A STUDY ON TEXTURE FILTERING USING JPEG 2000 SCALABILITIES

Yuki KOBAYASHI, Masaaki FUJIYOSHI, and Hitoshi KIYA

Department of Information and Communication Systems, Tokyo Metropolitan University
6–6 Asahigaoka, Hino-shi, Tokyo 191–0065, Japan

ABSTRACT

In computer graphics (CG), mipmapping is used for texture filtering. This paper studies a utilization of JPEG 2000 (JP2) features in mipmapping. JP2's spatial scalability allows us to easily and efficiently generate mipmaps for mipmapping which mipmaps consist of images with various resolutions. In addition, other scalabilities of JP2 offers further versatile and efficient texture rendering. In experimental results, it is verified that some relations between JP2 encoding parameters and texture rendering results, and is also verified the effectiveness of the utilization.

***Index Terms*—** Mipmapping, Interpolation, Geographic information systems

1. INTRODUCTION

According to the development of hardware, computer graphics (CG) has become to be more and more leveraged for a lot of purposes. For cinema or video games, the *texture* of geometry model's surfaces are rendered by mapping texture images to the surfaces with interpolation referred to as *texture filtering*. Among texture filtering techniques, *mipmapping* [1] is attractive for its low cost performance, but mipmapping requires texture images with various resolutions that are referred to as *mipmap*s. This fact brings a problem such as an increase of required storage capacity and/or transmission channel capacity. For instance, applications such as Google Earth [2] and Nasa World Wind [3] render the texture with high quality satellite images which are transmitted through a network. Thus, a network bandwidth aware mechanism is demanded in which only the most important part of the image is transmitted.

To overcome these problems or demands, JPEG 2000 (JP2) [4, 5] is utilized to generate mipmaps [5]. By using the spatial scalability of JP2, various resolution images are decoded from one JP2 coded image. That is, mipmapping requires to store and/or transmit only one JP2 coded image for one texture. However, the effectiveness of this JP2 utilization for mipmapping has not been investigated. Moreover, other scalabilities of JP2 also offers us further versatile and efficient texture rendering. So, this paper investigates relations between encoding parameters in JP2 and rendering results, and it also verifies the effectiveness of the utilization.

Fig. 1. An example of mipmaps.

2. PRELIMINARY

This section briefly describes JP2 [4, 5] and mipmapping [1].

2.1. JP2

JP2 [4, 5] is the latest international standard for still image compression that has many features. By using the spatial scalability based on discrete wavelet transformation (DWT), images with $1/2^n$ resolution can be decoded from one JP2 coded image where n is the DWT decomposition level. Moreover, the SNR scalability that serves progressive image quality improvement is served by quality layers. Furthermore, nonoverlapping spatial division referred to as tile division allows us to compress each tile independently with different parameters.

2.2. Mipmapping

As previously mentioned, mipmaps, e.g., Fig. 1, are required by mipmapping per texture [1]. The image with the particular resolution in mipmaps is specified as mipmap level k. Mipmap level 0 has the original resolution and mipmap level k's resolution is scaled down according to k increasing. So mipmapping consumes huge storage area and huge channel capacity in applications where many texture images are required, e.g., geographical information systems.

3. MIPMAPPING UTILIZING JP2 SCALABILITY

Introducing JP2 coded images to mipmapping has been proposed [5], in which mipmaps consisting of $1/2^n$ resolution images are obtained by decoding only one JP2 coded image using the spatial scalability. This approach serves efficient texture rendering; only one compressed image is stored and

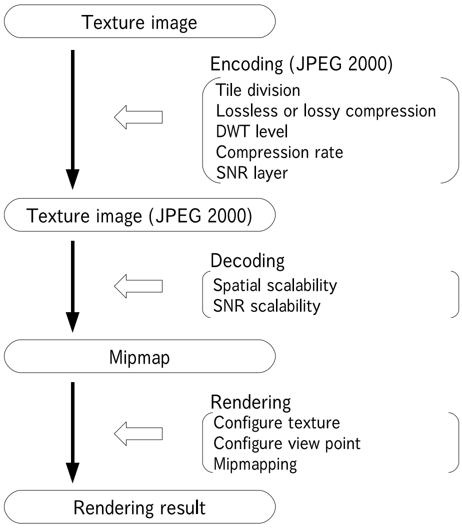


Fig. 2. The process of mipmaping utilizing JP2's scalabilities.

Table 1. Settings for XNA.

Screen resolution	800×600
Effect	BasicEffect
Field of view angle	90°
Point of gaze	the center of the plate

transmitted rather than storing and transmitting many images with various resolution.

Furthermore, this paper introduces other JP2 scalabilities to mipmapping. The process of mipmapping utilizing the scalabilities of JP2 is summarized as Fig. 2. By using such scalabilities of JP2, applications in which texture images are transmitted from a server to a terminal to render textures at the terminal side are available for poor environments with a slow network and/or an poor terminal.

JP2 has encoding parameters that also control the scalabilities. The parameters related for mipmapping are tile size, lossless or lossy compression, DWT level, compression rate, and the number of layers. These parameters affect the quality of rendering result.

4. EXPERIMENTAL RESULTS

This section investigates the relations between JP2 encoding parameters and rendering result qualities for the process shown in Fig. 2.

Results were obtained by a Windows XP computer with GeForce 9500 GT GPU. Kakadu [6] is used as the JP2 codec. Parameters being not referred to are implicitly set to the default in it. Microsoft XNA 3.1 [7] is used for rendering and its settings are shown in Table 1.

Two 1024×1024-sized 8-bits grayscale images, "Pentagon" and "Man" that are shown in Fig. 3, are used as texture images. Textures are mapped on a square shaped flat plate, represented with two triangle polygons, by mipmapping where the mipmaps are decoded from JP2 coded texture images.

(a) Man (b) Pentagon

Fig. 3. Texture images.

Table 2. Encoding parameters for evaluating the compression rate.

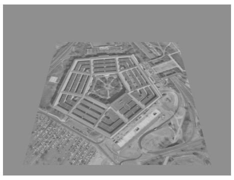
	Reference	Experiments
The number of tiles	1	1
Lossless or lossy compression	lossy	lossless or lossy
DWT level	5	5
Compression rate	not specified	0.1, 0.5, 1
The number of layers	5	5

This paper investigates the effect of the compression rate and the DWT level among various JP2 encoding parameters. In addition, the effectiveness of combination of the tile division and the SNR scalability is shown.

4.1. Compression Rate

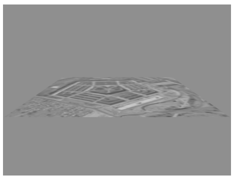
According to Fig. 2, texture images are compressed by JP2 with parameters shown in Table 2 where the parameters for the reference are used to generate the reference rendering result that to be compared with rendering result generated with various parameters indicated as Experiments. Compressed images are decompressed to images with various resolutions using spatial scalability to generate mipmaps. The surface of the plate is rendered by mipmapping using the generated mipmaps, from four viewpoints shown in Fig. 4. The quality of the rendering result is evaluated in terms of the PSNR between the reference and an experimental result, where only the texture drawn pixels are used to calculate the PSNR.

For each viewpoint, the PSNR's between the rendering result with various compression rates and the reference are shown in Fig. 5. From Fig. 5, it is found that the lossless compression is worse than the lossy compression. This is because the used DWT filters are different between lossless and lossy compression, and this difference decreases the PSNR for lossless compression. For lossy compression, the PSNR's decrease according to the compression rate decreasing. It is further found that the PSNR's of Fig. 5 (d) is the highest



(a) depression 53°, and distance 140.

(b) depression 53°, and distance 440.



(c) depression 14°, and distance 140.

(d) depression 14°, and distance 440.

Fig. 4. Viewpoints.

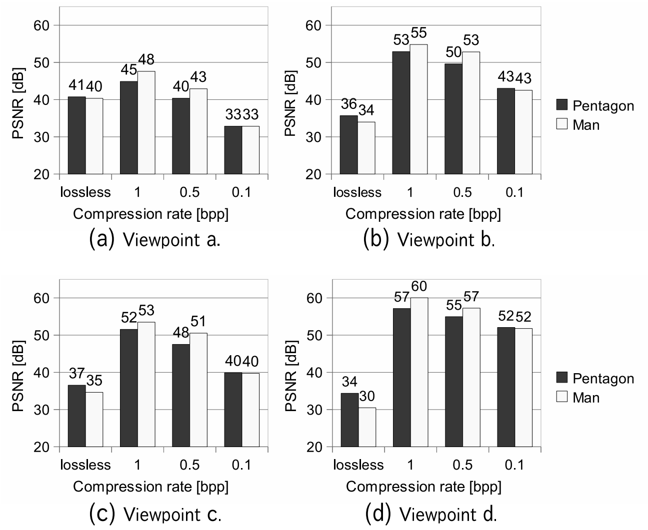


Fig. 5. The quality of rendering results versus the compression rate.

among different viewpoints. This means that PSNR becomes high in such viewpoint where the distance from the viewpoint to the plate is far or the angle of depression of the plate from the viewpoint is low. It is confirmed that both compression rate and the viewpoint affect the quality of the rendering result.

4.2. DWT Level

The DWT level that gives the number of available resolutions is varied in this section. The texture images are compressed using parameters shown in Table. 3.

The PSNR's are shown in Fig. 6. It is found that the PSNR's for DWT level 3 are smaller than others, in particular on Figs. 6 (c) and (d). DWT level 3 can provide only mipmap levels 0 to 3. Mipmap levels required to render the surface for the viewpoints like Fig. 4 (c) and (d) are greater than 3, so in such viewpoints, renderer uses further scaled mipmap level 3. This fact deteriorates the rendering results.

Table 3. Encoding parameters for evaluating the DWT levels.

	Reference	Experiments
The number of tiles	1	1
Lossless or lossy comrpession	lossy	lossy
DWT level	10	3, 5, 7, 9
Compression rate	not specified	1
The number of layers	5	5

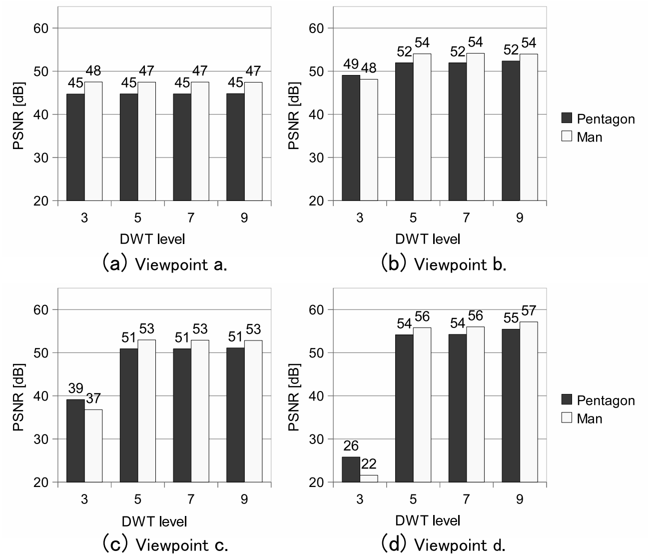


Fig. 6. The quality of rendering results versus DWT level.

Fig. 7 shows PSNR's where DWT level is from 0 to 10, the view angle is the same as that of the viewpoint shown in Fig 4 (d), the distance from the viewpoint to plate is 2240, and other parameters are the same as those shown in Table 3. It is confirmed that PSNR's increase according to the DWT level increasing. But the PSNR of DWT level 10 is smaller than that of DWT level 9. This is because the effect of compression efficiency decreased by using large DWT level is greater than the rendering result quality improvement by increasing mipmap level. Consequently it is confirmed that the appropriate DWT level for mipmapping exists.

4.3. Tile Division

In this section, it is demonstrated that the tile division and the SNR scalability of JP2 serve high quality rendering with a little decoded data in mipmapping.

The view point is shown as Fig. 8. Each texture image is compressed with tile configuration shown as Fig. 9 (a), and other parameters are shown in Table 4. In a generation of mipmaps, the number of decoded SNR layers is varied among

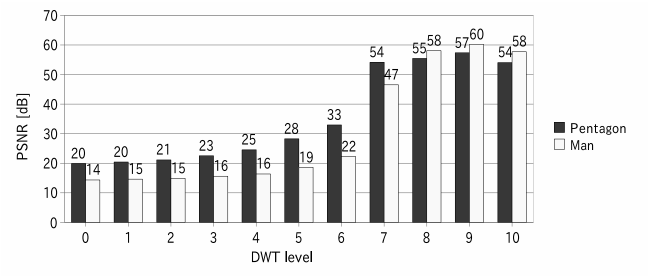


Fig. 7. The quality of rendering results versus DWT levels (0 to 10).

Table 4. Encoding parameters for evaluating the tile division.

	Reference	Experiments
The number of the tiles	1	4
Lossless or lossy compression	lossy	lossy
DWT level	5	5
Compression rate	1	1
The number of layers	5	5

tiles according to Table 5. In Table 5, condition “A” decodes all data, i.e., “A” gives the best quality, whereas “B” gives the worst quality. For condition “C,” the number of decoded layers is determined according to distance from the viewpoint to the plate. This is because farther distance makes some tiles on screen small as shown in Fig. 9 (b), so rendering such tiles with low quality images can reduce rendering cost without perceptual degradation. The PSNR’s are summarized in Fig. 10.

From Table 5 and Fig. 10, it is confirmed that rendering using appropriate quality images according to the distance from the viewpoint is achieved by the tile division and the SNR scalability of JP2, and it reduces required data about 40 %.

5. CONCLUSIONS

This paper has studied an utilization of the scalabilities of JP2 in mipmaping. Experimental results show that scalabilities of JP2 serves the versatile and efficient texture rendering.

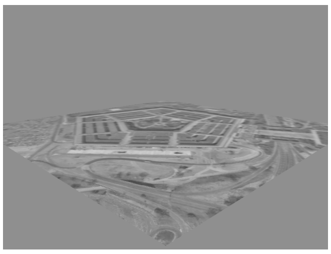
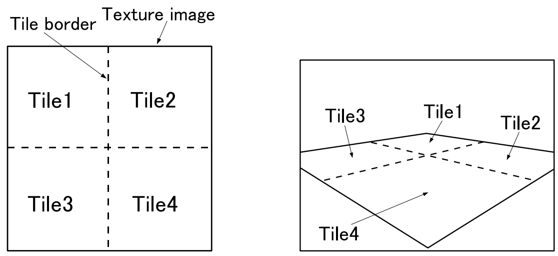


Fig. 8. depression 53°, azimuth 45°, and distance 140.



(a) Composition of tile division. (b) Tile positions on redering result.

Fig. 9. Example of tile division.

Table 5. The number of decoded layers.

	A	B	C
Tile1	5	1	1
Tile2	5	1	3
Tile3	5	1	5
Tile4	5	1	5
data amount ratio to A	100%	20%	60%

REFERENCES

[1] L. Williams, “Pyramidal parametrics,” in *Proc. ACM SIGGRAPH*, 1983, pp.1–11.

[2] “Google Earth,” http://earth.google.com/

[3] “NASA World Wind,” http://worldwind.arc.nasa.gov/

[4] D.S. Taubman and M.W. Marcellin, “JPEG2000 image compression fundamentals, standards and practice,” Springer Science+Business Media, 2002.

[5] P. Schelkens, A. Skodras, and T. Ebrahimi, “The JPEG 2000 suite,” Wiley, pp.421–439, 2009.

[6] “Kakadu,” http://www.kakadusoftware.com/

[7] “Microsoft XNA,” http://msdn.microsoft.com/xna/

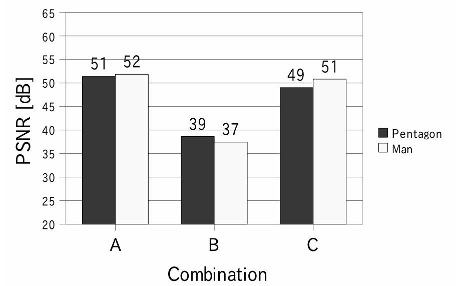


Fig. 10. The advantages of tile division.