A SHAPE ERROR CONCEALMENT TECHNIQUE FOR ROBUST MPEG-4 SYSTEM

Shih-Chang Hsia and Cheng Hung Hsiao
Department of Computer and Communication Engineering, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, R.O.C.

Keywords: MPEG-4, Shape information, Error concealment, Interpolation.

Abstract: This paper presents a fast-efficient error concealment method for recovering the shape information. The proposed technique consists of the block classification, the edge direction interpolation and the filtering interpolation. The missing block with the logic criterion is classified to four types that are transparent, opaque, edge and isolated blocks. Most of the computations cost on the edge block and the isolated block to obtain better cost-performance tradeoffs. For the edge-block recovery, the edge slope is computed by referring the near available block, and then the missing shape is interpolated along the edge direction. We deal with the isolated block using the cascaded filter to approximate the real shape. The experimental results show that the proposed method can achieve better cost-performance to restore the shape information compared to the other competing algorithms in both of the numerical parameters and the shape images. The processing speed is faster about 2~3 times to the well-known methods. The adaptive algorithm employed the low computational load overhead to make it applicable to a real-time MPEG-4 coding system.

1 INTRODUCTION

Recently the MPEG-4 system enables content-based functionality, as well as high coding efficiency, by taking into account shape information for the low bit-rate system (Jan, 1999). If the shape information cannot be reconstructed, the decoded object would produce very serious distortions. Recently, there are some literatures focused on the error concealment techniques for shape recovery (Lee et al., 2001, Salama and Huang, 2002, Li et al., 2002, Soares and Pereira, 2004). Lee, etc. al. used the fuzzy theory to recover the shape error (Lee et al., 2001). First, the error block is computed with the bilinear interpolation. Then the result is modified according to their fuzzy membership function to truncate the pixel to a binary form with one threshold. Shirani etc. al. presented a novel error concealment method for recovering shape information (Salama and Huang, 2002), which iteratively interpolate the missing pixel with a set of weight along eight directions. With iterative procedures, the error shape is recovered better and better. Li etc. al. proposed recursive error concealment to recover the shape information (Li et al., 2002). With line-correction, line-prediction and line detection approaches, the recovered shape error can be kept in small distortion. However, we hardly estimate the computational time for this approach, which is away from real-time applications.

This paper presents a new method to recover the object shape for the intra frame and still image. The proposed algorithm consists of the block classification scheme and the edge interpolation, to achieve better quality but using low computational cost. This paper is organized as follows. The proposed algorithm for shape recovery is presented in Section II. The simulation results and comparisons with other approaches are shown in Section III. Concluding remarks are outlined in Section IV.

2 THE PROPOSED ALGORITHM

Figure 1 illustrates the processing flow of the proposed algorithm. According to the feature of neighboring blocks, the processed block can be classified into four groups. If the neighboring blocks all are dark or bright, the current block should be an opaque or transparent block. This block is easily recovered since we can directly replicate the data

Hsia S. and Hsiao C. (2009).
A SHAPE ERROR CONCEALMENT TECHNIQUE FOR ROBUST MPEG-4 SYSTEM.
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications*, pages 103-106
DOI: 10.5220/0002164001030106

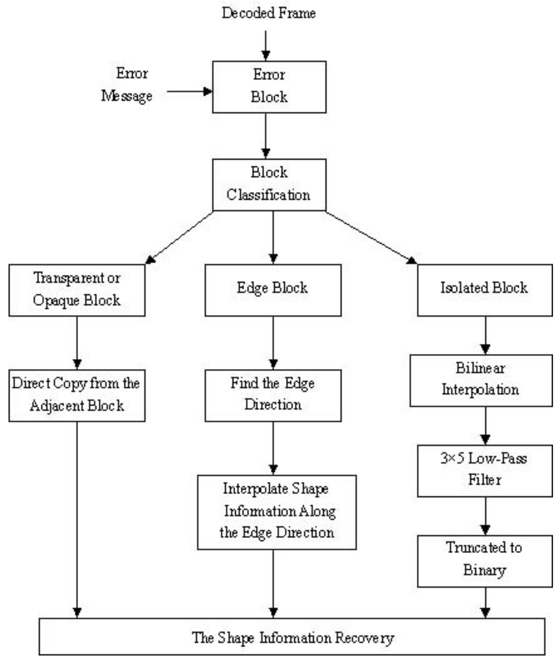

Figure 1: The flowchart of the proposed shape recovery.

with the adjacent pixels. However, when the neighboring blocks contain the edge information that is partial dark and partial bright, we try to find the edge direction for interpolating the shape. The last is an isolated block that its neighboring blocks are different groups. This block always locates at the top or bottom of object, which is difficult to be recovered perfectly.

Figure 2 illustrates the position of the missing block processing. If the B_T, B_B, B_{TL}, B_{TR}, B_{BL} and B_{BR} blocks all are zero (or one), the current block belongs to an opaque (or transparent) block. We only fill the zero or one for all missing pixels without computations. If the T-block and B-block have the edge information, the lost block would be an edge block, where T-block is defined by consisting of blocks B_{TL}, B_T and B_{TR}, and B-block combining with blocks B_{BR}, B_B and B_{BR}. The block can be classified according to the information of T-block and B-block. To fasten processing, the simple AND and OR operations are used. We define "T_{and}" function that the boundary pixels of the B_{TL}, B_T and B_{TR} blocks (T-Block) are performed with AND function as

$$T\_\{and\} = (B_{TL}^0 \bullet B_{TL}^1 \bullet B_{TL}^2 ... \bullet B_{TL}^{N-1} \bullet B_T^0 \bullet B_T^1 \bullet B_T^2 ... \bullet B_T^{N-1} \bullet B_{TR}^0 \bullet B_{TR}^1 \bullet B_{TR}^2 ... \bullet B_{TR}^{N-1}) \qquad (1)$$

,where B_{TL}^i , B_T^i *and* B_{TR}^i is the i^{th} boundary pixel of blocks B_{TL}, B_T and B_{TR}, respectively. If the block size is N×N, we need to perform (3N-1)-bit AND

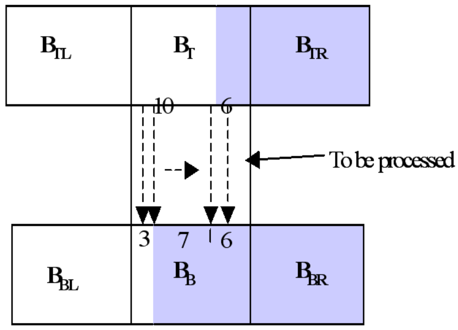

Figure 2: The block position.

operations. By the same concept, "B_{and} ", "T_{or}" and "B_{or}" are defined to perform the AND function of B-block, the OR function of T-block and the OR function of B-block, respectively. Table-1 lists the block classification with T_{and}, B_{and}, T_{or} and B_{or}. Obviously, if all operation results are zero or one, which be an opaque or transparent block. While T-block and B-block have the edge information, the partial data is zero or one. Hence the T_{and} and B_{and} should be zero, but T_{or} and B_{or} are one. There are four cases. For case 1, the background is dark and the missing block is at the top of object, the result of T_{and}, B_{and}, T_{or} and B_{or}={0001}. For case 2, the missing block changes to at the bottom of object, and the following result is {0010}. If the background is bright, the results of the bottom and top of object are {0111} and {1011} for case 3 and 4 respectively.

For an edge block, XOR operation is used to estimate the edge direction. If the block size is N×N, the XOR of the block B_T and B_B is performed by

$$ES = (B_T^0 \oplus B_B^0, B_T^1 \oplus B_B^1, ..., B_T^{N-1} \oplus B_B^{N-1}) \qquad (2)$$

to denote the edge shift (ES) between the two blocks. B_T^i *and* B_M^i is the i^{th} boundary pixel of blocks B_T and B_B. For example, in Figure 2, *ES* =(0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0). *ES_one*=7 is by counting the number of one. The *ES* value is used to find the vector distance for the missing block. The missing block is interpolated with along the edge direction of the blocks B_T and B_B. Since the shape information is binary, first we can fill 0 and 1 for two-line values along the edge direction. The proposed algorithm is available for the vertical edge blocks, but fail for horizontal edges. The horizontal edge block is classified to the isolated block type since its adjacent top and bottom blocks belong to different block groups.

Table 1: Block Classification with the Boundary Pixels.

	Opaque Block	Transparent Block	Edge Block	Isolated Block					
				Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
T-and	0	1	0	0	0	0	1	0	1
B-and	0	1	0	0	0	1	0	1	0
T-or	0	1	1	0	1	1	1	0	1
B-or	0	1	1	1	0	1	1	1	0

For isolated block processing, we proposed a filtering interpolation to recover the isolated block efficiently. With the processing flow of Figure 1, first we do the linear interpolation for the lost block with the boundary pixels of B_T and B_B blocks. As N=16, the pixel of the first vertical line can be interpolated by line interpolation. The interpolation result will be gray level since the top block B_T is high and the bottom block B_B is zero level. To recover the error blocks to near real shapes, the interpolated value is further processed with a low-pass filter. Our motivation is that we attempt to use the low-pass filter to spread the normal shape information of the neighbor blocks to modify the gray level of the error block to recover the shape. If the processing window covers the object pixels that its level is 255, the gray level of error pixel after low-pass filtering will increase. As closer to the background, the level will decrease since the window covers more black pixels. This approach can modify the bilinear interpolation result for the lost block to approach the practical object shape. Finally, the gray-level is truncated with a threshold to become the binary shape data. The threshold value is adaptive according to the neighboring available shape block. If the processed block is close to the object, the threshold will be increased. However, when the processed block possibly cannot be decided according to the case of the Table-1. The non-defined block types are also processed with the same as the processing flow of the isolated block.

3 SIMULATIONS

To evaluate the performance of the proposed algorithm, three video sequences “TV”, “Salesman” and “Irene” are selected. The data is encoded with MPEG-4 encoder. There are possibly partial or complete slice errors by error patterns. The block size used 16×16, so there are 16×16×m data corrupted, where m is the number of error blocks. The error rate is about in the bounding box. Figure 3 and 4 shows the original shape and error shape with 20% ~30% error for “Salesman”. First, we employ the bilinear interpolation to fill the missing block and then to truncate the interpolated results with the

Figure 3.

Figure 4.

Figure 5.

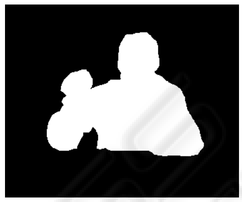

Figure 6.

threshold 127. We find that the performance is poor since the shape outline has great distortion, as shown in Figure 5. The fuzzy theory is used to refine the result of bilinear processing. The data is first processed with bilinear interpolation and then the result is revised with its fuzzy decision rule. The final result is also truncated to get the binary shape information. The result is better than that of the bilinear method. Next, the shape information is recovered with our proposed algorithm. First, we classify the processed block with our algorithm. The complete result is shown in Figure 6 for “Salesman” sequence from the error shape in Figure 4. The proposed technique outperformed all the others against which it was compared. The edge error concealment algorithm can recover most of shape information. The 3-stage low-pass filter is used for isolated blocks in experiments. If the lost block has high-variance in its shape, it faces difficulty to conceal errors for the block from neighboring slices. In the special case, the missing block in bottom is simulated with the Figure 5. Clearly the B-blocks are not available. However, we need two-boundary data for the bilinear interpolation. For simulations, the non-available pixels use the zeros for bilinear computations. The results from bilinear and fuzzy interpolations are poor. The proposed algorithm recovers the shape information for the missing block in bottom along the edge direction.

For numerical analysis, we also can compute the error value by comparisons with the original shape image and the restored image. The number of error points is checked by XOR operation of the original shape image and the restored image. If the pixel of the restored image is different from the pixel of the original shape image at the same position, the result of XOR operation becomes high. When all pixels are compared with XOR operation, the number of "high" is counted to denote the number of error points that is used to adjudge the performance of error recovery. Besides, we can segment the object from the image according to the decoding shape information after error concealment processing.

The error is estimated with the distortion ration and the number of error points. Table-2 lists the DR result

Table 2: Comparisons with DR and Recovery Error Points from Various Algorithms.

Images / Methods	TV		Salesman		Irene	
	DR	Error Points	DR	Error Points	DR	Error Points
Bilinear	0.396	401	0.511	518	0.249	252
Fuzzy Theory [2]	0.389	395	0.516	523	0.233	236
Our Method	0.347	352	0.489	496	0.217	220

and the error points from recovery with the bilinear, fuzzy (Lee et al., 2001) and proposed methods, respectively. The distortion rate (DR) can be evaluated with

$$DR = \frac{Numuber\ of\ Error\ Point}{Frame\ Size} \times 100\% \tag{3}$$

If the shape is more complex, the performance of error concealment will be reduced since the detail content is difficult to recover. The results shown that our performance is better than the bilinear interpolation and the fuzzy theory. The computational complexity is an important term for real-time applications. The computational load of the proposed algorithm is adaptively dependent on what kind of the block.

With the logic criterion to classify the block type, we can save the computational power for the opaque and transparent block since these blocks can be simply restored without any distortion. As for the edge block, we only employ logic operations to find the edge direction and to interpolate the shape data between the T-block and B-block. The computational load is not high. Most of the computations cost for the isolated block to achieve better quality. We employ the local filtering approximated approach for the isolated block processing. In fact, the number of edge and the isolated blocks in the error sequence will dominate both the complexity and the reconstructed quality. The statistic analysis from the practical experiments, the transparent and opaque blocks occupy about 60% of the full missing data. The edge block and isolated block is about 30% and 10% respectively, but being dependent on the error pattern. Due to most of the processing blocks without using computations, the computational complexity of the proposed system is low. Experiments result that our computation time is about 50% and 30% of the bilinear and fuzzy method respectively. Therefore, the proposed method is suitable applied on low-cost and high-speed video coding products.

4 CONCLUSIONS

This paper presents a fast high-efficiency method for the shape information recovery. First, the processed block is classified into four catalogs and then we employ various techniques to process the different block to restore the shape information with better cost-performance. We do not waste any computational power for the simple opaque and transparent blocks since these blocks are easily to be restored by copying the neighboring pixels. Instead, we make more computations for the edge block and isolated block because these blocks quality will greatly determine the recovery performance. To restore the edge block, we proposed an edge-oriented algorithm that employed the boundary matching with the simple logic operations rather than complex computations. Then, we used spatial interpolation to recover the edge information of a missing block from the result of boundary search. But the boundary matching method fails to recover the isolated block. Instead, the filtering interpolation is used to recover the shape information for the isolated block. The results demonstrate that the shape information can be efficiently restored, but using low computational load to make it applicable to real-time systems. Hence the proposed algorithm is very valuable for the shape recovery while applying for high robust MPEG-4 systems.

REFERENCES

Jan. 1999. Coding of audio-visual objects: video,MPEG 4 , ISO/IECJTC/SC29/WG11.

P. J. Lee, L. G. Chen, 2001. W.J. Wang and M.J. Chen," Robust error concealment algorithm for MPEG-4 with the aids of fuzzy theory", *IEEE International Conference on Consumer Electronics*, PP. 154-155.

P. Salama and C. Huang, 2002. " Error concealment for shape coding", *International Conference on Image Processing,* September 22-25, pp.II-701-704.

X. Li, G.M. Schuster and A.K. Katsaggelos, 2002. "A recursive shape error concealment method," *IEEE International Conference on Image Processing,* September 22-25, pp.I-177-180.

L. D. Soares and F. Pereira, Apr. 2004. " Spatial shape error concealment for object-based image and video coding", *IEEE Trans. Image Processing*, vol. 13, no.4 pp. 586-599.