International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.1 (2014), pp.45-52
http://dx.doi.org/10.14257/ijsip.2014.7.1.05

An Enhanced Hybrid Content-Based Video Coding Scheme for Low Bit-Rate Applications

Wendan Xu[1, 2, a], Xinquan Lai[1, b], Donglai Xu[3, c] and Nick A. Tsoligkas[3, d]

[1]*Institute of Electronic CAD, Xidian University, Xi'an, 710071, P.R. China*
[2]*Xi'an Aeronautical University, Xi'an, 710077, P. R. China*
[3]*School of Science & Engineering, Teesside University, Middlesbrough, TS1 3BA, UK*
[a]*xuwendan24@163.com,* [b]*xqlai@mail.xidian.edu.cn,*
[c]*d.xu@tees.ac.uk,* [d]*tsoligas@teihal.gr*

Abstract

This paper presents a hybrid content-based video coding scheme that encodes arbitrary shaped objects instead of blocks of images. The scheme achieves efficient compression for low bit-rate applications by separating moving objects from stationary background and transmitting the shape, motion and residuals for each segmented object. Furthermore, a new content-based object segmentation algorithm is proposed in the scheme, which does not assume any prior modeling of the objects being segmented. The algorithm is based on a threshold function that calculates block histograms and takes image noise into account. The experimental results show that the scheme proposed outperforms the classical object-based coding methods in terms of PSNR or the average number of bits required for coding a single frame.

Keywords: *Video Compression, Object Segmentation, Motion Estimation, Motion Compensation*

1. Introduction

With modern multimedia communication applications, there has been considerable research in the area of the content-based coding for efficient video compression. The compression is achieved by separating coherent moving objects from stationary background and compactly representing their shapes, motions and contents [1-3]. However, most of the content-based coding techniques have two major drawbacks. Firstly, object segmentation and motion estimation methods are computationally very intensive. Secondly, accurately representing the shapes of moving objects would result in insufficient number of bits for coding the content of a low bit-rate encoder. Although the shape coding can be avoided by using a fixed block-based partitioning technique [4], such as the overlapped block motion compensation, which also help reduce prediction error significantly. But in general the block-based coding suffers from blocking artifacts due to the use of a fixed block partitioning on a fixed grid. Also the block-based coding is susceptible to channel noise.

In this paper, a hybrid content-based video coding scheme is presented, incorporating a new object segmentation tool that retains the characteristics of both object-based and block-based coding. The object segmentation technique is based on the generation of a binary mask of the objects being segmented. The position of an object is unknown and has to be

ISSN: 2005-4254 IJSIP

determined based on its motion. Two successive frames are subtracted to generate a change detection mask [5, 6], and then the shape, motion and residuals of each object are coded accordingly.

2. Structure of Hybrid Object-based Coding

2.1. Overview

The block diagram of the proposed hybrid content-based video coding scheme is shown in Figure 1. It includes the following main components.

- Object motion detection;
- Motion estimation and compensation;
- Motion failure region detection;
- Residual encoding;
- Shape representation and coding;

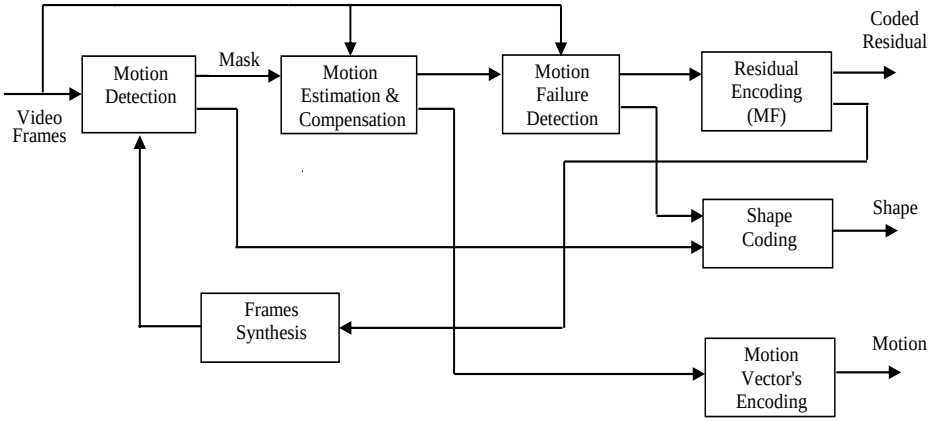

Figure 1. Hybrid Object-based Coding

The first frame of a video sequence is intra-frame coded using DCT transform, quantization, run-length coding and Huffman coding. The subsequent frames are then inter-frame coded with the process described below.

2.2. Object Motion Detection

The purpose of the motion detection is to distinguish the temporally changed regions between two successive frames. Here the selection of an appropriate threshold is key because a low threshold may cause over-segmentation, while a high threshold could result in incomplete objects. There have been many techniques developed to determine a threshold for binarization of an intensity image [7, 8]. However, most of these are based on the histogram of the intensity signal. Since a difference image differs from an intensity image, the threshold methods for the intensity images may not be appropriate to be used for difference images. The motion detection method in Figure 1 explores the statistical properties of the difference image between two consecutive frames, which include three major steps as follows.

Image Differencing. The objective of differencing images is to identify the areas of images, which have changed from one frame to the next. This is accomplished by computing the absolute difference value of two images pixel by pixel.

Image Filtering and smoothing. The absolute difference image normally includes artifacts due to illumination and noise changes. By averaging the difference, the addition of correlated pixel values is achieved while uncorrelated noises are reduced. Therefore the image becomes smooth. The use of a maximum filter limits the motion detection to the neighborhood of the current pixel and improves the stability around the object boundaries.

Image Threshold. Threshold computation and image binarization are designed to be tolerant of large variations in image intensity and contrast. Knowing that the thresholding methods can be divided into two categories, global and local, we combine the global method (block-based) with the local method (block-histogram-based) to calculate threshold. Assume that the filtered image is divided into L equal blocks with the size of each block being $W \times H$, the threshold T_h is calculated as follows.

$$T_h = \frac{\sum_{n=1}^{L}(\sum_{i}^{2} g_i) + \mu_n}{2 \times L + L} \tag{1}$$

$$\mu_n = \frac{\sum_{i}^{W}\{\sum_{j}^{H}(FD(i,j))\}}{W \times H} \tag{2}$$

where g_i is the most frequent gray level (pixel intensity) of a block and μ_η is the average gray level of the block. Here, the histogram of each block is computed. Each histogram is then clustered into two (or more) equal clusters and for each cluster the most frequent gray level g_i is found. Further, the average gray level μ_η of each block is calculated. The threshold T_h is calculated by averaging all the g_i and all the μ_η of the L blocks. In order to adapt to image noise, the threshold is further expressed by

$$T_t = T_t + c \times \sigma^2 \tag{3}$$

where σ^2 is image noise variance [9,10] and c is a constant less than 1. In order to determine the final threshold, T_t is compared with the previous calculated threshold T_{t-1}. If the current threshold T_t is greater than the previous threshold T_{t-1}, T_t is selected; otherwise, T_{t-1} is selected. When a small motion or no motion is detected (threshold is below a certain value) T_t is calculated by taking the average value of all the previous threshold values. Thus, the detection of the binary object mask is stabilized. Each pixel of the frame difference *FD(i,j,t)* is classified as either belonging to an object and labelled white in a binary image mask, *M(i,j,t)* or belonging to the background and labeled black.

The binary mask *M(i,j,t)* resulted from the binarization process may contain artifacts, especially around objects boundaries, so a post-processing step is applied to the binary image. First, a median filtering is performed. And then, three successive morphological closing operations with 3×3 kernel, followed by three morphological opening operations with the same kernel, are applied to clean up raw intensity differences and then to cluster moving objects regions. Small regions (smaller than 100 pixels) are eliminated.

2.3. Motion Estimation and Compensation

To remove temporal redundancies of a video, motion estimation needs to be performed to predict the contents of the segmented regions based on previously synthesized frame. Each macroblock in current frame is compared with the macroblocks within a search range in the reference (previous) frame by measuring the error through computing the sum of absolute differences. The best matching macroblock is selected. For the current frame, this macroblock can be encoded as a motion vector, which denotes only the translational displacement of the macroblock in the reference frame in new position. In order to improve estimation accuracy, the fractional-pixel motion estimation approach (quarter-pixel or half-pixel) [11] is adopted in the scheme proposed. After motion estimation is carried out, motion vectors are predictively coded using fixed predictor coefficients, and the prediction errors for the displacement vectors undergo Huffman coding. Now the block correlation is performed inside the change detection mask and the resulting coarsely sampled motion vector field is interpolated to a dense field and passed through the iterations of the Horn-Schunck algorithm [12]. This process smoothes the vector field used to predict the frame.

2.4. Motion Failure

Motion failure region detection refers to clusters of pixels in an area where motion compensation alone was inadequate. The boundaries of these regions can be estimated by thresholding the displaced frames' difference that is computed from the forward dense motion field estimation, *i.e.*, from the frame k to $k+1$. Clearly, the accuracy of these boundaries depends on the accuracy of dense motion estimation. Here, a further motion segmentation is performed between the current frame and the motion compensated frame. The motion failure region detection threshold used is defined by:

$$| f_{k+1}(x,y) - f'_{k+1}(x,y) | < T_{MF} \text{ and } T_{MF} = c \times \frac{1}{N} \sum_{n}^{N} | f_{k+1}(x_n, y_n) - f'_{k+1}(x_n, y_n) | \quad (4)$$

where the subscript n denotes the index of pixels forming the moving area, N is number of the pixels included in the moving area, $f'_{k+1}(x,y)$ denotes the intensity value of the $(k+1)^{th}$ reconstructed frame, and c is a constant.

2.5. Residual Encoding

The residual information inside the motion failure regions is used to correct the errors produced during the motion estimation. The coding technique applied here is the wavelets transform, which consists of a wavelet decomposition of the entire error image (five level decomposition for the luminance pixels), followed by quantization and Huffman coding. In our implementation, a bi-orthogonal filter has been used. The synthesized image is passed through a spatial filter before its subtraction from the incoming video sequence. Compared to DCT technique, the quality of the synthesized image is improved.

2.6. Shape Analysis

The boundaries of the detected moving regions need to be approximated by a shape model that can be represented with a few parameters. Here, the polygon approximation algorithm [13] and cubic smoothing spline data interpolation are adopted to approximate shapes. The traced contours are differentially coded and then run-length coded. In both encoding and

decoding, the contours are approximated by fitting splines to the corner points. While applying this algorithmic procedure to motion failure object shapes, the number of vertices may be reduced (sampled), and therefore the simplification of shape can be introduced. If the objects shapes are interpreted as a set of blocks, the analysis-synthesis codec technique is the same as the hybrid codec technique.

3. Experimental Results and Analysis

To test the proposed scheme, the experiments are carried out using the test sequence Miss America, which is characterized as having slow motions and simple spatial details, typical for videoconferencing and videophone applications. Figure 2 and Figure 3 show the 81st frame of this video sequence and the 80th synthesized (reconstructed) frame. Figure 4 shows the change regions resulted from applying the moving object detection method described above. The motion failure objects shown in Figure 5 are detected by applying the threshold values of Eq. 4.

Figure 2. Original Current Frame

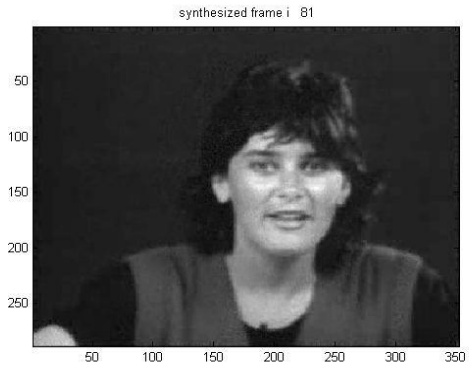

Figure 3. Reconstructed Frame

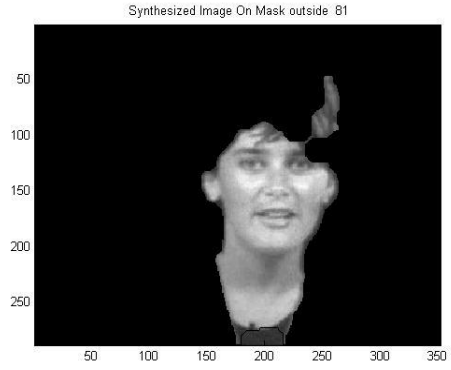

Figure 4. Detected Moving Areas

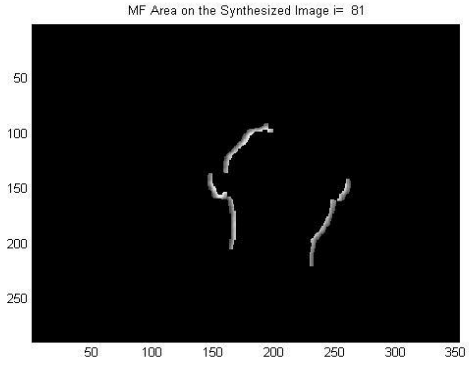

Figure 5. Motion Failure Areas

The criterion used to measure reconstructed image's quality is the PSNR (peak signal to noise ratio) that is defined by:

$$PSNR = 10 \times \log_{10} \frac{255^2}{(1/(W \times H)) \sum \sum [f(x,y) - f'(x,y)]^2} \quad (5)$$

where $W \times H$ is image size, and $f(x,y)$ and $f'(x,y)$ are the pixel intensities of the original and reconstructed images at the coordinate point (x,y), respectively. The luminance and chrominance components are predicted using the same set of motion vectors. So no additional overhead is required to predict the chrominance. The motion vectors are estimated using the luminance information only. However, the chrominance mismatch correction for the chromo-blocks has not been applied here. The first frame of the sequence is intra-frame coded and decoded and used as the first synthesized frame. In order to demonstrate effectiveness of the proposed scheme, Table 1 gives the performance comparisons between the proposed scheme and the H. G. Musmann [14], K. Grotz [15] and M. Hotter [16] methods in terms of the average number of bits required for coding one frame and PSNR.

Table 1. Performance Comparisons of Coding Methods

Scheme	Number of bits	PSNR (dB)
H.G Musmann	1040	38.16
K. Grotz	1222	39.77
M. Hotter	1350	41.20
The proposed	**1228**	**40.35**

In the proposed scheme, the number of bits required for coding one frame is similar to or slightly higher than that of other schemes. This is because in the shape analysis, encoder detects the corner points along contours and sends these to the decoder that uses a differential coding technique, and also the maximum number of corner points, rather than the reduced one, has been used for decoding in our experiments. The decoder approximates the contours by fitting splines to the corner points so that both encoder and decoder agree precisely. The proposed scheme yields a marginally higher PSNR than other methods requiring the lower number of bits. However, the algorithm to find the Minimum Perimeter Polygon (MPP) [17, 18] of a region has not been applied yet. With use of the MPP, it is expected that PSNR will improve further significantly.

The number of bits required for the motion failure region coding is usually large, so the extracted area should be relatively small. The minimum size of the motion failure region is normally set to 0.2% of the total number of pixels in an image. For the CIF 'Miss America' sequence with frame size 352×288, the minimum size is about 200 pixels. The regions with a size smaller than 200 pixels are eliminated for coding efficiency. The average size of the motion failure area over an entire frame is normally set to 1% of the total number of pixels in an image, *e.g.*, 1000 pixels in the above example. Therefore, the amount of bits required for coding is actually controlled. Examples of motion failure regions detected using this technique is shown in Figure 5. The shapes of these motion failure areas are coded by polygon or cubic smooth spline approximation. The resultant percentage of the motion failure area to the total image area is shown in Figure 6, with a mean value of 0.7%.

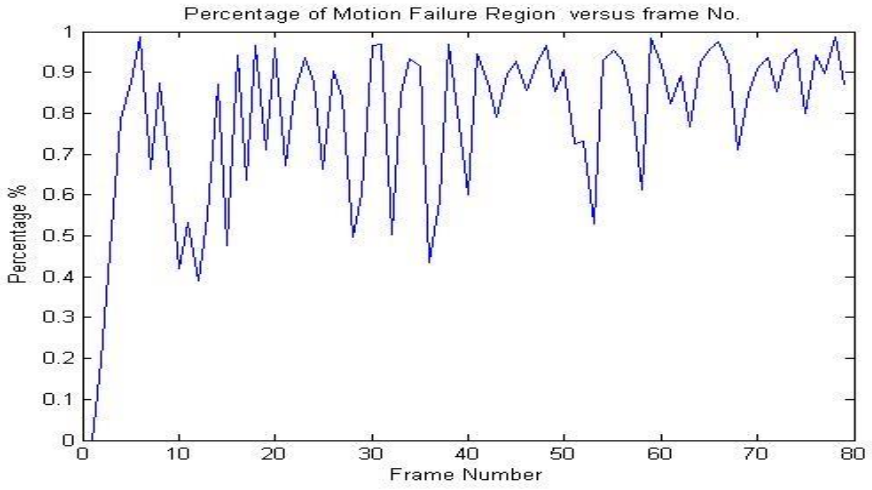

Figure 6. Percentage of Motion Failure Area to Total Image Area

4. Conclusions

The work presented above describes a hybrid video coding scheme that combines traditional block-based method with content-based coding. Primarily, it aims at low bit-rate transmission applications, such as video telephone and video conferencing. In the segmentation algorithm of the scheme, which is developed to be robust to additive noise, both illumination changes and noise are taken into account by way of dividing image into blocks and clustering the histogram of each block into multiple clusters. Uses of the maximum and average filters in the segmentation have produced smooth and stable object boundaries.

In the scheme, a coarse sampled motion field is generated by the block correlation performed inside the change detection mask. The motion field is then interpolated to a dense field by passing through 15 iterations of the Horn-Schunck algorithm. The resultant smooth dense motion estimation is therefore used for motion compensation to provide high accuracy for prediction. Using spline-based shape representation (or bounding rectangles), the scheme also achieves improved PSNR or coding efficiency compared to the classical object-based coding methods.

In addition, the modular nature of this coding scheme enables the development and use of other coding techniques, *e.g.*, shape adaptive-DCT and adaptive vector quantization in the residual encoding module, thus providing flexibility and the scope for further improvement in the future.

References

[1] S. and J. A. Robinson, "Object Based Video Coding by Global-to-Local Motion Segmentation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 12, **(2002)**.
[2] X. Cai, F. H. Ali and E. Stipidis, "Object-Based Video Coding with Dynamic Quality Control, Image and Vision Computing", vol. 28, no. 3, **(2010)**.
[3] P. Gergen, "Object-Based Analysis-Synthesis Coding of Image Sequences at Very Low Bit Rates", IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, no. 3, **(1994)**.
[4] Y. Zhang, W. Wang, L. Zheng and M. Wu, "Motion Compensation Using Polyline Based Block Partition", Proceedings of the 2nd International Congress on Image and Signal Processing, Tianjin, China, October **(2009)**.

[5] P. L. Rosin and E. Ioannidis, "Evaluation of Global Image Thresholding for Change Detection", Pattern Recognition Letters, vol. 24, **(2003)**.
[6] N. A. Tsoligkas, D. Xu and I. French, "Hybrid Object-Based Video Compression Scheme Using a Novel Content-Based Automatic Segmentation Algorithm", Proceedings of IEEE International Conference on Communications, Glasgow, UK, June **(2007)**.
[7] T. R. Singh, S. Roy, O. I. Singh, T. Sinam and K. M. Singh, "A New Local Adaptive Thresholding Technique in Binarization", International Journal of Computer Science Issues, vol. 8, no. 6, 2, **(2011)**.
[8] M. Dai, P. Baylon, L. Humbert and M. Najim, "Image Segmentation by A Dynamic Thresholding Using Edge Detection Based on Cascaded Uniform Filters", Signal Processing, vol. 52, **(1996)**.
[9] S. Olsen, "Estimation of Noise in Images: An Evaluation", Graphical Models and Image Processing, vol. 55 **(1993)**.
[10] S. C. Tai and S. M. Yang, "A Fast Method For Image Noise Estimation Using Laplacian Operator and Adaptive Edge Detection", Proceedings of the 3rd International Symposium on Communications, Control and Signal Processing, Malta, **(2008)** March.
[11] P. Kuhn, "Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation", Kluwer Academic Publishers, The Netherlands, **(2003)**.
[12] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow", Artificial Intelligence, vol. 17, **(1981)**.
[13] R. C. Gonzalez, R. E. Woods and S. L. Eddins, "Digital Image Processing Using Matlab", Gatesmark Publishing, Knoxville, TN, **(2009)**.
[14] H. G. Musman, M. Hotter and J. Osterman, "Object-Oriented Analysis-Synthesis Coding of Moving Images", Signal Processing: Image Communication, vol. 1, **(1989)**.
[15] K. Grotz, J. U. Mayer and G. K. Suessmeier, "A 64 kbits/s Video Phone Codec with Forward Analysis and Control", Signal Processing: Image Communication, vol. 1, **(1989)**.
[16] M. Hotter, "Object-Oriented Analysis-Synthesis Coding Based on Moving Two-Dimensional Objects", Signal Processing: Image Communication, vol. 2, **(1990)**.
[17] F. Hassanzadeh and D. Rappaport, "Approximation Algorithms for Finding a Minimum Perimeter Polygon Intersecting a Set of Line Segments", Algorithms and Data Structures - Lecture Notes in Computer Science, vol. 5664, **(2009)**.
[18] J. Sklansky, R. L. Chazin and B. J. Hansen, "Minimum-Perimeters Polygon for Digitized Silhouettes", IEEE Transactions on Computers, vol. C-21, no. 3, **(1972)**.

