Enhanced Frame-Based Video Coding to Support Content-Based Functionalities

Prabhudev I. Hosur and Rolando A. Carrasco

Abstract

This paper presents the enhanced frame-based video coding scheme that enables some useful content-based functionalities such as user-interactivity with video content, assignment of different quality to video objects, and content-based video search and retrieval. The input video source for the proposed enhanced frame-based video encoder consists of rectangular-size video frames and shapes of arbitrarily-shaped video objects in the frames. The rectangular frames are encoded using a scheme similar to the conventional frame-based video coding and the shapes of video objects are encoded using the contour-based vertex coding. It is possible to achieve several useful content-based functionalities by utilizing the shape information in the bitstream at the cost of a very small overhead of shape bits.

1: Introduction

Over the years, the digital video technology has been evolving quickly in terms of the ways video content is produced, delivered, and consumed. So the users are not content with only the compression efficiency in a video coding technology; they are now demanding more features such as content-based interactivity and content-based video indexing and retrieval. Furthermore, the new video coding schemes are required to cope up with bandwidth and bit error rates of various transmission networks and achieve backward compatibility with existing video coding schemes in addition to providing a very good compression efficiency.

For the conventional frame-based video encoders, the input video source is in the form of a sequence of rectangular frames. Whereas, the new video encoders which support content-based functionalities require the source input video to be in such a form that the video content can be easily identified and characterized. The video source to the object-based video coding approach in MPEG-4 standard [1] is in the form of arbitrarily shaped foreground and background video objects and their shapes. We have not yet seen the widespread usage and adoption of object-based video coding probably because the accurate segmentation of semantically meaningful objects from a rectangular video still remains a challenging problem. Furthermore, the bitmap representation of shape employed by MPEG-4 is not well suited for semantic shape characterization for content-based video indexing and retrieval. In the sub-picture coding [2] proposed towards the H.264 standardization activities, a video frame is partitioned into one or more user-defined non-overlapping rectangular foreground objects called sub-pictures and a remaining background picture. However, the rectangular sub-pictures do not generally represent the true video objects with which a user would like to interact. The proposed enhanced frame-based video coding technique is described in Section 2.
2: Proposed Enhanced Frame-Based Video Coding

In the proposed enhanced frame-based video encoding scheme (EFBE), the input source video consists of rectangular-size video frames and shapes of arbitrarily-shaped objects in each video frame. We define the object of interest (OOI) in a rectangular frame video as the arbitrarily-shaped semantically meaningful video object which is of interest to a user. There are three steps involved in the process of obtaining the coded representation of video using the enhanced frame-based video coding scheme: 1) pre-processing, 2) encoding shape and texture, and 3) post-processing. In the first step, the OOI is identified in each video frame and its shape information is obtained in the form of either an outline contour sketch or a segmentation mask. The outline contour sketch of an OOI can be obtained by marking the outline of OOI on the screen manually; this way the shape information can be easily generated by a user. Several techniques have been proposed to obtain the segmentation masks using chroma-key [3], automatic segmentation [4] and semi-automatic segmentation techniques [5]. For the proposed EFBE, the boundaries of segmented video objects need only represent an approximate outline of the area belonging to a video object and there is no requirement for segmentation to be accurate. In the second step, the rectangular frame texture and the video object’s shape are encoded. The block diagram of the EFBE is shown in Fig. 1(a). The major components of the EFBE are frame-based texture coding and shape coding. The post-processing step involves linking the shape bits of OOI with the texture bits belonging to the OOI region on the rectangular frame. The link information is multiplexed into the bitstream along with texture and shape information. The video bitstream in which a link and content information is associated with a region on a video frame is generally referred to as HyperVideo [6]. In this paper, we focus mainly on the second step and
propose a new video encoding scheme to enable content-based functionalities whatever the methods used for pre-processing and post-processing.

2.1: Shape coding

The proposed shape coding in the EFBE employs the contour-based technique which lends itself to the semantic shape characterization. The shape contour in the form of either the segmentation mask's boundary or the outline contour sketch of OOI is approximated by a polygon such that the distance between the polygon and the contour is less than or equal to a given tolerable approximation error $\delta$. The vertices of the polygon are coded using the object-adaptive vertex encoding method described in [7]. The amount of shape distortion is controlled by varying the value of $\delta$; the larger the value of $\delta$, the higher the shape distortion. Lossless shape coding is achieved by setting $\delta = 0$.

The distance between the polygonal approximations of OOI shape in the current and the reference frame is used to detect the amount of temporal variation in shape. The distance between two polygons is computed as follows. Let $P_c$ and $P_r$ be the polygonal approximations of the current and the reference OOI shape. Let $h_i$ and $v_i$ be the horizontal and vertical distance of $i$th vertex of $P_c$ from $P_r$. Then the distance of $P_c$ from $P_r$ is defined as

$$ D(P_c, P_r) = \max_{i} (d_i), $$

where $d_i = \min(h_i, v_i)$. For the objects of interest such as talking-head, the variation in OOI shape over a sequence of contiguous frames is usually very small. Therefore when the lossy shape coding is desired, the shape information is not transmitted with every frame. Instead, the polygonal approximation of the current OOI shape is coded and transmitted only if the distance between the polygonal approximations of the current and reference OOI shapes is greater than or equal to a threshold $T$ (i.e., if $D(P_c, P_r) \geq T$). Otherwise no shape information is transmitted in the current frame. At the decoder, the most recently decoded shape is used to identify the OOI if no shape information is present in the current frame.

2.2: Texture coding

The basic steps of texture coding in the EFBE are essentially the same as those in a typical rectangular frame-based video encoder. These basic steps consist of dividing a video frame into an array of basic units called macroblocks and processing each macroblock by applying discrete cosine transform, quantization and variable length coding. In fact, the texture coding block in the EFBE can be any rectangular frame-based video encoder (e.g., MPEG-1, MPEG-2 or MPEG-4 (simple profile) video encoder). In our implementation of the EFBE, we have used MPEG-4 (simple profile) encoder for frame-based video texture coding.

In the DST mode, the shape information is used to adjust the texture coding parameters. The main idea is to encode the texture in the region belonging to the polygonal approximation of the OOI shape with a finer quantization as compared to the rest of the video frame. The quantizer values used for the two regions are embedded in the header information. At the decoder, the decoded shape information is utilized to correctly identify the region belonging to video object on the video frame and the quantizer information in the bitstream header is utilized for decoding the frame texture.
Figure 2. The original rectangular frame texture and OOI shape contour of (a) Akiyo, and (b) Foreman.

2.3: Multiplexing shape and texture bits

We call the combined bits consisting of the shape bits, the bits required for coding the quantization parameters in DST mode and the bits that are generated by the post-processing stage for linking the objects of interest as additional bits. In order to achieve backward compatibility with the conventional frame-based coding, we need to place the additional bits into the bitstream such that the conventional frame-based decoders would simply ignore these additional bits and decode the texture bits as usual. The proposed enhanced frame-based video decoders would utilize the additional bits to provide the content-based functionalities. We employ the user data packet insertion scheme as described in [8] for this purpose. We combine the additional bits into user data and place the user data into the bitstream generated by the frame-based texture coding block of the EFBE.

3: Features of the EFBE

The architecture of the proposed EFBE is designed such that it can be implemented as a simple extension of the existing frame-based encoder architecture. The embedded shape information in the bitstream of proposed EFBE can be utilized to support several content-based functionalities. At the receiver, the shape information of a video object in the bitstream facilitates the identification of the region belonging to the arbitrary-shaped video object as a hotspot on the rectangular frame for content-based interactivity (see Fig. 1(b)). For example, a hyperlink can be provided for the object on the rectangular frame when a user activates the object by clicking on it. Such application does not require the shape of OOI to be accurate. Furthermore, lossy shape coding can be employed to achieve higher compression. The annotation of video hyperlinks in the form of small icons containing the polygonal approximation of object contour can be displayed at the bottom of the rectangular video for a user to identify the hot spots in a frame. During fast-forward or fast-reverse operations, only shape can be decoded and displayed instead of decoding the entire rectangular frame texture. The vertex-based shape representation carries the semantic meaning of the object and therefore it can be made use of in the future multimedia applications such as video indexing and retrieval.

4: Experimental results

The conventional frame-based video encoder (CFBE) in the MPEG-4 Verification Model (VM) software [1] is used as the basis to implement the proposed EFBE. The CFBE forms
Table 1. Comparison of performance of CFBE and the IST mode of EFBE.

| Coding scheme | Quantizer (Q) | CFBE Av. Bits/frame | EFBE (IST) Av. Bits/frame |
|---------------|---------------|---------------------|--------------------------|
|               | Texture       | Shape               | Texture                  | Shape                  |
| Akiyo         | 8             | 102961              | 102961                   | 21                      |
|               | 16            | 5924                | 5924                     | 21                      |
| Foreman       | 8             | 176584              | 176584                   | 220                     |
|               | 16            | 16941               | 16941                    | 220                     |

the texture coding block of the EFBE. We modified the VM software to incorporate the following: 1) our contour-based shape coding technique, 2) algorithm for adjustment of quantization step based on polygonal approximation of shape contour in the DST mode of operation, and 3) the technique of multiplexing the additional header information, encoded shape information and the link and content information of hotspots.

In our experiments we use the first 100 frames of the 30Hz CIF-size Akiyo and Foreman video sequences and associated OOI shape contours (see Fig. 2). The Akiyo is a low-motion talking-head video and the Foreman is a moderate-motion video. The video sequences are encoded at 10 frames/sec; so there are 34 coded frames in the bitstream. In our experimental results presented in this paper, we do not consider the bits required for embedding the link and content information of hotspots.

First, we compare the performance of the IST mode of the EFBE with the performance of the CFBE. A fixed quantization step of $Q=16$ is used for all the macroblocks. In the case of our EFBE, we set the shape coding parameters $\delta$ and $T$ as follows: \{$\delta = 10, T = 0$\} for Foreman and \{$\delta = 10, T = 5$\} for Akiyo. The texture coding in the IST mode of EFBE is the same as that in the CFBE; therefore obviously both the encoders yield the same video quality. The comparison of number of encoded bits is shown in Table 1. The number of additional shape bits in the EFBE bitstream are shown separately. Since $T = 5$ was used for coding OOI shape of Akiyo in our tests, we observed that the OOI shape was encoded in only four out of the 100 frames and the total number shape bits in the entire bitstream was 720; thus, the average number of shape bits per frame is $720/34 \approx 21$, which is a negligibly small value as compared to the average number of texture bits. In case of the Foreman, the shape is encoded for the OOI in each frame in the 100 frames because we set $T = 0$. We observe that the shape bit is 0.125% and 1.29% of the texture bits for $Q = 8$ and $Q = 16$, respectively. However, this additional overhead of shape bits in the EFBE bitstream as compared to the CFBE bitstream is greatly justified by the benefit achieved in terms of several useful content-based functionalities that the shape information enables.

In Table 2, we present the comparison of the performance of the DST mode of the EFBE with that of the CFBE using the Foreman video. In the case of CFBE, a fixed quantization step of $Q=16$ is used for all the macroblocks in a frame. Whereas in the DST mode operation of EFBE, a lower quantizer ($Q=8$) is used for OOI region and higher quantizer ($Q=31$) is used for the remaining part of the frame. For the shape coding in the DST mode of the EFBE, we set \{$\delta = 10, T = 0$\}. The quality and bitrate for the first Intra-frame encoded with the two encoders are presented in Table 2. Using the EFBE, a higher PSNR for the OOI region is achieved at the cost of lower PSNR for the background region as compared to the overall PSNR obtained with the CFBE.
Table 2. Comparison of performance of CFBF and the DST mode of EFBE when the first frame of the Foreman is encoded in intra-mode.

| Coding scheme | Quantizer (Q) | Bits | PSNR in dB |
|---------------|---------------|------|------------|
|               |               | Texture | Shape |           |
| CFBF         | 16            | 34248  | -    | 30.11     |
| EFBE (DST)   | 8 (OOI)       | 34532  | 220  | 31.85 (OOI) |
|              | 31 (Background)|       |      | 29.11 (Background) |

5: Conclusions

The architecture and design of the proposed enhanced frame-based video encoder is presented. The main aim of the proposed encoder is to provide an enhancement to the conventional frame-based coding. By embedding the coded representation of a video object’s contour along with the coded texture in the bitstream, it is possible to achieve several useful content-based functionalities. In our experimental results, the overhead of additional bits required for shape coding is less than 2% of the total bits of the conventional frame-based video coding.

References

[1] MPEG Video Group, “MPEG-4 Verification Model (VM) Version 14.0,” ISO/IEC JTC1/SC29/WG11 N2932, Oct. 1999.
[2] M. M. Hannuksela, Ye-Kui Wang, and M. Gabbouj, “Sub-picture video coding for unequal error protection,” XI European Signal Processing Conference (EUSIPCO), France, 03-06 Sep. 2002.
[3] K. Panoutsopoulos and X. Chen, “A Modified Chroma Key Technique,” Thirty-Third Asilomar Conference on Signals, Systems and computers 1999, pp. 735-739, vol. 1, 1999.
[4] T. Meier and K. N. Ngan, “Automatic Segmentation of Moving Objects for Video Object Plane Generation,” IEEE Trans. Circuits Syst. Video Technol., pp. 525-538, vol. 8, Sep. 1998.
[5] C. Toklu, M. Tekalo and T. Erdem, “Semi-automatic Video Object Segmentation in the presence of occlusion,” IEEE Trans. Circuits Syst. Video Technol., pp. 624-629, vol. 104, Jun. 2000.
[6] F. Bota, F. Corno, and L. Farinetti, “Hypervideo: a Parameterized Hotspot Approach,” TICW2002: IADIS International Conference WWW/Internet 2002, Nov. 2002.
[7] K. J. O’Connell, “Object-adaptive vertex-based shape coding method,” IEEE Trans. Circuits Syst. Video Technol., vol. 2, no. 1, Feb. 1997, pp. 251-255.
[8] S. T. Worrall, A. H. Sadka, P. Sweeney, and A. M. Kondoz, “Backward compatible insertion of user defined data into MPEG-4,” IEE Electronics Letters, vol. 36, no. 12, Jun. 2000, pp. 1036-1037.