

# COMPRESSION OF CONCENTRIC MOSAIC WITH REBINNING OF SLITS (ROSS)

*Yunnan Wu\**

Univ. of Sci. & Tech. of China  
Hefei, 230026, China

*Jin Li*

Microsoft Research China  
Sigma Ctr., 49 Zhichun Road, Haidian, 100080, China

## ABSTRACT

Concentric mosaic offers a quick solution for the construction and navigation of a virtual environment. However, the associated huge amount of data presents a heavy burden for its application. In this paper, we propose the rebinning of slits (ROSS) approach for the compression of concentric mosaic. The proposed scheme decorrelates the 3D concentric mosaic data set by rebinning the vertical slits from all captured image shots into a 2D panorama, and then compresses the panorama with a highly efficient still image coder, such as JPEG2000. Unlike typical video coders that use motion compensation to decorrelate multiple frames, the rebinning converts 3D spatial correlation into stronger 2D correlation. Experimental results show that ROSS outperforms MPEG-2 coding of concentric mosaic by an average of 1.0dB.

## 1. INTRODUCTION

Concentric mosaic[1] is an image based rendering (IBR) scheme with 3D parameterization of the plenoptic function[2]. It is very useful for generating photo realistic views of the synthetic and real world scenery in real time. By rotating a single off-center camera and recording the captured images at regular intervals, a concentric mosaic scene is built up quickly, and new views can be easily created by interpolating nearby light rays. Concentric mosaic easily models the 3D environment and enables virtual walkthrough. However, the amount of data is huge, which is a heavy burden for storage, transmission and display. A sample concentric mosaic scene from [1] includes 1350 RGB images with resolution 320x240 and occupies a total of 297MBs. Efficient compression is thus essential.

Since concentric mosaic consists of a sequence of images, its compression resembles that of the video. However, concentric mosaic bears unique characteristics, which lead to new challenges in compression. First, concentric mosaic consists of shots of a static environment with a swinging camera, which implies more redundancies than an ordinary video sequence. Second, the distortion tolerance of concentric mosaic is low, as each rendered image is to be viewed statically, and the human visual system (HVS) is much more sensitive to static distortions than time-variant distortions. Since a rendered view of concentric mosaic is not known at the compression stage, certain HVS properties such as spatial and temporal masking may not be used. Moreover, a compressed video bitstream is played frame by frame, while access to the compressed concentric mosaic is dynamically determined by the user's navigation. Fully decompressing the entire data volume is often beyond the memory capability of an ordinary PC and will introduce a long initialization delay. It is therefore essential to maintain the data in compressed form, and to decode only the content nec-

essary to render the current view. We call such concept just-in-time (JIT) rendering. JIT rendering is a key to the concentric mosaic compression algorithm. To facilitate JIT rendering, the bitstream needs to be random accessible and the decoder should be reasonably fast to accommodate the real-time decoding need.

In [1], a spatial vector quantization (SVQ) scheme has been proposed to compress concentric mosaic. SVQ is fast in decoding, and the compressed SVQ index can be easily accessed at arbitrary location. However, SVQ is complex in the encoding stage, and the compression performance of SVQ is limited, e.g., 12:1. If we raise the compression ratio higher, the scene quality degrades quickly. An alternative approach is to treat the concentric mosaic as a video and compress it with a video coder, such as MPEG2[4]. The approach does not take the random access requirement into consideration, and thus may not be practical for concentric mosaic rendering. In addition, although MPEG-2 is very efficient in video coding, it may not be the most efficient when applied to concentric mosaic. We have proposed a 3D wavelet approach in the compression of concentric mosaic[3]. The 3D wavelet algorithm achieves good compression ratio, and the multi-resolution capability offered is an attractive feature in the rendering and Internet streaming application. However, the complexity of the 3D wavelet approach is high.

Concentric mosaic may be compressed frame by frame with a high performance still image coder, such as the JPEG 2000[5]. The compressed bitstream can be easily accessed. However, since concentric mosaic consists of multiple correlated shots, this may not be very efficient. In this work, we propose a preprocessing technique to efficiently realign the concentric mosaic scene from 3D to 2D. We call the approach rebinning of slits (ROSS). In ROSS, each concentric mosaic image is split into vertical slits which are also the elementary access units in the rendering operation. Instead of using frame prediction as in MPEG-2 or 3D wavelet to explore the redundancies across image shots, the slits are rebinned into a 2D panorama so that the cross-frame redundancy is converted into intra-frame redundancy. The rebinned panorama is then encoded with a still image coder. In this work, we use the JPEG 2000 to compress the rebinned image, however, other state-of-the-art image compression schemes may be used as well. The proposed approach is shown to be superior to MPEG-2 in compressing the concentric mosaic.

The rest of the paper is organized as follows. The acquisition and display of the concentric mosaic is covered in Section 2. Section 3 describes the details of the slits rebinning operation. In Section 4, we give a performance evaluation of the ROSS algorithm. Finally, a conclusion is drawn in Section 5.

\* This work was performed during Mr. Wu's internship at Microsoft Research China. Please direct all correspondences to Dr. Jin Li. Tel: +86 1062617711 Ext. 5793. Fax: +86 1062555337, Email: jinl@microsoft.com.

## 2. THE CONCENTRIC MOSAIC SCENE

A concentric mosaic scene is captured by mounting a camera at the end of a level beam, and shooting images at regular intervals as the beam rotates, as shown in Figure 1. Let the camera shots taken during the rotation be denoted as  $F_n = \{c(n, w, h) / w, h\}$ , where  $n$  indexes the camera shot,  $w$  indexes the horizontal position within a shot, and  $h$  indexes the vertical position. Let  $N$  be the total number of camera shots,  $W$  and  $H$  be the horizontal and vertical resolution of each camera shot, respectively. Concentric mosaic can be treated as a series of camera shots  $F_n$ , or alternatively be interpreted as a series of mosaic panoramas  $P_w = \{c(n, w, h) / n, h\}$ , each of which consists of vertical slits at position  $w$  of all camera shots. Panorama  $P_w$  can be considered as taken by a virtual slit camera rotating along a circle co-centered with the original beam with a radius  $r = R \sin \theta$ , where  $R$  is the radius of the rotation beam,  $r$  is the equivalent radius of the slit camera, and  $\theta$  is the angle between ray  $w$  and the camera normal. Since the entire data volume  $P_w$ ,  $w = 0, \dots, W-1$  can be considered as a stack of co-centered mosaic panoramas with different radius, it is named concentric mosaic [1].

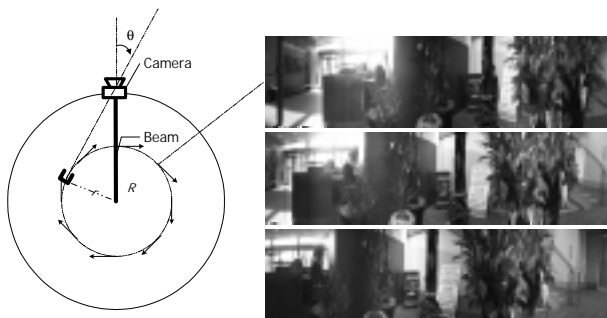


Figure 1 The concentric mosaic imaging geometry

Rendering concentric mosaic involves reassembling slits from the captured data set. Shown in Figure 2, let the horizontal field of view of the camera be  $FOV$ , concentric mosaic can render arbitrary view within an inner circle of radius  $r = R \sin(FOV/2)$ . Let  $P$  be a novel viewpoint and  $AB$  be the field of view to be rendered. We split the view into multiple vertical slits, and render each slit independently. A basic hypothesis behind concentric mosaic rendering is that the intensity of any ray does not change along a straight line unless blocked. Thus, when a slit  $PV$  is rendered, we simply search for the slit  $P'V$  in the captured dataset, where  $P'$  is the intersection point with the camera track. Because of the discrete sampling, the exact slit  $P'V$  might not be found in the captured dataset. Let the four sampled slits closest to  $P'V$  be  $P_1V_{11}$ ,  $P_1V_{12}$ ,  $P_2V_{21}$  and  $P_2V_{22}$ , where  $P_1$  and  $P_2$  are the two nearest captured shots,  $P_1V_{11}$  and  $P_1V_{12}$  are the slits closest to  $P_1V$  in shot  $P_1$ , and  $P_2V_{21}$  and  $P_2V_{22}$  are closest to  $P_2V$  in shot  $P_2$ . We may choose only the slit that is closest to  $P'V$  to approximate the intensity of  $PV$ . However, a better approach is to use bilinear interpolation, where all four slits are employed to interpolate the rendered slit  $PV$ . The environmental depth information may assist in finding the best approximating slits and alleviate the vertical distortion. More detailed description of concentric mosaic rendering may be found in [1]. Our key observation is that concentric mosaic data is accessed by slits, and to render a view, only a partial subset of the concentric mosaic data set needs to be accessed.

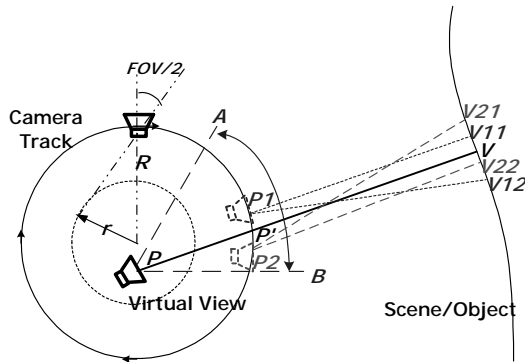


Figure 2 Rendering with concentric mosaic

## 3. SLITS REBINNING AND COMPRESSION OF CONCENTRIC MOSAIC

### 3.1 Motivation

We first analyze the volume formed by the concentric mosaic data set, which is shown as a cube in Figure 5. The front slice of the cube corresponds to a mosaic panorama  $c(n, ; h)$ , the side gives an image shot  $c(, w, h)$ , and the top shows a cross-section slice along the vertical direction  $c(n, w, ;)$ . Notice that the top slice in Figure 5 comprises primarily of straight lines with different slopes or slightly bending curves. Such cross-section is called epipolar plane image (EPI). Since each object is captured multiple times during the camera movement, a spatial point leaves a continuous trace in the EPI. As a camera moves, each point in the image moves at a different rate depending upon its distance to the camera. The close-by object moves faster, while the faraway background moves slower. The traces in the EPI thus possess distinct slopes or curvatures.

The lines and curves in the EPI imply vast amount of redundancy in the data volume of the concentric mosaic. The objective of compression is to reduce such redundancy, both within and across shots. Correlation within frame can be easily reduced through a 2D transform, such as DCT or wavelet; correlation across shots is normally addressed through a 3D transform or predictive coding. However, in this work we investigate an alternative approach. We propose the rebinning of slits (ROSS) to convert the cross-frame correlation into stronger intra-frame correlation. Figure 3 illustrates the rebinning operation, where the 3D concentric mosaic (the upper block) is rebinned into a large 2D panorama (bottom line). We look downward at the concentric mosaic volume, so that each rectangular block in Figure 3 represents a vertical slit. The image shots are split into vertical slits, and combined into a 2D panorama, clustering multiple occurrences of the same object together. The goal is to generate an image panorama which comprises all the content of the original concentric mosaic and yet is as smooth as possible. After such 3D to 2D conversion, still image compression techniques may be applied to compress the resultant concentric mosaic very efficiently.

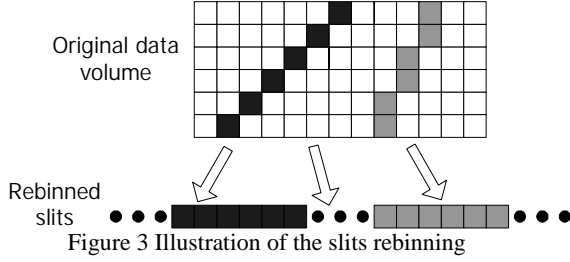


Figure 3 Illustration of the slits rebinning

We may perform the clustering by analyzing the structure of the EPI and rearrange the slits accordingly. However, recovering the structure information is usually difficult. Here the rebinning is performed through a simple slit-by-slit insertion. We insert each slit to a “best” position with regard to some smoothness criteria. Part of the rebinned panorama of scene *Lobby* is shown in Figure 4. We observe that the rebinned panorama is very smooth which is suitable for high ratio compression. The step-by-step description of the slit rebinning process is provided in the following section.

### 3.2 The Slit Rebinning Process

Recall that  $C=\{c(n,m,k)|n=1\cdots N, m=1\cdots M, k=1\cdots K\}$  represents the entire concentric mosaic, where  $n,m$  and  $k$  index the image shot, the horizontal and vertical positions, respectively. Let  $F=\{f(p,k)|p=1\cdots P, k=1\cdots K\}$  be the resultant panorama, where  $P$  is the number of slits in the panorama, which satisfies:  $P=N \times M$ . Let  $G(n,m)$  be an index function which provides the location of the slit  $C(n,m,k)$  in the panorama. The ROSS algorithm can be described as follows:

#### Step 1. Initialization

We initialize the set  $F$  with all the slits from the first concentric mosaic shot, i.e.,

$$f(m,k)=C(1,m,k), \quad \text{where } m=1\cdots M, k=1\cdots K.$$

#### Step 2. Insertion of slits one by one into $F$ .

We enumerate shot by shot, and within each shot, we enumerate slit by slit. For each slit  $c(n,m, \cdot)$  in consideration, it is inserted into  $F$  between slits  $p$  and  $p+1$  where the insertion point  $p$  can be calculated as follows:

$$\min_p \sum_{k=1}^K \{|f(p,k)-C(n,m,k)|+|f(p+1,k)-C(n,m,k)|\}$$

That is, we insert the current slit into a position where the sum of absolute difference between the slit and its two neighboring slits in the panorama is minimized. An exhaustive search may lead to an optimal solution, however, it is computationally expensive. Assuming that concentric mosaic is shot with a clockwise swinging camera, and shot  $c(n,m, \cdot)$  precedes  $c(n+1,m, \cdot)$ , we know that the panorama grows at the right when more and more camera shots are added. We may significantly speed up the search with the constraint:

$$p > G(n-1,m) \quad \text{and} \quad p > G(n,m-1). \quad (1)$$

#### Step 3. Coding of the rebinned panorama

After the rebinning process, a still image coder is used to compress the rebinned 2D panorama. In this work, we use the JPEG 2000 for its high compression and region of interest access functionality.

To avoid buffering a full rebinned panorama, the coding is performed before the rebinned panorama is completely gener-

ated. Since the panorama grows to the right, we can start coding those panorama slits that are already stable at the left end side. During the insertion of frame  $n$  slit  $c(n,m, \cdot)$ , all slits left to  $G(n,1)$  in the rebinned panorama can be coded. Coupled with the pipeline coding of JPEG 2000, we do not need a full buffer to hold the rebinned panorama, but only a relatively small buffer to hold the panorama left of  $G(n,1)$ .

#### Step 4. Encoding of the mapping index.

We encode the inverse of the mapping index,  $G^{-1}(p)$ ,  $p=1\cdots P$ , which records the source shot and slit number of each rebinned slit  $f(p,k)$ . With constraint (1), the left-to-right order of slits in the same shot is preserved, i.e., if slit A is on the right of slit B of the same source concentric mosaic shot, the mapping of slit A will be on the right of slit B in the rebinned panorama. Consequently, we need only to record the source shot number of  $G^{-1}(p)$ . To further reduce the entropy of mapping index coding, the second order differentials of the shot number of  $G^{-1}(p)$  is calculated:

$$H(p)=G^{-1}(p)-2G^{-1}(p-1)+G^{-1}(p-2),$$

and the result is compressed by LZW. With the above operation, the entropy of the mapping index has been reduced to around 2.75 bits per slit, which occupies only a small portion of the overall bit-stream.

### 3.3 Decoding and rendering of concentric mosaic

We do not expand the ROSS compressed concentric mosaic and then render it. Instead, a selective decompressor is implemented. The user decides the current viewing position and angle. The rendering engine then finds in the original panorama a set of slits which are used to render the current view. By decoding the index function  $G^{-1}(p)$ , the relationship between the concentric mosaic data set and the rebinned panorama can be uniquely determined. We thus transform the slit coordinate from the concentric mosaic data set to the rebinned panorama. Finally, the selective ROI decoder from JPEG 2000 can be used to selectively access and decompress bitstream segments correspond to the accessed slits.

## 4. EXPERIMENTAL RESULTS

The performance of the proposed rebinning of slits (ROSS) for the compression of concentric mosaic is evaluated with experimental results. The test datasets are the scene *Lobby* (1350x320x240) and *Kids* (975x360x288). The *Lobby* scene is compressed at 0.2bpp and 0.4bpp, respectively. The *Kids* scene has more details, and is thus compressed at 0.4bpp and 0.6bpp, respectively. The objective peak signal-to-noise ratio (PSNR) is measured between the original and decompressed scene. We convert the scene from RGB to YUV color-space with 4:2:0 sub-sampling. Since it is the Y component that matters most in compression, we comment only on the Y component PSNR in the following, though the results of all three components Y, U and V are shown in Table 1.

We compare the ROSS with two benchmark algorithms. The first benchmark algorithm treats the entire concentric mosaic as a video and compresses it with a MPEG-2 codec, which we download from [www.mpeg.org](http://www.mpeg.org). The second benchmark algorithm simply compresses each individual shot independently with JPEG 2000 VM 5.0.

It is observed that ROSS outperforms independent JPEG 2000 compression by 3.3-4.6dB, with an average performance gain of 4.0dB. It almost doubles the compression ratio. Since the rebinned panorama in ROSS is also compressed with the same JPEG 2000 codec, the comparison demonstrates the gain achieved through slit rebinning, and shows the effectiveness of the ROSS. The performance gain of ROSS versus MPEG2 ranges from 0.6-1.3dB, with an average of 1.0dB. ROSS thus presents an alternative way to explore the cross-frame redundancy in concentric mosaic. Moreover, concentric mosaic compressed by MPEG2 may not be accessed randomly, but the one compressed with ROSS may.

Table 1 Experimental results for compression of concentric mosaic

	LOBBY(0.4bpp)	LOBBY(0.2bpp)
MPEG2 (dB)	Y:35.3 U:41.7 V:40.7	Y:32.4 U:40.3 V:39.6
JPEG 2000 (dB)	Y:32.1 U:38.7 V:38.0	Y:28.4 U:36.6 V:36.2
ROSS (dB)	Y:36.5 U:42.4 V:41.7	Y:33.0 U:40.3 V:39.9
	KIDS(0.4bpp)	KIDS(0.6bpp)
MPEG2 (dB)	Y:29.9 U:38.1 V:38.1	Y:31.7 U:39.4 V:39.5
JPEG 2000 (dB)	Y:27.4 U:33.6 V:33.8	Y:29.4 U:35.3 V:35.3
ROSS (dB)	Y:30.7 U:36.8 V:37.4	Y:33.0 U:38.3 V:38.8

## 5. CONCLUSION

In this paper, we propose the rebinning of slits (ROSS) to compress concentric mosaic. ROSS reduces the spatial redundancy of concentric mosaic via clustering the similar content. The 3D concentric mosaic data set is rebinned (realigned) into a smooth 2D panorama, which can be efficiently compressed. Experimental results show that rebinning significantly im-

proves the compression performance and achieves 1.0dB gain over MPEG-2.

## 6. ACKNOWLEDGEMENT

The authors would like to acknowledge Harry Shum, Honghui Sun and Minsheng Wu for the raw concentric mosaic data and the source code of the concentric mosaic browser.

## 7. REFERENCES

- [1] H.-Y. Shum and L.-W. He. "Rendering with concentric mosaics", in *Computer Graphics Proceedings, Annual Conference series (SIGGRAPH'99)*, pp. 299-306, August 1997, Los Angeles.
- [2] L. McMillan and G. Bishop, "Plenoptic modeling: an image-based rendering system", *Computer Graphics Proceedings, Annual Conference series (SIGGRAPH'95)*, pp. 39-46, Aug. 1995.
- [3] L. Luo, Y. Wu, J. Li, and Y.-Q. Zhang, "Compression of concentric mosaic scenery with alignment and 3D wavelet transform", *SPIE Image and Video Communications and Processing*, SPIE 3974-10, San Jose, CA, Jan. 2000.
- [4] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG video compression standard*. New York: Chapman and Hall, 1996.
- [5] Verification model ad-hoc group, "JPEG 2000 verification model 5.0", *ISO/IEC JTC 1/SC 29/WG 1, WG1N1420*, Los Angeles, CA, Oct. 1999.

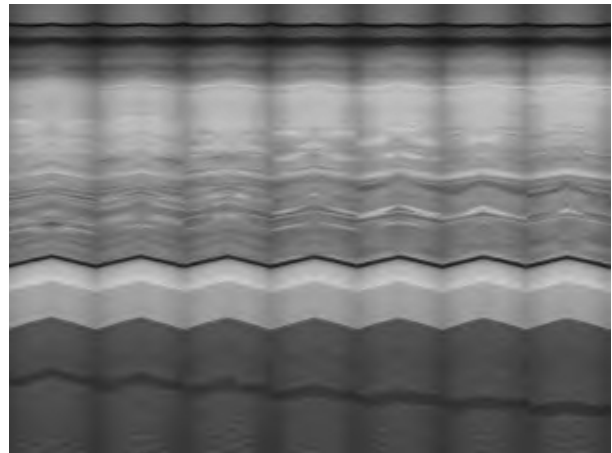


Figure 4 Part of the rebinned panorama (LOBBY)



Figure 5 The data volume of the concentric mosaics (KIDS)