Motion-Oriented Coding Scheme for Compression of Concentric Mosaic Scene Representations

Kehua Jiang and Eric Dubois
School of Information Technology and Engineering, University of Ottawa
800 King Edward Ave., Ottawa, Ontario, K1N 6N5, Canada
{kjiang, edubois}@site.uottawa.ca

Abstract

A motion-oriented coding scheme for compressing concentric mosaic representations is designed based on modifications to a standardized block-based hybrid video coding scheme. The motion features of the camera capturing the concentric mosaic representations are exploited to enhance the coding efficiency and improve the decoding flexibility. The coded data is organized in a hierarchical structure into a bitstream. Two-level motion vector representation and estimation is used. Simple one-level side information is embedded to facilitate random access. The optimized coding pattern is investigated to enhance the coding efficiency. Only a small portion of reference frames needs to be accessed and decoded to get a target pixel column. This enables fast data retrieval and rapid rendering. The proposed coding scheme is capable of providing high compression efficiency and fast random selective decoding for concentric mosaic rendering.

1. Introduction

Concentric mosaics were first proposed for image-based rendering (IBR) by Shum and He [6]. It is easy to capture image sequences of environments for concentric mosaic representations. No geometric or photometric modeling is required for rendering novel views, and the rendering speed is fairly high by simply reassembling slits derived from pixel columns of the images in concentric mosaic representations. Concentric mosaics allow users to freely wander in a circular region on a plane, providing the experiences of appropriate spatial parallax and lighting variances. It is popularly used in some practical applications, such as virtual reality.

Although the one-dimensional image sequence of a concentric mosaic representation has smaller image file size than those of Lumigraph [1] and Light field [2] representations, it still represents a huge amount of data. For example, the concentric mosaic image sequence Kids consisting of 2967 images occupies a storage space of 860 MB. Consequently, effective compression schemes should be applied to efficiently store and access the concentric mosaic data for further image-based rendering and transmission.

Different kinds of compression techniques could be used for coding concentric mosaics. In [6], vector quantization (VQ) was used in the compression scheme with a compression ratio of about 12:1 reported. VQ is capable of selective decoding, and can easily facilitate the random access required by IBR. However its compression efficiency is limited and can not meet the requirement of higher compression ratios in many applications. On the other hand, wavelet transform proves to be an efficient technique [4, 8]. The advantages of image resolution and quality scalabilities can be taken from incorporating wavelet transform in the coding scheme. However wavelet transform has a complex implementation structure, and its processing is computation intensive and time consuming, preventing its use in some applications.

Some compression schemes for coding concentric mosaic representations have been designed as a result of modifications to standardized video coding strategy [3, 5]. These schemes are characterized by their higher compression efficiency than VQ schemes and less complexity than wavelet-based compression schemes. Zhang and Li [9] proposed a reference block codec (RBC) approach based on the standard algorithms of motion compensation and residue coding. It classifies input images into anchor A frames and predicted P frames. A two-level hierarchical index table is stored in the bitstream for random access decoding. The MPEG-like compression scheme developed by Shum et al. [7] is another instance of modified standardized schemes. It takes each normal planar image or panorama image

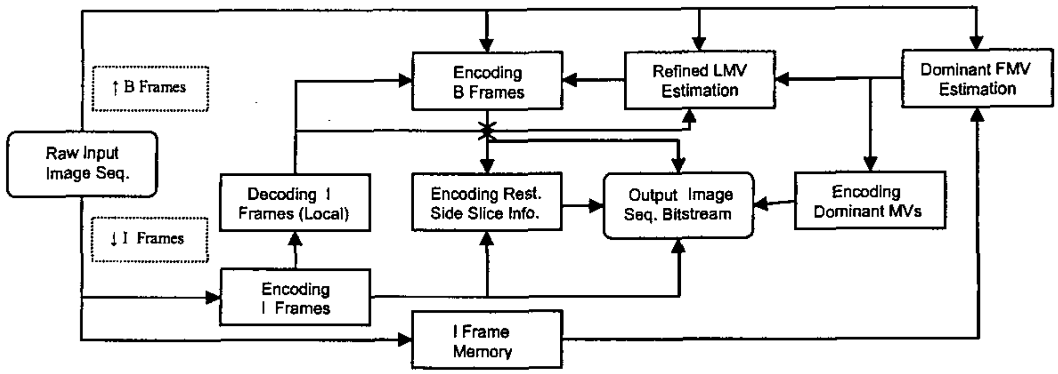

Figure 1. Motion-oriented coding scheme for compressing concentric mosaic representations

as a video frame, and exploits the image redundancies. A set of pointers to groups of blocks is embedded in the compressed bitstream for supporting random access. In order to further improve the coding performance, we propose here a motion-oriented coding scheme for compressing concentric mosaic representations. This scheme, also based on the standardized video coding strategy, has a different coded data structure and different bitstream syntax adapted for the requirements of concentric mosaic rendering. It includes two image coding modes: the intra coded I frame and the bidirectional predicted B frame and optimizes the coding pattern. The specific motion features of the camera in capturing the concentric mosaic image sequences are exploited to enhance the coding efficiency and improve the decoding flexibility. Simple one-level side information is embedded into the bitstream to facilitate the pixel-column random access. While the compression efficiency is kept at a high level, only a small portion of each reference frame needs to be accessed and decoded to get a target pixel column for synthesizing novel views. This speeds up the concentric mosaic data retrieval process and benefits the subsequent fast image-based rendering process.

In the next section, the motion-oriented coding scheme for compressing concentric mosaic representations is described in detail. In Section 3, the experimental results of the proposed scheme applied to compressing sample concentric mosaic image sequences are provided with some analyses and discussions. Finally, conclusions are drawn in Section 4.

2. Motion-Oriented Coding Scheme

The proposed motion-oriented coding scheme is based on the standardized video coding strategy which is referred to the block-based hybrid coding to exploit the intra-image redundancy by spatial transformation and the inter-image redundancy by temporal prediction enhanced by motion estimation and compensation. Some significant modifications to the standardized video coding scheme are made in order to facilitate the column random access required by concentric mosaic rendering and simultaneously keep the high coding efficiency. The block diagram of the encoding procedure of the proposed coding scheme is depicted in Fig. 1. It takes image input from the concentric mosaic representations, and processes images in two different ways according to the different image mode in the modified coded data structure. Two-level motion vector representation and estimation are used to enhance the coding performance. Side information is added for each slice to facilitate the required random access. The coded two-level motion vectors, the slice side information codes and the coded frame data for the two modes of images are all multiplexed to make up the bitstream output of the concentric mosaic representations.

2.1. Coded data structure

In order to support efficient coding, the coded data of the concentric mosaic representation is organized in a hierarchical structure with modifications based on the coded video data structure of the standardized video coding scheme. As shown in Fig. 2, the entire one-dimensional concentric mosaic representation is taken as an image sequence. This image sequence is then divided into groups of images (GOI). Each group of images with a uniform pattern begins with one image encoded as an intra coded I frame (IF) followed by a fixed number of images encoded as bidirectionally predicted B frames (BF) with reference to the nearby intra coded frames. No predicted frames are used as reference frames in order to avoid multiple dependency and to speed up the decoding process. Each I or B frame consists of modified restricted slices (RS). As novel views are synthesized in the rendering stage by combining slits derived from pixel columns of the images in the concentric mosaic image sequence, macroblocks (MB) in each frame are scanned and processed in an order starting from the macroblock on the top left corner, column by column. A modified restricted slice involves all the macroblocks in the same column to facilitate random access at the column level. In this proposed coding scheme, the macroblock which corresponds to an area of 16x16 luminance pixels is the basic unit for motion estimation and the 4:2:0 chrominance format is utilized. Consequently, each macroblock here comprises of four blocks of luminance samples and two blocks of chrominance samples. Motion compensation, spatial transformation and quantization are all performed block by block.

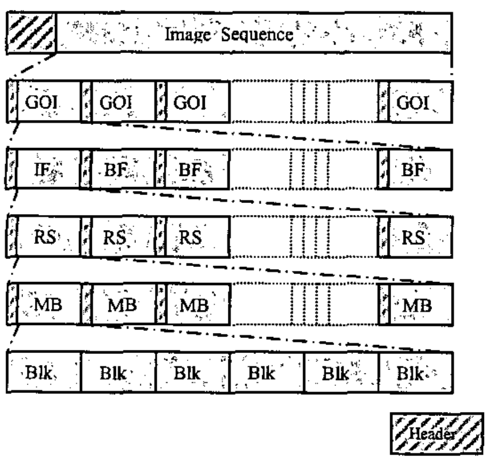

Figure 2. Coded data structure

2.2. Motion estimation and compensation

Motion estimation and compensation are applied in coding bidirectionally predicted frames to improve the prediction performance. For capturing the concentric mosaic representation of an environment, a camera is fixed at the end of a beam. The motion of the camera is constrained on a circular track in a horizontal plane by rotating the beam. The images in the concentric mosaic image sequence are shot at points evenly distributed on the circular track. The dominant motion of the camera can be regarded as a pan, resulting in a dominant horizontal image motion. These features are used to enhance the coding efficiency and to improve the random access decoding flexibility. First, the forward and backward dominant horizontal motion vectors of each bidirectionally predicted frame are determined with full pixel accuracy using the original input image data. The search ranges for dominant horizontal frame motion vectors are estimated by testing some sample images in the concentric mosaic image sequence. Then, with the central positions located by making use of the dominant frame motion vectors, full searches for the local motion vectors with full pixel accuracy are conducted for each macroblock. The sum of the absolute errors between the pixel values in the current predicted macroblock and those in each macroblock within the search range in the reference frame is computed and compared to determine the best matching local motion vector. Both the forward and the backward local motion vectors are examined for each macroblock to make the best prediction from the past reference frame, the future reference frame or both together. The size of the full search windows is 16 horizontal pixels by 10 vertical pixels in both of the past and future reference frames. The search of motion vectors is confined to the significant area of the image window. Any motion vector referring to pixels beyond the image boundaries is excluded. After that, more accurate searches for refined local motion vector with half pixel accuracy by using the locally decoded reference frame data are performed. All the eight adjacent half-pixel positions surrounding the full-pixel position located by the matching local motion vector determined from the previous step are checked in the decoded reference frames. This is done in an order line by line from upper left half-pixel position to lower right one. Motion estimation is carried out by using luminance component data, but the resulting motion vectors are also used for the block prediction and compensation of chrominance component blocks belonging to the same macroblock. The values of the motion vector used for luminance blocks are halved for applying to the corresponding chrominance blocks.

2.3. Spatial transformation and quantization

The 8x8 pixel blocks in intra macroblocks or the 8x8 pixel residue blocks in predicted macroblocks are transformed into 8x8 coefficient blocks by the two dimensional spatial discrete cosine transform (DCT). Then, linear quantization is performed for DCT coefficient blocks. The DC coefficients in intra blocks of both the luminance and chrominance components are quantized by a constant quantizer step size of 4, and all the other coefficients are quantized in two steps. First, quantizer matrixes are used to convert these coefficients into scaled DCT coefficients. An intra quantizer matrix with coarser quantization for higher frequency components is applied to the AC transform coefficients of intra blocks, and a non-intra quantizer matrix with all elements equal to a constant value is applied to the DC and AC transform coefficients of predicted blocks. After that, a uniform quantizer scale is utilized for quantizing these scaled transform coefficients. This quantizer scale is manually controlled to take different constant values to get steady image quality and achieve different compression ratios.

2.4. Bitstream Syntax

As the motion-oriented coding scheme is specially designed for coding concentric mosaic representations, its bitstream syntax is considerably simplified compared to the standard video bitstream syntax, which is designed for a much wider range of applications. Also, some modifications are made to the standard bitstream syntax corresponding to the modifications to the standardized coding scheme. The bitstream is produced by complying with the hierarchical data structure described in Section 2.1. The image sequence header is filled with the coded information of the parameters of the camera capturing the concentric mosaic representation, the radius of the camera track, the data structure parameters of the coded image sequence and the number of images in the sequence. Also, the coded dominant horizontal frame motion vectors of the bidirectionally predicted frames are put in the image sequence header. In order to facilitate the column random access and selective decoding, side information specifying the start position of each modified restricted slice in the bitstream is generated before coding the slice. This information is used in the decoding stage to locate the portion of the bitstream to be decoded. It is coded in a predictive way and the coded predictive residue is also put in the image sequence header.

2.5. Decoding Procedure

The image sequence header at the beginning of the bitstream is decoded first. It involves the configuration information needed for concentric mosaic image-based rendering. Also, the dominant frame motion vectors and the restricted slice side information are decoded at this stage. According to the requirement of the rendering technique, a specific modified restricted slice corresponding to a required pixel column is decoded by randomly accessing the bitstream and selectively decoding only a small portion of the bitstream. In order to get a pixel column in an intra-coded frame, the start point of the corresponding coded restricted slice in the bitstream is located by using the side information of the slice, and only one coded restricted slice needs to be decoded to reconstruct the required pixel column. For getting a pixel column in a bidirectionally predicted frame, the reference restricted slices in both the past and the future reference intra frames for the restricted slice in the predicted frame need to be determined first by using forward and backward dominant frame motion vectors of the predicted frame. According to the motion vector searching widow size we choose, reference is possibly made from within three successive modified restricted slices in each reference frame. The position of the reference restricted slice in the middle is determined by the dominant frame motion vector. Then, the start point of the coded restricted slice on the left is located by its side information. Three successive restricted slices beginning from this start point are decoded. This is done in the past and future reference frames separately. After that, the predicted restricted slice is selectively decoded with the assistance of its side information, and motion compensation is performed by making use of the decoded reference slices to reconstruct pixel column in the predicted restrict slice.

3. Experimental Results

The proposed motion-oriented coding scheme was implemented based on a modified video codec with standardized structure designed for MPEG-2 software simulation. Two concentric mosaic image sequences Kids and Toy were used for testing the effectiveness and efficiency of the scheme. The image sequence Kids consists of 2967 images with a resolution of 352 x 288. One sample image in Kids is shown in Fig. 3 (a). Another image sequence Toy consists of 1882 images with a resolution of 352 x 288 and a sample image shown in Fig. 4 (a).

It is expected that different coding patterns with different numbers of bidirectionally predicted frames have different impact on the coding performance. Determining the best number of B frames corresponding to a coding pattern with an optimized coding performance is something to be investigated. To answer this question, the proposed scheme was applied to process the sample sequences by coding different number of images as B frames in a group of images, under the condition that a constant quantizer scale (Q = 20) was used. The experimental results of compression ratio versus number of B frames are presented in Fig. 5. Both the compression efficiency in term of compression ratio and the image quality in term of peak signal-to-noise ratio (PSNR) should be evaluated to decide on a good coding pattern. But with an unchanged quantizer scale applied, the reconstructed images are in a constant quality not affected by the number of B frames. In this case, we can evaluate the coding performance only based on the results of compression efficiency. According to Fig. 5, the number of B frames corresponding to an optimized coding pattern for Kids is 14, and for Toy, this number is 10. Here, we only present the experimental results for the luminance component, as the coding performance is mainly determined and well represented by the luminance component. The obtained B frame numbers corresponding to the optimized coding performance are only suitable for the given sample image sequences. Generally speaking, the proper number of B frames depends on the image sampling rate and the depth range of the captured scenes. Indeed, the optimized coding pattern needs to be determined by investigating the actual image sequence to further reduce the data size and improve the coding performance.

In our motion-oriented coding scheme, no bit rate control technique is incorporated, as steady and high quality images need to be reconstructed at the rendering stage. A uniform quantizer scale parameter is applied for the whole image sequence. It is set to different values to change the compression ratio. This leads to different image qualities as well. The experimental results of compression ratio versus quantizer scale are presented in Fig. 6 for the two sample image sequences, and the results of the PSNR versus quantizer scale are presented in Fig. 7. These results are obtained by setting the coding patterns with 14 B frames for Kids and 10 B frames for Toy, which corresponds to the appropriate B frame number of the optimized coding pattern. It is shown that the compression ratio is approximately linearly proportional to quantizer scale. Thus, the coding efficiency can be well controlled by selecting proper quantizer scale values. Meanwhile, the quality of the reconstructed images gets worse with the increase of the quantizer scale value. Consequently, a tradeoff should be made between high compression efficiency and a good quality of reconstructed images, and the suitable quantizer scale value determined accordingly.

Two samples of reconstructed images compared with the encoded input image are shown in Figure 3 and Figure 4. The reconstructed image of Kids in Figure 3 (b) is encoded with a coding pattern of 14 B frames and a quantizer scale of 20, and the compression ratio is 44. The reconstructed image of Toy in Figure 4 (b) is encoded with a coding pattern of 10 B frames and a quantizer scale of 20, and the compression ratio is 67. In image-based rendering, the requirement for steady and high quality of reconstructed images limits the increase of the compression ratio. If the quality requirement is lowered, the compression ratio can be higher.

Figure 3. An image in Kids: (a) input image (b)decoded image (Q=20, N=14, CR =44)

Figure 4. An image in Toy: (a) input image (b)decoded image (Q=20, N=10, CR =67)

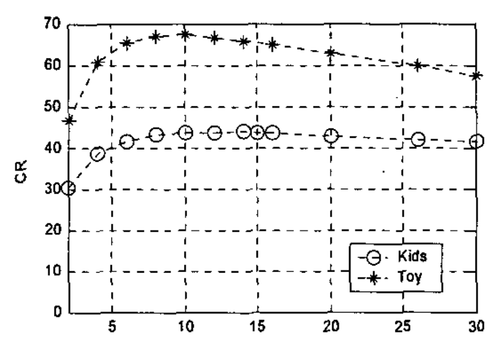

Figure 5. Compression ratio vs. B frame number (Q = 20)

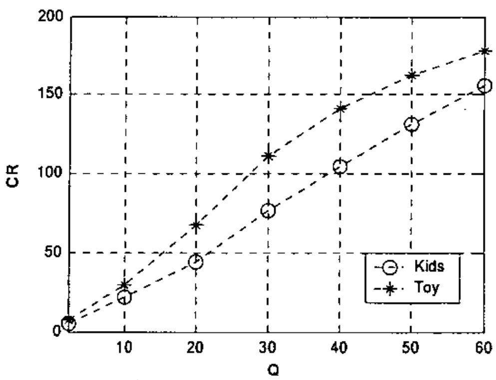

Figure 6. Compression ratio vs. quantizer scale (Kids: N = 14, Toy: N = 10)

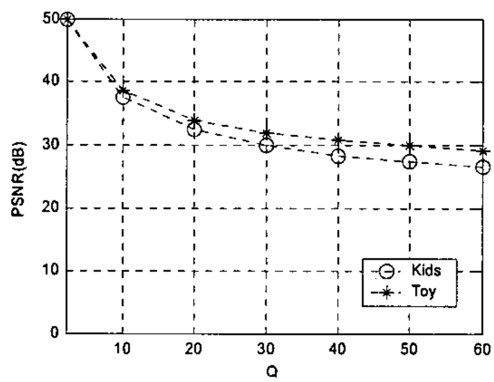

Figure 7. PSNR vs. quantizer scale (Kids: N = 14, Toy: N = 10)

4. Conclusions

A motion-oriented coding scheme for compressing concentric mosaic image-based representations was designed by exploiting the motion features of the camera capturing the concentric mosaic image sequences. This scheme is based on the standardized video coding strategy to take advantage of its high coding efficiency. Significant modifications were made to the standardized video coding scheme to meet the requirements of concentric mosaic image-based rendering. A different coded data structure and different bitstream syntax were adopted for compressing concentric mosaic representations. Simple one-level side information is embedded into the bitstream to facilitate the pixel-column random access at the decoding stage. A proper coding pattern of the group of images should be determined to optimize the coding performance. Only a small portion of each reference frame needs to be accessed and decoded to get a target pixel column for synthesizing novel views. This speeds up the concentric mosaic data retrieval and benefits the subsequent fast image-based rendering. The proposed motion-oriented coding scheme is capable of providing high compression efficiency and random selective decoding for concentric mosaic representations. The compression efficiency can be controlled by choosing appropriate quantizer scale values.

Acknowledgements

The authors wish to acknowledge the funding from LORNET, an NSERC research network, and Microsoft for providing the test concentric mosaic image sequences.

References

[1] S. Gortler, R. Grzeszczuk, R. Szeliski and M. F. Cohen, "The Lumigraph," *Proc. ACM Annu. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH'96)*, New Orleans, LA, USA, Aug. 1996, pp. 43-52.

[2] M. Levoy and P. Hanrahan, "Light Field Rendering," *Proc. ACM Annu. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH'96)*, New Orleans, LA, USA, Aug. 1996, pp. 31-42.

[3] J. Li, H. Shum and Y. Q. Zhang, "On the Compression of Image Based Rendering Scene," *Proc. IEEE Int. Conf. Image Processing (ICIP'00)*, Sept. 2000, vol. 2, pp. 21-24.

[4] L. Luo, Y. Wu, J. Li and Y. Q. Zhang, "Compression of Concentric Mosaic Scenery with Alignment and 3D Wavelet Transform," *Proc. SPIE Image and Video Communications and Processing Symp.*, San Jose, CA, USA, Jan. 2000, vol. 3974, pp. 89-100.

[5] K. T. Ng, S. C. Chan, H. Y. Shum and S. B. Kang, "On the Data Compression and Transmission Aspects of Panoramic Video," *Proc. IEEE Int. Conf. Image Processing (ICIP'01)*, Thessaloniki, Greece, Oct. 2001, vol. 2, pp. 105-108.

[6] H. Shum and L. He, "Rendering with Concentric Mosaics," *Proc. ACM Annu. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH'99)*, Los Angeles, CA, USA, Aug. 1999, pp. 299-306.

[7] H. Shum, K. Ng and S. Chan, "Virtual Reality Using the Concentric Mosaic: Construction, Rendering and Data Compression," *Proc. IEEE Int. Conf. Image Processing (ICIP'00)*, Sept. 2000, vol. 3, pp. 644-647.

[8] Y. Wu, C. Zhang and J. Li, "Smart Rebinning for the Compression of Concentric Mosaic," *IEEE Trans. on Multimedia*, vol. 4, no. 3, pp. 332-342, Sept. 2002.

[9] C. Zhang and J. Li, "Compression and Rendering of Concentric Mosaics with Reference Block Codec (RBC)," *Proc. SPIE Visual Communications and Image Processing Symp.*, Perth, Australia, June 2000, vol. 4067, pp. 43-54.