SHAPE ERROR CONCEALMENT BASED ON A SHAPE-PRESERVING BOUNDARY APPROXIMATION

Evaggelia Tsiligianni, Lisimachos P. Kondi

University of Ioannina
Dept. of Computer Science
Ioannina, Greece

Aggelos K. Katsaggelos

Northwestern University
Dept. of EECS
Evanston, IL, USA

ABSTRACT

In error–prone communications, packet loss results in missing information of shape, motion and texture of a video object (VO). Error concealment refers to the recovery of lost information at the decoder. In this paper, we propose a spatial shape error concealment technique. We consider a geometric representation of the shape of a VO consisting of its boundary, which can be extracted from the received α-plane. Some boundary parts are missing due to errors. We propose a method for modeling the received boundary based on a shape-preserving approximation that uses T–splines. Such an approximation provides a good estimation of the direction of a missing boundary segment, which we use to construct a concealment spline that joins smoothly with the received boundary parts.

Index Terms— Shape coding, error concealment, T-splines.

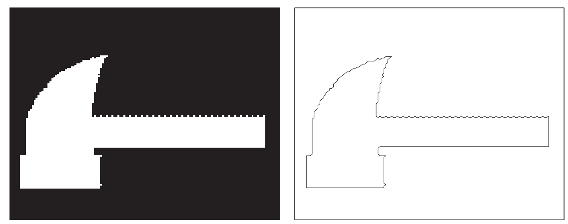
1. INTRODUCTION

In wireless networks and the Internet, transmitted information is subject to errors. As retransmission of lost or damaged packets may incur delay, error resilient methods have been developed to detect and correct transmission errors. Post processing error concealment includes estimation of the lost information by making use of the inherent correlation among spatially and temporally adjacent samples.

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

Three categories of techniques for shape error concealment have been developed. Temporal techniques use information from previous video frames. Spatial techniques are based on information of the neighbouring to the lost part area. Techniques combining both temporal and spatial information are referred to as temporal-spatial techniques.

In this paper, we propose a spatial error concealment technique based on a contour representation of the object shape, i.e., the boundary of its texture (Fig. 1). Channel errors result in broken parts of the boundary (Fig. 2). The techniques proposed to solve this problem include the modeling of the received boundary parts and the construction of a concealment curve based on the modeling curve (Fig. 3). In [1], Hermite cubic polynomials are used as concealment curves. An estimation of the first derivatives of the received boundary is necessary to construct the polynomial curve. On each side of the missing segment, a boundary part is approximated by a quadratic polynomial. A similar reconstruction based on cubic Bezier curves is proposed in [2], where the approximation error is minimised by an iterative algorithm. However, a small approximation error does not guarantee a good estimation of the first derivative.



(a) *hammer* α-plane (b) *hammer* boundary

Fig. 1: Binary (a) and contour (b) representation of an object shape.

As simple polynomials often fail to represent a natural boundary [3], we propose a concealment technique based on a spline representation. Moreover, we need an approximation that does not introduce changes in the boundary slope. We propose an appropriate modeling of the received boundary based on a spline approximation that preserves the characteristics defining the shape. Such an approximation provides a good estimation of the first derivative, leading to the construction of a smooth and natural concealment spline curve.

Our problem formulation is presented in section 2. The proposed boundary modeling and concealment methods are described in sections 3 and 4, respectively. In section 5, error concealment is applied to a boundary encoded with B–splines. In section 6, experimental results are presented. Finally, in section 7, conclusions are drawn.

2. PROBLEM FORMULATION

The basic unit of coding in MPEG-4 is the Video Object Plane (VOP). A binary shape is described by a binary mask. In the binary α-plane, 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. In this paper, we use a geometric description of the object shape, its boundary. We define the boundary of an α-plane as the collection of points belonging to the background, which have at least one 4-connect neighbour (that is with pixels above, below, to the left and to the right) that belongs to the object (that is their value is 0 in the α-plane) (Fig. 1). We assume that the boundary is a closed non-intersecting curve.

A broken boundary can be caused by the loss of a packet containing information corresponding to several boundary points, yielding one or more missing boundary segments. We refer to the points that touch a missing segment as "connecting points". We also re-

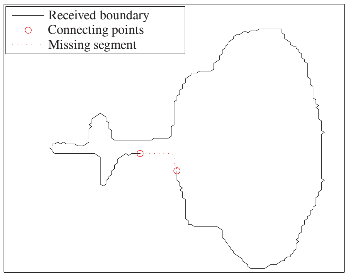


Fig. 2: Broken boundary

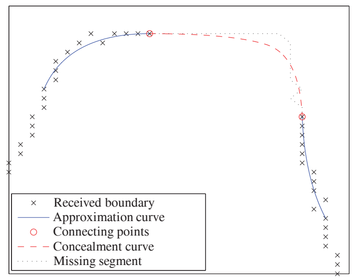


Fig. 3: Boundary modeling and error concealment

fer to the boundary slope at each connecting point as the *boundary direction*. Our shape error concealment technique consists of two steps: First, we model the received boundary by solving an approximation problem; second, we estimate the boundary direction and use it to construct a curve representing the missing segment (Fig. 3). We require the new curve direction to coincide with the received boundary direction, so that C^1 continuity is achieved. Consequently, the direction estimation is critical to the construction of a smooth concealment curve.

3. RECEIVED BOUNDARY MODELING

Regarding the modeling of the received boundary, traditional B–splines could be a solution, if they didn't fail to preserve essential properties of the boundary points [4]. In Fig. 4, a B–spline approximation is decreasing to the right of the connecting point, not preserving the zero slope of the boundary at the connecting point. In order to prevent the approximation from introducing changes of the boundary direction, we will use monotone least squares splines.

In [4], 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; for such problems effective and robust methods exist.

3.1. Least squares spline approximation

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

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

which forms a solution to the problem

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

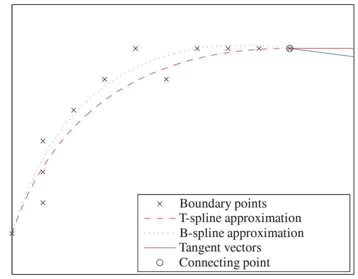


Fig. 4: T–spline approximation and estimation of the tangent vector at the connecting point, where the boundary slope is zero. T–spline approximation provides the right slope estimation as it preserves boundary monotonicity. B-spline approximation results in a wrong (negative) slope estimation.

where f_i are the spline basis functions and b_i are the unknown spline coefficients or approximation control points. For the calculation of the values of $\bar{u}_k$, $k = 0, \ldots, N_B$ that affect the parameterization of the curve, the "chord length" method could be used [3].

3.2. Monotone spline approximation using T–splines

According to [4], if we use second degree T–splines basis functions, we can construct an increasing quadratic approximation. The necessary and sufficient condition for monotonicity is $b_i \geq 0$, $i = 0, \ldots, n$. For a definition of T–splines see [5]. Under the monotonicity condition, the linear least squares problem (2) becomes a non-negative least squares problem.

Morever, in our problem we need the approximation to interpolate 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 for w.

Under all restrictions above, (2) finally becomes

$$\min_{b_i \geq 0} \left[\sum_{k=1}^{N_B - 1} \left(Q_k - C(\bar{u}_k)\right)^2 + w \cdot \left(Q_{N_B} - C(\bar{u}_{N_B})\right)^2 \right], \tag{3}$$

which can be solved using Matlab's subprogram *lsnonneg*.

Having computed the control points b_i, we can use B–spline conversion formulas [4] to get a stable and effective calculation of the approximation curve C. For monotonically decreasing splines, the results are analogous.

Fig. 4 shows an approximation curve that was constructed with the above method, providing a good estimation of the boundary direction, a goal that cannot be achieved by a B–spline approximation.

3.3. Shape-preserving boundary approximation

In order to apply the T–splines approximation method, we have to select a suitable boundary part. 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. 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 $x_i \geq x_{i+1}$, $\forall i = 0, \ldots, N_B$. Second, we have to select a part of increasing or decreasing monotonicity. However, a

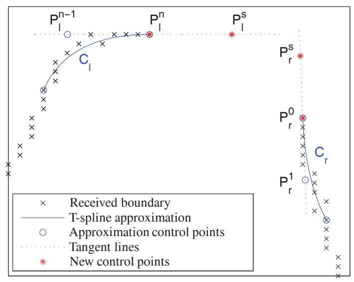


Fig. 5: Approximation of the received boundary on each side of the missing segment and determination of new control points. The approximation control points and the estimated tangents at the connecting points are used to find the new control points.

minor variation in monotonicity is allowed as it does not affect the T–spline approximation results. In Fig. 4, the change in monotonicity introduced by the fourth point before the connecting point is ignored while selecting the boundary points.

Because of the free form of a boundary, it is difficult to predefine the complexity of the approximation curve, which is determined by the number of the spline segments or equivalently, by $n+1$, the number of the control points b_i (Eq. (1)). In order to minimise the approximation error, we may need to execute a few iterations of the approximation method, in which we gradually increase $n+1$. If it reaches half the number of the boundary points and 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 detailed results of the approximation method can be seen in Fig. 5.

4. CONCEALMENT CURVE CONSTRUCTION

4.1. New control points

Suppose we have found the left boundary approximation, C_l, which is represented by the control points $(P_l^0, \dots, P_l^n)$. We recall that C_l interpolates P_l^n, the left connecting point. The tangent line at P_l^n passes through control point P_l^{n-1} [3]. We can find a new control point, P_l^s, on the tangent vector in the way that was proposed in [2]; namely P_l^s is symmetric to P_l^{n-1} with respect to P_l^n (Fig. 5). This way the tangent vectors on each side of the connecting point P_l^n coincide and C^1 continuity between the approximation and the concealment curve is achieved. Similarly, we obtain another control point P_r^s at the right boundary part.

4.2. Concealment curve

Having two connecting points and two new control points, we can use four control points $(P_0, P_1, P_2, P_3) = (P_l^n, P_l^s, P_r^s, P_r^0)$ to get a concealment curve. We could use various types of curves such as cubic polynomials [1] or cubic Bezier [2]. We propose the quadratic B–spline curve because it has a more natural appearance compared to a cubic polynomial, passes closer to new control points compared to a cubic Bezier and joins smoothly to the known boundary approximation curves. The concealment B–spline curve is obtained by

$$C(u) = \sum_{i=0}^{3} P_i \cdot N_{i,2}(u), \quad 0 \le u \le 1, \tag{4}$$

with $N_{i,2}$ the i-th quadratic B–spline basis function. Figs. 3 and 6 show the concealment results for the example illustrated in Fig. 2.

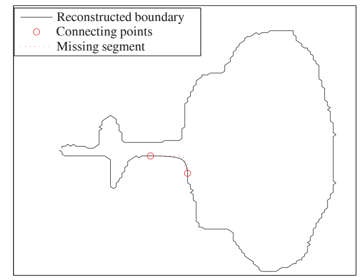


Fig. 6: Boundary after concealment.

	$D_{n,\text{avg}}$	$D_{n,\text{low}}$	$D_{n,\text{high}}$
hammer	0.011	0.000	0.025
fountain	0.004	0.000	0.010
fork	0.013	0.004	0.025

Table 1: Arithmetic results of the proposed method.

5. 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. If the object boundary encoding scheme is based on a spline approximation, the received information consists of the approximation control points. If the introduced distortion is small, the approximation curve 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.

In [6] 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 distortion band width defines the approximation quality. If the width is kept small, the approximation curve does not introduce changes in boundary direction.

In order to apply the proposed concealment method, the received control points representing the boundary parts on each side of the missing boundary segment can be used. New control points can be determined in the way described in section 4.1 and a concealment curve can be constructed according to (4).

6. EXPERIMENTAL RESULTS

A number of experiments were performed, some of which are presented here. In order to quantify the performance of the proposed concealment method, we will 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. We will compare our method to the error concealment method proposed by Soares and Pereira [2]. Finally, reconstructed α-planes will be illustrated for subjective evaluation.

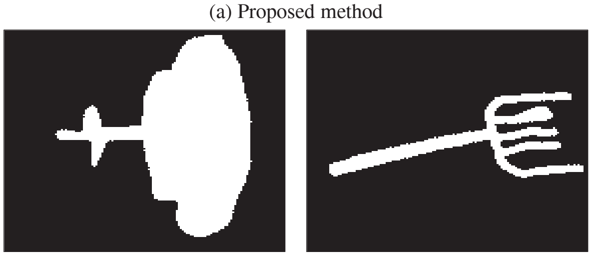
In our experiments we used three object shapes with different smoothness level and concealing difficulty, namely *hammer*, *fountain* and *fork* (Figs. 1 and 7). For every boundary, we assumed a missing segment consisting of 20 points and applied the proposed method. The corresponding α-plane was extracted after boundary

(a) Original *fountain* and *fork*

(b) Corrupted *fountain* and *fork*

Fig. 7: Shapes used in experiments.

(a) Proposed method



(b) Soares-Pereira method

Fig. 8: Reconstructed α-planes for *fountain* and *fork*.

reconstruction. We repeated the experiment on many different missing boundary segments. The average D_n values associated with every object are shown in Table 1. We also show D_n low and high values observed for every object, corresponding to the best and the worst concealment results. As can be seen in Table 1, only a small percentage of the reconstucted object pixels differs from the original ones. In most cases, such small differences are hardly visible.

The same samples were used in order to compare our method to the method of Soares and Pereira. Results of this method are illustrated in Table 2. Comparing Tables 1 and 2, the proposed method gives better D_n values for all three shapes. The improvement is greater in the case of a less smooth boundary like *fork*. This is explained by the fact that a cubic Bezier curve can be effective in modeling smooth boundaries, however, for complex boundaries a spline curve is more appropriate.

In Table 3, we present experimental results for the case where the original boundary was encoded using a quadratic B–spline approximation [6]. The spline lies in a band of one pixel width

	$D_{n,\text{avg}}$	$D_{n,\text{low}}$	$D_{n,\text{high}}$
hammer	0.014	0.002	0.035
fountain	0.004	0.000	0.013
fork	0.017	0.007	0.048

Table 2: Arithmetic results of Soares–Pereira method.

	$D_{n,\text{avg}}$	$D_{n,\text{low}}$	$D_{n,\text{high}}$
hammer	0.006	0.001	0.015
fountain	0.011	0.006	0.024
fork	0.036	0.005	0.090

Table 3: Arithmetic results of the proposed method (B–splines encoding scheme)

along the original boundary, therefore, it can be considered shape-preserving. Thus, as discussed in section 5, 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. We assume that information of two control points are lost during transmission, which results in a missing segment consisting of 30 original points. The low D_n values again indicate successful concealment.

Obviously, arithmetic results cannot express the subjective impact of the reconstructed boundary. As our approximation method preserves original boundary characteristics, such as boundary direction, it leads to better subjective results than [2]. Fig. 8 illustrates concealed α-planes of the corrupted samples of Fig. 7(b), for our method and the method of [2].

7. CONCLUSIONS

We proposed a shape concealment method that replaces missing boundary information with a natural spline curve. Our method is based on the approximation of the received boundary in a way that can represent its complexity level and preserve its direction at the connecting points. The concealment curve is a quadratic B–spline curve having the same direction at the connecting points. Our method leads to better objective and subjective results than the current state of the art.

8. REFERENCES

[1] G.M. Schuster, X. Li, and A.K. Katsaggelos, "Error concealment using Hermite splines," *IEEE Transactions on Image Processing*, vol. 13, pp. 808–820, 2004.

[2] L.D. Soares and F. Pereira, "Spatial shape error concealment for object-based image and video coding," *IEEE Transactions on Image Processing*, vol. 13, pp. 586–599, 2004.

[3] L. Piegl and W. Tiller, *The NURBS Book*, Springer, 1997.

[4] G. Beliakov, "Shape preserving approximation using least squares splines," *Approx. Theory & its Appl*, vol. 16, pp. 80–98, 2000.

[5] P. Dierckx, *Curve and Surface Fitting with Splines*, Clarendon Press, 1995.

[6] F.W. Meier, G.M. Schuster, and A.K. Katsaggelos, "A mathematical model for shape coding with B-splines," *Signal Processing:Image Communication*, vol. 15, pp. 685–701, 2000.