ELSEVIER

Signal Processing: *Image Communication* 15 (2000) 631–641

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.nl/locate/image

Error concealment in video coding of arbitrarily shaped objects

M.R. Frater*, W.S. Lee, M. Pickering, J.F. Arnold

School of Electrical Engineering, Australian Defence Force Academy, Canberra ACT 2600, Australia

Abstract

Recent trends in video coding technology have included the coding of arbitrarily shaped objects and the improvement of the robustness of video bitstreams when subjected to errors during transmission. In this paper, we examine the interaction of these two technologies, using a coder and decoder based on MPEG 4-video. Effective techniques are proposed for concealing the effects of these transmission errors.

1. Introduction

Digital video communication has traditionally assumed that video from different sources is composited into a sequence of rectangular pictures before encoding and relied on having a relatively error-free transmission medium. There has recently been significant progress in overcoming both of these limitations. An important consequence of this is the MPEG-4 standard [1]. Systems based on this standard will incorporate video coding tools including:

- a basic video coder and decoder, based on traditional motion-compensated DCT coding, and capable of coding and decoding rectangular pictures;
- shape coding extensions, which allow video objects of arbitrary shape to be coded and decoded. The shape data is coded using a context-based arithmetic coder, in which the transparency of each pixel is predicted from surrounding pixels; and
- error-robustness tools, including data partitioning and variable length slices (which are known in MPEG-4 as video packets).

These advances in technology will enable many new audio-visual communication applications. The ability to code video objects of arbitrary shape and to associate audio with individual objects permits the compositing process to be moved from encoder to decoder, and provides a generalisation of the currently used blue-screen techniques for overlaying one object onto another. The incorporation of error resilience will permit these applications to be carried on wireless and other error-prone channels, particularly as higher bandwidth wireless channels become more commonly available.

In this paper, we introduce techniques for error concealment that can be applied by a video decoder on coded arbitrarily shaped video objects. The performance of these techniques is evaluated using a coder and decoder based on the MPEG-4 standard.

* Corresponding author. Tel.: +61-2-6268-8236; fax: +61-2-6268-8443.

E-mail address: m-frater@adfa.edu.au (M.R. Frater)

0923-5965/00/$ - see front matter
PII: S0923-5965(99)00042-9

This paper is structured as follows. The basic principles of error resilience are discussed in Section 2. Section 3 describes the video coding algorithms used for coding video containing arbitrarily shaped pictures. Strategies for concealing transmission errors are set out in Section 4. Experimental conditions used in the evaluation of the effectiveness of these techniques are summarised in Section 5, followed by the presentation of the results obtained in Section 6.

2. Error resilience

The provision of error resilience can be conveniently divided into a number of parts which are discussed briefly below.

Error detection

Even a single error in a video bitstream can have a large effect on video quality, especially if it causes the decoder to lose synchronisation with the arriving bitstream. In order to minimise this effect, it is desirable to detect quickly that an error has occurred and immediately take steps to prevent catastrophic degradation in video quality. Error detection can be based on syntactic (e.g. an illegal variable length codeword) or semantic (e.g. attempting to decode more than 64 DCT coefficients in a block) considerations.

Resynchronisation

When synchronism is lost with the arriving bitstream, the decoder will normally go looking for a unique resynchronisation codeword within the arriving bitstream. The codeword is unique in the sense that it cannot occur other than at a resynchronisation point within a non-errored bitstream. The resynchronisation word is followed by sufficient data to allow the decoder to continue the decoding process from that point.

Data recovery

Techniques exist (such as two-way variable length codeword decoding [2]) which allow some of the data between the point where decoding synchronism is lost to the point where it is regained to be utilised.

Concealment

Finally, the impact of any errors in decoded pictures which have occurred as a result of transmission errors (and in particular lost data) can be concealed using correctly decoded information from the current or previous pictures.

While all of these parts are important, it is probably true to say that the use of good error concealment techniques will lead to the greatest improvement in subjective quality. It is this topic which forms the basis of this paper.

3. Error resilient coding of arbitrarily shaped video objects

3.1. Video coding

Conventional video consists of a sequence of pictures, each of which consists of a two-dimensional array of pixels. For each picture, the coded bitstream typically begins with configuration information, including the picture size. For coding, pixels are grouped into macroblocks of 16×16 pixels. A motion vector can be used to specify the offset to the 16×16 pixel region in the previous picture that best matches each macroblock. The discrete cosine transform, followed by run-level and Huffman coding, is used to code the residual difference in 8×8 blocks. In normal coding, all of the data associated with a macroblock are transmitted before beginning to transmit the following macroblock.

Because variable length codes are used extensively in the bitstream, a single bit error can cause the decoder to lose synchronisation in the bitstream. Even if a decoder were to accidentally resynchronise, correct decoding may be prevented by certain parameters required for prediction being unknown. For example, the number of bits required to transmit motion vectors is usually reduced by predicting their value from surrounding motion vectors and transmitting only the residual difference. Resynchronisation is achieved by placing a unique code that cannot be emulated by any other codes, known as a start code, in the bitstream. Following this start code, all prediction is reset to a known value. The group of macroblocks lying between two start codes is usually referred to as a slice. This structure is illustrated in Fig. 1.

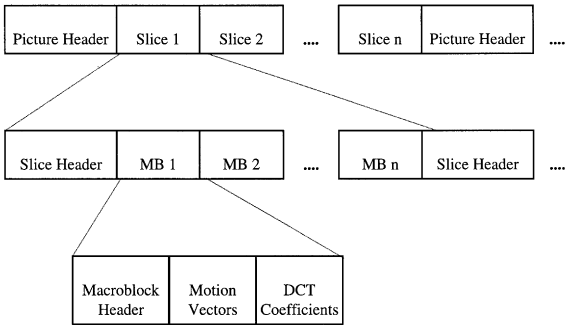

Fig. 1. Structure of a coded video bitstream.

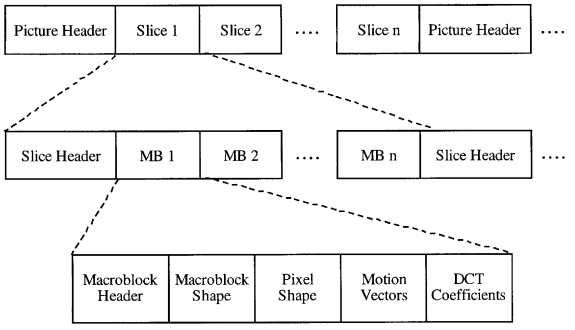

Fig. 2. Structure of a coded video bitstream incorporating arbitrarily shaped objects.

A coded video bitstream, therefore, consists of motion information, DCT coefficients representing the residual difference and various configuration data, which includes resynchronisation information.

3.2. Coding of video objects with arbitrary shape

When the restriction to rectangular pictures is removed, further information is required to specify which pixels within the bounding box lie inside the picture and which lie outside. In the terminology of MPEG 4, a sequence of pictures of arbitrary shape is referred to as a "video object", and several video objects may be decoded together and composited to form a composite sequence. In the MPEG-4 approach to shape coding used here, the coding of shape is broken into two parts:

1. Macroblock shape
   Each macroblock is classified as either transparent (i.e. completely outside the object), opaque (i.e. completely inside the object) or boundary (i.e. both pixels inside and outside the object lie within the macroblock). A Huffman code is used to represent the value of this field.
2. Pixel shape
   Within each boundary macroblock, a context-based arithmetic coder is used to identify which pixels lie inside the object. The context for the arithmetic coder is derived from surrounding pixels in the current and prediction pictures.

The structure of a coded video bitstream incorporating arbitrarily shaped objects is shown in Fig. 2.

Further details of MPEG-4's shape coding can be found in [1].

The extension of video coding to support objects of arbitrary shape raises a number of issues. One of the most important is how to perform motion compensation.

In conventional video coding, motion compensation is performed on a macroblock basis, using motion vectors transmitted in the bitstream. These motion vectors specify the offset to the location of the prediction in the reference frame. In the case of arbitrarily shaped objects, the same motion vectors are transmitted. The reference frame, however, is padded around the edge of objects in boundary macroblocks using a technique called 'repetitive padding'. An example of the results of repetitive padding in the "Weather" sequence is shown in Fig. 3.

Repetitive padding is carried out in two steps. The first, known as horizontal repetitive padding, replicates each boundary pixel of the object in boundary macroblocks horizontally towards the edge of the macroblock. In the second step, known as vertical repetitive padding, this process is repeated in the vertical direction, extending from both boundary pixels and from pixels padded using horizontal repetitive padding. All transparent pixels in boundary macroblocks of prediction pictures are assigned a value for use in forming the motion compensated prediction.

With this extension to video objects of arbitrary shape, a coded video bitstream consists of shape information, motion information, DCT coefficients representing the residual difference and various

Fig. 3. Example of the results of repetitive padding.

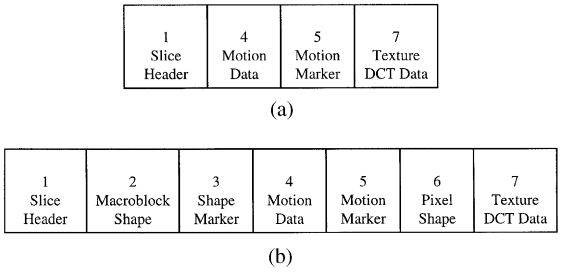

Fig. 4. Slice structure of coded bitstream; (a) rectangular case; (b) arbitrary-shape case.

configuration data, which includes resynchronisation information.

3.3. Data partitioning for error resilience

When an error occurs, all data in the bitstream from that point until the next resynchronisation point is lost. Often this means that some macroblocks in a slice can be correctly decoded but that no information is available for the remaining macroblocks. It has been found (see e.g. [1]) that not all data in the bitstream is equally important to the quality of decoded video. For example, macroblock shape data and motion vectors are more important than other data. Consequently, data partitioning, in which important information for all macroblocks in a slice is transmitted before any less important information for any macroblock, can be used to improve decoded video quality in the presence of errors with only a very small overhead. An outline of the syntax of a slice employing data partitioning is shown in Fig. 4.

The functions of the various syntax elements are:

1. Slice header
   A start code that is unique in the bitstream is used to identify the beginning of a slice. This is followed by other header information, such as the step size used in quantisation.
2. Macroblock shape
   Each macroblock is labelled as either transparent, opaque or lying on a boundary, i.e. containing some transparent and some opaque pixels. This data is the most important in the slice. Without it, the remainder of the slice cannot be decoded. As will be seen below, no further information is transmitted for transparent macroblocks.
3. Shape marker
   The end of the macroblock shape data is signalled by a code that cannot be emulated by this shape data. Once the shape marker is received, the number of macroblocks contained in the slice can be determined.
4. Motion data
   The prediction mode and any motion data associated with each non-transparent macroblock is the next most important, and therefore the next highest priority, information in the slice.
5. Motion marker
   The end of the motion information is identified by the presence of the motion marker. This is a special code that cannot be emulated by the motion information in the bitstream.
6. Pixel shape
   The location of the boundary for each boundary macroblock is specified using a context-based arithmetic coder [1]. All pixel shape data were intra-coded, i.e. no prediction from previous pictures was used.
7. DCT coefficients
   This data partitioning approach, which is similar to one of the approaches used in the MPEG-4 standard, is used for all the simulations described later in this paper.

4. Concealment strategies

The decoding process begins with the slice header. After the header is decoded, the macroblock shape information is decoded until the shape marker is detected. Once the shape marker has been found, the decoder finds the following slice header. If an error is detected (e.g. because of an illegal variable length code or the number of macroblocks implied by the macroblock shape information differs from the number implied by the two slice headers), the decoder moves immediately to the next slice and concealment (defined below) is used for the macroblocks of the current slice. If no error is detected, the motion data is decoded. This ends at the motion marker. Errors in the motion data can be identified by either illegal VLCs in the bitstream, the absence of a motion marker before the next slice header or a disagreement between the number of macroblocks implied by the motion data and the number identified in decoding the macroblock shape information. If an error is detected, the decoder conceals the macroblocks of the current slice, and moves to the next slice. Following successful decoding of motion data, the pixel-level shape and texture data (i.e. DCT coefficients) is decoded. Once again, various syntactic and semantic checks are employed for detecting errors.

This decoding process is essentially identical to that used in the development of the error resilience tools for the MPEG-4 standard [1].

Using the syntax outlined above and this error discard strategy, there are four possible outcomes:

1. The whole slice is received correctly.
2. Only the pixel shape and DCT data are lost.
3. Motion data plus pixel shape and DCT data are lost.
4. Whole of slice data are lost.

In later sections, the following approaches to the concealment of a lost macroblock are compared:

1. "Frame replacement" [3], in which each lost pixel is replaced by the pixel at the corresponding location in the previous frame,
2. "Above MV" [3], in which the motion vector from the macroblock above the lost macroblock is used to perform motion-compensated concealment,
3. "DMVE" (decoder motion vector estimation) [4,5], which uses pixels surrounding a lost macroblock to perform motion estimation in the decoder. Separate motion estimation is performed for shape data and texture data. This is possible because there the shape and texture data are independently transmitted, and has been found to provide better performance than using only a single motion vector.

In the remainder of this section, concealment strategies are proposed for each case in which data are lost.

4.1. Loss of pixel shape and DCT data

No attempt is made to estimate the lost DCT data. In opaque macroblocks that are coded in the predictive mode, motion-compensated prediction is performed using the received motion vectors. In boundary macroblocks, the pixel shape data are estimated by motion-compensated concealment using the received texture motion vector and the shape data in the previous picture.

4.2. Loss of motion, pixel shape and DCT data

Once again, no attempt is made to estimate the value of the transmitted DCT coefficients that have been lost. The only information available from the decoded bitstream is the macroblock shape (i.e. opaque, transparent or boundary). Pixel-shape data in boundary macroblocks are estimated in the same manner as the previous section. This same concealment technique is applied to the texture data.

4.3. Total loss of slice

When the whole of a slice is lost, one of the concealment techniques described above is applied to both the shape and texture data.

5. Experimental conditions

In this section, we describe the experimental conditions used for assessing the performance of the arbitrary shape video coding algorithms described above. The performance of block replacement versus motion-compensated concealment using the "DMVE" and "Above MV" approaches will be evaluated.

5.1. Test sequences

The test sequences used in this paper are listed in Table 1. Each sequence consists of a segmentation mask, indicating a foreground object and background, and texture information. The foreground object of sequence was coded at the frame rate and with the bit-rate and PSNR shown in Table 1. Every 5th picture was coded in intra-mode, all others were coded using motion-compensated prediction from the previous picture.

The first frames of the foreground object for each of these sequences are shown in Figs. 5–7. The "Weather" (Fig. 5) and "Bream" (Fig. 6) sequences contain a single object that remains relatively stationary with respect to the display frame and whose bounding box changes significantly in size over the length of the sequence. In the "Weather" sequence this is caused by the arm of the presenter extending to the left; in the "Bream" sequence this is caused by the fish turning around.

The "Cyclamen" sequence also contains a single object, as illustrated in Fig. 7. In this case, the bounding box almost fills the display frame. In the course of the sequence, the camera pans to the right, revealing parts of the object not seen in the first frame. The shape information in this sequence is much more complex than for the other two sequences.

Table 1
Test sequences

Sequence	Resolution	Frame rate	Rate (kbit/s)	PSNR (dB)
Weather	QCIF	10	46	36.69
Bream	QCIF	10	59	35.07
Cyclamen	SIF	30	260	35.39

Fig. 5. First frame of "Weather".

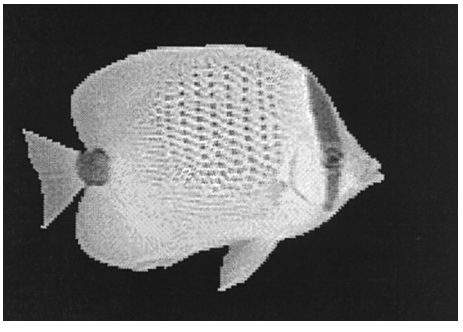

Fig. 6. First frame of "Bream".

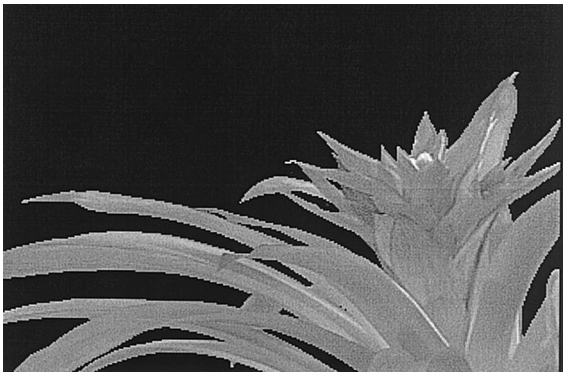

Fig. 7. First frame of "Cyclamen".

5.2. Test conditions

The following three distinct error conditions, which are the same as those being used in MPEG

to evaluate error resilience tools for MPEG 4, were used:

1. Random bit errors
   In video transmission, a single bit error can cause the decoder to lose synchronisation with the arriving bitstream. For a given bit error rate, this case usually presents the greatest challenges to a decoder. In practice, forward error correction is usually used to remove these errors. Cases 2 and 3 below, therefore, are more typical of the error conditions observed in real transmission systems. Results for bit error rates of 10^{-2} and 10^{-3} are presented below.
2. Burst bit errors
   In many transmission systems, such as wireless mobile networks, bit errors tend to occur in bursts. These might be caused by interference or fading of a wireless channel. Results are presented for error probabilities of 10^{-2} and 10^{-3} and burst lengths between 1 and 20 ms.
3. Packet loss
   This might occur in transmission of a video service over a packet network, such as the internet. In wireless audio-visual services packet loss can also occur due to transmission errors causing the multiplexing layer to lose packets. Results for packet loss with loss probabilities of 10^{-2} and 10^{-3} with packet lengths randomly chosen between 96 and 400 bits are shown.

These error conditions are chosen to emulate the types of errors that might be present in a bitstream passed from a multiplexing layer decoder (e.g. MPEG-4 systems) to a video decoder.

For each sequence at each error condition, 50 independent runs were performed. The resulting average, maximum and minimum PSNR was recorded, along with the standard deviation.

For each error condition the following results are quoted:

1. average PSNR from the 50 independent runs,
2. maximum PSNR,
3. minimum PSNR,
4. standard deviation of the PSNR values, and
5. average number of pixels whose shape data are incorrectly decoded (PSE).

The PSNR is measured over those pixels lying within the boundary of the arbitrarily shaped object.

6. Results

In this section we present results comparing the performance of the data partitioning and combined syntaxes and the use of block replacement and DMVE for concealment.

6.1. Objective results

Tables 2–4 show results comparing the three concealment techniques described above. The results for "Weather" suggest that no advantage is obtained by the use of intelligent concealment for this sequence. This is probably because there is little motion either within the object itself or between the object and the background. For the other two sequences, significant improvement in PSNR is obtained by the use of motion-compensated concealment, especially in the packet-loss error cases. For these two sequences, there is regular motion both within the object and between the object and the background. As a result, it is not surprising that motion vectors transmitted in the macroblock above a lost macroblock provide a good estimate of the motion for use in motion-compensated concealment. While the differences in PSNR observed are not large, their subjective impact is quite significant, as will be discussed below. A measure of the reliability of the PSNR results can be obtained from the fact that both DMVE and "Above MV" concealment produced a higher PSNR than frame replacement in 95% of error patterns for "Bream" and 100% of error patterns for "Cyclamen".

No significant objective difference exists between the number of pixels for which the shape is in error.

These results suggest that there is much to be gained from the use of motion-compensated concealment in improving the quality of arbitrarily shaped video objects transmitted over lossy channels, but that the added complexity of the DMVE approach is not justified, except possibly for

Table 2
Results for "Weather" – "Frame replacement" versus "Above motion vector" and "DMVE"

	Concealment	Mean	σ	Min	Max	PSE
Random 10^{-3}	Frame rep	28.93	0.80	26.73	30.48	243
	Above MV	29.04	0.83	26.73	30.48	243
	DMVE	29.04	0.83	26.75	30.52	256
Random 10^{-2}	Frame rep	25.29	0.77	22.71	26.51	574
	Above MV	25.48	0.79	22.78	26.73	572
	DMVE	25.54	0.73	22.97	26.58	631
Burst 10 ms, 10^{-2}	Frame rep	29.32	1.08	26.63	31.58	232
	Above MV	29.44	1.12	26.68	31.90	231
	DMVE	29.42	1.21	26.60	32.09	248
Burst 20 ms, 10^{-2}	Frame rep	30.26	1.29	26.91	32.60	197
	Above MV	30.36	0.135	26.83	32.88	197
	DMVE	30.33	1.40	26.87	32.90	211
Burst 1 ms, 10^{-3}	Frame rep	32.42	0.82	29.60	33.54	138
	Above MV	32.48	0.82	29.65	33.57	138
	DMVE	32.53	0.84	29.66	33.74	146
Burst 10 ms, 10^{-3}	Frame rep	35.26	1.04	32.06	36.23	109
	Above MV	35.29	1.05	32.06	36.27	109
	DMVE	35.30	1.05	32.07	36.32	110
Packet loss 10^{-3}	Frame rep	34.54	1.92	28.32	36.28	133
	Above MV	34.60	1.93	28.33	36.27	133
	DMVE	34.60	1.93	28.31	36.30	134
Packet loss 10^{-2}	Frame rep	32.46	1.80	28.22	34.71	173
	Above MV	32.56	1.82	28.22	34.80	173
	DMVE	32.53	1.82	28.22	34.90	177

higher-rate coding experiencing packet loss. The improvement resulting from the employment of DMVE is much less than that observed previously for rectangular pictures [4]. The most likely reasons for this are that the segmentation has created objects within which the motion is regular and that in many cases the PSNRs are quite close to the unerrored PSNR.

6.2. Subjective performance

6.2.1. Errors in shape

It was observed above that there is little difference between the number of errors in the shape between the three concealment methods investigated. Certainly, it could not be claimed that the number of shape errors had been reduced by the use of motion-compensated concealment. Subjective evaluation suggests that these objective results may be misleading, at least in some circumstances.

Fig. 8 shows one frame from the "Bream" sequence, in which a large number of shape errors can be seen along the top edge of the fish. The same frame (using the same error pattern) is shown for the "Above MV" concealment approach in Fig. 9. It can be seen clearly from this example that there are significantly more shape errors in the DMVE case, but that the DMVE algorithm has resulted in a smooth boundary that is subjectively less annoying. (For this frame, 307 shape errors were measured for the DMVE concealment, while only 169 were measured for the "Above MV" and frame replacement approaches (Fig. 10).) This effect was commonly observed, leading to the conclusion that the smoothing effect of the DMVE on shape errors often led to a significant subjective improvement in decoded video quality. On the other hand,

Table 3
Results for "Bream" — "Frame replacement" versus "Above motion vector" and "DMVE"

	Concealment	Mean	σ	Min	Max	PSE
Random 10^{-3}	Frame rep	24.63	0.77	21.99	26.09	423
	Above MV	25.17	0.89	22.14	27.01	405
	DMVE	25.24	0.93	22.44	27.47	404
Random 10^{-2}	Frame rep	22.82	0.70	20.73	23.92	1038
	Above MV	23.03	0.77	20.70	24.32	1024
	DMVE	23.01	0.80	20.65	24.42	1008
Burst 10 ms, 10^{-2}	Frame rep	24.10	0.89	21.92	25.72	471
	Above MV	24.62	0.93	22.45	26.17	450
	DMVE	24.73	0.96	22.22	26.23	448
Burst 20 ms, 10^{-2}	Frame rep	25.06	0.84	23.28	27.26	400
	Above MV	25.64	1.03	23.59	28.05	385
	DMVE	25.83	1.09	23.75	28.43	387
Burst 1 ms, 10^{-3}	Frame rep	27.47	1.07	25.20	29.38	162
	Above MV	27.91	1.17	25.29	29.70	155
	DMVE	27.97	1.19	25.34	29.93	155
Burst 10 ms, 10^{-3}	Frame rep	31.65	1.57	27.49	34.22	54
	Above MV	32.04	1.63	27.69	34.34	52
	DMVE	32.23	1.67	27.59	34.44	51
Packet loss 10^{-3}	Frame rep	30.89	1.76	26.68	33.01	94
	Above MV	31.36	1.94	26.75	33.77	91
	DMVE	31.50	1.98	26.78	33.88	91
Packet loss 10^{-2}	Frame rep	27.79	1.48	24.53	30.02	224
	Above MV	28.43	1.73	24.75	30.96	217
	DMVE	28.73	1.81	24.82	31.22	214

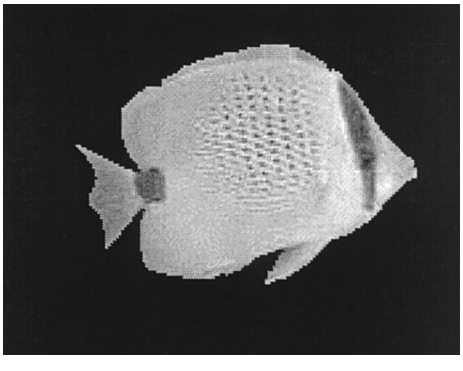

Fig. 8. Impact of errors on shape with DMVE concealment.

little difference is observed subjectively between the frame replacement and "Above MV" approaches.

6.2.2. Errors in texture

It has been shown previously that the DMVE algorithm provides significantly improved quality (both objectively and subjectively) for conventional video [4]. In the results presented above, significantly less improvement in PSNR was observed than would be expected from this previous work. This may be partly caused by use of data partitioning and the highly effective resynchronisation provided in MPEG-4, which means that the PSNR of the errored sequences is often quite close to the no-error PSNR.

Subjectively, however, the difference is often much more apparent, as illustrated in Figs. 11–13. It can be seen quite clearly that the amount of distortion in the upper right of the fish is very much less in the DMVE case than either of the other two cases. In this case, the removal of a visually significant artefact in a small portion of the picture will not result in a large improvement in PSNR, but does result in a large improvement in subjective quality.

Table 4
Results for "Cyclamen" — "Frame replacement" versus "Above motion vector" and "DMVE"

	Concealment	Mean	σ	Min	Max	PSE
Random 10^{-3}	Frame rep	24.45	0.45	23.82	25.13	2765
	Above MV	25.19	0.47	24.42	25.89	2621
	DMVE	25.27	0.50	24.49	26.02	2741
Random 10^{-2}	Frame rep	22.98	0.40	22.45	23.66	4735
	Above MV	23.69	0.44	23.09	24.29	4514
	DMVE	23.69	0.50	23.02	24.25	4724
Burst 10 ms, 10^{-2}	Frame rep	26.05	0.57	24.65	26.70	991
	Above MV	27.34	0.90	25.47	28.61	885
	DMVE	27.69	0.88	26.01	29.00	1034
Burst 20 ms, 10^{-2}	Frame rep	26.36	0.80	25.33	27.48	1001
	Above MV	27.44	1.01	25.76	28.68	923
	DMVE	27.76	1.09	26.30	28.98	1116
Burst 1 ms, 10^{-3}	Frame rep	29.99	0.66	28.54	30.81	425
	Above MV	30.96	0.81	29.16	32.29	391
	DMVE	31.11	0.82	29.25	32.46	403
Burst 10 ms, 10^{-3}	Frame rep	32.75	0.75	31.21	33.86	62
	Above MV	33.30	0.89	31.28	34.37	53
	DMVE	33.46	0.92	31.54	34.75	71
Packet loss 10^{-3}	Frame rep	31.26	1.46	28.90	32.87	365
	Above MV	32.18	1.78	29.08	34.01	339
	DMVE	32.42	1.78	29.22	34.29	339
Packet loss 10^{-2}	Frame rep	28.45	0.80	26.77	29.21	912
	Above MV	29.61	1.14	27.67	31.19	846
	DMVE	30.00	1.15	28.08	31.53	860

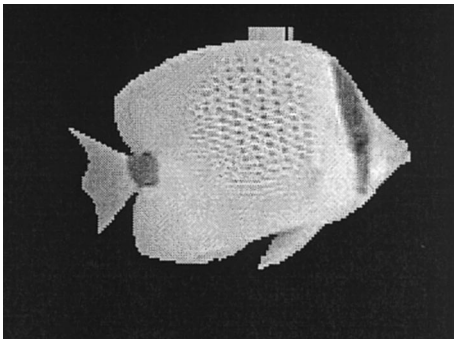

Fig. 9. Impact of errors on shape with "Above MV" concealment.

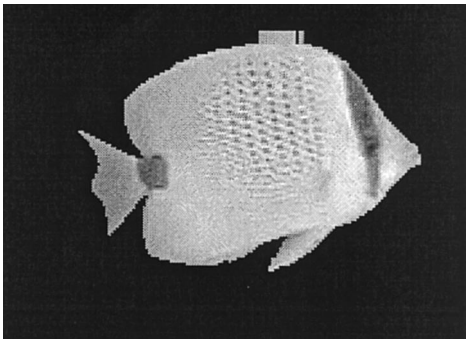

Fig. 10. Impact of errors on shape with frame replacement concealment.

7. Conclusion

In this paper, we have examined the application of error concealment techniques to arbitrarily shaped video objects. It has been shown that the use of motion-compensated concealment provides superior performance to direct replacement from the previous frame using both objective and subjective measures. It was also shown that the use of the decoder motion vector estimation technique

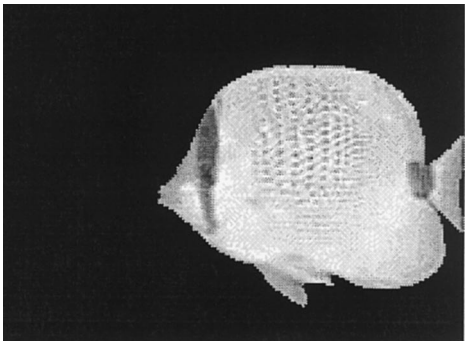

Fig. 11. Impact of errors in texture with DMVE concealment.

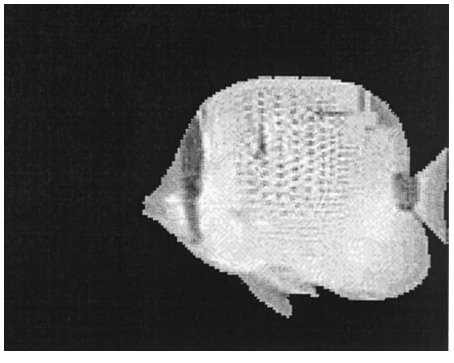

Fig. 12. Impact of errors in texture with "Above MV" concealment.

provided significantly improved performance over other motion-compensated concealment techniques in certain circumstances. In some circumstances it was found that separate motion estimation was required for shape and texture data (e.g. where all received data are corrupted), while in other situations texture motion vectors could be satisfactorily applied to shape data (e.g. where texture motion vectors are correctly received but shape data are corrupted). The trade-off between complexity and performance requires further study.

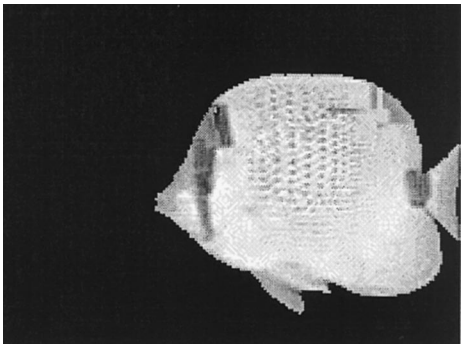

Fig. 13. Impact of errors in texture with frame replacement concealment.

References

[1] Very Low Bitrate Audio Visual Coding: Part 2 Video, ISO/IEC Committee Draft 14496-2. November 1997.

[2] Y. Takishima, M. Wada, H. Murakami, Reversible variable length codes, IEEE Trans. Commun. 43 (2/3/4) (1995) 158–162.

[3] Generic Coding of Moving Pictures and Associated Audio Information: Part 2 – Video, ISO/IEC International Standard 13818-2. 1995.

[4] J. Zhang, J.F. Arnold, M.R. Frater, Improved motion compensated concealment for MPEG-2 coded video, in: Proceedings of the International Workshop on Audio-Visual Services over Packet Networks, Aberdeen, Scotland, 15–16 September 1997.

[5] J. Zhang, M.R. Frater, J.F. Arnold, T.M. Percival, MPEG 2 video services for wireless ATM networks, IEEE J. Selected Areas Commun. 15 (1) (1997) 119–128.