TEMPORAL SHAPE ERROR CONCEALMENT BY GLOBAL MOTION COMPENSATION

Luis Ducla Soares, Fernando Pereira
Instituto Superior Técnico / Instituto de Telecomunicações

ABSTRACT

In this paper, an original temporal shape error concealment technique, to be used in the context of object-based video coding schemes, is proposed. In this technique, it is assumed that the shape of the corrupted object at hand is in the form of a binary alpha plane, in which some of the shape data is missing due to channel errors. Additionally, the technique considers that the changes, in consecutive time instants, of the shape of typical video objects can be described by a global motion model. This way, when errors occur and corrupt the shape data in several places, global motion parameters can be estimated at the decoder from the corrupted alpha plane. Then, the decoder can easily (global) motion compensate the shape data from the previous time instant and restore the corrupted shape, thus improving the final subjective impact.

1. INTRODUCTION

With the availability of the MPEG-4 object-based audiovisual coding standard [1], new multimedia services and devices are making their way into the market. In particular, there is an increasing interest in mobile multimedia services, such as mobile videotelephony and mobile video streaming, also based on scenes understood as a composition of objects. However, mobile networks are highly error-prone environments, which makes it difficult, if not impossible, to transmit highly compressed video with an acceptable quality without appropriate error concealment techniques, that deal with both shape and texture data.

While (frame-based) texture concealment schemes are abundant in the literature and can with more or less adjustments be adapted to work for object-based video, shape concealment schemes are just starting to appear. Nevertheless, several techniques have already been proposed. In particular, the one proposed in [2] is based on the idea that the changes of the contour (i.e., the boundary of the shape data) of a given video object, in consecutive time instants, can be described by a global motion model. Based on this assumption, the global motion parameters are estimated at the encoder and sent along with the encoded bitstream to the decoder. This way, when errors occur and cause the contour to be broken in several places, the decoder can easily (global) motion compensate the contour from the previous time instant using the information sent by the encoder and restore the corrupted contour. To recover the shape, the decoder has simply to fill in the concealed contour.

The main drawback of this technique is that the global motion estimation parameters are computed at the encoder, when the sequence is being encoded. Since these parameters are needed at the decoder for the concealment itself, they must be transmitted somehow. To do so, a separate stream is used, called USER_DATA stream in the MPEG-4 nomenclature [1], which according to the authors of [2] represents a 5% increase in terms of bit rate. This can be a very serious limitation because the technique can then only be used if both the encoder and decoder support the technique, which is very unlikely in such a competition-driven market with many manufacturers. Moreover, even if two different manufacturers decide to implement this technique, since it is not normative, it is unlikely that interoperability will be achieved due to the lack of a common specification. In addition, other issues must be considered, such as how to synchronize this new stream with the visual data stream itself and what should be done if this stream is corrupted.

2. PROPOSED TEMPORAL SHAPE ERROR CONCEALMENT ALGORITHM

In this paper, the idea of using global motion to conceal shape errors at the decoder is extended while eliminating the major drawback mentioned above. In the proposed technique, everything is done at the decoder and, therefore, the decoder does not have to rely on any additional (non-normative) information sent from the encoder. This allows the proposed technique to be used even if the encoder knows nothing about the concealment techniques implemented at the decoder, which is usually the case when terminals from different manufacturers are used. This way, the additional bit rate that was used in [2] for the extra stream can be used for something else, like increasing the shape intra refreshment rate or improving the texture quality.

In this paper, it is assumed that the alpha planes have been encoded with some kind of block-based technique before being delivered, such as (but not necessarily) the MPEG-4 Visual standard [1]. It is also considered that bitstream errors will manifest themselves in the form of bursts of consecutive erroneous blocks.

Since the global motion parameters are no longer sent by the encoder, this means that they have to be locally computed at the decoder. This is only possible because the decoded video data at a given time instant is usually not completely corrupted, only some parts of it are. This happens because, as mentioned above, errors typically manifest themselves as bursts of consecutive erroneous blocks. With the remaining correctly decoded shape and texture data, the decoder can extract the necessary global motion parameters.

After the global motion parameters have been determined, they can be used to motion compensate the alpha plane of the previous time instant. Here, also a different approach than what was used in [2] was adopted. Whereas in [2] the contour of the previous time instant is motion compensated, here the motion compensation is applied to the whole alpha plane, which avoids a number of complex problems typically associated with the processing of contours. For instance, a contour segment that is copied from the reference contour to replace a missing contour segment in the corrupted VOP may not connect with the already existing correctly decoded contour, especially if the global motion model is not able to perfectly describe the existing motion. This will create serious problems when generating the concealed alpha plane from the fixed contour, which do not happen with the proposed technique by directly working with the alpha planes.

The block diagram for the proposed temporal shape error concealment technique is presented in Figure 1; the input is a corrupted alpha plane with several lost blocks, typically arranged in bursts (see Figure 2). The correctly decoded blocks in the same alpha plane will be used to compute the global motion parameters with respect to the previously decoded VOP. Then, the computed parameters will be used for the concealment process, whose output is a fully concealed alpha plane. The three main steps in Figure 1 are conceptually similar to those in [2], but since the concealment is applied to alpha planes instead of contour images and the global motion is computed at the decoder, the techniques associated to each processing module are completely different. The three steps in the proposed shape concealment process are:

- **Global motion parameters computation** – This module has the task of computing the global motion parameters by considering the corrupted alpha plane and the previously decoded alpha plane;
- **Global motion compensation** – This module is responsible for compensating the previously decoded alpha plane with the global motion parameters determined above;
- **Replacement of corrupted alpha plane blocks** – This module is in charge of replacing the corrupted alpha plane blocks with the corresponding blocks from the global motion compensated previous alpha plane.

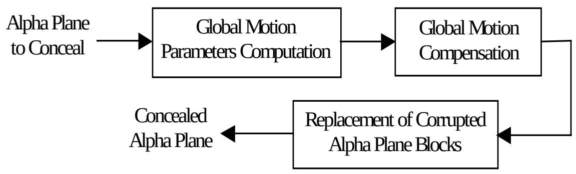

Figure 1 – Proposed temporal shape error concealment process

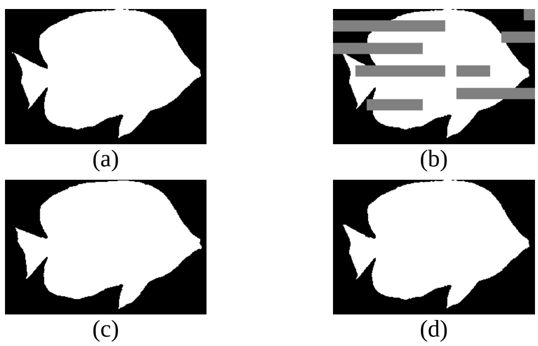

Figure 2 – Temporal shape concealment process for the Bream video object: (a) Original uncorrupted alpha plane; (b) Corrupted alpha plane; (c) Global motion compensated previous alpha plane; (d) Concealed alpha plane

In order to understand better the shape concealment process, an illustrative example will be given before detailing each of the steps above in the following sections; each step corresponds to a well defined and rather independent operation. As explained above, the input of the concealment chain depicted in Figure 1 is an alpha plane where some blocks have been lost (i.e. the lost area), which are shown in gray throughout the example. For this example, it will be considered that the alpha plane in Figure 2 (a) has been corrupted, as illustrated in Figure 2 (b). The first step is to consider the previous VOP and determine the global motion parameters. With these parameters, the previous VOP is motion compensated and Figure 2 (c) is obtained. After that, the corrupted blocks are replaced with the corresponding ones in the global motion compensated previous VOP, which gives the concealed alpha plane shown in Figure 2 (d).

2.1 Global motion parameters estimation

Before computing the global motion parameters, a global motion model has to be chosen from the ones available. The differences between the existing motion models are basically related to their complexity and, thus, their capacity to describe complicated motion trajectories. Since the error concealment technique proposed in this paper only deals with the motion of the shape data in consecutive time instants, which is typically quite simple, a simple model will suffice. However, it is always possible to replace the used motion model with a more complex one, since the proposed technique does not directly depend on the type of motion model that is used, as long as one is used. This way, the affine four parameter model, which is certainly the simplest and most widely used global motion model, was adopted. A description of how this model is derived can be found in [3], as well as the type of motion that is possible to describe with it, which is: i) change of the camera focal length (i.e., zoom or scale), ii) rotation around an axis normal to the camera axis (i.e., pan) and iii) rotation around the camera axis.

With this model, when two time instants are considered, as well as forward motion, the transparency value of a shapel with coordinates (x’,y’) in the most recent time instant can be computed from the shapel with coordinates (x,y) in the previous time instant by the following expression:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} c_1 & c_2 \\ -c_2 & c_1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c_3 \\ c_4 \end{bmatrix} \tag{1}$$

where c_1, c_2, c_3 and c_4 are the global motion parameters.

The target of the technique proposed here is to be able to conceal errors in the shape data of video objects and, therefore, it is important that the determined parameters for the adopted global motion model be able to accurately describe the motion of the alpha plane. Since the alpha plane is completely uniform on either side of the video object contour (opaque inside the object and transparent outside) it does not have any internal motion and, therefore, its motion basically corresponds to the motion of the contour. This way, to determine the motion of the shape data, the first step is to extract what is left of the video object contour from the correctly decoded shape data.

The next step is to determine, for each point of the extracted contour, the corresponding point in the previous alpha plane; this is the same as determining a motion vector for each contour point. For this, a shape context around the considered contour point is considered; here, the used context is a block of 16x16 shapels and the search range is 32 shapels (16 to each side) in both directions, but these values can easily be changed. Here, the accuracy of the estimation is favored and, therefore, an exhaustive search pattern is used. However, this could be replaced with a faster, but sub-optimal, algorithm such as a three step hierarchical search [4]. This search for the corresponding point in the previous alpha plane can be quite a daunting task because, in many cases, several perfectly matching candidates can be found, leading to disastrous results in terms of global motion parameters. Therefore, to improve the global motion estimation, the luminance around the considered contour points will also be used because the alpha plane alone gives too many inconsistencies, leading to very inaccurate motion parameters.

After all the corresponding points have been found for the current and previous alpha planes, the global motion parameters are determined by finding linear least squares estimates of c_1, c_2, c_3 and c_4, as described in [3].

2.2 Global motion compensation

Now that the global motion parameters are known, the motion compensation itself is quite straightforward. The objective of this module is to take the correctly decoded (or concealed) alpha plane from the previous time instant and global motion compensate it to the current time instant, so that it can be used to conceal the corrupted alpha plane. To do this, all the decoder has to do is consider all the points with coordinates (x,y) in the previous alpha plane and compute their new coordinates (x',y') in the current time instant where the concealment is to be applied, by using Equation (1). To make the motion compensation more efficient, only the points with coordinates (x,y) that correspond to opaque shapels in the previous alpha plane have to be considered. Of course, motion compensation can be applied to the shape data as well as to the texture data.

2.3 Replacement of corrupted alpha plane blocks

After the previous alpha plane has been motion compensated, the concealment itself, which is simply a cut and paste operation, can start. In the corrupted alpha plane, all the blocks that were considered corrupted are simply replaced with the corresponding alpha plane blocks from the motion compensated previous alpha plane. The same can be done for the texture data.

To better understand this procedure, Figure 3 should be considered. In Figure 3 (a), the corrupted alpha plane that is going to be concealed is shown. The gray areas correspond to the corrupted alpha blocks that have to be concealed. In Figure 3 (b), the previous global motion compensated alpha plane is shown, where the shaded areas are the alpha blocks that correspond to the corrupted alpha blocks in the corrupted alpha plane. Therefore, these blocks simply have to be copied to the corrupted alpha plane. The result is shown in Figure 3 (c), where the concealed blocks are shaded.

(a) (b) (c)

Figure 3 – Replacement of corrupted alpha plane blocks: (a) Corrupted alpha plane; (b) Previous global motion compensated alpha plane; (c) Concealed alpha plane

3. PERFORMANCE EVALUATION

In order to evaluate the proposed shape concealment technique, several MPEG-4 bitstreams have been tested, each bitstream containing one video object encoded at a given bit rate, according to the Core Visual Object Type. The *Akiyo*, *Bream* and *Stefan* video objects were tested. Although the actual bit rate used does not influence the quality of the shape (since lossless shape coding was used), it influences the quality of the texture data, which is used for estimating the global motion parameters. Therefore, the used MPEG-4 bitstreams had to be encoded at a bit rate that lead to an acceptable decoded texture quality, thus leading to more accurate global motion parameter values and improved shape concealment. Notice that MPEG-4 streams were just used as the most representative streams at hand; the proposed technique just requires a block-based coding approach and does not depend in any way on MPEG-4 video coding.

In terms of MPEG-4 error resilience tools, resynchronization markers and data partitioning with reversible variable length codes were used at the encoder, thus making the tested bitstreams more resilient to errors. Additionally, since it was important to avoid that the decoded texture quality degrade too much, an intra refreshment scheme was used at the encoder. As for the shape data, it was refreshed at each time instant.

Here, instead of adding errors directly to the bitstreams, the decoder simply ignored the video packets randomly with a given packet loss rate, thus allowing to evaluate the performance of the concealment technique independently of the decoder error detection capabilities. For each one of the studied loss rates, each one of the above bitstreams has been decoded 50 times (i.e., corresponding to 50 different error patterns or runs). Then, the proposed shape error concealment technique has been applied to the corrupted decoded alpha planes to recover the missing shape data. To evaluate the shape quality, the *Dn* metric defined by MPEG is used, which is defined as the number of different shapels between the decoded and original alpha planes divided by the total number of opaque shapels in the original alpha plane. *Dn* can also be expressed as a percentage, $Dn[\%] = 100 \times Dn$. Additionally, since the proposed technique can also be applied to the texture of the video object, whose decoded texture quality is here very important (for the estimation of the global motion parameters), numerical texture quality results will also be presented using the *PSNR* metric. However, since arbitrarily shaped video objects are used, the definition of *PSNR* has to be adjusted and thus, instead of being computed over all the pixels in a rectangular frame, it is computed over the pixels that belong to both the decoded and concealed VOP being evaluated and the original VOP (the parts where there is no overlapping due to errors are not taken into account).

Due to space limitation, results are only shown here for the CIF version of the *Bream* video object; the lowest acceptable frame rate for this sequence (i.e., 10 fps) was used because it corresponds to the most critical situation in terms of temporal error concealment. At this frame rate, an acceptable texture quality can be obtained by encoding the sequence at 128 kbps. As for the size of the video packet, this was chosen to be 1600 bits, which corresponds to eight video packets per VOP. With these parameters, an average error-free decoded texture quality of 30.01 dB is obtained. The obtained *Dn* is obviously 0.00% because the used shape coding is lossless.

In the following numerical results, three different types of *Dn* and *PSNR* values are shown for the tested video object sequences, and for four different video packet loss rates: Dn_{low}, Dn_{avg}, Dn_{high} and $PSNR_{low}$, $PSNR_{avg}$, $PSNR_{high}$. In order to define them, the *Dn* and *PSNR* values associated with a given run have to be considered, which are simply the temporal average of the *Dn* and *PSNR* values, respectively, for all the VOPs in that video object sequence. This way, while Dn_{low} and Dn_{high} values correspond, respectively, to the average *Dn* value associated with the best and the worst runs in terms of shape quality, Dn_{avg} corresponds to the mean of the average *Dn* values associated with the 50 different runs for each test case. Similarly, $PSNR_{low}$ and $PSNR_{high}$ values correspond, respectively, to the average *PSNR* value associated with the worst and the best runs in terms of texture quality and $PSNR_{avg}$ corresponds to the mean of the average *PSNR* values associated with the 50 different runs for each test case.

As can be seen in Table 1, even for relatively high video packet loss rates, the *Dn* values remain relatively low. In most cases, the *Dn* values are so low that they correspond to hardly noticeable artifacts (i.e., Dn_{avg} values around 1% and below).

This is the case for the *Bream* video object with packet loss rates of 1%, 5% and 10%. For higher packet loss rates, such as 20% for *Bream*, the artifacts start to become more visible (i.e., Dn_{avg} values higher than 1% but still below 3%). This can be explained by considering the type of motion associated with the tested object: the *Bream* object moves quite a lot but suffers almost no deformation along time. In terms of texture quality, whose results are included in Table 2, the same comments that were made for the *Dn* values can be also made for the *PSNR* values.

Table 1 – Dn values for the Bream video object

Video packet loss rate	Dn_{low} [%]	Dn_{avg} [%]	Dn_{high} [%]
1%	0.01	0.09	0.34
5%	0.28	0.51	0.88
10%	0.64	1.03	1.56
20%	1.62	2.22	3.33

Table 2 – PSNR values for the Bream video object

Video packet loss rate	$PSNR_{low}$ [dB]	$PSNR_{avg}$ [dB]	$PSNR_{high}$ [dB]
1%	27.48	28.90	29.80
5%	24.98	26.13	27.36
10%	22.58	23.96	25.31
20%	19.43	21.36	22.95

In order to illustrate the results in Table 1, the shape and texture of a given decoded VOP (corresponding to VOP 3 in the original 300 VOP sequence) is used, which is shown in Figure 4 (a) and (b). In Figure 4, three different corrupted versions of the alpha plane in Figure 4 (a) and the texture in Figure 4 (b) are shown, in addition to the corresponding concealed alpha planes and textures. These three versions correspond to three of the 50 different error patterns used above for a video packet loss rate of 20%. As can be seen in Figure 4, in terms of shape, the obtained results have a pleasing subjective impact on the viewer, in the sense that no annoying artifacts appear, except for error pattern 3, which corresponds to a higher *Dn* value. As for the *Dn* values corresponding to the shown alpha planes, the results are also very low (typically below 1%): 0.99% for error pattern 1, 0.97% for error pattern 2 and 2.56% for error pattern 3. The exception is error pattern 3, which is higher because the back fin of the fish violates the global motion assumption and is regularly deformed along time. In terms of texture, the same comments that were made for shape can also be made. As for the corresponding *PSNR* values, they are 28.34 dB for pattern 1, 30.55 dB for pattern 2 and 21.55 dB for pattern 3. The uncorrupted texture has a *PSNR* value of 32.01 dB.

4. FINAL REMARKS

In this paper, one temporal technique was proposed to conceal shape errors in binary alpha planes or in the binary supports of gray scale shapes corresponding to the objects in object-based video coding systems, such as MPEG-4. Results have been presented showing that this technique is able to recover lost shape data with rather small distortion, as long as the basic global motion assumption is verified. However, if the global motion model is not able to adequately describe the alpha plane changes in consecutive time instants, some visible artifacts may appear.

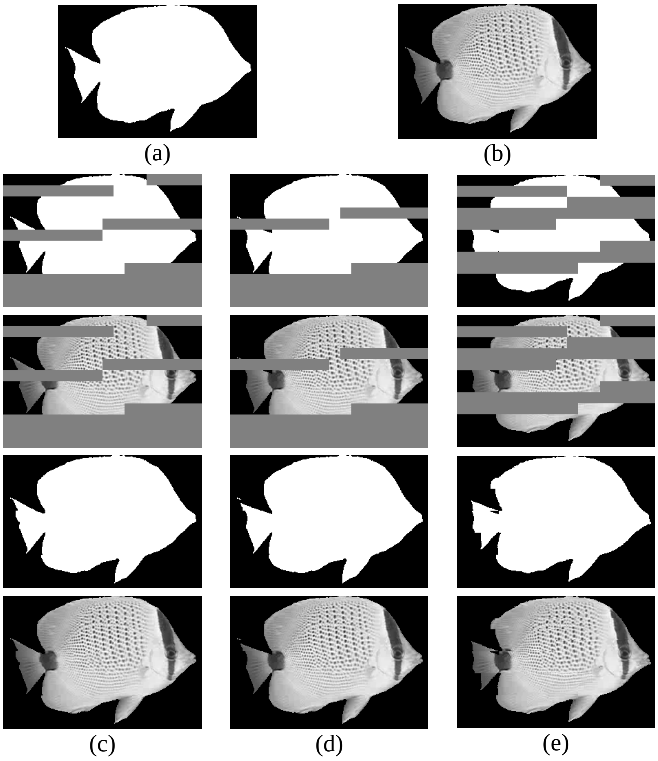

Figure 4 – Corrupted and respective concealed alpha planes and textures for the Bream video object with a video packet loss rate of 20% – (a) Uncorrupted shape; (b) Uncorrupted texture; (c) Error pattern 1; (d) Error pattern 2; (e) Error pattern 3

Finally, to end this paper, it is important to emphasize the relevance of shape concealment techniques, not only important to achieve an acceptable shape quality, but also because the decoded texture quality achieved is highly dependent on the quality of the shape data. Therefore, for object-based video applications to be actually used in error-prone environments, robust shape error concealment techniques will have to be available.

5. ACKNOWLEDGMENT

The authors acknowledge the support provided by the European Network of Excellence VISNET (IST Contract 506946).

6. REFERENCES

[1] ISO/IEC 14496-2, “Information Technology – Coding of Audio-Visual Objects,” Part 2: Visual, December 1999.

[2] P. Salama, C. Huang, “Error Concealment for Shape Coding,” *Proc. of ICIP 2002*, Rochester, NY, Vol. 2, pp. 701-704, September 2002.

[3] A. Zakhor, F. Lari, “Edge Based 3-D Camera Motion Estimation with Application to Video Coding,” *IEEE Transactions on Image Processing*, Vol. 2, No. 4, pp. 481-498, October 1993.

[4] M. Bierling, R. Thoma, “Motion Compensating Field Interpolation Using a Hierarchical Structure Displacement Estimator,” *Signal Processing*, Vol. 11, No. 4, pp. 387-404, December 1988.