1]–[3]. Post processing error concealment includes estimation of the lost information by making use of the inherent correlation among spatially and temporarily adjacent samples.

In MPEG-4 Part-2, a video coder is composed of two parts: the shape coder and the motion and texture coder. Shape, motion, and texture information can be encoded and transmitted separately. Due to the encoding of arbitrary shape video objects (VOs), shape information is critical for the representation of a

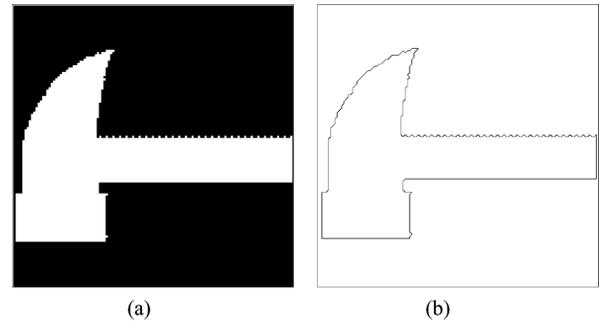


Fig. 1. (a) Binary and (b) contour representation of an object shape (a) *hammer* α -plane and (b) *hammer* boundary.

VO. If only texture is lost, shape and motion can be tapped to conceal texture, while if shape/motion is lost, the whole packet is discarded. For these reasons shape error concealment techniques are of great importance.

Shape error concealment can be achieved by exploiting shape information from the current or previous video frames. Three categories of techniques have been developed. Temporal techniques [4]–[6], which are based on motion compensation and exploit inter-frame correlation; they perform poorly when objects appear/disappear or are deformed. Spatial techniques, which use information from the neighboring to the lost part area and, besides video frames, they can also be applied to still images. They rely on object shape statistics [7], combination of image and shape data [8], or employ a geometric shape representation [9]–[13]. Techniques combining both temporal and spatial methods are referred to as spatio-temporal techniques [14], [15].

In this paper, we propose a spatial error concealment technique based on a contour representation of the object shape. Let us consider a binary shape representation like the one presented in Fig. 1(a), which represents the α -plane of the object *hammer*. From it we can extract a contour corresponding to the border of the object, i.e., the boundary of its texture, as shown in Fig. 1(b). Channel errors may result in a corrupted α -plane, where one or more blocks of binary information are missing [see Fig. 2(a)], yielding a broken boundary [see Fig. 2(b)]. Error concealment includes the construction of a new curve that successfully replaces the missing boundary parts and joins smoothly with the received parts. The existing geometric concealment approaches build a polynomial concealment curve based on conditions that arise from smoothness requirements between the new curve and the curve representing the received boundary points, i.e., the boundary modeling curve [see Fig. 2(c)].

Manuscript received October 27, 2011; revised February 09, 2012; accepted February 09, 2012. Date of publication April 03, 2012; date of current version July 12, 2012. This work was supported by a Marie Curie International Reintegration Grant within the 7th European Community Framework Programme.

E. Tsiligianni and L. P. Kondi are with the Department of Computer Science, University of Ioannina, Ioannina 45110, Greece (e-mail: etsiligia@cs.uoi.gr; lkond@cs.uoi.gr).

A. K. Katsaggelos is with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: aggk@eecs.northwestern.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2012.2192850

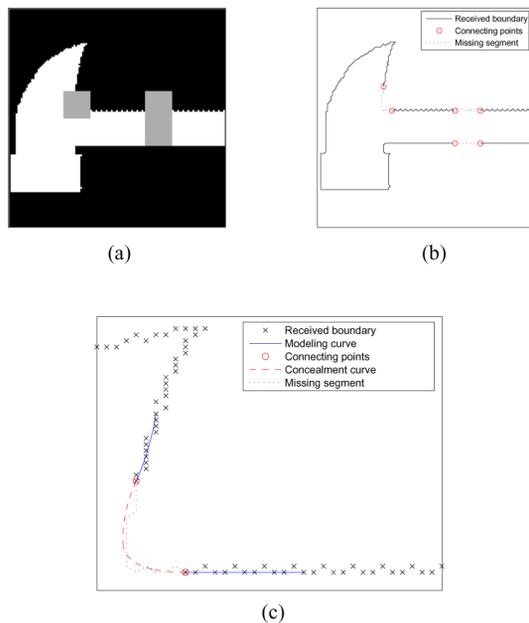


Fig. 2. Error concealment for a broken boundary. (a) A corrupted α -plane. (b) A broken boundary. (c) Boundary modeling and error concealment.

A boundary error concealment method based on first degree line segments appeared in [9]. Higher order curves are utilized in [10], [11] and [12]. In [12], quadratic Bezier curves are used for concealment. However, these curves can only represent convex or concave boundaries. In [10] and [11], the authors use second and third degree polynomials to model the received and the lost boundary, respectively, yielding better results. The missing boundary is concealed by a Hermite cubic polynomial that joins smoothly with the original boundary. The polynomial coefficients are specified by requiring first-order (C^1) continuity between the concealment curve and a quadratic polynomial approximation representing the original boundary. The approximation is obtained using least squares in a boundary part consisting of 20 boundary points on each side of the missing segment. The estimation of the first derivative at the joining points is based on this quadratic approximation. A similar reconstruction based on cubic Bezier curves is proposed in [13]. A cubic polynomial approximation is used for the representation of the received parts. The approximation algorithm uses an iterative process in order to select an appropriate boundary part leading to a minimum approximation error. For concealment, a cubic polynomial is constructed using the form of Bezier curves, requiring the tangent vectors at the joining points to coincide.

The existing solutions basically differ in the degree of the polynomials proposed to represent the received and the lost part of the boundary. The above methods fail when either the received or the missing boundary part is not characterized by quadratic or cubic behavior, which is common in natural boundaries. Moreover, their success mainly depends on the estimation of the first derivatives at the points where the received boundary joins with the new curve. As the estimation is derived from the received boundary approximation, the concealment result is determined by the quality of the approximation. However, even a small approximation error of the modeling curve may result in

large estimation errors if the approximation fails to capture the shape of the original boundary.

In this paper, we overcome these limitations using splines. Spline curves can represent complex boundaries better than simple polynomial functions. Due to their simplicity of construction and accuracy of evaluation, they have been extensively used for shape representation in various fields such as shape based image retrieval, image processing and computer-aided design (CAD) [16]–[20]. Here, our goal is to construct a shape-preserving modeling curve that can provide a good estimation of the first derivative, which is crucial for the error concealment. An appropriate representation should not introduce changes in the boundary slope. If the modeling curve preserves the monotonicity of the original boundary, changes in the boundary slope can be avoided. For this reason, we use monotone least squares splines to model the received boundary. Thus, we get a reliable first derivative estimation enabling us to construct a natural concealment spline that joins smoothly with the original boundary. An early version of our work appears in [21].

The rest of this paper is organized as follows. In Section II, the shape error concealment problem is defined. The proposed boundary modeling and concealment methods are described in Sections III and IV, respectively. In Section V, error concealment is applied to a boundary encoded with B-splines. In Section VI, we discuss error concealment for the case of more than one missing segments and we summarize the error concealment process. In Section VII, experimental results are presented. Finally, in Section VIII, conclusions are drawn.

II. PROBLEM FORMULATION

In this paper, we use a geometric description of the object shape, that is, its boundary. Depending on the shape coding technique, the object boundary can be represented either in a binary form (bitmap-based coding) or in a contour form (contour-based coding). In this section, we present the binary shape coding scheme used in the MPEG-4 video standard. We will discuss a contour-based encoding scheme in Section V.

The basic unit of coding in MPEG-4 is the video object plane (VOP) [22], [23]. A binary shape is described by the binary α -plane (see Fig. 1). Pixels of a VOP belonging to the object are assigned an α -value equal to 1, whereas pixels belonging to the background an α -value equal to 0. We define the boundary of an α -plane as the collection of points belonging to the background, which have at least one four-connect neighbor (that is, with pixels above, below, to the left and to the right) that belongs to the object (that is, their value is 1 in the α -plane). Fig. 1(b) shows the boundary as defined for the α -plane of Fig. 1(a). The α -plane is divided into small blocks of information, namely the macroblocks consisting of 16×16 pixels. Depending on the packetization scheme, one packet contains information corresponding to one or more macroblocks. A packet loss can result to one or more missing boundary segments (see Fig. 2). We refer to the points on the received boundary that touch a missing segment as “connecting points”. We also refer to the boundary slope at each connecting point as “boundary direction”. An example is shown in Fig. 2(b) where missing segments are represented

by a dotted line. The received boundary is represented by a solid line, while the connecting points as open circles.

In shape error concealment, a missing segment of a broken boundary is concealed by a new curve that joins smoothly with the received part. As splines have been used to represent natural boundaries efficiently [19], they are expected to provide accurate results as concealment curves. Smoothness plays an important role in the visual outcome of the concealment and can be achieved by requiring some degree of continuity between the concealment spline and a curve representing the received boundary. Hence, before proceeding to the construction of a concealment spline, it is necessary to find a representation of the received boundary.

The proposed shape error concealment technique consists of two steps. First, we solve the problem of modeling the received boundary. Second, we exploit the information given by the modeling curve to construct a curve representing the missing segment [see Fig. 2(c)]. In order to achieve a smooth transition between the two curves, restrictions must be applied. We require the new curve to pass through the connecting points and the new curve direction to coincide with the received boundary direction. The first derivative of the concealment curve and the estimated first derivative of the received boundary at each connecting point must be equal so as C^1 continuity is achieved. Thus, the received boundary direction estimation is crucial for a successful solution to our problem.

In the rest of this paper we assume that the boundary is a closed non-intersecting curve. Considering a video frame, this assumption results in one concealing boundary out of several possible ones and in non-intersecting concealing curves. If the above assumption is not valid, we must know how many boundaries we have to conceal and how many points they have in common. Then the proposed method can be applied to every boundary separately.

III. RECEIVED BOUNDARY MODELING

Assuming that a boundary consists of an ordered set of points, we can obtain a suitable representation of the known boundary by solving a curve fitting problem. The existing methods propose a polynomial boundary representation using least squares. Second order polynomials have been used in [12] for the received and the missing segment representation. As these curves are either convex or concave, they do not lead to an appropriate representation, if the boundary is neither. In [10], a piecewise polynomial curve is constructed by smoothly joining a quadratic polynomial curve, used for the received boundary modeling, with a cubic polynomial curve, used for the missing segment representation. While this method results in a more appropriate concealment curve compared to [12], the modeling curve is still convex or concave. Considering that a natural boundary may have a complex form, we propose piecewise polynomial curves to represent the received boundary and the missing segment as well.

Regarding the received boundary representation, least squares splines could produce an acceptable model. However, splines' ability to represent complex forms is not sufficient for a successful approximation. We need a solution that can capture

the original shape and preserve fundamental characteristics of the original boundary. Traditional B-splines cannot produce a shape-preserving approximation and may introduce major changes in the boundary direction. As the concealment curve must satisfy at least a C^1 continuity condition, an erroneous estimation of the boundary direction at a connecting point may lead to concealment failure.

In the existing error concealment methods, an erroneous direction estimation is frequently related to changes in the boundary monotonicity introduced by the approximation. Given a set of boundary points $S_B = (Q_0, Q_1, \dots, Q_{N_B})$, $Q_i = (x_i, y_i)$, $i = 0, \dots, N_B$, we define boundary monotonicity as

- (i) increasing, if $x_i \leq x_{i+1}$ implies
 $y_i \leq y_{i+1}$, $i = 0, \dots, N_B$
- (ii) decreasing, if $x_i \leq x_{i+1}$ implies
 $y_i \geq y_{i+1}$, $i = 0, \dots, N_B$.

Considering the importance of the direction estimation to error concealment results, we employ monotone least squares splines to model the received boundary, so as the modeling curve preserves original monotonicity and provides successful estimation of the boundary direction.

A. Least Squares Spline Approximation

Let S_B be an ordered set of points representing a natural boundary, $S_B = (Q_0, Q_1, \dots, Q_{N_B})$, $Q_i = (x_i, y_i)$, $i = 0, \dots, N_B$. A least squares spline approximation is a piecewise polynomial $C(\cdot)$

$$C(u) = \sum_{i=0}^n b_i \cdot f_i(u), \quad u \in [0, 1] \quad (1)$$

which forms a solution to the problem

$$\min_{b_i \in \mathbb{R}^2} \sum_{k=0}^{N_B} [Q_k - C(\bar{u}_k)]^2 \quad (2)$$

where $f_i(\cdot)$ are appropriate basis functions, $b_i = (x_i, y_i)$, $i = 0, \dots, n$ are the unknown spline coefficients or approximation control points and \bar{u}_k , $k = 0, \dots, N_B$ are parametric values affecting the form of the curve.

The quality of a spline approximation depends on the type and the degree p of the basis functions, the number of the spline segments and the parameterization of the curve. If C^r continuity is required then p should satisfy

$$p \geq r + 1.$$

The parameterization of the curve is related to the values of \bar{u}_k , $k = 0, \dots, N_B$, and the position of the approximation *knots*. The knots form the *knot vector* $U = (u_0, u_1, \dots, u_m)$ and divide the parameterization space into the knot spans. They are involved in the computation of the basis functions and determine the way the control points affect the curve. The curve continuity depends on the knot multiplicity as well [16]. For a given type

and degree of basis functions, the number of the spline segments is determined by the number of the control points.

In this paper we assume that $u \in [0, 1]$ and we use the ‘‘chord length’’ method [16] to calculate the knot vector and the parametric values \bar{u}_k . Let d be the ‘‘chord length’’ given by the data points Q_i , $i = 0, \dots, N_B$, that is

$$d = \sum_{k=1}^{N_B} |Q_k - Q_{k-1}| \quad (3)$$

where $|Q_k - Q_{k-1}|$ is the Euclidean distance between Q_k and Q_{k-1} . Then we obtain the parametric values by

$$\bar{u}_k = \begin{cases} 0, & k = 0 \\ \bar{u}_{k-1} + \frac{|Q_k - Q_{k-1}|}{d}, & k = 1, \dots, N_B - 1 \\ 1, & k = N_B. \end{cases} \quad (4)$$

In order to get a p -degree curve, we need $n+p+2$ knots. $2(p+1)$ knots are positioned at the edges of $[0, 1]$. We obtain the internal $n-p$ ones according to

$$u_{j+p} = (1 - \alpha)\bar{u}_{j+p} + \alpha\bar{u}_i, \quad j = 1, \dots, n-p, \quad i = \text{int}(jD), \quad \alpha = jD - i \quad (5)$$

where

$$D = \frac{N_B + 1}{n - p + 1}$$

and $\text{int}(\cdot)$ denotes the integer part of the argument. As far as the degree of the curve is concerned, a quadratic spline is a choice that can meet our continuity requirements while keeping the computational cost low. In the following, we present a discussion of the basis functions type and the number of spline segments constituting the approximation.

B. Boundary Approximation Using B-Splines

B-spline basis functions have the minimal support with respect to a given degree and smoothness and can be evaluated in a numerically stable way by the de Boor algorithm [24]. These reasons have made B-splines popular in shape modeling [25]–[29] and shape coding [18]–[20], [30]–[34]. However, if we use B-splines basis functions in (2), we obtain solutions that cannot preserve essential boundary properties like monotonicity and inflection points, yielding changes in the boundary direction.

Fig. 3 illustrates the approximation of a boundary part with a least squares quadratic B-spline curve. Although it seems that the approximation forms a good representation of the original boundary, focusing at the connecting point, we detect a wrong direction estimation. The curve is decreasing in contrast to the original increasing monotonicity. The representation is compared to a least squares quadratic T-spline approximation that preserves monotonicity.

C. Monotone Boundary Approximation Using T-Splines

Changes in monotonicity imply changes in the boundary direction. An approximation that can provide a good estimation of the boundary direction must preserve the original boundary

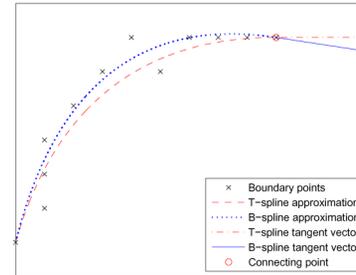


Fig. 3. Spline approximation and estimation of the tangent vector at the connecting point, where the boundary slope is zero. T-spline approximation provides the right slope estimation as it preserves boundary monotonicity. B-spline approximation results in a wrong (negative) slope estimation.

monotonicity. In [35], Beliakov proposed a simple way of producing a monotone least squares spline by selecting T-splines as basis functions and imposing linear inequality restrictions on spline coefficients. The linear least squares problem becomes a non-negative least squares problem; robust methods exist for the effective solution of such problems.

Definition of T-Splines: Given a knot vector $U = (u_0, u_1, \dots, u_m)$, $u_0 \leq u_1 \leq \dots \leq u_m$, trapezoidal or T-splines functions, of degree p , are defined as a sum of B-splines basis functions

$$T_{i,p}(u) = \sum_{j=i}^m N_{j,p}(u), \quad u \in [0, 1], \quad i = 0, \dots, m \quad (6)$$

where $N_{j,p}(\cdot)$ is the j th B-spline basis function, of degree p ($j = 0, \dots, m$). Therefore, the T-splines are linearly independent and form a basis of the space of p -degree piecewise polynomial functions [36]. A spline $C(\cdot)$ can be represented as

$$C(u) = \sum_{i=0}^n b_i \cdot T_{i,p}(u), \quad u \in [0, 1]. \quad (7)$$

1) *Monotone Spline Approximation:* According to [35], if we use second degree ($p = 2$) T-splines, we can construct an increasing quadratic approximation. The necessary and sufficient condition for monotonicity is $b_i \geq 0$, $i = 0, \dots, n$ [see (7)]. The linear least squares problem (2) under monotonicity condition becomes a non-negative least squares problem.

Moreover, in our problem we need the approximation to pass through the connecting points. Interpolating the first point is achieved by setting $b_0 = Q_0$. A practical way to force the approximation to pass through the last point, without making the above problem more complicated, is to assign a weight w to the last term of the sum in (2). A very large value for w may lead to trivial solutions. A value that is a little greater than the number of data points is an acceptable choice.

Under all restrictions above, (2) finally becomes

$$\min_{b_i \geq 0} \left[\sum_{k=1}^{N_B-1} (Q_k - C(\bar{u}_k))^2 + w \cdot (Q_{N_B} - C(\bar{u}_{N_B}))^2 \right]. \quad (8)$$

This is a weighted nonnegative least squares problem (weighted-NNLS) and can be solved using MATLAB’s subprogram *lsqnonneg*, which is a modification of Lawson and

Hanson approach [37], [38]. For a survey of other methods (Bro and Jong's Fast NNLS, Projected Quasi-Newton NNLS) the reader is referred to [39].

Having computed the control points b_i , we can use (7) to obtain the approximation curve $C(\cdot)$. However, for stable and effective calculations, B-splines basis functions are more appropriate than T-splines. Beliakov [35] suggests obtaining the approximation curve using B-splines instead of T-splines by converting the approximation T-spline coefficients b_i to B-spline coefficients P_i according to

$$P_i = \sum_{j=0}^i b_j, \quad i = 0, \dots, n. \quad (9)$$

Then, using quadratic B-spline basis functions $N_{i,2}(\cdot)$, we get the approximation curve $C(\cdot)$

$$C(u) = \sum_{i=0}^n P_i \cdot N_{i,2}(u), \quad u \in [0, 1]. \quad (10)$$

We have concluded the description of the approximation method proposed to represent an increasing boundary part with an increasing spline. For monotonically decreasing splines the results are analogous. The importance of preserving monotonicity can be seen in Fig. 3, where the T-spline approximation curve constructed with the above method leads to a good estimation of the boundary direction at the connecting point, which cannot be achieved by a B-spline approximation.

D. Shape-Preserving Boundary Approximation

1) *Selection of the Appropriate Boundary Part*: In order to apply the T-splines approximation method, we have to select a suitable boundary part. Starting from a connecting point, we search along the boundary to choose $N_B + 1$ consecutive points. $N_B + 1$ should be equal to the estimated number of lost points. We make a rough estimation of this number by calculating the length of the diagonal of the lost block. This value may be reduced as the selected part must satisfy the following conditions. First, we have to ensure that this part can be approximated by a function, i.e., to select points $Q_i = (x_i, y_i)$ that satisfy the condition $x_i \leq x_{i+1}$ or $x_i \geq x_{i+1}$, $\forall i = 0, \dots, N_B$. Second, we have to select a part of increasing or decreasing monotonicity. However, a minor variation in monotonicity is allowed as it does not affect the T-spline approximation results. Suppose we move along the boundary collecting points that satisfy a certain monotonicity condition (see Section III). If we meet a point that does not satisfy this condition and the change is not bigger than a threshold (e.g., 1% of the frame height), we "ignore" it and continue checking a few more of the following points. If they do not satisfy the initial monotonicity condition either, then we stop. Else, we keep collecting points ignoring the "minor" variation we have detected. In Fig. 3, the change in monotonicity introduced by the fourth point before the connecting point is ignored while selecting the boundary points.

2) *Approximation Algorithm*: Because of the free form of a natural boundary, it is difficult to predefine the complexity of an appropriate approximation, which is determined by the number of the spline segments or equivalently, by $n + 1$, the number of the control points b_i [see (1)]. Increasing complexity

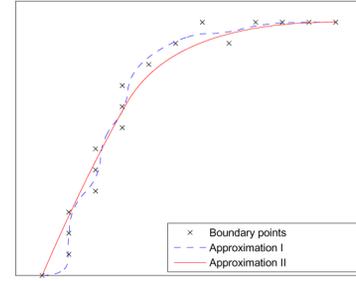


Fig. 4. Least squares T-spline approximations for a given data set. The number of the control points used to construct the first curve (*Approximation I*) is equal to 3/4 of the number of data points. Although the large number of the control points results in a small approximation error, the approximation "wiggles" through data points failing to capture the boundary direction. By reducing the number of the approximation control points to 1/4 of the number of data points, we get a relative coarse approximation (*Approximation II*) that provides a better direction estimation.

may improve the accuracy of the representation, by means of the approximation error. However, large values of n produce solutions that "wiggle" through boundary points, introducing major changes in the boundary direction (see Fig. 4). Even though preserving monotonicity reduces "wiggling" compared to a B-spline approximation, experimental results show that small values of n lead to better modeling curves.

Considering that our approximation should capture the shape of the data without lacking accuracy, we will try to specify appropriate thresholds for the approximation error and the number of the control points. We define the approximation error as the maximum value of the smaller Euclidean distance between the approximation curve and the boundary points. An acceptable approximation should not exceed a specified error threshold. In order to satisfy this condition we may need to execute a few iterations of the approximation method, in which we gradually increase n . Starting from a minimum value of $n = 2$, we propose to keep the number of the approximation control points smaller than half the number of the boundary points. If we still cannot obtain an acceptable approximation, we gradually shorten the selected boundary part. By removing boundary points we get a simpler boundary form, easier to approximate.

The curve complexity should be treated with the specified error threshold. Both parameters affect the approximation quality in a similar way. For the proposed values of n , experiments have shown that 1 pixel is an appropriate error threshold. Algorithm 1 summarizes the steps needed for boundary approximation.

Algorithm 1 Received boundary modeling

- 1: select an appropriate boundary part
- 2: initialize the number of the approximation control points
($n + 1$): $n = 2$
- 3: find an approximation curve using T-splines
- 4: calculate the approximation error
- 5: **if** approximation error > error threshold **then**
- 6: **if** $n < \lfloor 1/2 \text{ length of boundary part} \rfloor - 1$ **then**
- 7: increase the number of control points: $n = n + 1$
 and go to step 3
- 8: **else**

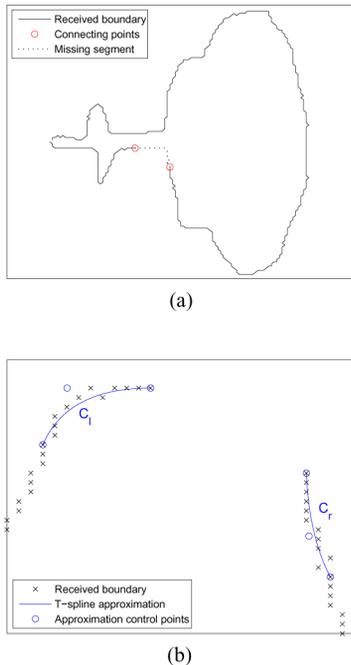


Fig. 5. First step of the proposed error concealment method. We construct a T-spline approximation of the received boundary, on each side of the missing segment. (a) A broken *fountain* boundary. (b) Boundary modeling curves.

```

9:   reduce the length of the selected boundary part and
      go to step 2
10:  end if
11:  else
12:   return the approximation control points
13:  end if

```

Fig. 5 illustrates the results of the boundary modeling step. The broken boundary of the object *fountain* is illustrated in Fig. 5(a). The T-spline approximation method is applied to an appropriate set of boundary points, automatically selected, so as the approximation curves preserve the original boundary characteristics [see Fig. 5(b)].

IV. CONCEALMENT CURVE CONSTRUCTION

We will use the example of Fig. 5 to present the construction of the proposed concealment curve. Suppose we have obtained the left boundary approximation, C_l , described by the control points (P_l^0, \dots, P_l^n) . We recall that C_l passes through P_l^n , the left connecting point. Similarly, the right boundary part is approximated by C_r , described by (P_r^0, \dots, P_r^m) (see Fig. 6). Due to smoothing restrictions that have already been discussed in Section II, the concealment curve should pass through the connecting points, P_l^n and P_r^0 , and match the corresponding boundary tangent lines. Therefore, the concealment curve should satisfy four conditions, two at each connecting point. This information could be used to produce various types of curves such as polynomials of third degree [10], cubic Bezier [13] or spline curves. As splines can represent natural boundaries better than polynomials, we expect them to give better concealment results. Four control points are sufficient to

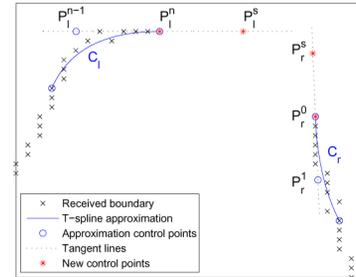


Fig. 6. Determination of the new control points. Based on the T-spline approximation of the received boundary, we estimate the tangent lines at the connecting points, on each side of the missing segment. We locate the new control points on the estimated tangent lines.

construct a quadratic B-spline and will be obtained in the way proposed in [13]. The experiments we performed to test our choice against alternative solutions have shown that it yields the best concealment.

A. New Control Points

As we have seen in Section III, a spline curve, C , is determined by the control points (P_0, \dots, P_q) and the knot vector $U = (u_0, \dots, u_k)$, $0 \leq u_0 < \dots < u_k \leq 1$. In [16] the first derivatives at the end points of a B-spline curve are given by

$$C'(0) = \frac{p}{u_{p+1}}(P_1 - P_0) \quad (11)$$

$$C'(1) = \frac{p}{1 - u_{k-p-1}}(P^q - P^{q-1}) \quad (12)$$

with p the degree of the spline. The above equations imply that the tangent line at the end point, $u = 0$, passes through the second control point, P_1 , and the tangent line at the end point, $u = 1$, passes through the control point P^{q-1} , that is the one before the last control point of the curve.

Fig. 6 illustrates the tangent lines at the connecting points of the *fountain* broken boundary. Assume that the left boundary approximation, C_l , is a B-spline curve determined by the control points (P_l^0, \dots, P_l^n) . Then the tangent line at the left connecting point, P_l^n , passes through the previous control point, P_l^{n-1} . Similarly, if the right boundary approximation, C_r , is described by the control points (P_r^0, \dots, P_r^m) , the tangent line at the right connecting point, P_r^0 , passes through the next control point P_r^1 .

Let C be the new concealment spline described by the control points (P_0, \dots, P_q) and the knot vector $U = (u_0, \dots, u_k)$, $0 \leq u_0 < \dots < u_k \leq 1$, with $P_0 = P_l^n$ and $P_q = P_r^0$. Then (11) gives the tangent of the new curve at the left connecting point $P_0 = P_l^n$. We specify a new control point, P_l^s , on the tangent line of C_l , which is symmetric to P_l^{n-1} with respect to P_l^n (see Fig. 6). We consider P_l^s as the second control point of C , i.e., $P_1 = P_l^s$. Consequently, the tangents of the approximation and the concealment curve at the left connecting point, P_l^n , coincide and C^1 continuity between the two curves is achieved.

Similarly, we obtain another control point, P_r^s , at the right boundary part. Using the new control points we can produce a new spline having the same tangent with the approximation curve at each connecting point. The actual value of the corresponding first derivative is not needed.

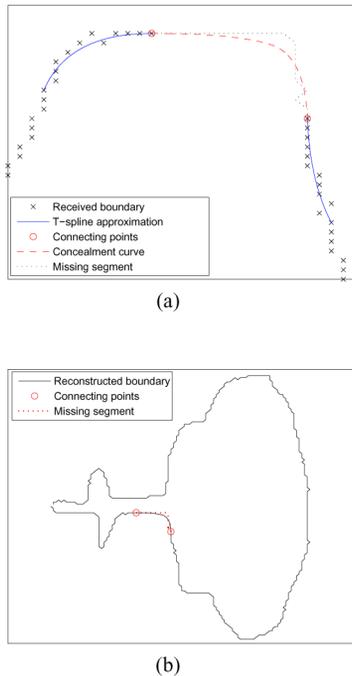


Fig. 7. New control points determined in Fig. 6 produce a B-spline concealment curve that joins smoothly with the boundary modeling curves. The broken boundary is reconstructed successfully. (a) Boundary approximation and error concealment curves. (b) Reconstructed *fountain* boundary.

B. Concealment Curve

The connecting points and the new control points constitute a set of four control points $(P_0, P_1, P_2, P_3) = (P_l^n, P_l^s, P_r^s, P_r^0)$. Four control points can give a quadratic B-spline as the concealment curve according to

$$C(u) = \sum_{i=0}^3 P_i \cdot N_{i,2}(u), \quad 0 \leq u \leq 1 \quad (13)$$

with $N_{i,2}(\cdot)$ the i th quadratic B-spline basis function. The new curve passes through the first and the last control points, i.e., the connecting points, and joins smoothly with the boundary approximation curves. The proposed quadratic B-spline solution can represent more complex boundary forms compared to a second order polynomial and has a more natural appearance compared to a cubic polynomial. Besides smoothness, the method used to find the new control points targets at preserving the control polygon of the received boundary. A B-spline passes closer to new control points compared to a cubic Bezier, thus, it preserves the original boundary with greater success. Fig. 7 illustrates the modeling and concealment curves and the reconstructed boundary.

In Fig. 8 we present another example of the proposed method and compare it to the method of [13]. A broken boundary (*fork*) is concealed with B-splines and with cubic Bezier Fig. 8(b). The details of the approximation and the determination of the new control points are illustrated in Fig. 8(c) and (d). The length of the received boundary selected to be modeled is determined by the proposed approximation algorithm. In both cases, the specified error threshold is 1 pixel. Apparently, T-splines provide

an approximation that leads to a better direction estimation of the received boundary. Smoothness is achieved by locating the new control points on the estimated tangent lines. The number and the position of the control points used from the proposed method depend on the form of the received boundary, whereas the method of [13] always uses four control points. However, the number of the approximation control points affects the distance between them and, consequently, the distance between each symmetric control point and the corresponding connecting point. Thus, the proposed approximation curve reflects the original boundary complexity by keeping the control points close to each other, a property that is transferred to the new control points [see Fig. 8(c)]. On the other hand, the large distance between the four control points of the cubic Bezier curve is unavoidable in order to represent a complex boundary [see Fig. 8(d)]. The impact on the concealment curve is obvious.

V. ERROR CONCEALMENT FOR A BOUNDARY ENCODED WITH B-SPLINES

The previous discussion applies to the error concealment of any boundary, regardless of how it was encoded. In many shape encoding schemes, the object boundary is represented by a spline approximation [31]–[34]. The representation introduces distortion; however, if the distortion is small, the approximation can be considered a shape-preserving representation of the original boundary. In such a case, the boundary modeling step is not necessary for error concealment. The received information consisting of the approximation control points is sufficient to apply the proposed concealment step.

In [31] a shape coding method is proposed, based on a boundary approximation that uses quadratic B-splines. The approximation lies inside a distortion band along the original boundary. The control points describing the B-spline curve are encoded and transmitted. The distortion band width defines the approximation quality. If the width is kept small, the approximation curve does not introduce changes in boundary direction.

The example of Fig. 9 illustrates error concealment for a B-spline encoded boundary. We assume that some control points have been lost during transmission [see Fig. 9(a)]. As we consider differential encoding the successful decoding of every control point depends on the successful decoding of the previous control points. If some of the control points are lost, concealment is feasible only if some of the transmitted control points are encoded directly.

In order to apply the proposed concealment method, we can use the received control points, representing the boundary parts on each side of the missing segment, to estimate the tangent lines at each connecting point. Fig. 9(b) illustrates the new control points determined in the way described in Section IV-A and the quadratic B-spline concealment curve constructed according to (13). The reconstructed boundary is shown in Fig. 9(c).

VI. MULTIPLE MISSING SEGMENTS

A packet loss during transmission may result in one or more lost macroblocks. Moreover, a lost block of shape information, regardless its size, may yield more than one missing boundary

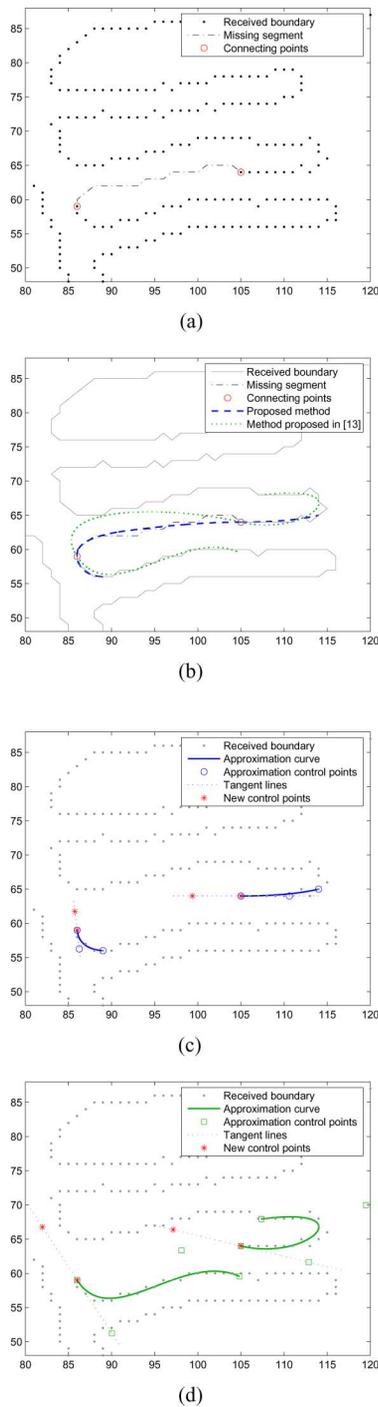


Fig. 8. Error concealment with the proposed method and with cubic Bezier curves [13] for *fork*. The corrupted boundary shown in (a) is concealed in (b) using the proposed method and the method proposed in [13]. In (c) and (d) detailed results of the approximation of the received boundary and the determination of the new control points are shown for every method. At each connecting point the tangents of the received boundary are estimated. The new control points, located on the tangents, determine the form of the concealment curve. Thus, the proposed method leads to better concealment results in (b), due to successful tangent estimation and location of the new control points in (c), a goal that is not achieved with great success by the cubic Bezier curves in (d). (a) Corrupted *fork*. (b) Approximation and concealment comparison. (c) Proposed method. (d) Cubic Bezier curves [13].

segments (see Fig. 10). Before we apply the proposed error concealment method, we should take into consideration that we may need to conceal more than one missing boundary segment.

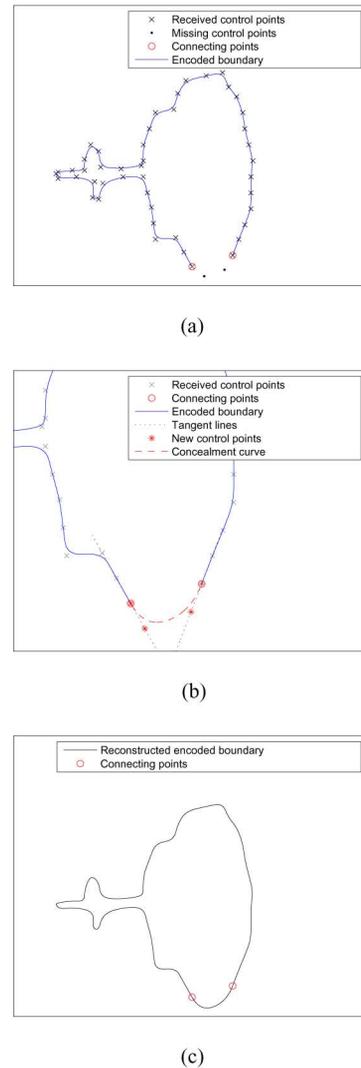


Fig. 9. Proposed error concealment method for a boundary encoded with B-splines according to [31]. Given the received control points, we estimate the tangent lines at the connecting points. We locate the new control points on the estimated tangent lines and use them to construct the B-spline concealment curve. (a) Corrupted encoded *fountain*. (b) Error concealment. (c) Reconstructed encoded *fountain*.

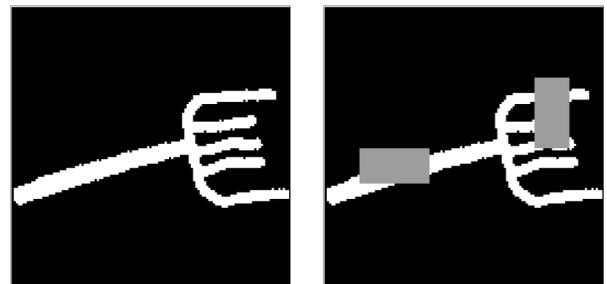
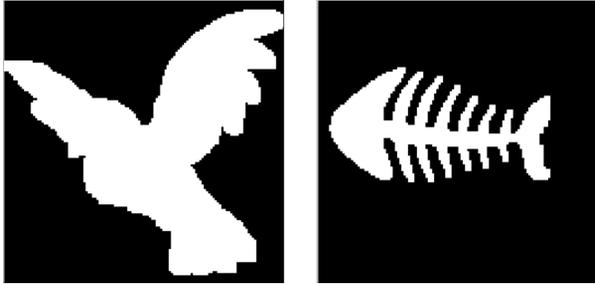


Fig. 10. Multiple missing segments for *fork*.

First, if more than one macroblock are lost, we need to group lost macroblocks together. Four-connected macroblocks constitute a lost group of shape information that has to be treated in a unifying way. Second, if the lost information yields multiple missing segments, we need to apply error concealment for every missing segment separately. As we have already mentioned, a

Fig. 11. Original *bird* and *fishbone*.

missing segment is defined by the start and the end point, that is the connecting points. In order to find pairs of connecting points, we use the algorithm proposed in [10], which is based on the assumption that the boundary is a closed non-intersecting curve. Since every line that goes into a lost block must come out of it, there is always an even number of connecting points. However, N connecting points lead to $N!$ possible pairs. The connecting points are coupled recursively, under the condition that no pair can lead to intersected straight line segments. The reader is referred to [10] for more details. After concealment splines construction, the above assumption is checked again, since non-intersecting straight line segments do not necessary lead to non-intersecting splines. Finally, when the concealed boundary is constructed, we can extract the corresponding α -plane by assigning the value 1 to the pixels that are in the interior of the boundary and 0 to the others (including the ones belonging to the boundary). Algorithm 2 summarizes the whole error concealment process.

Algorithm 2 Shape error concealment

- 1: find the received boundary of a binary encoded shape (α -plane)
 - 2: group 4-connected lost macroblocks together
 - 3: **for** every lost group **do**
 - 4: find pairs of connecting points according to [10]
 - 5: **for** every pair of connecting points **do**
 - 6: apply the proposed error concealment
 - 7: **end for**
 - 8: **end for**
 - 9: extract the reconstructed α -plane or calculate the B-spline encoding curve
-

VII. EXPERIMENTAL RESULTS

To test the proposed concealment method, a number of experiments were performed, some of which are presented here. Besides numerical results, examples showing the visual outcome are shown. In order to quantify the performance of the proposed method, we use a relative measure, the ratio D_n , of the number of different pixels in the original and reconstructed α -plane divided by the total number of object pixels in the original α -plane. This is a quality metric used in MPEG-4 to evaluate shape coding techniques. Other metrics such as recall, precision, and distortion were also utilized in our experiments but

TABLE I
NUMERICAL RESULTS IN CASE OF 1 LOST MB

	D_n	fork	fishbone	bird	fountain	hammer
Proposed	avg	0.020	0.015	0.003	0.003	0.003
	low	0.002	0.001	0.000	0.000	0.000
	high	0.049	0.041	0.019	0.011	0.006
[13]	avg	0.022	0.016	0.004	0.003	0.004
	low	0.001	0.000	0.000	0.000	0.000
	high	0.069	0.042	0.019	0.011	0.015
[10]	avg	0.024	0.016	0.004	0.003	0.003
	low	0.003	0.000	0.000	0.000	0.000
	high	0.066	0.046	0.022	0.011	0.015
# of exper.		18	21	32	21	10

TABLE II
NUMERICAL RESULTS IN CASE OF 2 LOST MBs

	D_n	fork	fishbone	bird	fountain	hammer
Proposed	avg	0.056	0.041	0.009	0.010	0.011
	low	0.020	0.004	0.001	0.003	0.003
	high	0.084	0.078	0.040	0.045	0.037
[13]	avg	0.060	0.042	0.009	0.011	0.012
	low	0.024	0.007	0.001	0.004	0.001
	high	0.135	0.080	0.042	0.040	0.038
[10]	avg	0.062	0.044	0.009	0.011	0.010
	low	0.018	0.005	0.000	0.003	0.001
	high	0.110	0.084	0.044	0.042	0.038
# of exper.		20	24	36	14	12

TABLE III
NUMERICAL RESULTS IN CASE OF A B-SPLINE ENCODED BOUNDARY

	D_n	fork	fishbone	bird	fountain	hammer
Proposed	avg	0.036	0.031	0.004	0.011	0.010
	low	0.005	0.017	0.000	0.006	0.000
	high	0.090	0.042	0.010	0.024	0.024
[13]	avg	0.040	0.032	0.005	0.012	0.011
	low	0.006	0.021	0.000	0.003	0.000
	high	0.090	0.047	0.011	0.034	0.036
[10]	avg	0.036	0.011	0.005	0.012	0.011
	low	0.009	0.002	0.001	0.004	0.004
	high	0.092	0.028	0.011	0.022	0.023
# of exper.		18	29	15	13	12

they did not accurately depict the differences between the proposed and existing methods.

Our experiments are divided into two categories. First, we assume that the decoder receives a corrupted α -plane. The tested shapes are described by an α -plane of 128×128 pixels and the missing information block consists of either one macroblock (16×16 pixels) or two neighboring macroblocks (16×32 or 32×16 pixels). All possible loss patterns that lead to a broken boundary have been considered. We have also tried loss patterns with larger groups of lost macroblocks. However, they do not express the differences among the methods under testing, so they are not presented here. Subsequently, we consider a boundary encoded with B-splines, as described in Section V. Here, we assume that some control points are lost during transmission and error concealment is applied. After boundary reconstruction, the corresponding α -plane is extracted. The concealed α -plane is compared to the original. Pixels that are assigned 1(0)

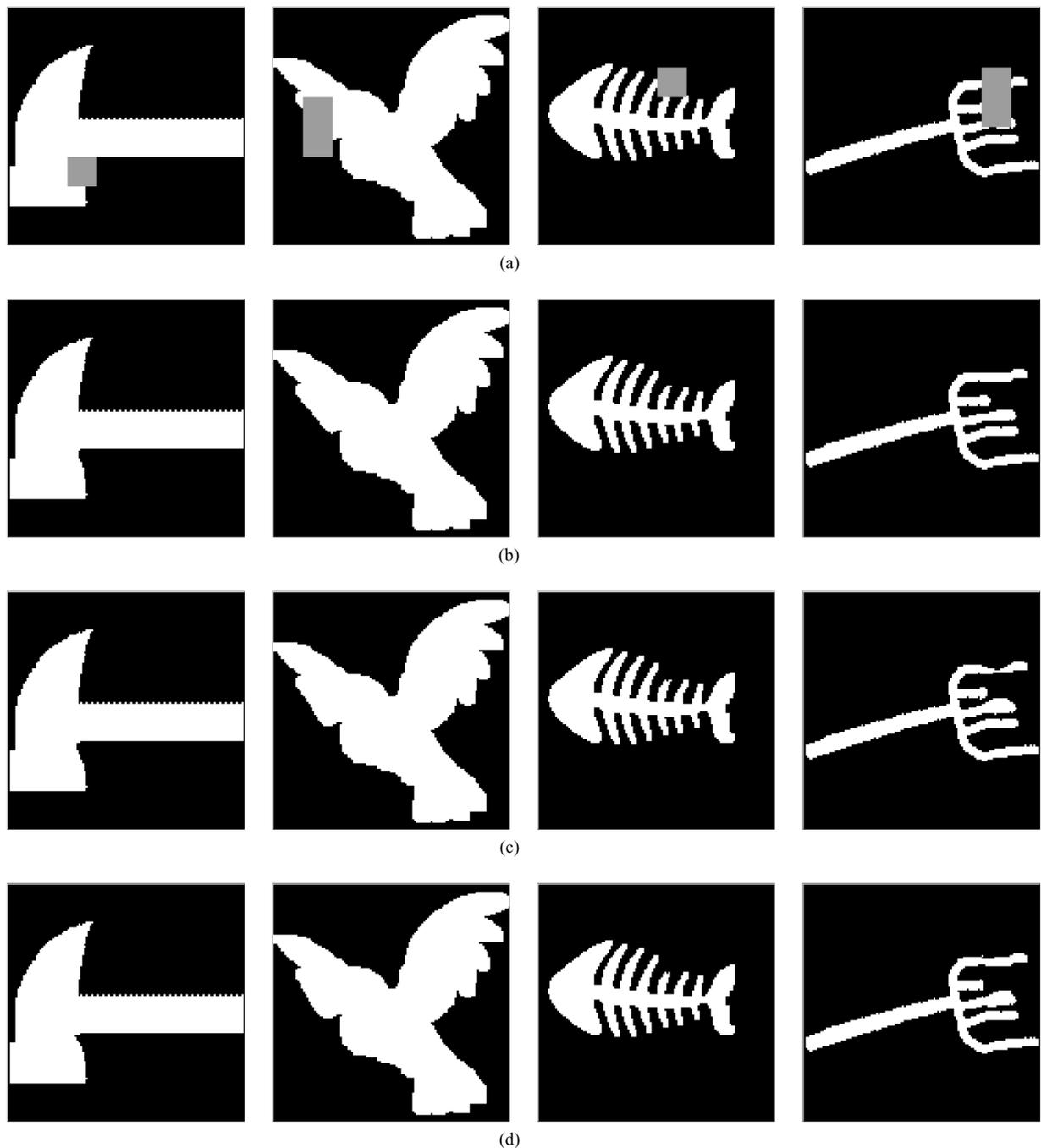


Fig. 12. Error concealment visual results. (a) Corrupted *hammer*, *bird*, *fishbone*, and *fork*. (b) Concealment with the proposed method. (c) Concealment with [13]. (d) Concealment with [10].

in the new α -plane while their value is 0(1) in the original are considered erroneous.

In our experiments we used object shapes with different smoothness level and concealing difficulty, namely *hammer*, *fountain*, *bird*, *fishbone*, and *fork* as shown in Figs. 1 and 5, Figs. 10 and 11, respectively. Besides the proposed error concealment method, we also applied the methods of [10] and [13]. Tables I and II show the average D_n values associated with every object. We also show the low and high of D_n values observed for every object, corresponding to the best and worst concealment results.

As can be seen in both tables, as far as the proposed method is concerned, only a small percentage of the reconstructed object pixels differs from the original ones. In most cases, such small differences are hardly visible. Comparing the proposed method to [13] or [10], in Table I, we can see that the proposed method gives better average D_n values for three of the five shapes, namely *fork*, *fishbone* and *bird* and similar for the remaining. The proposed method also yields better results as far as the worst (high) D_n values are concerned. The difference between the proposed and the existing methods is more obvious in the case of a less smooth boundary like *fork*. This is explained

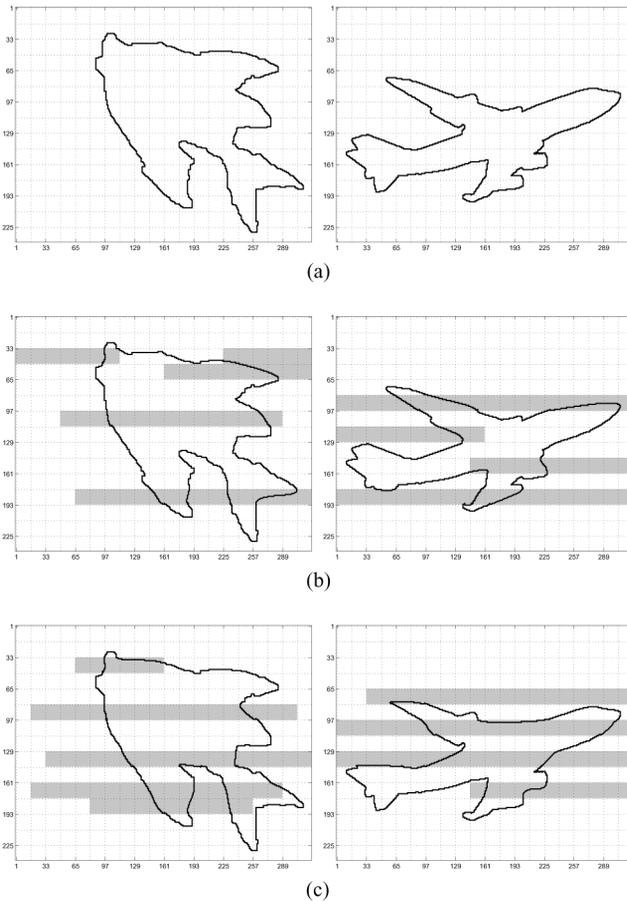


Fig. 13. Realistic error concealment scenario for 320X240 video objects. The reconstructed boundary is drawn on the top of the missing MBs. (a) Original boundary for *fish* and *airplane*. (b) First loss scenario and reconstructed boundary. (c) Second loss scenario and reconstructed boundary.

by the fact that a quadratic [10] or a cubic [13] curve can be effective in modeling smooth boundaries; however, for complex boundaries a spline curve is more appropriate.

In Table II, we demonstrate the results for the case where two neighboring macroblocks of the transmitted α -plane are lost. Here, we expect larger D_n values compared to Table I. Increasing the lost information block results in larger difficulties in error concealment and leads to smaller differences among the methods tested. Still, the proposed method yields better average results for three of the five shapes and smaller worst D_n values for most of the shapes under testing.

In the second category of experiments we assume that the original boundary was encoded using a quadratic B-spline approximation [31]. The spline lies in a band of one pixel width along the original boundary, therefore, it can be considered shape-preserving. Thus, as discussed in Section V, the computational complexity of the error concealment algorithm is reduced by not performing received boundary modeling. The number of the control points is approximately 1/10 of the total number of the original boundary points. Moving along the boundary, we assume that two consecutive control points are lost during transmission, which results in a missing segment consisting of 30 original points. Spatial concealment in case

of more lost control points is not possible, as it introduces great distortion. We apply error concealment using the received control points as described in Section V. The low average D_n values demonstrated in Table III indicate successful concealment for the proposed method. Moreover, compared to [10] and [13], our method yields better results for most of the tested shapes.

Obviously, numerical results cannot express the subjective impact of the reconstructed boundary. Besides the example in Fig. 8, we illustrate some more examples for which we construct the concealed α -planes in Fig. 12. The proposed method yields natural and pleasant visual results. In most examples a small distortion can be noticeable only if we compare the reconstructed α -plane to the original one. It is clear that the error concealment preserves the original boundary form. This is not true when the concealment is based on [10] or [13]. The results may sometimes look awkward as these methods fail in capturing the original shape.

In the above experiments we have chosen small objects in order to reduce evaluation time. Towards a more realistic error concealment scenario, in Fig. 13, we also present some visual results for 320×240 frames considering several missing macroblocks. Fig. 13(a) shows the original boundary for *fish* and *airplane*. Fig. 13(b)–(c) illustrate the loss scenarios; the lost macroblocks are represented as grey shadowed blocks and the reconstructed boundary is drawn on top of them.

VIII. CONCLUSION

Motivated by splines' ability to model natural forms successfully, we develop a new error concealment method based on a geometric representation of a VO shape. Considering a broken boundary, the construction of a quadratic B-spline curve is proposed to replace a missing segment. The new curve is obtained by applying smoothness conditions at the connecting points touching the missing segment. We use T-splines to produce an approximation of the received boundary, aiming at a representation that preserves the original boundary characteristics and provides a reliable estimation of the direction at the connecting points. At these points, the original boundary approximation and the concealment curve have the same direction. Under these conditions, we construct a concealment curve yielding better objective and subjective results than the current state of the art.

ACKNOWLEDGMENT

The authors would like to thank Prof. G. Schuster for providing them with the code for implementing shape error concealment according to [10].

REFERENCES

- [1] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Process. Mag.*, vol. 17, no. 4, pp. 61–82, Jul. 2000.
- [2] I. Moccagatta, S. Soudagra, J. Liang, and H. Chen, "Error resilient coding in JPEG-2000 and MPEG-4," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 899–914, Jun. 2000.
- [3] S. K. Kwon, A. Tamhankar, and K. R. Rao, "Overview of H.264/MPEG-4 part 10," *J. Visual Commun. Image Representation*, vol. 17, pp. 186–216, 2006.

- [4] G. M. Schuster and A. K. Katsaggelos, "Motion compensated shape error concealment," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 501–510, Feb. 2006.
- [5] L. D. Soares and F. Pereira, "Temporal shape error concealment by global motion compensation with local refinement," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1331–1348, Jun. 2006.
- [6] P. Salama and C. Huang, "Error concealment for shape coding," in *Proc. IEEE Int. Conf. Image Process.*, 2002, vol. 2, pp. 701–704.
- [7] S. Shirani, B. Erol, and F. Kossentini, "A concealment method for shape information in MPEG-4 coded video sequences," *IEEE Trans. Multimedia*, vol. 2, no. 3, pp. 185–190, Sep. 2000.
- [8] F. A. Sohel, G. C. Karmakar, and L. S. Dooley, "Image-dependent spatial shape-error concealment," in *Proc. IEEE Int. Conf. Signal Process.*, 2008, pp. 753–756.
- [9] X. Li, A. K. Katsaggelos, and G. M. Schuster, "A recursive shape error concealment algorithm," in *Proc. IEEE Int. Conf. Image Process.*, 2002, vol. 1, pp. 177–180.
- [10] G. M. Schuster, X. Li, and A. K. Katsaggelos, "Shape error concealment using Hermite splines," in *Proc. IEEE Trans. Image Process.*, 2004, vol. 13, pp. 808–820.
- [11] G. M. Schuster, X. Li, and A. K. Katsaggelos, "Spline based boundary loss concealment," in *Proc. IEEE Int. Conf. Image Process.*, 2003, vol. 3, pp. 671–674.
- [12] M. J. Chen, C. C. Cho, and M. C. Chi, "Spatial and temporal error concealment algorithms of shape information for MPEG-4 video," in *Proc. IEEE Int. Conf. Consumer Electron.*, 2002, pp. 224–225.
- [13] L. D. Soares and F. Pereira, "Spatial shape error concealment for object-based image and video coding," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 586–599, Apr. 2004.
- [14] L. D. Soares and F. Pereira, "Combining space and time processing for shape error concealment," presented at the Picture Coding Symp. (PCS), San Francisco, CA, 2004.
- [15] L. D. Soares and F. Pereira, "Spatio-temporal scene level error concealment for shape and texture data in segmented video content," in *Proc. IEEE Int. Conf. Image Process.*, 2006, pp. 2249–2252.
- [16] L. Piegl and W. Tiller, *The NURBS Book*. New York: Springer, 1997.
- [17] S. Biswas and B. Lovell, *Bezier and Splines in Image Processing and Machine Vision*. London, U.K.: Springer, 2008.
- [18] F. W. Meier, G. M. Schuster, and A. K. Katsaggelos, "An efficient boundary encoding scheme which is optimal in the rate distortion sense," in *Proc. IEEE Int. Conf. Image Process.*, 1997, vol. 2, pp. 9–12.
- [19] G. M. Schuster, G. Melnikov, and A. K. Katsaggelos, "Operationally optimal vertex-based shape coding," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 91–108, Nov. 1998.
- [20] G. Melnikov, G. M. Schuster, and A. K. Katsaggelos, "Inter mode vertex-based optimal shape coding," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 1999, vol. 6, pp. 3125–3128.
- [21] E. Tsiligianni, L. P. Kondi, and A. K. Katsaggelos, "Shape error concealment based on a shape-preserving boundary approximation," in *Proc. IEEE Int. Conf. Image Process.*, 2010, pp. 457–460.
- [22] T. Ebrahimi and C. Horne, "MPEG-4 natural video coding—An overview," *Signal Process.: Image Commun.*, vol. 15, pp. 365–385, 2000.
- [23] Y. Wang, J. Ostermann, and Y. Zhang, *Video Processing and Communications*. Englewood Cliffs, NJ: Prentice-Hall, 2001.
- [24] C. de Boor, *A Practical Guide to Splines*. New York: Springer, 2001.
- [25] F. Mokhtarian and A. K. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 8, pp. 789–805, Aug. 1992.
- [26] Y. Wang and E. K. Teoh, "A novel 2D shape matching algorithm based on B-spline modeling," in *Proc. IEEE Int. Conf. Image Process.*, 2004, vol. 1, pp. 409–412.
- [27] P. Mongkolnam, T. Dechsakulthorn, and C. Nukoolkit, "Image shape representation using curve fitting," in *Proc. 6th WSEAS Int. Conf. Signal, Speech, Image Process.*, 2006, pp. 1–6.
- [28] Y. C. Hwang, T. L. Shieh, and L. S. Lan, "A weighted least-squares approach for B-spline shape representation," in *Proc. 3rd Int. Symp. Commun., Control, Signal Process.*, 2008, pp. 894–898.
- [29] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recog.*, vol. 37, pp. 1–19, 2004.
- [30] A. K. Katsaggelos, L. P. Kondi, F. W. Meier, J. Ostermann, and G. M. Schuster, "MPEG-4 and rate-distortion based shape coding techniques," *Proc. IEEE*, vol. 86, no. 6, pp. 1126–1154, Jun. 1998.
- [31] F. W. Meier, G. M. Schuster, and A. K. Katsaggelos, "A mathematical model for shape coding with B-splines," *Signal Process.: Image Commun.*, vol. 15, pp. 685–701, 2000.
- [32] L. P. Kondi, G. Melnikov, and A. K. Katsaggelos, "Joint optimal shape estimation and encoding," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 14, no. 4, pp. 528–533, Apr. 2004.
- [33] S. K. Bandyopadhyay and L. P. Kondi, "Optimal bit allocation for joint texture-aware contour-based shape coding and shape-adaptive texture coding," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 18, no. 6, pp. 840–844, Jun. 2008.
- [34] J. Zaletejl and J. F. Tasic, "Rate-distortion snake: A tool for optimal shape coding," in *Proc. IEEE Int. Conf. Image Process.*, 2005, vol. 1, pp. 1-565–I-568.
- [35] G. Beliakov, "Shape preserving approximation using least squares splines," *Approx. Theory its Appl.*, vol. 16, pp. 80–98, 2000.
- [36] P. Dierckx, *Curve and Surface Fitting With Splines*. Oxford, U.K.: Clarendon Press, 1995.
- [37] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [38] L. Shure, "Brief history of nonnegative least squares in MATLAB," 2006. [Online]. Available: <http://blogs.mathworks.com/loren/2006/>
- [39] D. Chen and R. Plemmons, "Nonnegativity constraints in numerical analysis," presented at the Symp. Birth Numerical Anal. Leuven, Belgium, 2007.



Evaggelia V. Tsiligianni (S'10) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 2001 and the M.S. degree in computer science from University of Ioannina, Greece, in 2010, where she is currently pursuing the Ph.D. degree from the Department of Computer Science.

Her research interests include video coding, sparse representations and compressive sensing.



Lisimachos P. Kondi (SM'11) received the Diploma in electrical engineering from the Aristotle University of Thessaloniki, Greece, in 1994, and the M.S. and Ph.D. degrees in electrical and computer engineering from Northwestern University, Evanston, IL, in 1996 and 1999, respectively.

During the 1999–2000 academic year, he was a Postdoctoral Research Associate at Northwestern University. He is currently an Assistant Professor with the Department of Computer Science, University of Ioannina, Greece. He was previously with the faculty of the University at Buffalo, The State University of New York, and has also held summer appointments at the Naval Research Laboratory, Washington, DC, and the Air Force Research Laboratory, Rome, NY. His research interests include the general areas of signal and image processing and communications, including image and video compression and transmission over wireless channels and the Internet, super-resolution of video sequences, and shape coding.

Dr. Kondi is an Associate Editor of the *EURASIP Journal on Advances in Signal Processing* and an Associate Editor of the *IEEE SIGNAL PROCESSING LETTERS*.



Aggelos K. Katsaggelos (F'98) received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Greece, in 1979, and the M.S. and Ph.D. degrees in electrical engineering from the Georgia Institute of Technology, in 1981 and 1985, respectively.

In 1985, he joined the Department of Electrical Engineering and Computer Science at Northwestern University, where he is currently a Professor holder of the AT&T chair. He was previously the holder of the Ameritech Chair of Information Technology (1997–2003). He is also the Director of the Motorola Center for Seamless Communications, a member of the Academic Staff, NorthShore University Health System, an affiliated faculty at the Department of Linguistics and he has an appointment with the Argonne National Laboratory. He has published extensively in the areas of multimedia signal processing and communications (over 180 journal papers,

400 conference papers and 40 book chapters) and he is the holder of 19 international patents. He is the co-author of *Rate-Distortion Based Video Compression* (Kluwer, 1997), *Super-Resolution for Images and Video* (Claypool, 2007), and *Joint Source-Channel Video Transmission* (Claypool, 2007).

Prof. Katsaggelos was Editor-in-Chief of the *IEEE Signal Processing Magazine* (1997–2002), a BOG Member of the IEEE Signal Processing Society (1999–2001), and a member of the Publication Board of the IEEE Proceedings (2003–2007). He was a recipient of the IEEE Third Millennium Medal

(2000), the IEEE Signal Processing Society Meritorious Service Award (2001), the IEEE Signal Processing Society Technical Achievement Award (2010), an IEEE Signal Processing Society Best Paper Award (2001), an IEEE ICME Paper Award (2006), an IEEE ICIP Paper Award (2007), and an ISPA Paper Award (2009). He was a Distinguished Lecturer of the IEEE Signal Processing Society (2007–2008). He is a fellow of the SPIE (2009).