Shape error concealment using Hermite splines

Guido M. Schuster, Xiaohuan Li, and Aggelos K. Katsaggelos

***Abstract*— The introduction of Video Objects (VOs) is one of the innovations of MPEG-4. The α-plane of a VO defines its shape at a given instance in time and hence determines the boundary of its texture. In packet-based networks, shape, motion, and texture are subject to loss. While there has been considerable attention paid to the concealment of texture and motion errors, little has been done in the field of shape error concealment.**

In this paper we propose a post-processing shape error concealment technique that uses geometric boundary information of the received α-plane. Second order Hermite splines are used to model the received boundary in the neighboring blocks, while third order Hermite splines are used to model the missing boundary. The velocities of these splines are matched at the boundary point closest to the missing block.

There exists the possibility of multiple concealing splines per group of lost boundary parts. Therefore we draw every concealment spline combination that does not self-intersect and keep all possible results until the end. At the end we select the concealment solution that results in one closed boundary. Experimental results demonstrating the performance of the proposed method and comparisons with prior proposed methods are presented.

I. Introduction

One of MPEG-4's most prominent features, as compared to MPEG-1 and 2, is the presentation of video objects (VOs) [1], [2]. Based on this, MPEG-4 video sequences are interpreted and manipulated in terms of visual objects. These objects are defined by their motion, texture and shape. Their particular shape at a given instance in time is described as a gray-scale or binary α-plane (see for example Figure 1). In this paper, a bounding box is put around the shape in the α-plane. The bounding box width and height must be integer multiples of 16 pixels. With this constraint, the position of the bounding box and its width and height are selected such that it is as small as possible, but still completely contains the shape, and stays within the original frame. In the figures, the tick marks are such that the left and upper side of the bounding box would fall exactly onto these, as can be seen in Figure 1. The interpretation of the binary α-plane is that a 1 indicates a pixel that is part of the VO at that given time instance, while a 0 identifies a pixel belonging to the background. A gray-scale α-plane allows for a blending of the object with the background. In this paper we are concentrating on the more common binary α-planes. In error-prone communication environments, MPEG-4 video packets are often coded in the data-partitioning mode, under which shape and motion information is separated from texture. As suggested by annex E1.2 of [1], 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. This provides motivation to design a shape concealment technique, in addition to all existing algorithms for texture and motion concealment [3].

Various approaches to conceal shape information have been proposed. Some of them are based on a motion compensation scheme [4], [5], [6]. These concealment methods are limited by the intrinsic setbacks of motion compensation, such as error propagation, and perform poorly when objects appear/disappear, rotate or are distorted. Spatial concealment methods, by nature, bypass these problems, although they only have the information of the current α-plane available, which, in high loss scenarios, can be a drawback. As we will show in the experimental section, this seems to be enough information to conceal the lost boundary parts quite well.

A maximum *a posteriori* (MAP) estimator with a Markov random field (MRF) designed for binary shape information of neighboring blocks was proposed in [7]. It uses the spatial redundancy in the transmitted shape information on a statistical basis. This approach is based on the statistics of the α-plane and does not use the object boundary directly as the salient feature. As we will show in the experimental section, the proposed boundary-based approach is able to significantly outperform this statistical approach.

In the spatial concealment part of [6], a more geometric point of view was taken and the binary α-plane was interpreted as a boundary that is missing some segments. This is similar to the interpretation used in this paper and in [8] and [9]. In [6], the missing boundary segments were estimated using a second order Bezier curve that matched the estimated derivatives at both connecting ends. There are several problems with this method. First, second order Bezier curves are either convex or concave, although there is no reason why the lost boundary segment must be convex or concave. In our experience this constraint often results in incorrect approximations. For example, Figure 3 shows the reconstructed boundary using second order Bezier concealment splines for the received α-plane shown in Figure 2. In Figure 3, in the right upper corner, the missing boundary segment is convex and hence the second order Bezier curve approach provides a satisfactory result. In the middle lower part, however, the missing boundary segment is neither convex nor concave, and forcing a second order Bezier curve that matches the derivatives results in a rather strange concealment spline. Second, in [6] there is no solution offered for the problem when there is more than one spline necessary to conceal the lost boundary area, as for example shown in Figure 4. Since an entire row of the α-plane is lost, there are now many possible splines that could be drawn

G. M. Schuster is with the Abteilung Elektrotechnik, Hochschule für Technik Rapperswil, Oberseestrasse 10, CH-8640 Rapperswil, Switzerland, T +41 55 222 4513, F +41 55 222 4400, guido.schuster@hsr.ch

X. Li is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA, T 404 894 2977, F 404 894 8363, xiaohuan@ece.gatech.edu

A. K. Katsaggelos is with the Department of Electrical and Computer Engineering, Northwestern University, 2145 Sheridan Road, Evanston, Illinois 60208-3118, USA, T 847 491 7164, F 847 491 4455, aggk@ece.nwu.edu

to conceal the lost boundary. In Figures 5 and 6, two such possible solutions are shown. Third, the estimation of spatial derivatives on the boundary is not numerically robust. As we will show in later sections, the proposed algorithm overcomes all these limitations.

The rest of the paper is organized as follows: In section II, we look at the case of one missing macro block (MB) and show how the Hermite spline concealment technique works. In section III, we introduce the concept of connected groups of lost MBs. The main problem is now that there can be many possible concealing splines. This case also includes the possibility that there are many possible concealing splines when only one MB is lost. In section III, we solve this problem by considering only the spline combinations that do not intersect. This reduces the number of test cases by a large factor. We then keep every spline combination that does not intersect and introduce the final test, according to which the boundary which results in the shortest closed non-intersecting curve, is kept. In section IV, we present experimental results and compare the proposed algorithm with others in the literature. In section V, we summarize the paper and draw our conclusions.

II. Case I: One lost macro block having only two connecting points

In this section we introduce the core idea of the paper, which is the concealment of the missing boundary segment using a cubic Hermite spline. We define the boundary of an α-plane as the collection of points that belong to the background (that is their value is 0 in the α-plane) which have at least one 4-connect neighbor (that is the 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). Figure 7 shows the boundary as defined for the α-plane shown in Figure 1. Note that we could equally well define the boundary as belonging to the object and being 4-connected to the background. A basic assumption is that the boundary is a closed non-intersecting curve. We use the closed curve assumption at the end to select one concealing boundary out of several possible ones and the non-intersecting assumption to remove a large number of possible, but intersecting, concealment splines. If these are not appropriate assumptions, it becomes necessary to know how many boundaries there are and how large (in number of points) they are relative to each other. If this knowledge is known, then the proposed algorithm is still applicable. In addition we also assume that only entire MBs can be lost and the dimension of an MB is 16×16 pixels. In other words, the smallest unit of lost information is a MB. Clearly, depending on the particular packetization scheme, one packet loss can result in several MBs losses. For example, it is quite common to put an entire row of MBs into one so called MPEG-4 video packet and then put this video packet into one RTP/UDP/IP (Real Time Protocol/User Datagram Protocol/Internet Protocol) packet. Hence if this packet is lost during transmission, an entire row of α-plane data is lost (see for example Figure 4). First we simplify the problem by considering the case when only one MB of the α-plane containing part of the boundary is lost.

A. Connecting points

Consider Figure 8, which shows an α-plane that is missing a MB containing part of the boundary. First we detect the part of the boundary that is not missing, using the above definition of a boundary, as seen in Figure 9. Now we need to detect boundary points that touch the missing MB. We call these points “connecting points”. These are important points, since they are the start and the end of the concealing splines. This is achieved by first finding points that belong to the detected boundary which also have at least one 4-connect neighbor belonging to the missing MB. Out of these points, we then select the points that have only one 8-connected neighbor belonging to the detected boundary. This excludes points that run parallel to a missing MB and hence are not really connecting points. We collect the connecting points in a clockwise fashion. This is not important with only two points, but will become important in the next section when we consider many connecting points. As we will also see in the next section, there is always an even number of these connecting points, since every line that enters a MB must come out of it.

B. Associated points

After the two (in this case) connecting points are identified, we collect for each connecting point a set of points, henceforth called “associated points”. Associated points are points that belong to the boundary and are 8-connected with a connecting point or another associated point. Hence we start at each of the two connecting points and collect associated points in an 8-connect fashion, moving forward or backward on the boundary. From Figure 10 it is clear that there may be more than one boundary point which is 8-connected to the current associated point, say point 6. It becomes necessary to select them in the proper order, since we want to create an ordered connected set of associated points. This is achieved by selecting the 8-connected point that has the most object pixels as 8-connect neighbors within the 8-connect neighborhood of the current associated point. Hence in this example, the next associated point would be point 7. With this strategy, we force the associated points to stay as close as possible to the object, which is especially important in sharp turns of the boundary. The ordering shown in Figure 10 is achieved with this strategy. Figure 11 shows the connecting points and the associated points for the current example. It is important to note that the connecting points are the first points in the ordered associated points set and that the further away from the connecting point (along the boundary) a point is in the associated point set, the farther back in the ordered set it is. One parameter that has not been discussed so far is the number of points that should be collected for the associated points set. This depends on the future use of these points. Since we will employ these points to estimate second order Hermite splines, we collect quite a few points. Our experiments have shown that an associated point set of 20 points is a good choice. In the case where many MBs are lost, fewer points are available and then we collect as many as we can.

C. Second order Hermite spline model for the associated points set

Now that the two associated points sets have been defined, a mathematical model is chosen to describe each of them. One common approach is to utilize a straight line that goes through the connecting point. For example, a minimum mean-squared-error (MSE) fit can be used, to fit the data to the model. We do not adopt this model primarily since it represents a poor description of the associated points set. In addition, since we plan to utilize cubic Hermite splines for concealing the lost boundary segment, a second order Hermite spline is used to model each associated points set. We have carried out experiments comparing the performance of first, second and third order Hermite splines as mathematical models of the associated points sets. The second order model resulted in the best overall performance. A second order Hermite spline has the form:

$$l(s) = \begin{bmatrix} l_x(s) \\ l_y(s) \end{bmatrix} = \begin{bmatrix} e_x s^2 + f_x s + g_x \\ e_y s^2 + f_y s + g_y \end{bmatrix}, \quad (1)$$

where s is the free parameter, l is used to indicate the associated points set to the left of the missing MB (the same equation with $r(s)$ is utilized for the right associated points set), and e_x, e_y, f_x, f_y, g_x and g_y represent the parameters to be evaluated. The velocity vector $lv(s)$ for this model is equal to:

$$lv(s) = \begin{bmatrix} \frac{dl_x(s)}{ds} \\ \frac{dl_y(s)}{ds} \end{bmatrix} = \begin{bmatrix} 2e_x s + f_x \\ 2e_y s + f_y \end{bmatrix}. \quad (2)$$

The goal of the mathematical model for the associated points set is to provide accurate velocity vectors that will be matched by the concealing curve, as explained in the next section. This goal can be achieved even with a first order Hermite spline, which is a straight line. A straight line is, again, too simple of a model to capture the true nature of the associated points set, and hence a second order Hermite spline is used.

The parameters in Equation 1 are evaluated by fitting the spline to the data in the MSE sense and forcing it to go through the connecting point. Since the spline has to go through the connecting point (where the free parameter s is zero) with the coordinates (x_0,y_0), the estimates of the parameters g_x and g_y must be equal to x_0 and y_0 respectively. For the MSE estimate of the other parameters, we use the facts that the associated point set is ordered and the distance between consecutive points is known. The distance is 1 pixel for horizontal/vertical neighbors and 1.41 pixels for diagonal neighbors. In other words, the values of the parameter s start at 0 and are incrementing by 1 or 1.41 pixel units as it moves along the associated points set. This can be written in matrix form as follows:

$$\begin{bmatrix} s_1^2 & s_1 \\ s_2^2 & s_2 \\ s_3^2 & s_3 \\ \vdots & \vdots \\ s_N^2 & s_N \end{bmatrix} . \begin{bmatrix} e_x \\ f_x \end{bmatrix} = \begin{bmatrix} x_1 - x_0 \\ x_2 - x_0 \\ x_3 - x_0 \\ \vdots \\ x_N - x_0 \end{bmatrix} \quad (3)$$

and

$$\begin{bmatrix} s_1^2 & s_1 \\ s_2^2 & s_2 \\ s_3^2 & s_3 \\ \vdots & \vdots \\ s_N^2 & s_N \end{bmatrix} . \begin{bmatrix} e_y \\ f_y \end{bmatrix} = \begin{bmatrix} y_1 - y_0 \\ y_2 - y_0 \\ y_3 - y_0 \\ \vdots \\ y_N - y_0 \end{bmatrix} \quad (4)$$

where (x_0,y_0) are the coordinates of the connecting point, (x_N,y_N) the coordinates of the last point in the associated points set, and s_i, ($i = 1, \ldots, N$), the value of s at point (x_i,y_i) of the associated points set. The MSE estimates ($[\hat{e}_x, \hat{f}_x]^T$ and $[\hat{e}_y, \hat{f}_y]^T$) for the above parameters can now be found using the generalized inverse. To get an estimate of the velocity at the connecting point, that is at $s = 0$, the values of the estimated parameters are substituted into Equation 2. This value of the velocity will be matched by the concealing spline, as explained in the next section. Figure 12 shows two second order Hermite splines (the left and the right) and also the estimated velocities at the connecting points. Note that the velocities are amplified by a factor 10 and the velocity at the left connecting point had its sign changed. This is necessary since the parameter s is always equal to zero at the connecting point and increases when moving along the ordered associated points set. That is, the direction of s is different for the right and left splines, and therefore one of the velocities needs its sign changed.

D. Cubic Hermite spline for boundary concealment

We propose to use a cubic Hermite spline for concealing the missing boundary, which goes through the left and right connecting points and matches the corresponding velocities of the left and right second order splines modelling the associated points set at these points. The ability of a curve to match the starting and stopping points (left and right connecting points, respectively, in the current example) is absolutely necessary and many curves can do that. Being able to match the left and right velocities along the concatenated curve (left spline, middle spline, right spline) leads to visually pleasing results. One could increase the requirements on the concealing curve and also match, for example, the acceleration along the concatenated curve. We have experimented with higher order splines and found out that the results were similar, while the computational complexity was increased. So the cubic Hermite spline appears to be the best compromise in matching the concealing spline to the available data.

A cubic Hermite spline $m(s)$ is defined by

$$m(s) = \begin{bmatrix} m_x(s) \\ m_y(s) \end{bmatrix} = \begin{bmatrix} a_x s^3 + b_x s^2 + c_x s + d_x \\ a_y s^3 + b_y s^2 + c_y s + d_y \end{bmatrix}, \quad (5)$$

where s again is the free parameter and a_x, a_y, b_x, b_y, c_x, c_y, d_x and d_y are the parameters to be estimated. It has the interesting property that the parameters can be selected so that one can exactly match a start and a stop point ($m(start)$, $m(stop)$) and a start and a stop velocity ($mv(start)$, $mv(stop)$) given by

$$mv(s) = \begin{bmatrix} \frac{dm_x(s)}{ds} \\ \frac{dm_y(s)}{ds} \end{bmatrix} = \begin{bmatrix} 3a_x s^2 + 2b_x s + c_x \\ 3a_y s^2 + 2b_y s + c_y \end{bmatrix}. \quad (6)$$

The equations for estimating the unknown parameters in Equation 5 are given by

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ s_m^3 & s_m^2 & s_m & 1 \\ 0 & 0 & 1 & 0 \\ 3s_m^2 & 2s_m & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} = \begin{bmatrix} l_x(0) \\ r_x(0) \\ -\frac{dl_x}{ds}|_{s=0} \\ \frac{dr_x}{ds}|_{s=0} \end{bmatrix} \quad (7)$$

and

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ s_m^3 & s_m^2 & s_m & 1 \\ 0 & 0 & 1 & 0 \\ 3s_m^2 & 2s_m & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_y \\ b_y \\ c_y \\ d_y \end{bmatrix} = \begin{bmatrix} l_y(0) \\ r_y(0) \\ -\frac{dl_y}{ds}|_{s=0} \\ \frac{dr_y}{ds}|_{s=0} \end{bmatrix}. \quad (8)$$

The terms $l_x(s)$, $l_y(s)$, $\frac{dl_x}{ds}|_{s=0}$ and $\frac{dl_y}{ds}|_{s=0}$ have been defined in Equations 1 and 2. The terms $r_x(s)$, $r_y(s)$, $\frac{dr_x}{ds}|_{s=0}$ and $\frac{dr_y}{ds}|_{s=0}$ are also defined by Equations 1 and 2 by replacing "l" by "r", as mentioned earlier, where s_m is the distance along the final cubic Hermite spline in pixels. Now since this distance is not known *a priori*, we start with an s_m that is equal to the distance between the two connecting points. This is the shortest distance possible. Based on our experimental evidence the performance of the scheme can be slightly improved if an iteration is used, where we start with the minimum spline length, we obtain an estimate of the spline, measure the new spline length, and if this new length is more than 1% different from the old length, we use this new length and iterate again. This is done until there is no significant change in the lengths, which is the case after a few iterations.

The solution of the two systems of equations in Equations 7 and 8 is straightforward. The resulting cubic Hermite spline from the current example is shown in Figure 13. Note the matching velocities, which are again amplified by a factor 10. The concealing spline is now connected to the remaining boundary, as shown in Figure 14. To get the concealed α-plane we simply flood the interior of the object with 1s and set everything else (including the reconstructed boundary) to 0, as shown in Figure 15. Figure 16 shows the pixels that were incorrectly concealed, that is they are 1 when they should have been 0 or they are 0 when they should have been 1.

III. Case II: Multiple connecting points

Losing a single MB is possible, but clearly, depending on the loss rate of the network and the packetization scheme, it is also possible to lose a number of MBs. As can be seen in Figure 17, we need to group lost MBs together since 4-connected MBs create a lost group of MBs that have to be concealed together. So the proposed algorithm detects these groups of 4-connected lost MBs and treats them independently, one group at the time. Only at the end, when we utilize the assumption that the boundary is a single closed non-intersecting curve, do we actually consider all groups simultaneously. After the received α-plane has been split up into groups of 4-connected lost MBs, we conceal each group as follows.

If the group has only two connecting points, we use the concealment method discussed so far. The problem arises when there are more than two connecting points, which could also happen with only one MB, and in fact, this section applies to that case too. Since every line that goes into the group must come out of it, there is always an even number of connecting points. If we assume that a spline drawn from connecting point A to connecting point B is not the same as a spline drawn from connecting point B to connecting point A (in our case they are the same) then there are $N!$ possible splines, if there are N connecting points. This is the case, since $N!$ is the number of possible permutations of the points, and each permutation represents a possible way of connecting up points by splines if the first point is connected with the second point, the third point with the fourth and so on. Since A to B is the same as B to A in our case, the number is smaller, that is, $K = \frac{N!}{2^{N/2}(N/2)!}$. This is still a rather large number; for example, for $N = 2, 4, 6, 8, 10$, $K = 1, 3, 15, 105, 945$. Based on our initial assumptions that the boundary is not self-intersecting, excluding all intersecting combinations further reduces the value of K.

This reduction is based on the fact that if straight-line connections intersect, then spline connections will intersect too. On the other hand, if straight-line connections do not intersect, does not mean that spline connections will not intersect. First, we exclude all possible combinations of connecting points that result in straight-line intersections. We derive this recursively, by starting with 2 connecting points as shown in Figure 18 in the first circle with two dots (the connecting points) and one dotted line (the straight-line connection between the connecting points). Note that we use a circle as a representative group of lost MBs and use the notation GX, where X is the number of connecting points. Since we collect the connecting points in an ordered (in our case clockwise) direction, the effective shape of the group does not matter and hence we use a circle for this derivation. For the 2 connecting points, there is only one way to connect them, which does not intersect. For 4 connecting points this is more interesting, since there are now 3 ways to connect them but, by inspection of the second row in Figure 18, only 2 of them do not intersect. For 6 connecting points there are 15 ways to connect them but by inspection of the third row in Figure 18, only 5 do not intersect. For 8 connecting points there are already 105 ways to connect them but only 14 of them do not intersect. This is not shown anymore in Figure 18, but instead we use the trees drawn in Figure 19 to explain how we derived the number 14. The G4 tree is made up of 2 branches of G2s. This is based on the observation that the dashed line connecting the uppermost point in the circle with another point can be in two positions. Hence we have two branches in the G4 tree. Now if a point is connected to its neighbor in the counter clockwise direction, then we have left 2 points on the right, which form a G2 group. If it is connected to its neighbor in the clockwise direction, then we have left 2 points on the left, which again form a G2 group. Hence, through this tree, a G4 group can be represented recursively in terms of G2 groups. Since there is only one valid combination per G2 group and the G4 group is made up of 2 G2 groups, there are 2 valid G4 combinations for lines that do not intersect. This is the same result we obtained in the second row of Figure 18. This idea can be extended to the G6 group below the G4 group in Figure 19. The dashed line now creates 3 branches, where the

leftmost branch is a G4 group, the middle branch is a product of 2 G2 groups and the rightmost branch is another G4 group. Hence the number of combinations for a G6 group is 2+1*1+2 = 5, which again we can also see in row 3 of Figure 18. Clearly this recursive argument can be also used for a G8 and a G10 group, which is done in Figure 19. The G8 group results in 14 combinations and the G10 group in 42. In practice, such a large number of connecting points per group is rather unlikely but the presented recursive procedure not only can be used to count the number of combinations, but also to draw them. We need to draw all these combinations, since the fact that a straight-line does not result in an intersection on a circle does not guarantee that splines in an arbitrarily shaped group of 4-connected lost MBs do not intersect. We test this by drawing the splines of all these combinations and only keeping the ones which do not intersect each other. This is shown for the loss pattern of Figure 17 in Figures 20 and 21. In Figures 22 and 23 we show the resulting reconstructed boundary. Note that in Figures 20 and 21 the concealment spline for the right upper group is unique, since it is a G2 group, while for the left lower group, 2 possible solutions exist since it is a G4 group and the splines did not intersect. As mentioned before, we collect each non-intersecting solution per group and all combinations of all these solutions are then considered at the end to select a final reconstructed boundary. In our example we have two groups, the first of which created one solution and the second, two solutions. Therefore the total number of solutions is 1*2=2, as seen in Figures 20 and 21.

After concealing splines have been generated for all groups, a final solution needs to be selected. We use the assumption that the boundary is a closed curve and check for all solutions (after the received boundary and the concealment splines have been concatenated as shown in Figures 22 and 23) to see if they represent a closed boundary. Since it is possible that there is more than one closed boundary among the feasible solutions, we select the shortest one. It remains to fill the interior of the final reconstructed boundary with 1s and set everything else to 0, which results in the reconstructed α-plane, shown in Figure 24. In Figure 25 we show the incorrectly concealed pixels, which will be counted in the next section to arrive at a performance measure for the proposed concealment algorithm.

IV. Experimental results

Clearly, there are many different ways to quantify the performance of the proposed concealment algorithm. As an absolute measure, we use the total number of incorrectly concealed pixels in the concealed α-plane. If many MBs are lost though, this number tends to be higher than when few MBs are lost. Therefore we also use a relative measure, the ratio of the incorrectly concealed pixels divided by the total number of lost pixels. Unfortunately, both objective measures do not take into consideration the smoothness of the concealment, which in the proposed approach is inherently given and which is important to a human observer. Such measures, however, are difficult to devise and hence we will simply show several examples of reconstructed boundaries. Objectively we measure the performance of the concealment algorithms either in absolute terms, i.e., the number of incorrectly concealed pixels, or in relative terms, using the ratio of the number of incorrectly concealed pixels over the total number of missing pixels.

Before we can run tests on the proposed concealment algorithm, we need a model for the packet network, particularly for the packet loss. We assume that the Internet would be used to transport the data packets as RTP/UDP/IP packets. Therefore we use a model studied in [10] and shown in Figure 26 as the packet loss model. This model is a two state Markov chain, where the state “L” is the loss state and the state “NL” is the not-loss state. In the loss state, the packet is lost and in the not-loss state, the packet arrives at its destination. This model allows for correlated losses and one of the parameters of the model is the so called conditional loss probability, *clp*, which is the probability that a packet is lost, given that the previous packet was lost. This is equivalent to the probability that the current state is the loss state, given that the previous state was the loss state. The higher this probability, the more correlated the losses are. The other parameter of the model is the unconditional loss probability, *ulp*, which is the overall probability that a packet is lost, or equivalently, that the Markov chain is in the loss state. For the presented simulation results, we use the parameters given in Table 3 of [10]. We selected the third column, which represents a *ulp* of 12% and a *clp* of 27%. This corresponds to a relatively heavily loaded Internet. We also use the parameters *ulp* of 4% with a *clp* of 27% which represents a relatively lightly loaded Internet. Such lighter congestions were not studied in [10], but the trend in Table 3 of [10] is such that the *clp* is getting smaller as the *ulp* gets smaller. By not reducing the *clp* but keeping it equal to 27% corresponds therefore to a worst case scenario *clp* for the 4% *ulp* Internet.

Next, we need a model of how the MBs are put into RTP/UDP/IP packets. MPEG-4 uses a concept called application layer framing, by defining video packets (VP). In application layer framing, the application layer manages how the data is put into transport packets and the application is also responsible for synchronization. A video packet can contain one or more MBs and it is the smallest unit of synchronization, since the MBs do not carry an MB address. In the experiments, we assume two packetization modes.

In the MB mode, one MB is put into one VP and that VP is put into one RTP/UDP/IP packet. This mode is inefficient, since the overhead of the VP/RTP/UDP/IP packet is now added to each MB. It would be more efficient to put several MBs into one VP/RTP/UDP/IP packet. On the other hand, this mode is good for concealment, since most of the time only one MB is lost and the neighboring MBs are available for the concealment scheme. We use this mode as a baseline to show what the algorithm could achieve in the best packetization case.

In the slice mode, we put all MBs of a row (a so called slice) into one VP which in turn is put into one RTP/UDP/IP packet. This mode is clearly more efficient, since the overhead is amortized over several MBs. Unfortunately this efficiency makes the concealment of a lost packet more difficult, since now MBs are lost in slices, which means there is a larger connected area of lost data that needs to be concealed. This model though is more realistic to the packetization done in MPEG-4 systems.

In an error prone packet-based environment, it is important, that the information in a given video packet is independent of the information in any other video packet. If this is not the case, error propagation occurs as pointed out in [11]. As also suggested in [11], in MPEG-4 shape coding this independence is achieved if the intermode is not used. Hence in our experiments, we only use intramode shape coding.

The proposed concealment algorithm is compared to a modified MAP and a modified median algorithm. These two methods are modified versions of the algorithms introduced in [7]. The modification consists of first scanning the α-plane for missing MBs of which all four neighbors (up, down, left, right) have been received such that the missing MB can be filled in with 1s, as shown in Figure 27. This idea can be extended to include cases where only three neighboring blocks have been received, as shown in Figure 28. Clearly we can also fill in the missing MBs in this case. Note that we then apply this filling-in iteratively until there are no more MBs that can be filled in with this simple method. For example, this iteration allows us in Figure 28 to also fill in the middle MB, which has only two solid sides at the beginning, since after the first pass, its two missing neighbors have been filled in and hence it can also be filled. As also pointed out in [7], the inability to fill such holes is a major drawback of the median method and the proposed modification solves this problem and makes the median method much more competitive to the MAP method. Since we use this simple pre-processing for the median method, we also use it for the MAP method so as to have a fair comparison.

As the name suggests, the median method replaces a missing pixel with the median of a neighborhood. We use the same 16-pixel neighborhood as proposed in [7]. This median filtering is repeated iteratively until the concealed α-plane does not change anymore. The MAP method uses a Markov random field (MRF) as the image model. In [7] the model is selected such that the resulting estimator is simply a weighted version of a median filter. The weighting is proportional to the frequency of occurrence of boundary lines (of a given angle) in the four neighboring MBs that would extend into the missing MB. In other words, the pixels in the missing MB are selected such that dominant edges are conserved.

Both, the modified median method and the modified MAP method are based on an iterative procedure. The iteration converges, when the concealed α-plane does not change anymore. While this median based iteration is straight forward to implement, it is also relatively slow to execute and to converge. This is true even though we do not sort the data to find the median (since the α-plane is a binary image), but simply count the numbers of 1s in the 16-pixel neighborhood . If this number is greater or equal to eight, we set the value to 1, otherwise we set it to 0. Nevertheless these iterative methods are two to four times slower than the proposed Hermite spline method, for which no iteration is needed. Note that all three algorithms are not optimized for speed and hence the above ratio could change somewhat when optimized implementations are compared. This ratio is also not a fixed number, but changes, since the number of operations of all methods depends on the particular loss pattern and the particular α-plane.

Tables I and II summarize the performance of the proposed Hermite Spline, the modified MAP and the modified median algorithms. The results reported are the averages over 100 realizations for each unconditional loss probability (4% and 12%), packetization mode (MB and slice), concealment method (median, MAP and Hermite) and for every third frame of the 300 frames QCIF sequences Bream, Akiyo and Weather. From these tables, it becomes clear that the proposed Hermite spline method significantly outperforms the other two methods using both, the absolute and relative objective measures. It is interesting to note that the relative performance measure is almost independent of the packet loss rate.

Often more important than the numerical objective results are the visual subjective ones. Since the proposed method has the smoothness of the reconstructed boundary as a built-in feature, it comes as no surprise that the concealed α-plane created by the Hermite spline method is visually pleasing. To illustrate this, we compare the Hermite spline method with the modified median method, which, in our experiments, consistently outperformed the modified MAP method. For this comparison, we use the slice loss pattern shown in Figure 29. In Figure 30, the resulting concealed α-plane by the modified median method is shown, and in Figure 31, we see the corresponding error pixels. In Figure 32, the resulting concealed α-plane by the proposed Hermite spline method is shown, and in Figure 33 the corresponding error pixels are shown. From this visual subjective inspection of the resulting concealed α-plane, the smooth boundary resulting from the proposed method is clearly the more pleasing one. To further illustrate this point, we show four more reconstructed boundaries (Figures 34, 35, 36, and 37) of the bream object on top of the corresponding lost MBs.

V. Summary and conclusions

In this paper we propose a novel shape error concealment method based on Hermite splines. We use second order Hermite splines to estimate the velocities at the connecting points, and a cubic Hermite spline matching these two velocities to conceal the lost boundary segment. We propose a solution when there are more than two connecting points per group using a recursive generation of all possible non-intersecting solutions. Finally we collect all possible solutions that result from the local decisions made during the concealment of the groups and make a final decision on the reconstructed boundary based on the assumption that the boundary is a closed non-intersecting curve.

The proposed shape concealment method interprets shape information in terms of a closed non-intersecting boundary. It explores the known shape information on a relatively high level such as velocity along a curve, thus making better use of the available information than the statistical methods. The Hermite spline method experimentally outperformed the modified median and the modified MAP methods based on objective performance metrics. What is even more important is the improvement in subjective quality, which is difficult to measure. The obtained results demonstrated that the Hermite

spline method is also superior to the statistical methods based on subjective evaluations.

REFERENCES

[1] ISO/IEC JTC1/ SC29/ WG11 N2502, "Text for ISO/IEC FDIS 14496-2 Visual," November 1998.

[2] A. Puri and T. Chen, Ed., *Multimedia Systems, Standards, and Networks*. Marcel Dekker, Inc, 1999.

[3] Y. Wang, S. Wenger, J. Wen, and A. G. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Processing Magazine*, vol. 17, no. 4, pp. 61–82, July 2000, special issue on Multimedia Communications over Networks.

[4] M. R: Frater, W.S. Lee, M. Pickering, and J.F. Arnold, "Error concealment of arbitrarily shaped video objects," in *Proceedings of the International Conference on Image Processing*, vol. 3, Chicago, Illinois, USA, October 1998, pp. 507–511.

[5] P. Salama and C. Huang, "Error concealemt for shape coding," in *Proceedings of the International Conference on Image Processing*, vol. 2, Rochester, New York, USA, September 2002, pp. 701–704.

[6] M-J. Chen, C-C Cho, and M-C. Chi, "Spatial and temporal error concealment algorithms of shape information for MPEG-4 video," in *IEEE International Conference on Consumer Electronics*, June 2002.

[7] S.Shirani, B. Erol, and F. Kossentini, "A concealment method for shape information in MPEG-4 coded video sequences," *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 185–190, September 2000.

[8] X. Li, G. M. Schuster, and A. K. Katsaggelos, "A recursive shape error concealment method," in *Proceedings of the International Conference on Image Processing*, Rochester, New York, USA, September 2002, pp. 22–25.

[9] ——, "Curve-fitting algorithms for shape error concealment," in *Proceedings of the 5th Nordic Signal Processing Symposium*, on board the MS Trollfjord from Tromso to Trondheim, Norway, October 2002, pp. 4–7.

[10] J.C. Bolot, "Characterizing end-to-end packet delay and loss in the Internet," *Journal of High-Speed Networks*, vol. 2, no. 3, pp. 305–323, December 1993.

[11] N. Brady, "MPEG-4 standardized methods for the compression of arbitrarily shaped video objects," *IEEE Transactions of Circuit and Systems for Video Technology*, vol. 9, no. 8, pp. 1170–1189, December 1999.

Guido M. Schuster received the Ing HTL degree in Elektronik, Mess- und Regeltechnik in 1990 from the Neu Technikum Buchs (NTB), Buchs, St.Gallen, Switzerland. He then received the M.S. and Ph.D. degrees, both from the Department of Electrical and Computer Engineering, Northwestern University, Evanston, Illinois, in 1992 and 1996, respectively. In 1996 he joined the Network Systems Division of U.S. Robotics in Mount Prospect, Illinois (later purchased by 3Com). He co-founded the 3Com Advanced Technologies Research Center and served as the Associate Director of the Center. He also co-founded the 3Com Internet Communications Business Unit and developed the first commercially available SIP IP Telephony system. He was promoted to the Chief Technology Officer and Senior Director of this Business Unit. During this time, he also served as an Adjunct Professor in the Electrical and Computer Engineering Department at Northwestern University. He is currently a Professor of Electrical and Computer Engineering at the Hochschule für Technik Rapperswil (HSR), Rapperswil, St.Gallen, Switzerland, were he focuses on Digital Signal Processing and Internet Multimedia Communications.

Guido M. Schuster holds 30 U.S. patents in fields ranging from adaptive control over video compression to Internet telephony. He is the co-author of the book "Rate-Distortion Based Video Compression", published by Kluwer Academic Publishers and has published 46 peer reviewed journal and proceedings articles. Furthermore he is the recipient of the gold medal for academic excellence at the NTB, the winner of the first Landis & Gyr fellowship competition, the recipient of the 3Com inventor of the year 1999 award and the recipient of the IEEE Signal Processing Society Best Paper Award 2001 in the multimedia signal processing area. His current research interests are operational rate-distortion theory and networked multimedia.

Xiaohuan Li received her B.S. degree from the Department of Automation of Tsinghua University in China in 2000 and her M.S. degree from the Department of Electrical and Computer Engineering of Northwestern University in U.S.A. in 2002. She is currently a Ph.D. candidate in the Center of Image and Signal Processing of the School of Electrical and Computer Engineering of Georgia Institute of Technology in U.S.A. Her research interests include MPEG/H.26L video coding and communication.

Aggelos K. Katsaggelos 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 both in electrical engineering from the Georgia Institute of Technology, in 1981 and 1985, respectively. In 1985 he joined the Department of Electrical and Computer Engineering at Northwestern University, where he is currently professor, holding the Ameritech Chair of Information Technology. He is also the Director of the Motorola Center for Communications and a member of the Academic Affiliate Staff, Department of Medicine, at Evanston Hospital.

Dr. Katsaggelos is a member of the Publication Board of the IEEE Proceedings, the IEEE Technical Committees on Visual Signal Processing and Communications, and Multimedia Signal Processing, the Editorial Board of Academic Press, Marcel Dekker: Signal Processing Series, Applied Signal Processing, and Computer Journal. He has served as editor-in-chief of the IEEE Signal Processing Magazine (1997-2002), a member of the Publication Boards of the IEEE Signal Processing Society, the IEEE TAB Magazine Committee, an Associate editor for the IEEE Transactions on Signal Processing (1990-1992), an area editor for the journal Graphical Models and Image Processing (1992-1995), a member of the Steering Committees of the IEEE Transactions on Image Processing (1992-1997) and the IEEE Transactions on Medical Imaging (1990-1999), a member of the IEEE Technical Committee on Image and Multi-Dimensional Signal Processing (1992-1998), and a member of the Board of Governors of the IEEE Signal Processing Society (1999-2001). He is the editor of Digital Image Restoration (Springer-Verlag 1991), co-author of Rate-Distortion Based Video Compression (Kluwer 1997), and co-editor of Recovery Techniques for Image and Video Compression and Transmission, (Kluwer 1998). He is the co-inventor of eight international patents, a Fellow of the IEEE, and the recipient of the IEEE Third Millennium Medal (2000), the IEEE Signal Processing Society Meritorious Service Award (2001), and an IEEE Signal Processing Society Best Paper Award (2001).

LIST OF FIGURES

LIST OF TABLES

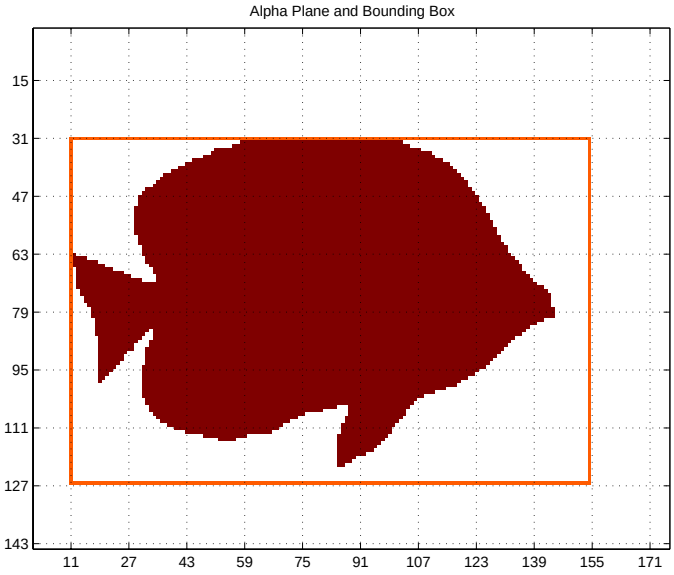

Fig. 1. α-plane of frame 0 of the bream object with bounding box

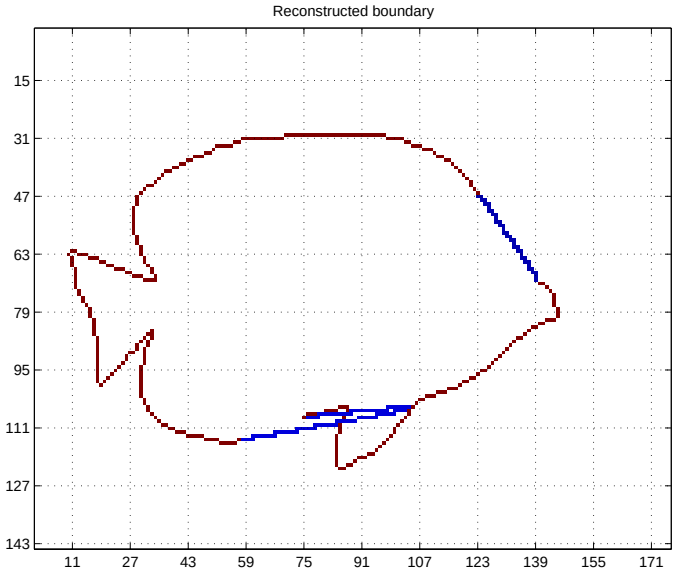

Fig. 3. Second order Bezier curve concealment of figure 2

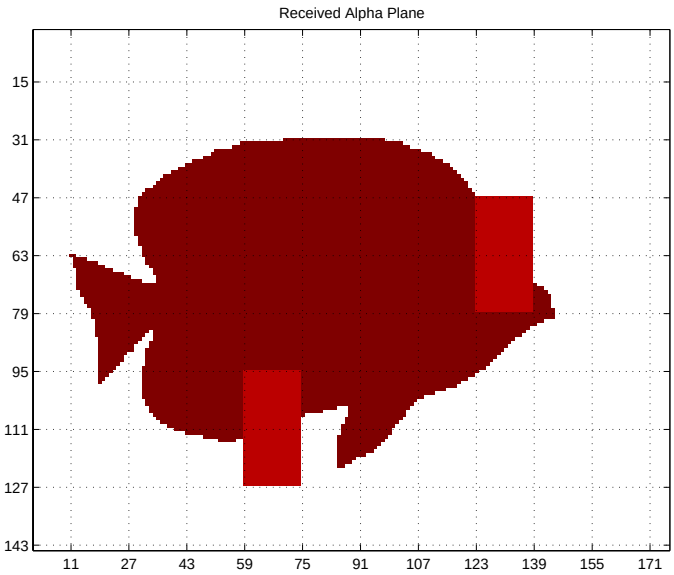

Fig. 2. Received α-plane with four macro blocks missing

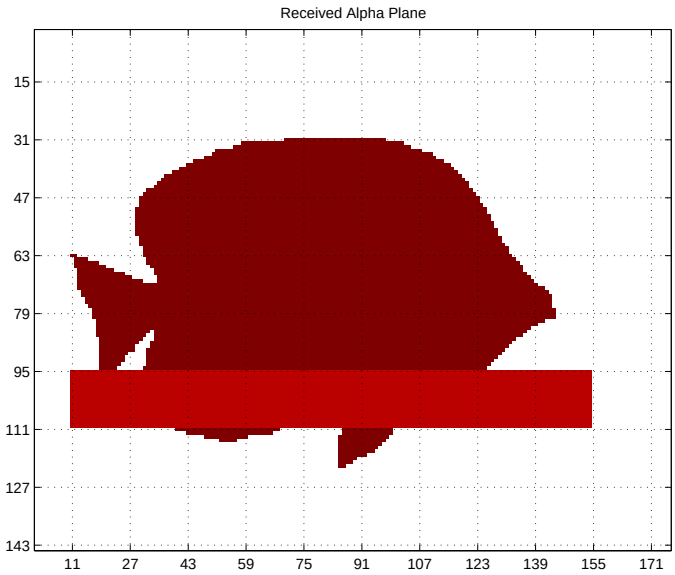

Fig. 4. Received α-plane with one slice missing

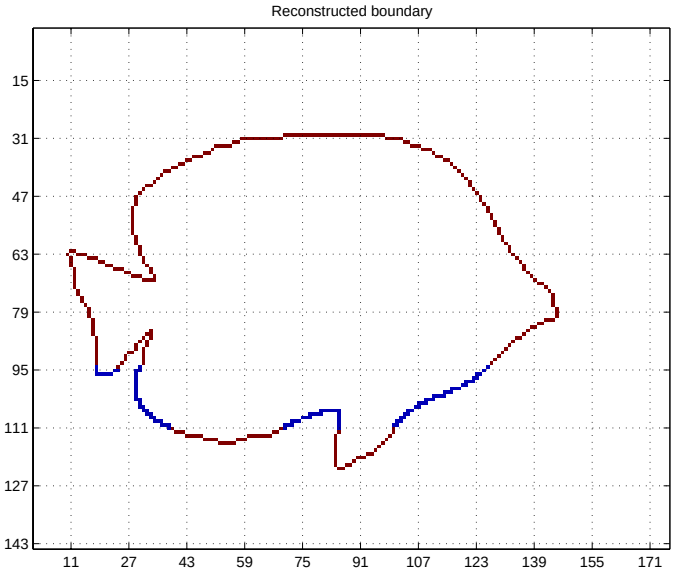

Fig. 5. One possible reconstructed boundary for figure 4

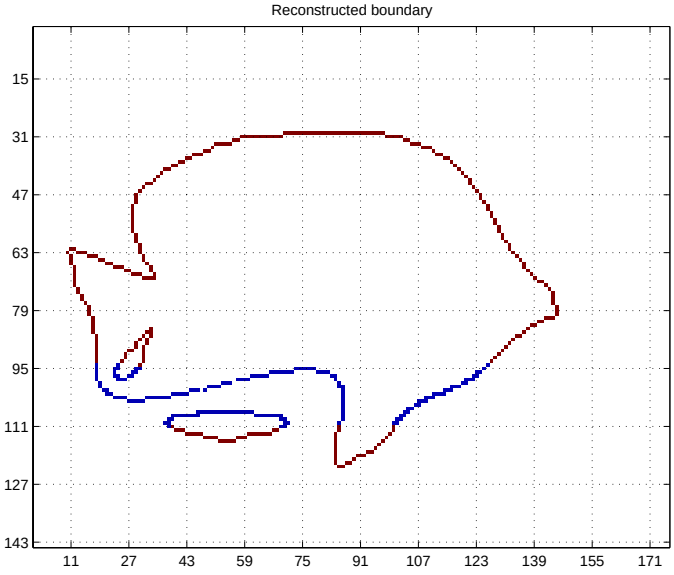

Fig. 6. Another possible reconstructed boundary for figure 4

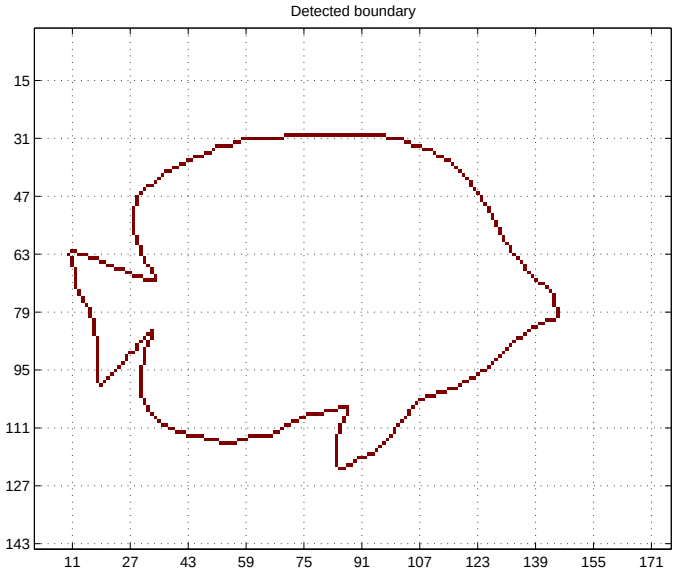

Fig. 7. Boundary of figure 1

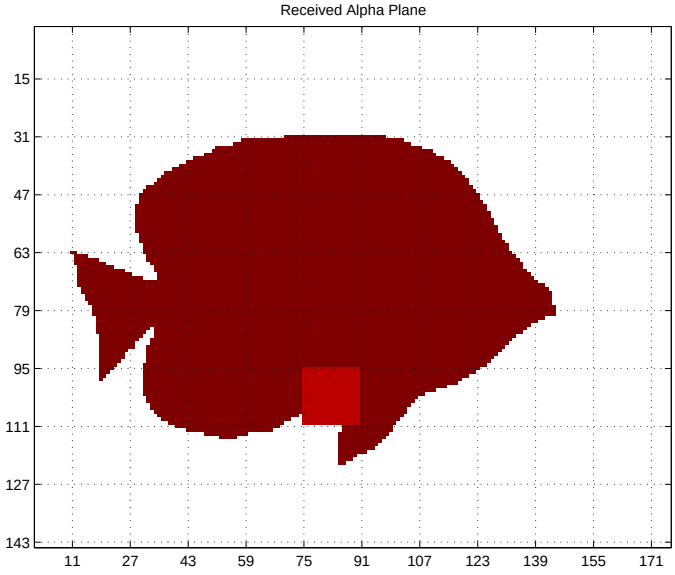

Fig. 8. α-plane with one missing macro block

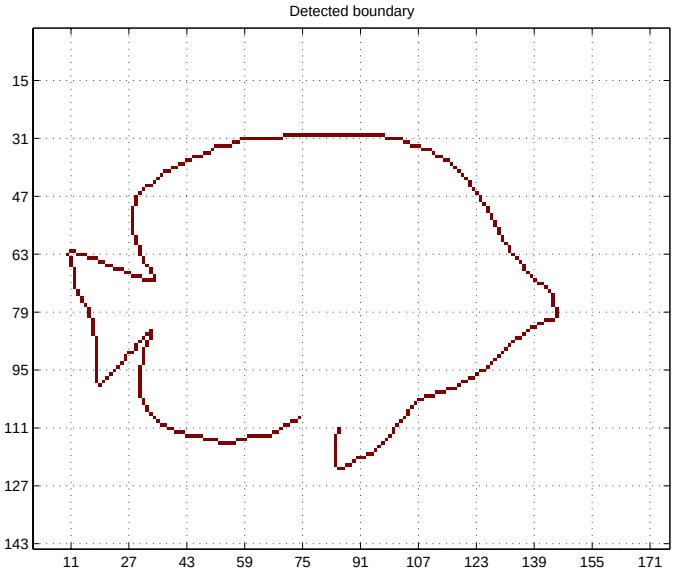

Fig. 9. Detected boundary of received α-plane in figure 8

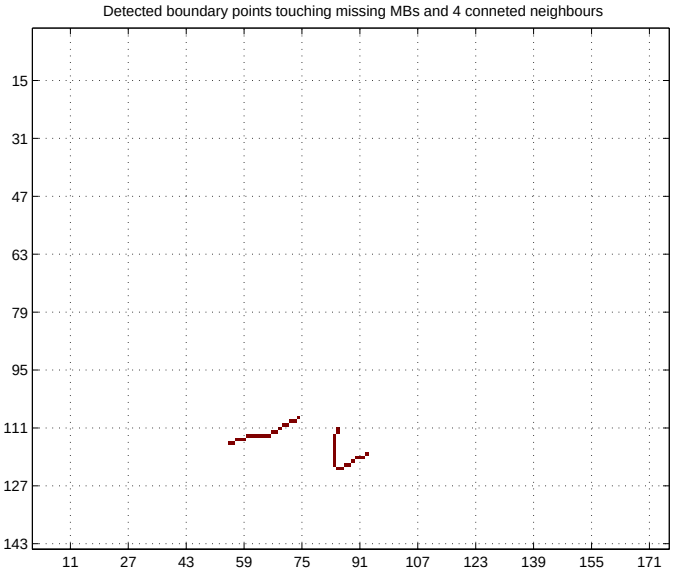

Fig. 11. Associated points set

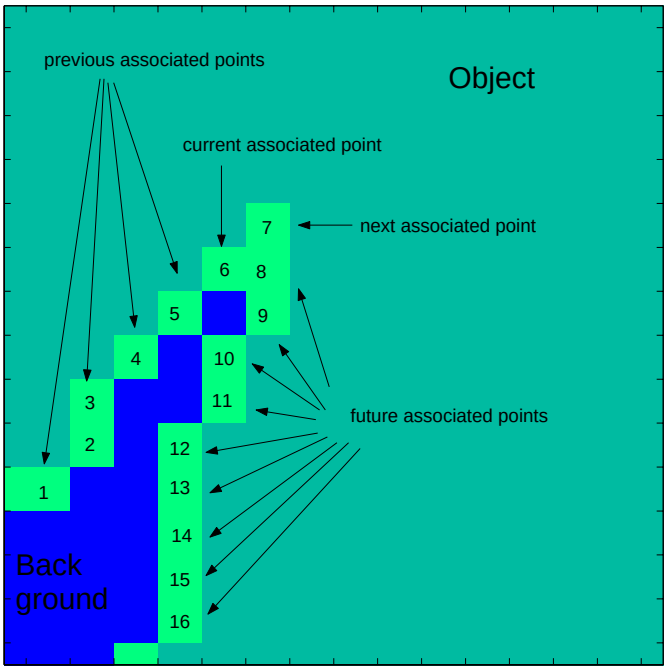

Fig. 10. The problem with sharp turns

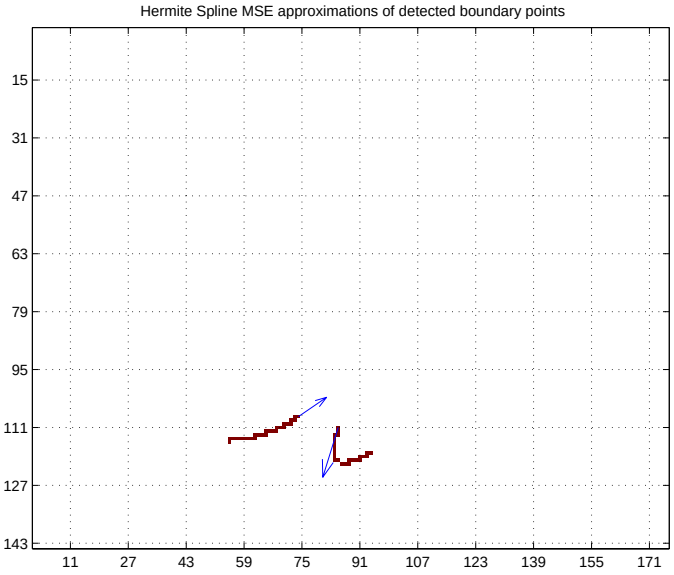

Fig. 12. Second order Hermite spline model of associated points set with velocity vectors amplified by a factor of 10

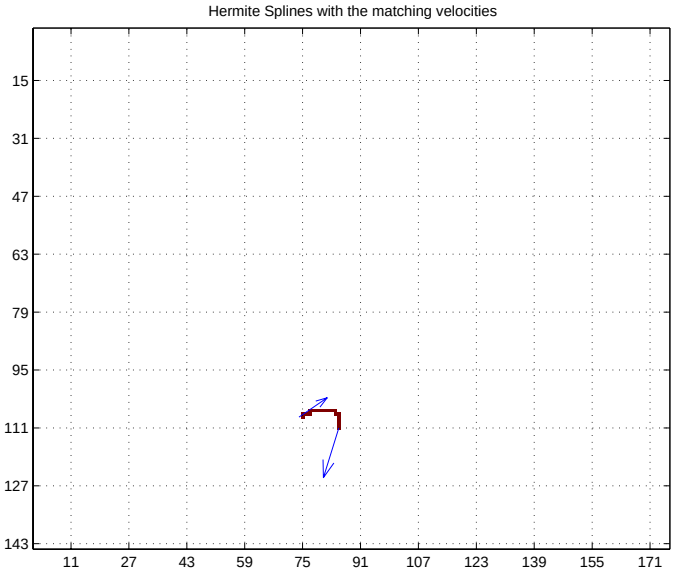

Fig. 13. Cubic Hermite spline, velocity amplified by 10

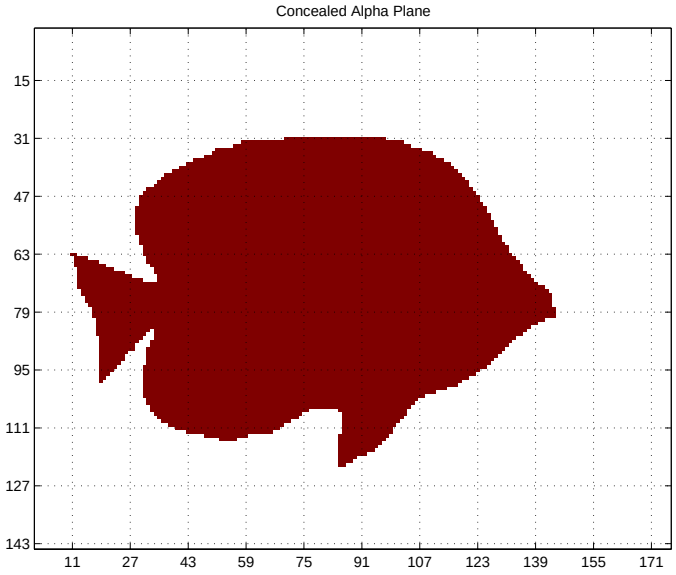

Fig. 15. Concealed α-plane

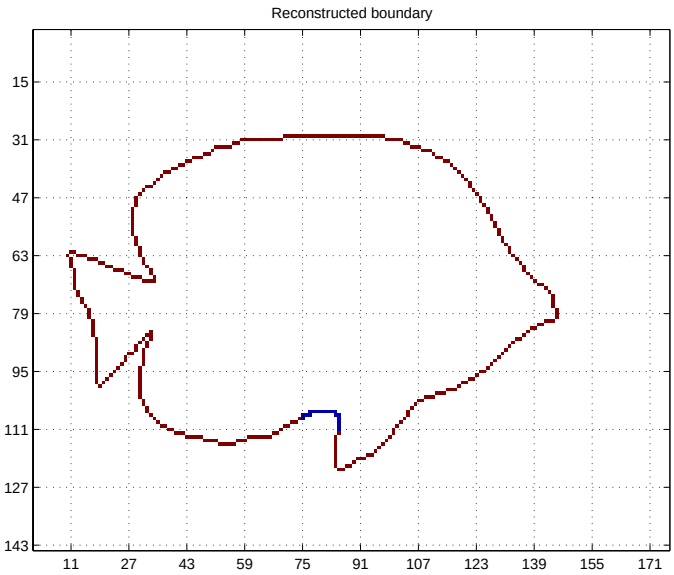

Fig. 14. Reconstructed boundary

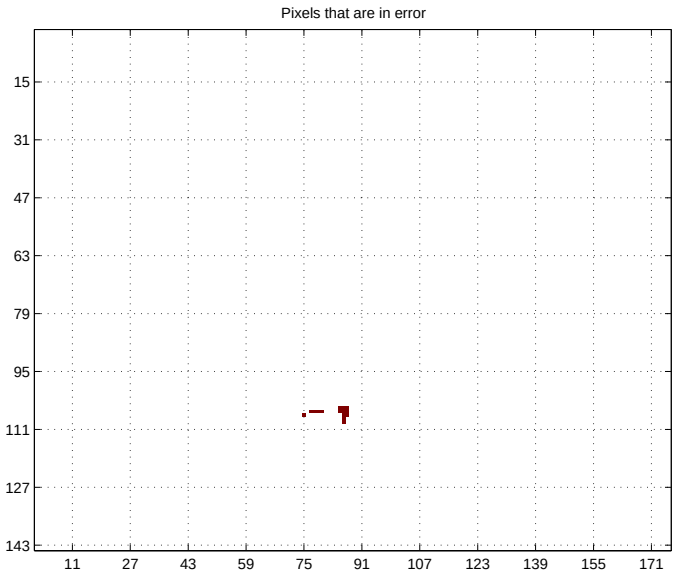

Fig. 16. Pixels that are in error

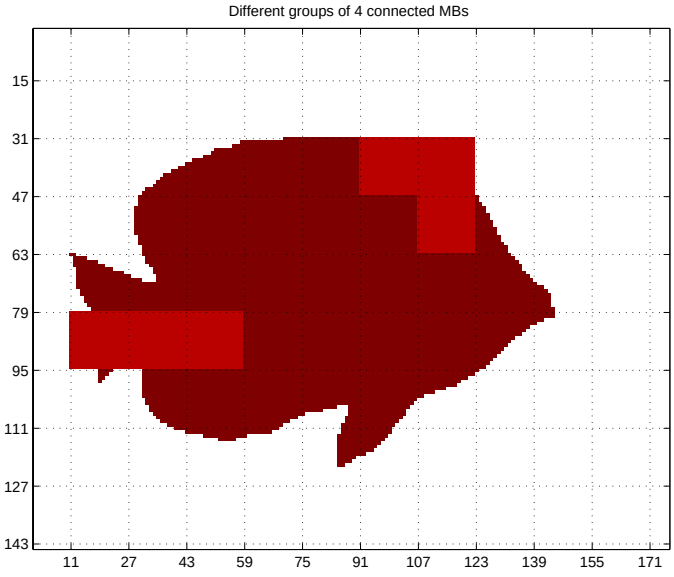

Fig. 17. Two 4-connect groups of lost macro blocks

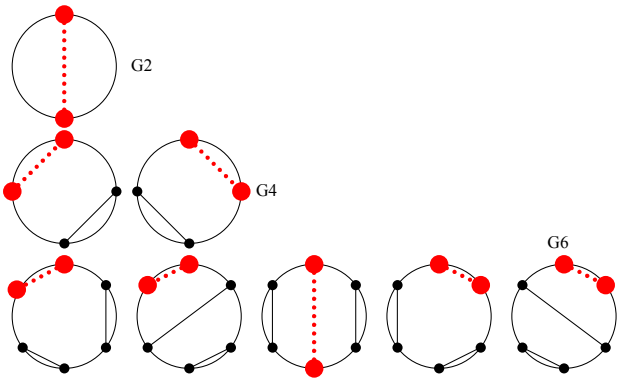

Fig. 18. Different groups GX with different numbers of connecting points X=2, 4 or 6 and all possible non-intersecting combinations

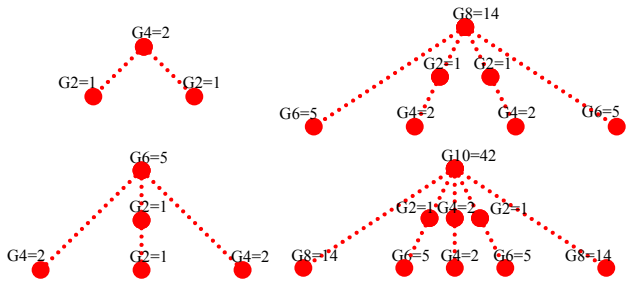

Fig. 19. Recursive calculation of the number of non-intersecting combinations for a given group

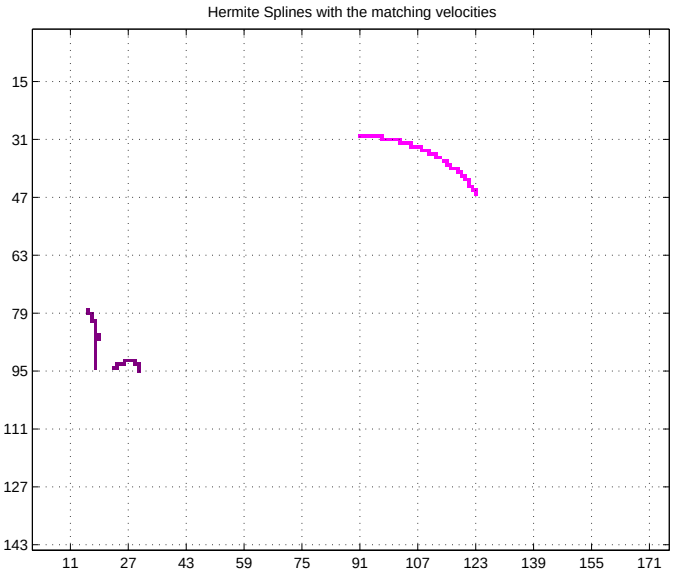

Fig. 20. One combination of non-intersecting concealment splines

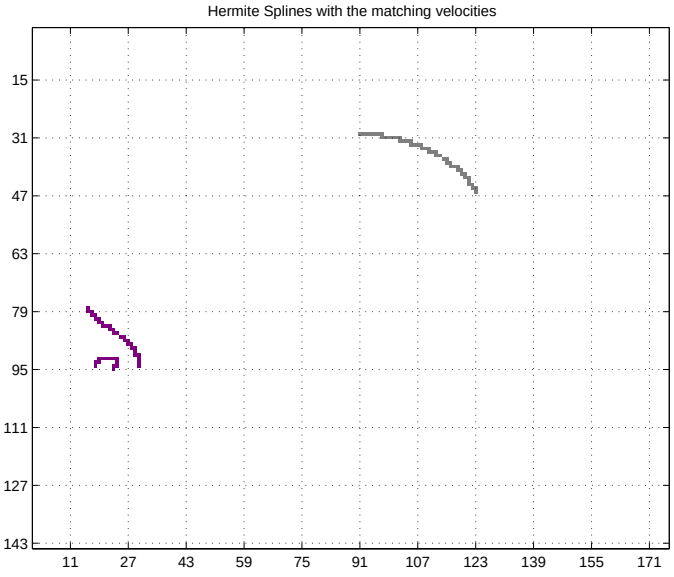

Fig. 21. Another combination of non-intersecting splines

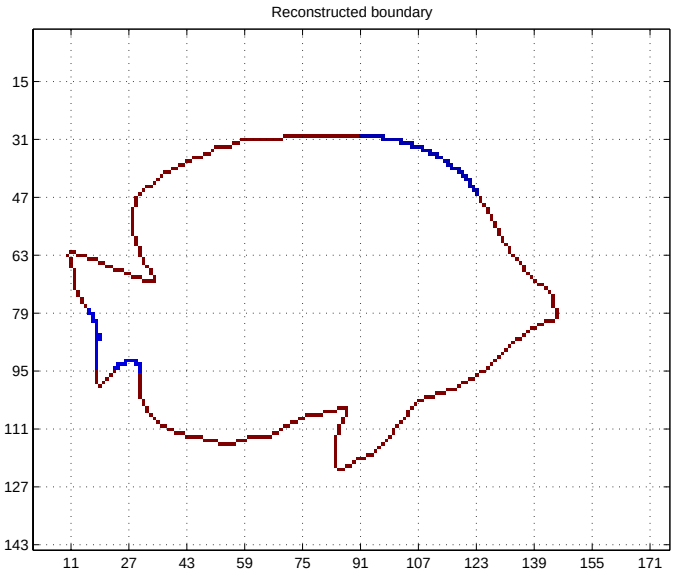

Fig. 22. Reconstructed boundary using the concealment splines of figure 20

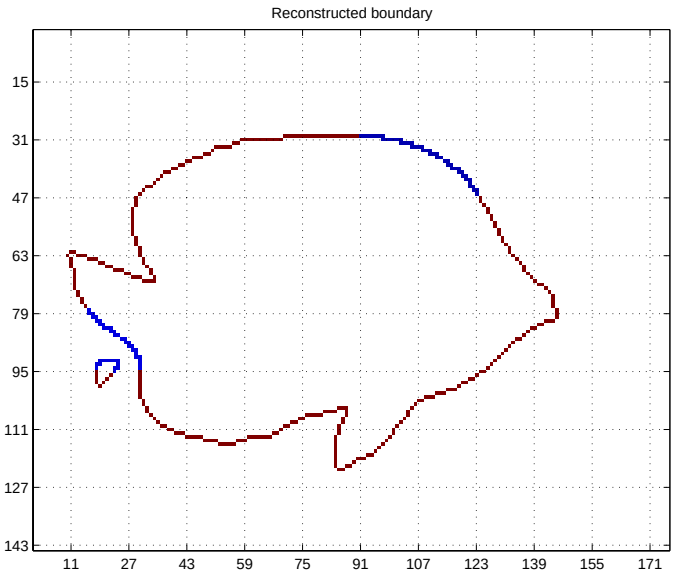

Fig. 23. Reconstructed boundary using the concealment splines of figure 21

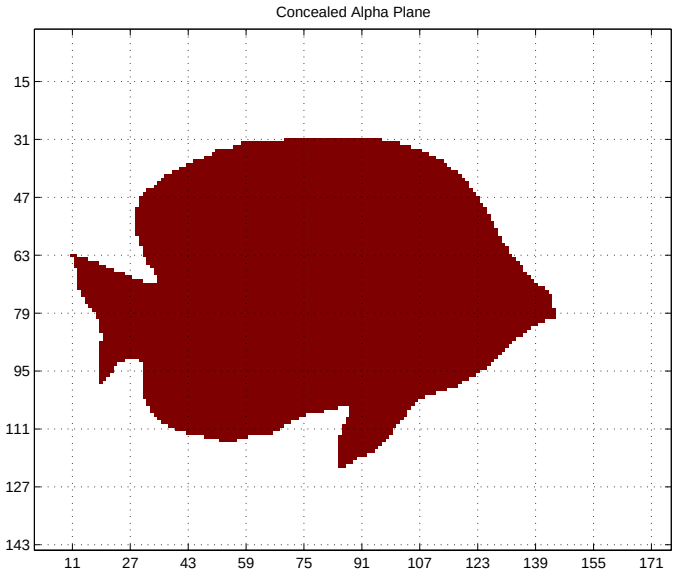

Fig. 24. Concealed α-plane

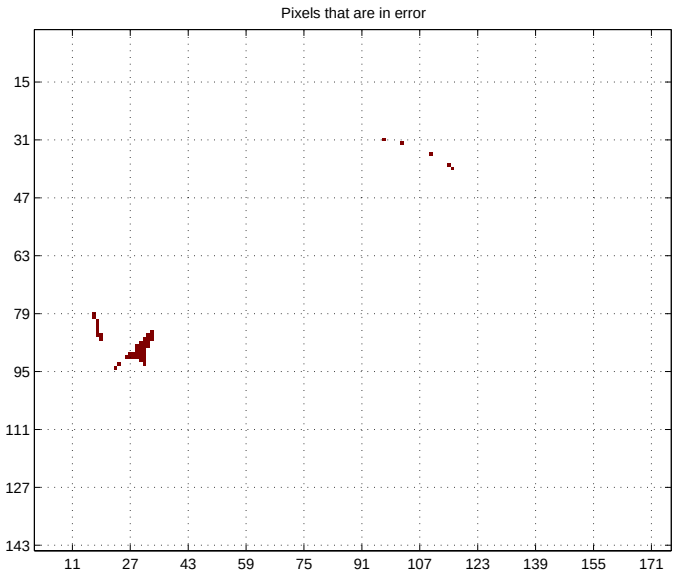

Fig. 25. Error pixels

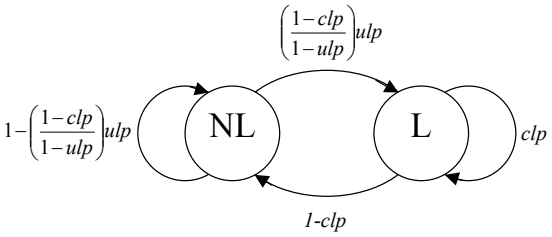

Fig. 26. Markov model of packet loss

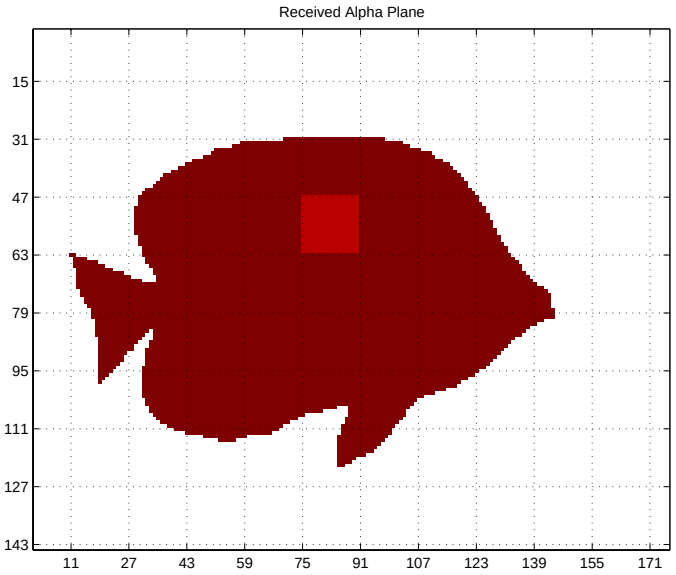

Fig. 27. Received α-plane where the lost MB can be filled before the median or MAP method is applied

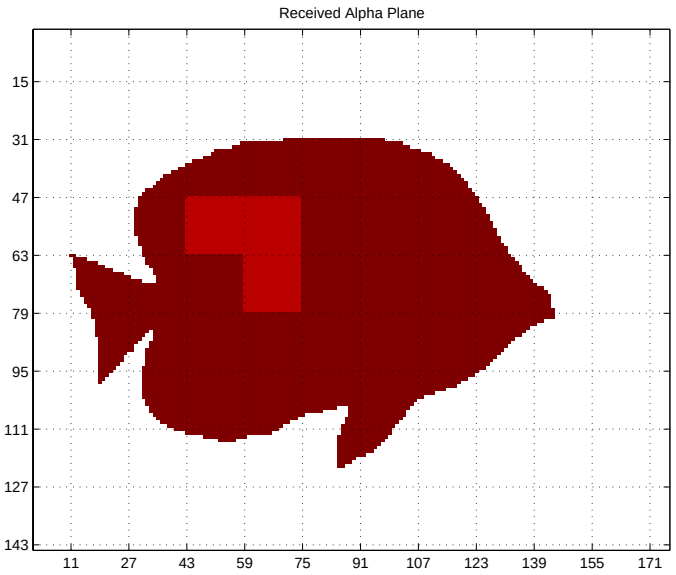

Fig. 28. Received α-plane where the lost MBs can be filled before the median or MAP method is applied

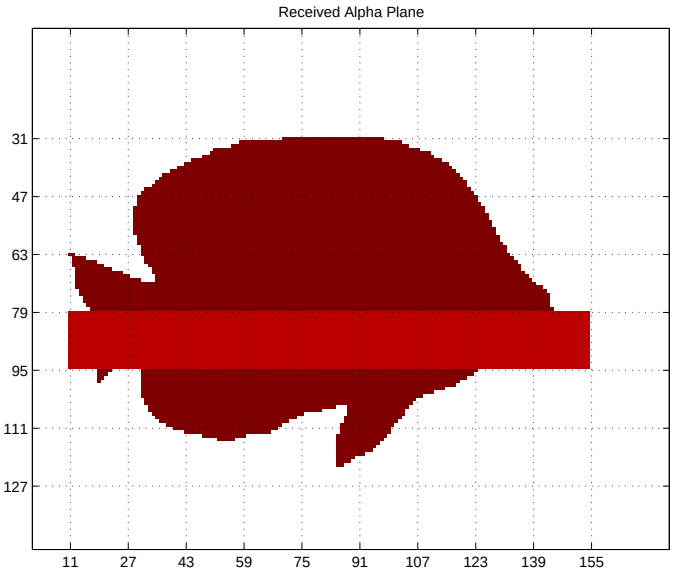

Fig. 29. Received α-plane with a slice loss pattern

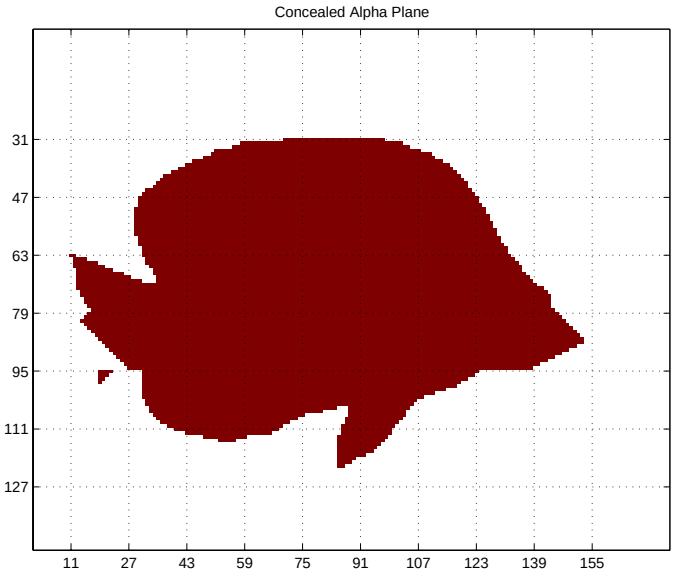

Fig. 30. Concealed α-plane using the modified median method

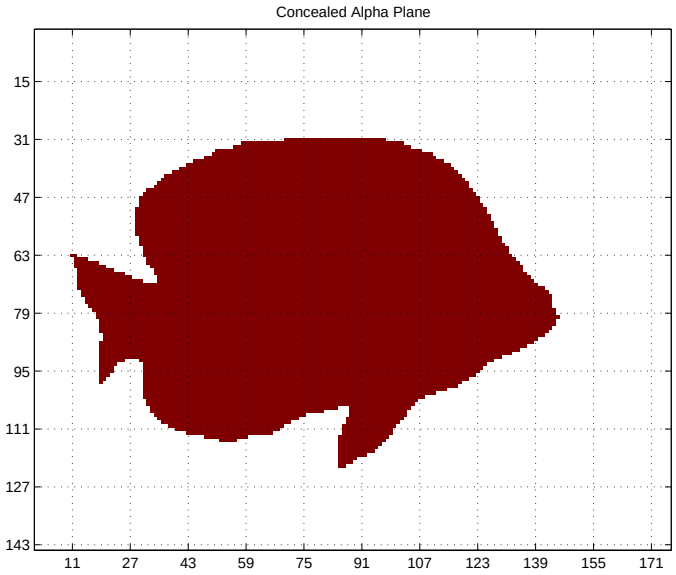

Fig. 32. Concealed α-plane using the Hermite Spline method

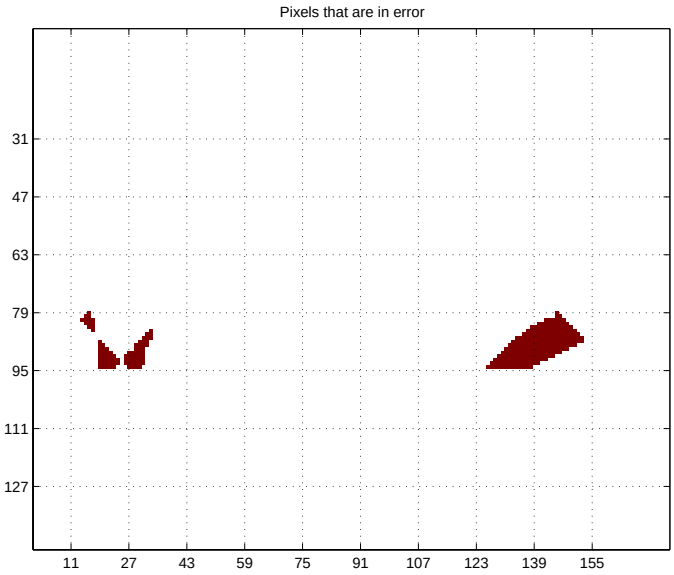

Fig. 31. Error pixels for the modified median method. Of the 2304 lost pixels, 299 are incorrectly concealed which corresponds to 13.0%

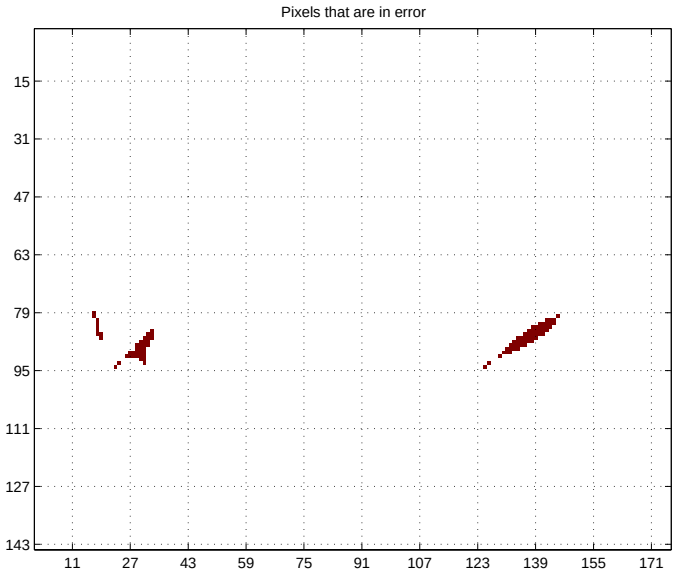

Fig. 33. Error pixels for the Hermite Spline method. Of the 2304 lost pixels, 95 are incorrectly concealed which corresponds to 4.1%

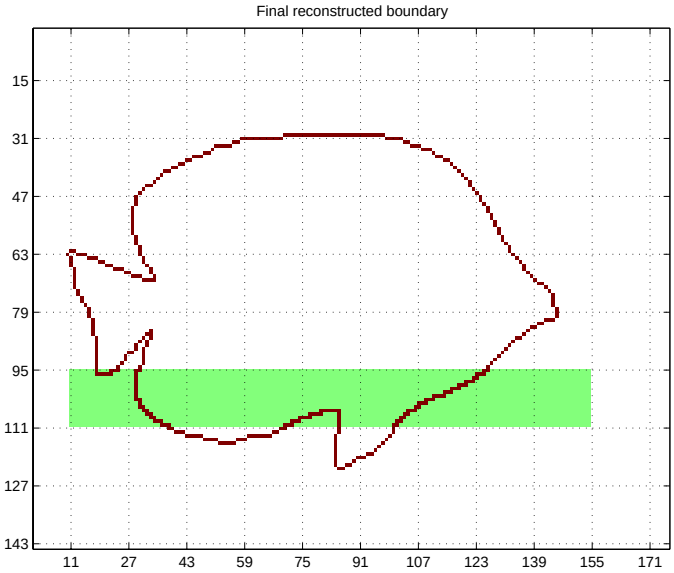

Fig. 34. First reconstructed boundary drawn on top of the missing MBs to illustrate the visually pleasing reconstruction

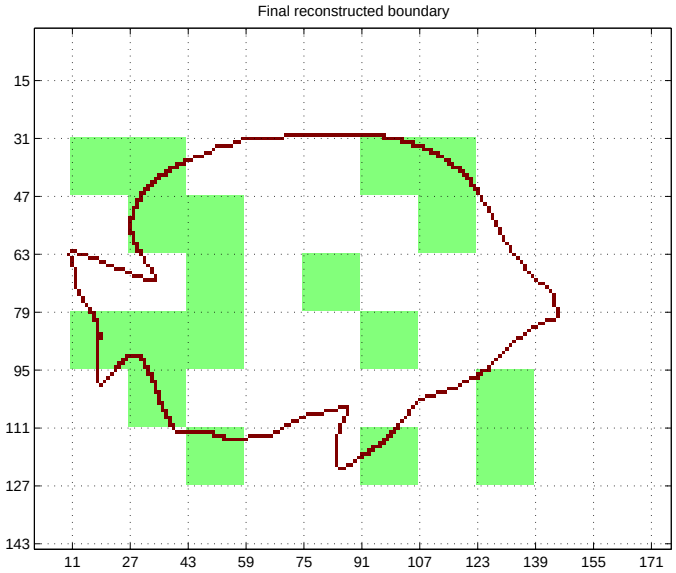

Fig. 36. Third reconstructed boundary drawn on top of the missing MBs to illustrate the visually pleasing reconstruction

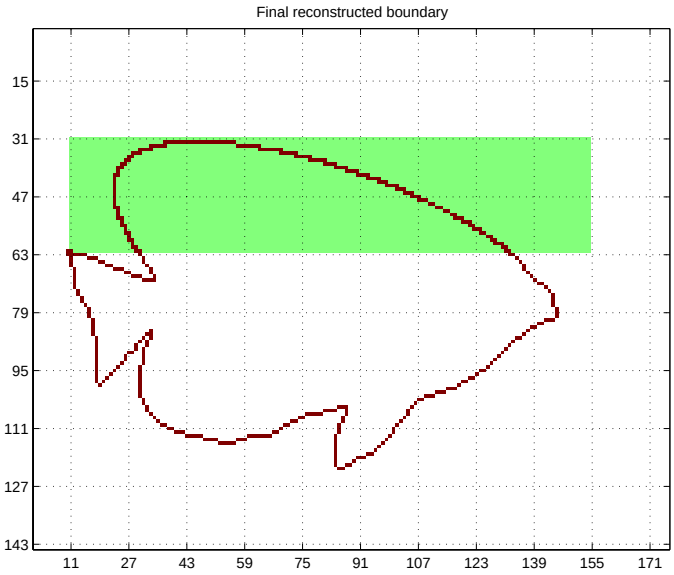

Fig. 35. Second reconstructed boundary drawn on top of the missing MBs to illustrate the visually pleasing reconstruction

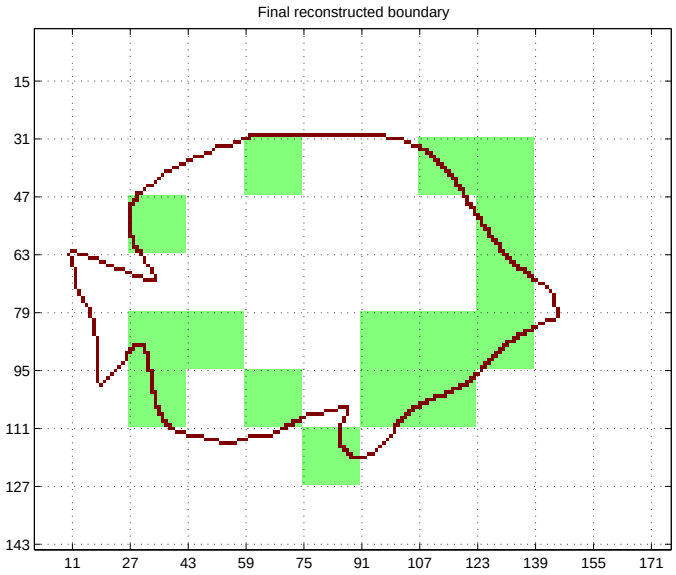

Fig. 37. Fourth reconstructed boundary drawn on top of the missing MBs to illustrate the visually pleasing reconstruction

Experiment	W.	B.	A.	Mean
MB4%MAP	40	27	33	33
MB4%Median	33	29	23	28
MB4%Hermite	16	18	9	14
MB12%MAP	128	94	116	112
MB12%Median	110	97	85	97
MB12%Hermite	54	59	30	48
Slice4%MAP	87	87	127	100
Slice4%Median	74	78	111	88
Slice4%Hermite	54	44	38	45
Slice12%MAP	255	271	416	314
Slice12%Median	218	243	364	275
Slice12%Hermite	161	137	133	144

TABLE I

AVERAGE NUMBER OF INCORRECTLY CONCEALED PIXELS OVER 100 REALIZATIONS FOR EACH PACKETIZATION SCHEME, UNCONDITIONAL LOSS PROBABILITY, CONCEALMENT METHOD AND FOR EVERY THIRD FRAME OF THE 300 FRAMES QCIF SEQUENCES WEATHER (W.), BREAM (B.) AND AKIYO (A.)

Experiment	W.	B.	A.	Mean
MB4%MAP	9.1%	5.5%	4.4%	6.4%
MB4%Median	7.5%	5.8%	3.1%	5.5%
MB4%Hermite	3.7%	3.6%	1.2%	2.8%
MB12%MAP	9.8%	6.2%	5.3%	7.1%
MB12%Median	8.5%	6.4%	3.9%	6.3%
MB12%Hermite	4.2%	4.0%	1.4%	3.2%
Slice4%MAP	19.9%	17.8%	17.9%	18.5%
Slice4%Median	17.1%	15.9%	15.6%	16.2%
Slice4%Hermite	12.3%	9.0%	5.3%	8.9%
Slice12%MAP	19.8%	18.1%	18.8%	18.9%
Slice12%Median	16.9%	16.3%	16.4%	16.5%
Slice12%Hermite	12.5%	9.1%	6.0%	9.2%

TABLE II

AVERAGE NUMBER OF INCORRECTLY CONCEALED PIXELS DIVIDED BY THE AVERAGE TOTAL NUMBER OF LOST PIXELS. THE AVERAGES ARE TAKEN OVER 100 REALIZATIONS FOR EACH PACKETIZATION SCHEME, UNCONDITIONAL LOSS PROBABILITY, CONCEALMENT METHOD AND FOR EVERY THIRD FRAME OF THE 300 FRAMES QCIF SEQUENCES WEATHER (W.), BREAM (B.) AND AKIYO (A.)