EFFICIENT MULTIVIEW VIDEO CODING BY OBJECT SEGMENTATION

Narasak Boonthep[1], Werapon Chiracharit[1], Kosin Chamnongthai[1] and Yo-Sung Ho[2]

[1]Department of Electronic and Telecommunication Engineering
Faculty of Engineering
King Mongkut's University of Technology Thonburi
126 Pracha-utit Road, Bangmod, Tung-kru, Bangkok 10140, Thailand
Tel: (+66) 2470-9064, Fax: (+66) 2470-9070
E-mail: s0403407@st.kmutt.ac.th, werapon.chi@kmutt.ac.th, kosin.cha@kmutt.ac.th

[2]Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong, Buk-gu, Gwangju, 500-712, Korea
Phone: (+82) 62-970 2258 Fax: (+82) 62-970 3164
E-mail: hoyo@gist.ac.kr

ABSTRACT

Multi-view video consists of a set of multiple video sequences from multiple viewpoints or view directions in the same scene. It contains extremely a large amount of data and some extra information to be stored or transmitted to the user. This paper presents inter-view correlations among video objects and the background to reduce the prediction complexity while achieving a high coding efficiency in multi-view video coding. Our proposed algorism is based on object-based segmentation scheme that utilizes video object information obtained from the coded base view. This set of data help us to predict disparity vectors and motion vectors in enhancement views by employing object registration, which leads to high compression and low-complexity coding scheme for enhancement views. An experimental results show that the superiority can provide an improvement of PSNR gain 2.5–3 dB compared to the simulcast.

Keywords: Multi-view Video Coding (MVC), Multi-view video (MVV)

1. Introduction

Nowadays, the world's technologies are developing and changing all the time. Visual Communication Technology is one of the most important technology that have be improve in many fields such as 3D image processing, Holography, Multi View Video – MVV. Especially, Multi-view video have been applied into many applications like Free Viewpoint Video (FVV), Free Viewpoint Television (FVT), Video-teleconference and 3 DTV.

In the past, users are limit to see only 2D so they can access to the image only one side but 3D technology allow them to access the image with freedom view so it seem more realistic video to users. A multi-view video consists of a set of video sequences capturing the same scene simultaneously from cameras at various view directions, containing a high degree of correlation within each view as well as among the views. A full-scale multi-view video system allows interaction with scene immersion, but also demands extensive storage and transmission bandwidth due to the massive data volume. A multi-view video contains multiple views of the same scene and some extra information capturing the correlation among these views. In addition to the techniques used in traditional single view video coding, MVC must exploit the redundancies among the views. However, the exploitation of extra redundancies incurs extensive computation, counteracting the benefit gained from coding efficiency. We need to compress the multi-view sequence efficiently without sacrificing its visual quality significantly.

The goal of object segmentation is to partition a video frame into a set of meaningful objects or regions. In general, the disparity of different views is caused by different camera position, which can be regarded as camera motion to the scene. Normally, the camera motion can be modeled a 2D geometric transform. The parameter sets of the camera transform are different for different objects due to different distances of the objects to the camera. The disparity differences within an object are relatively smaller than those of different objects or the object with the background. In addition, in most multi-view systems, cameras are stationary or their relative positions are fixed during video capturing. The disparity of static background area will have less variation along the sequences. Therefore, an object-based approach better exploits the temporal correlation of the disparity and the camera geometry between views for objects and background in inter-view prediction. However, in the traditional object-based coding, segmentation has to be applied to every view, and segmentation masks are encoded and

transmitted as side information, which leads to high overhead in terms of computational complexity and output bit rate. This paper presents a video segmentation system to extract semantic objects from video frames to segment video object from backgrounds.

In this approach is aimed to improve compressing efficiency. The basis of our approach is using the latest standard coding H.264 combine with temporal and inter-view prediction. The result of using H.264 coding standard the output of video signal is better than previous coding standard (H.263) with the lower bit rate.

This paper is organized as follows: Section 2 describe about requirements and conditions for MVC**.** Section 3 describe about Experimental. Section 4 present experimental results and section 5 concludes this paper.

2. Requirements and Conditions for MVC

In this paper we have been developed for MVC. Therefore, most of the requirements as well as test data and evaluation conditions are defined by the MVC project as presented in Section 2.1-2.3

2.1 Requirements

The central requirement for any video coding standard is high compression efficiency. In the specific case of MVC, this means a significant gain compared to independent compression of each view. Compression efficiency measures the tradeoff between cost and benefit. However, compression efficiency is not the only factor under consideration for a video coding standard. Some requirements of a video coding standard may even be contradictory such as compression efficiency and low delay in some cases. Then a good tradeoff has to be found. General requirements for video coding such as minimum resource consumption, low delay, error robustness, or support of different pixel and color resolutions, are often applicable to all video coding standards. For MVC, additionally view scalability is required. In this case a portion of the bit-stream can be accessed in order to output a limited number out of the original views. Also backward compatibility is required for MVC. This means that one bit-stream corresponding to one view that is extracted from the MVC bit-stream shall be conforming to H.264/AVC. Quality consistency among views is also addressed. It should be possible to adjust the encoding for instance to provide approximately constant quality over all views. Parallel processing is required to enable efficient encoder implementation and resource management. Camera parameters (extrinsic and intrinsic) should be transmitted with the bit-stream in order to support intermediate view interpolation at the decoder.

2.2 Test Data and Test Conditions

The proper selection of test data and test conditions is crucial for the development of a video coding standard. The test data set must be representative for the targeted area of applications, and therefore cover a wide range of different content properties. The Interactive Visual Media Group of Microsoft Research for providing the Bullet data set by eight different multi-view test data sets have been used with 8 cameras views with 20 cm. spacing; 1D/arc. Picture resolutions are 1024 × 768 samples, and picture rates are 15 fps. The applications rather target high-quality TV-type video than limited channel communication-type video. Therefore, smaller resolutions like CIF or QCIF are not considered. The MVC test data set covers a wide range of different content types, moving camera systems, and different complexities of motion and spatial detail. Fig. 2 shows some examples.
In order to perform comparative evaluations, the test conditions also have to be specified. For each test sequence, three bit rate s have been chosen corresponding to low but acceptable, medium and high quality, depending on the properties and content of the particular sequence. These fixed bit rate s allow a fair comparison of different approaches for MVC in objective and subjective tests as described below.

Fig.1 Examples of multi-view video test data [7]

The main goal of MVC is to provide significantly increased compression efficiency compared to individually encoding all video signals. Therefore, encoding all views using H.264/AVC with the same test conditions was considered as the reference for coding performance comparison. The resulting decoded video signals (anchors) serve as reference for objective and subjective comparison. Encoding was done using typical settings.

2.3 Evaluation

Evaluation of video coding algorithms can be done using objective and subjective measures. The most widely used objective measure is the peak signal-to-noise-ratio (PSNR) of the signal which is given as

$$PSNR_Y = 10.\log_{10}\left(\frac{255^2}{MSE}\right) \quad (1)$$

with being the mean squared error (MSE) between the original and decoded video samples. Typically, PSNR values are plotted over bit rate and allow then comparison of the compression efficiency of different algorithms. This can be done in the same way for MVC.

However, PSNR values do not always capture video quality as perceived by humans. Some types of distortions that result in low PSNR values do not affect the human

perception in the same way. One example is a shift of the picture by one sample side wards. Therefore, any video coding algorithm can finally only be judged in subjective evaluations. The formal MVC tests were conducted by MPEG using a Single Stimulus Impairment Scale (SSIS) test. In this subjective test, subjects are being shown the decoded video signal from a candidate codec. The subjects judge the quality of decoded video on a scale from bad to excellent. The votes of the subjects are statistically analyzed to quantify subjective quality. For statistical confidence, a large number of subjects need to be involved. Display conditions, viewing room conditions (including lighting and view distance), and execution of test sessions (order of presented video, display time, etc.) require careful design. In consequence such formal subjective tests require a tremendous effort.

3. Experimental Results

3.1 Foreground Extraction

The objective of foreground extraction is to detect the moving foreground objects from a static background. For most natural scene sequences, the foreground area has an intensity that can be distinguished from its neighborhood background area. The premise of our algorithm is that we first detect an initial foreground area that we call potential foreground area, which covers all of the blocks with motion. For the reconstructed frame

Fig.2: Foreground Extraction Algorithms:
Foreground objects for frame 1 of Bullet.

3.2 Foreground Objects Clustering Segmentation

We use properties including intensity and motion vectors from block matching to segment the extracted foreground into different objects by hierarchical clustering.

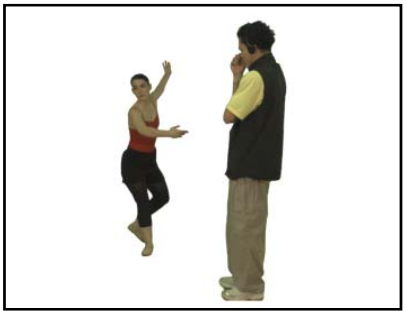

Fig.3: Foreground Objects Clustering Segmentation

The basis of the proposed algorithm is the exploitation of the intensity of the reconstructed frame and the already obtained motion vectors of the H.264/MPEG-4 AVC coded base view. The algorithm employs different procedures for intra-frame and inter-frame segmentation. For intra-frame segmentation, the algorithm uses a pixel-based foreground extraction technique when the encoded frames have low quantization parameters *(QPs)* and a block-based foreground extraction technique when the encoded frames have high *QPs*. These extracted foregrounds are feature segmented into moving objects by clustering algorithms that use motion vector and intensity characteristics to attain refined object masks. For inter frames, segmentation masks are obtained by inheriting the segmentation masks of the pixel in the previous frame indicated by its motion vector. In addition to object-based MVC, our segmentation algorithm can be applicable in the applications with pre-encoded video.

3.3 Experimental Results

Fig. 4 compares the rate-distortion performance of different multi-view video coding method average over 100 frames (at 15 frames per second) and 8 view of Bullet sequence. The lower curve (Labeled “Simulcast”) represents independent coding of each view using version JMVM 8.0 of the H.264 reference software. The Upper curve represents “Object segmentation”. It can provide an improvement of PSNR gain 2.5–3 dB compared to the simulcast.

4. Conclusions

We propose a multi-view video coding based on the H.264 which utilize both temporal/interview prediction by adaptive use object segmentation, aiming to exploit the camera geometry between views for different objects and background to achieve high coding efficiency and lower the prediction complexity. The experimental results showed that the proposed algorithm provides higher coding efficiency when compared to the segmentation-free scheme, while pertaining to lower computational complexity with better inter-view prediction. From experimental results, this leads to bit-rate savings and reduced computational complexity.

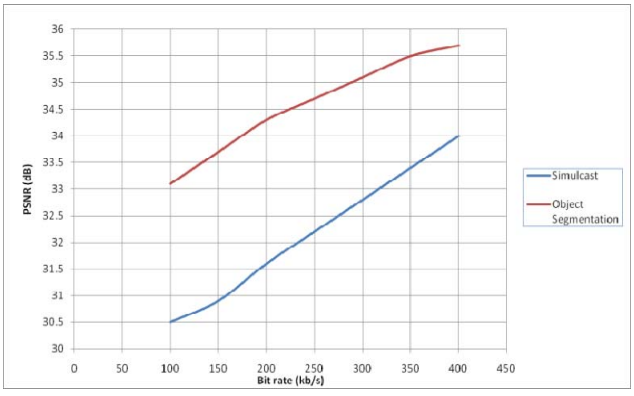

Fig.4: result for Multiview video coding

6. References

[1] V. Cheung, Brendan J. Frey, N. Jojic, “Video epitomes”, CVPR 2005. IEEE Computer Society Press, Los Alamitos, CA, June 2005.

[2] P. Merkle, A. Smolic´, K. Müller, and T. Wiegand, “Efficient Prediction Structures for Multi-view Video Coding”, IEEE Transactions on Circuits and Systems for Video Technology, VOL. 17, NO. 11, November 2007.

[3] K. Yamamoto, M. Kitahara, H. Kimata, T. Yendo, T. Fujii, M. Tanimoto, S. Shimizu, K. Kamikura, and Y. Yashima. “Multi-view Video Coding Using View Interpolation and Color Correction”, IEEE Transactions on Circuits and Systems for Video Technology, VOL. 17, NO. 11, NOVEMBER 2007.

[4] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, “Multi-view video plus depth representation and coding”. ICIP 2007 IEEE International Conference on Volume 1, pp. I - 201 - 204, Sept. 16 2007-Oct. 19 2007

[5] N. Jojic, B. J. Frey, and A. Kannan. Epitomic analysis of appearance and shape. In Proc. IEEE Intern. Conference on Computer Vision, 2003.

[6] ISO/IEC MPEG & ITU-T VCEG, Joint Multi-view Video Model (JMVM) 3.0, JVT-V207, Jan. 2007

[7] ftp://ftp.research.microsoft.com/users/sbkang/