A Solution to the Composition Problem in Object-Based Video Coding

Jeong-Woo Lee and Yo-Sung Ho

Kwangju Institute of Science and Technology (K-JIST)
1 Oryong-dong, Puk-gu, Kwangju, 500-712, Korea
{jeongwoo, hoyo}@kjist.ac.kr
http://vclab.kjist.ac.kr/

Abstract. In this paper, we introduce the composition problem associated with object-based video coding and propose a solution to this problem. Although the object-based rate control algorithm can provide the overall coding gain, it may create the composition problem due to different temporal resolutions in object encoding. By checking shape changes of video objects, we can minimize the possibility of the composition problem at the encoder. At the decoder, we can also apply hole detection and recovery algorithms to eliminate the effect of the composition problem to the human visual system.

1 Introduction

In MPEG-4 video coding [1], there are three types of coding scenarios. The first one is the simple frame-based coding. The second one is an object-based coding, where the temporal rates of all objects are constrained to be the same, but the bit allocation is performed at the object level. The third one is also an object-based coding, but the temporal rate of each object may vary. In each coding mode, we should consider trade-offs between spatial and temporal qualities to improve the overall coding efficiency. In the object-based framework, we have a freedom to choose different frameskip factors and corresponding quantization parameters for the objects in the scene.

However, the existing MPEG-4 rate control algorithms [2,3] do not address the case of coding several objects at different temporal rates. In other words, the objects in the current frame are either all coded or all skipped. As shown in Fig. 1, we can choose a different frameskip factor and corresponding quantization parameters for each object in the object-based coding framework.

Although an object-based rate allocation algorithm can provide a potential coding gain, it also makes the problem complicated significantly since we must track the individual time instant when each object is coded. The second difficulty to develop the object-based framework is a matter of procedure. It is not clear how we can determine the set of frameskip factors and quantization parameters; one possible approach is to break the main problem into smaller sub-problems [4]. The third difficulty in coding objects at different temporal rates is due to the composition problem, because holes may appear in the reconstructed frame

A. Yazici and C. Şener (Eds.): ISCIS 2003, LNCS 2869, pp. 683–690, 2003.

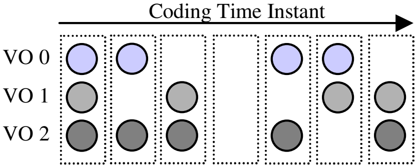


Fig. 1. Object-based coding framework

when objects are encoded at different temporal rates [5]. This can easily be avoided if all the objects in the current frame are constrained to have the same temporal rate, i.e., a *fixed* temporal rate. However, in order to fully explore the potential coding gain of object-based video coding, a different temporal rate for each object, i.e., *variable* temporal rates, should be allowed.

In this paper, after we describe the concept of the composition problem, we propose a solution to this problem both at the encoder and the decoder sides. While we can minimize the composition problem at the encoder, we can try to eliminate the problem at the decoder. For this objective, the encoder detects changes in the object boundaries over time and the decoder employs hole detection and recovery algorithms. Although the idea of detecting changes in object shape boundaries over time minimizes the effect on the average PSNR values in the combined objects [6], it does not consider the sensitivity of the human visual system. In other words, we should note that due to the sensitivity of the human visual system, holes may be easily detected even though they do not impact the average PSNR value significantly. In addition, there was no rate control algorithm that effectively utilizes this information for object-based coding.

This paper is organized as follows. Section 2 describes the concept of the composition problem. Section 3 provides new ideas to solve the composition problem at the encoding and the decoding parts. After we present simulation results to evaluate the performance of the proposed algorithms in Section 4, we draw conclusions of this work in Section 5.

2 Composition Problem

Fig. 2 illustrates the composition problem with the FOREMAN sequence that has two objects: foreground and background. The left image shows the decoded and composited sequence for the case when both objects are encoded at 30Hz. The right image shows the decoded and composited sequence when the foreground is encoded at 30Hz and the background at 15Hz.

When all objects are encoded at the same temporal resolution, there is no problem during the object composition for image reconstruction at the decoder, i.e., all pixels in the reconstructed frame are well-defined. However, the composition problem can occur if objects are encoded at different temporal rates. When the shapes of object boundaries are changing in the scene and these objects are encoded at different temporal rates, we may have some undefined pixels or holes

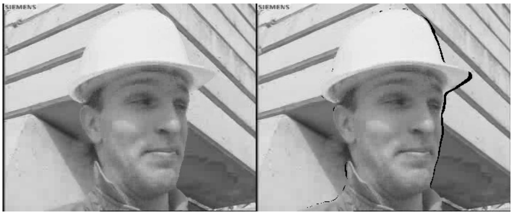


Fig. 2. Illustration of the composition problem

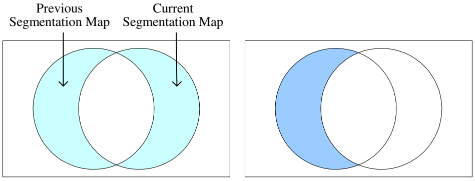
in the reconstructed frame. These holes are created due to the movement of one object without updating adjacent or overlapping objects. The holes are uncovered areas in the scene that cannot be associated with any previous information; therefore, those pixels are not well-defined. The holes are disappeared when objects are resynchronized, i.e., when both the background and the foreground objects are coded at the same time instant.

3 Solution to Composition Problem

3.1 Encoding Part: Minimization Process

In order to resolve the composition problem, we can try to minimize the possibility of the problem at the encoder. For this objective, we need to define the shape change of each object over time to identify its boundary information.

A well-known measure for the shape difference is the Hamming distance, which counts the number of different pixels between two shapes. Fig. 3(a) illustrates the concept of the Hamming distance.



(a) Hamming distance (b) Composition distance

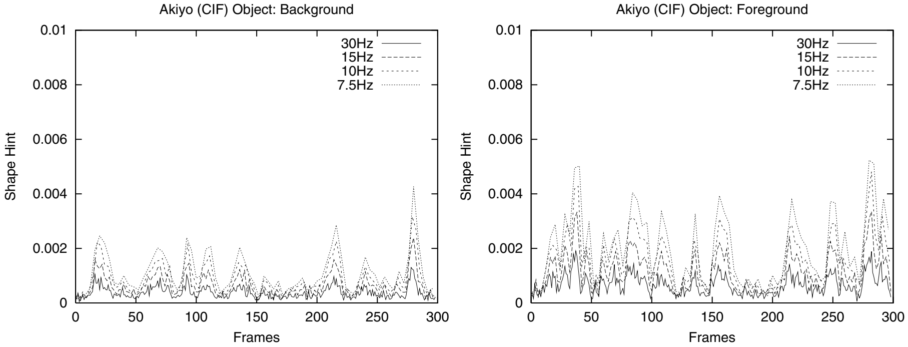
Fig. 3. Measure for shape hints

As shown in Fig. 3(b), however, we can observe that the region where the composition problem occurs consists of pixels that skipped objects do not cover. Therefore, the shape difference hint $d^c_{j,i}$ for the jth object at the current time index i is defined by

$$d^c_{j,i} = |A_{j,p} - A_{j,c}| \tag{1}$$

where $A_{j,c}$ and $A_{j,p}$ be sets that contain pixel positions of the segmentation map of the jth object at the current and the previous coding time instants, respectively. The sign '$-$' implies the *difference* of the set and $|\cdot|$ is the *cardinality* of the set. We note that all operations are performed by the set operation.

We then normalize the shape difference hint $d^c_{j,i}$ by the number of pixels in the object. The value of '0' indicates no change, while the value of '1' implies that the object is moving very fast. Fig. 4 shows shape hints of the AKIYO sequence, which indicate very small movement of objects. Since the shape hint is a good indicator of object movement, we can use it for the object-based allocation of variable temporal resolutions.



(a) Background (b) Foreground

Fig. 4. Shape hints for AKIYO

Based on the proposed measure, we can employ a simple algorithm to predict and avoid the problems at the encoder [5]. First of all, we need to determine existence of stationary objects, either rectangular background objects that cover the entire frame or arbitrarily shaped objects that do not move. Since we need to have at least two moving objects for the composition problem, we can apply any object-based rate allocation algorithm to the encoding process. In order to determine whether movement of the non-stationary objects is tolerable or not, we compare each shape hint to an empirically determined threshold value. If the threshold value is exceeded by any shape hint, we conclude that the composed scene contains too much distortion.

Although the shape hint of the non-stationary objects exceeds the threshold value, movement of objects may be tolerable if all objects move in the same direction. In order to test directions of the moving objects, we calculate the difference between the previous and the current segmentation maps. Although we consider only four directions for moving objects, any other methods can be employed to detect more detailed directions of the moving objects.

Fig. 5 explain how shape hints are used and how object-based rate allocation is performed. After shape hints for all objects are calculated by (1), collections of the extracted hints are passed into an analyzer where we estimate whether the

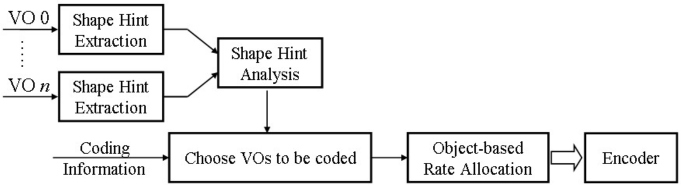


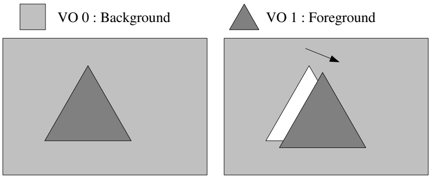
Fig. 5. Encoding process to reduce the composition problem

composition problem can occur at the decoder or not. Consequently, the possible coded object set M_L is determined based on values of shape hints and directions of the moving objects as well as the buffer occupancy [4]. For example, we do not allow variable temporal rates for those objects that move to the opposite direction each other.

3.2 Decoding Part: Elimination Process

If movement of shape boundaries is small, any hole in the reconstructed frame is negligible and does not effect the overall PSNR value significantly. However, we should note that due to the sensitivity of the human visual system, the hole may easily be detected. In order to overcome this problem, we can reduce its visual effect by recovering any undefined pixel values at the decoder.

In order to eliminate the visual effect of holes in the scene, we first check the existence of the holes and then delete hole regions. Fig. 6 illustrates how to detect holes in the reconstructed frame with the rectangular background object.



(a) Previous frame (b) Current frame

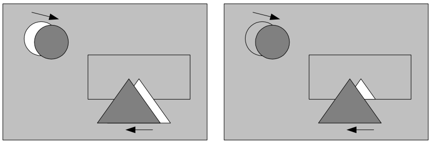
Fig. 6. Frame recovery with a rectangular background object

Let $A_{j,d}$ be the difference between $A_{j,p}$ and $A_{j,c}$. The set H_i representing hole regions in the reconstructed frame at the time index i is calculated by

$$H_i = \bigcup_{j=0}^{M-1} [A_{j,d} - \bigcup_{k=0,k \neq j}^{M-1} A_{k,c}] \tag{2}$$

where M is the number of objects. If the jth object is skipped, $A_{j,d}$ is empty.

If the reconstructed frame consists of arbitrarily-shaped objects without the background object, the proposed algorithm can detect the hole region indicated



(a) Initial hole region (b) Desired hole region

Fig. 7. Detection of the hole region

by the white space in Fig. 7(a). In order to detect the exact hole region, as shown in Fig. 7(b), we impose same restrictions on the hole detection algorithm.

For the set H_i, an element is discarded if there is no element of skipped objects within the search range around the pixel position of the element. Because the possibility of the composition problem is already reduced at the encoder, it is not necessary to select a wide search range. The search range empirically determined in our experiment is eight. When the object has a background, we do not need the search range.

Once we detect holes in the reconstructed frame, we need to recover the pixel values within the holes. When one object is coded and the other is skipped in the frame, because the coded object has the original shape of the object, changes of the coded object result in visual degradation. Therefore, the pixels within the holes must be recovered from the region of the skipped objects.

In order to recover pixel values in those holes, we use the Euclidean distance measure between pixels within the holes and pixels in the segmentation plane of the skipped objects. Each pixel within the holes is replaced by the pixel in the shape boundary of the skipped object that has the minimum distance between the pixel in the holes and the pixel in the skipped object.

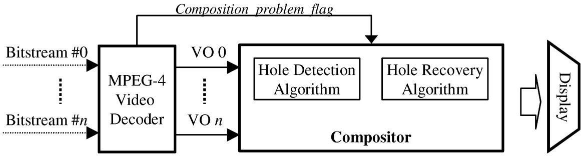
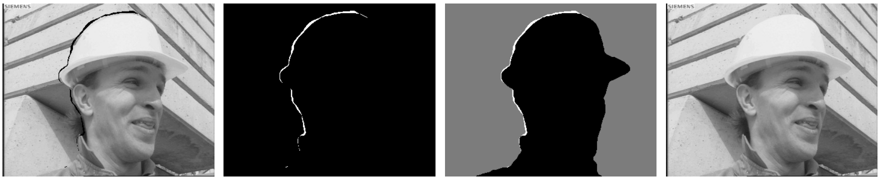


Fig. 8. Decoding process to reduce the composition problem

Fig. 8 shows the block diagram of the decoding process to reduce the composition problem. During the decoding process, each bitstream is decoded at the current decoding time. If there is no data to be decoded at the current time for the case of the skipped object, the object at the current time is replicated by the object at the previous decoding time. If all objects are decoded, the *composition_problem_flag* is set to be 0 and the reconstructed frame is directly displayed with no additional process. If the value of *composition_problem_flag* is 1, however, we apply the hole detection and recovery algorithm because of the composition problem at the reconstructed frame.

4 Experimental Results

In order to evaluate the performance of the proposed algorithm, we have tested the FOREMAN and COASTGUARD sequences. We have also applied the object-based rate allocation algorithm [4].



(a) Initial frame (b) Detected holes (c) Skipped object (d) Recovered frame

Fig. 9. Performance for a frame with a full-rectangular object

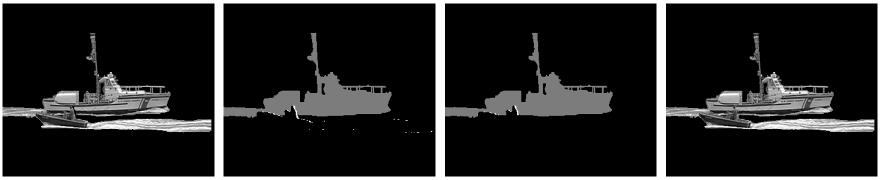
Fig. 9 shows an example of the composition problem generated by the foreground and background objects. If two objects construct the whole frame, as in the FOREMAN sequence, existence of the background object can be easily detected. Fig. 10 is another example of the composition problem generated by an arbitrarily-shaped object with no background object. Our test is conducted on two video objects (VO1 and VO2) of the COASTGUARD sequence. VO1 is a large boat with some motion and a complex shape, while VO2 is a small boat. VO2 moves in the opposite direction to VO1.

The white area in Fig. 9(b) shows the result of the hole detection algorithm by (2). The shaded or gray areas in Fig. 9(c) show the skipped object in the current frame. Undefined pixel values are recovered by the pixel values at the boundaries of the skipped object. Fig. 9(d) shows the result of the proposed hole recovery algorithm. As shown in Fig. 9(d), there is no problem with the object composition and no picture quality degradation is observed.

Fig. 10(b) shows the detected holes when the search range is not applied. In Fig. 10(b), the white and grey areas represent the detected holes and the skipped object, respectively. The pixel values in the white area should be recovered from the grey area. However, there exist some areas that can be against the human visual system when there is no background object. Fig. 10(d) and Fig. 10(e) show the results of the hole detection algorithm with the restriction and its reconstructed frame, respectively. These results indicate that the reconstructed frame is more comfortable to human eyes when the hole regions are recovered within the search range.

5 Conclusions

In this paper, we have proposed a solution to overcome the composition problem that can be generated in object-based video coding schemes. We define a shape hint measure to compute the change in the object shape over time and detect the



(a) Initial frame (b) Without restriction (c) With restriction (d) Recovered frame

Fig. 10. Performance for a frame without a full-rectangular object

direction of the moving object. The proposed idea is applied to the rate control part of the encoding process. Although any holes in the reconstructed frame are negligible and do not impact the PSNR value significantly, we have proposed the hole detection and recovery algorithm to minimize the effect of the visual distortion from the object composition problem. The proposed hole detection algorithm can be applied irrespective of existence of the background object.

Acknowledgments. This work was supported in part by Kwangju Institute of Science and Technology, in part by the Korea Science and Engineering Foundation (KOSEF) through the Ultra-Fast Fiber-Optic Networks (UFON) Research Center at K-JIST, and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.

References

1. ISO/IEC 14496–2 (MPEG-4 Video):Information Technology – Coding of Audio/Video Objects,(1998)
2. Vetro, A., Sun, H., and Wang, Y. :MPEG-4 Rate Control for Multiple Video Objects, IEEE Trans. Circuits Syst. Video Technol., Vol. 9, (1999) 186–199
3. Lee, J.W., Vetro, A., Wang, Y., and Ho, Y.S. :Bit Allocation for MPEG-4 Video Coding with Spatio-Temporal Trade-offs, IEEE Trans. on CSVT, Vol. 13, June (2003) 488–502
4. Lee, J.W., Vetro, A., Wang, Y., and Ho, Y.S. :MPEG-4 Video Object-based Rate Allocation with Variable Temporal Rates, ICIP, Sept. (2002) 701–704
5. Vetro, A. and Sun, H. :Encoding and Transcoding of Multiple Video Objects with Variable Temporal Resolution, Proc. IEEE Int'l Symp. on Circuits and Systems, May (2000)
6. Vetro, A., Sun, H., and Wang, Y. :Object-based Transcoding for Adaptable Video Content Delivery. IEEE Trans. on CSVT, Vol. 11, (2001) 387–401