28th Picture Coding Symposium, PCS2010, December 8-10, 2010, Nagoya, Japan

IMPROVED TEXTURE COMPRESSION FOR S3TC

Jiang Yifei [1,2], *Huan Dandan*[1]

[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2]Graduate University of Chinese Academy of Sciences, Beijing 100049, China
jiangyifei01@gmail.com, hdd@ict.ac.cn

ABSTRACT

Texture compression is a specialized form of still image compression employed in computer graphics systems to reduce memory bandwidth consumption. Modern texture compression schemes cannot generate satisfactory qualities for both alpha channel and color channel of texture images. We propose a novel texture compression scheme, named ImTC, based on the insight into the essential difference between transparency and color. ImTC defines new data formats and compresses the two channels flexibly. While keeping the same compression ratio as the de facto standard texture compression scheme, ImTC improves compression qualities of both channels. The average PSNR score of alpha channel is improved by about 0.2 dB, and that of color channel can be increased by 6.50 dB over a set of test images, which makes ImTC a better substitute for the standard scheme.

Index Terms— Computer graphics, texture compression, clustering algorithms

1. INTRODUCTION

Texture image is a special kind of still image widely adopted in 3D computer graphics systems, of which memory bandwidth is often the bottleneck. Reading from texture memory is often the main consumer of bandwidth, thus texture compression is widely used in graphics systems. Texture elements are usually expressed using RGBA tuples, among which RGB stand for color and A means transparency α. Therefore texture compression schemes generally handle both color channel and alpha channel. However, modern texture compression systems, including S3TC (S3 Texture Compression) [1] and ETC (Ericsson Texture Compression) [2], cannot generate satisfactory qualities for both the two channels.

We improve both the two channels' compression qualities while keeping the same bit rate as S3TC. Improved Texture Compression (ImTC) proposed in this paper exploits the essential difference between transparency and color, and defines new compression data formats for alpha channel. Meanwhile the unoccupied bits of alpha channel are flexibly utilized by color channel to improve the visual quality. ImTC introduces clustering algorithms into alpha compression besides the linear interpolation method used in S3TC. Color channel is compressed with three possible data formats, determined by the size of alpha block. Results show that ImTC outperforms S3TC in both the two channels' compression qualities.

Our major contribution is to propose new data formats for texture compression. Different implementations can be devised while sticking with the data formats. We modestly conclude that ImTC is a better alternative to DXT5 (a variation of S3TC), or at least the alpha compression of ImTC can substitute for that of DXT5.

Supported by the National Basic Research 973 Program of China (No. 2005CB321600), National High Technology Research and Development 863 Program of China (No.2008AA010901), National Sci&Tech Major Project of China (No.2009ZX01028-002-003, 2009ZX01029-001-003), and National Natural Science Foundation of China (No.60736012, 60921002, 61070025).

2. PREVIOUS WORK

S3TC and ETC are the most important texture compression systems. S3TC is a group of related texture compression algorithms originally developed by S3 Graphics, Ltd. Its subsequent inclusion in Microsoft's DirextX led to widespread adoption of the technology and made it the de facto standard. ETC, derived from iPACKMAN [3], is a relatively newer compression system used in high-end mobile phones, and has been standardized in OpenGL ES.

S3TC consists of 5 different variations, named DXT1 through DXT5, which differ in their handling of the alpha channel. Among the five variations, DXT2 and DXT4 are rarely used. DXT1 only handles RGB channel. A 4×4 pixel block is compressed to 64 bits of output, consisting of two RGB565 colors c_0, c_1 and 16 2-bit indices. If the first color c_0 as a 16-bit unsigned integer is greater than the second color c_1, two other colors are calculated, $c_2 = (2c_0 + c_1)/3$ and $c_3 = (c_0 + 2c_1)/3$. Otherwise if $c_0 \leq c_1$, $c_2 = (c_0 + c_1)/2$ and c_3 is transparent black. The indices are then consulted to determine the color value for each pixel.

DXT3 and DXT5 convert 16 input pixels into 128 bits, consisting of 64 bits of alpha data and 64 bits of color data. The color data are encoded the same way as DXT1, except that the 4 color version of DXT1 algorithm is always used.

DXT3 contains a 4-bit alpha value per pixel. DXT5 stores two 8-bit alpha values α_0, α_1 and 16 3-bit indices. If $\alpha_0 >$

978-1-4244-7135-5/10/$26.00 ©2010 IEEE

α_1, six other alpha values are calculated, such that $\alpha_2 = (6\alpha_0 + \alpha_1)/7$, $\alpha_3 = (5\alpha_0 + 2\alpha_1)/7$, $\alpha_4 = (4\alpha_0 + 3\alpha_1)/7$, $\alpha_5 = (3\alpha_0 + 4\alpha_1)/7$, $\alpha_6 = (2\alpha_0 + 5\alpha_1)/7$, and $\alpha_7 = (\alpha_0 + 6\alpha_1)/7$. Otherwise if $\alpha_0 \leq \alpha_1$, four other alpha values are calculated such that $\alpha_2 = (4\alpha_0 + \alpha_1)/5$, $\alpha_3 = (3\alpha_0 + 2\alpha_1)/5$, $\alpha_4 = (2\alpha_0 + 3\alpha_1)/5$, and $\alpha_5 = (\alpha_0 + 4\alpha_1)/5$ with $\alpha_6 = 0$ and $\alpha_7 = 255$. The indices are used to determine the alpha value for each pixel. Because DXT5 uses an interpolated alpha scheme, it generally produces superior results for alpha gradients than DXT3 and is considered the most flexible general purpose compression codec.

One interesting aspect of S3TC is that all the methods but DXT1 use four bits per pixel for alpha, and four bits per pixel for color. This means that the information in the alpha channel is only compressed down to 50% of its original size, with very little loss of quality. In contrast, the color channel is compressed down to 1/6 of its original size, resulting in a noticeable reduction in quality.

Based on this observation, ETC/TAC increases the compression ratio of alpha channel. ETC/TAC uses two bits or one bit per pixel for alpha, and keeps the same bit rate for color as S3TC. While the color channel's compression quality is improved a little, the compression quality of alpha channel is greatly reduced [3, 4].

3. IMTC DATA FORMATS

3.1. Motivation of ImTC

The essential difference between transparency and color is the motivation of ImTC. Both transparency and color are related to lighting but different in essence.

Transparent materials allow much of the incident light to be transmitted, with little being reflected. Absence of structural defects and molecular structure are mostly responsible for excellent optical transmission. Opaque materials have a chemical composition which includes absorption centers, and do not allow the transmission of light. They absorb certain portions of the visible spectrum, while reflecting others. The frequencies of the spectrum which are not absorbed are either reflected back or transmitted, which gives rise to color. Materials which only allow light to pass through diffusely are called translucency.

The International Union of Pure and Applied Chemistry (IUPAC) defines the ratio of the absorbed to the incident power as absorptance (α), which is identical with the transparency defined in graphics. Generally the alpha value of transparent fragment is 0.0, and the alpha value of opaque fragment is 1.0.

Transparency and color are in essence two different physical properties of materials. Transparency is mainly dependent on the materials, while colors are also affected by the ambient light. The diversity of alpha values is generally not as rich as that of color values, which conforms to our daily experiences.

3.2. Data formats of alpha channel

ImTC converts a 4×4 pixel block to 128 bits, which is the same as DXT5. Based on the perception of the essential difference, ImTC defines new alpha data formats as shown in Figure 1. There are 3 possible formats of the compressed 64-bit alpha block, distinguished by the relative values of the first two 8-bit data.

α_0	α_0	*00*	*00*	$(00)_2 \times 16$
α_0	α_1	$[\alpha_2]$	$[\alpha_3]$	2×16 *indices*
α_{max}	α_{min}	3×16 *indices*		

Fig. 1. ImTC alpha data formats. Top to bottom: format 1, format 2 and format 3.

The first two bytes are interpreted as two unsigned integers. If the two integers are equal (format 1), all the 16 pixels have the same alpha values. The rest 48 bits are zeros.

If the first integer is smaller than the second one (format 2), the following two bytes are another two base alpha values α_2 and α_3. The last 32 bits are 16 2-bit indices. One 2-bit index can decide a value from α_0, α_1, α_2, and α_3. α_2 and α_3 are not requisite. If they are present, the four base values make an increasing sequence, such that $\alpha_0 < \alpha_1 < \alpha_2 < \alpha_3$. Otherwise, zeros take the place of α_2 and (or) α_3.

If the first integer is greater than the second one (format 3), 8 base alpha values are calculated, such that $\alpha_0 = \alpha_{\min}$, $\alpha_7 = \alpha_{\max}$, $\alpha_1 = (6\alpha_0 + \alpha_7)/7$, $\alpha_2 = (5\alpha_0 + 2\alpha_7)/7$, $\alpha_3 = (4\alpha_0 + 3\alpha_7)/7$, $\alpha_4 = (3\alpha_0 + 4\alpha_7)/7$, $\alpha_5 = (2\alpha_0 + 5\alpha_7)/7$, $\alpha_6 = (\alpha_0 + 6\alpha_7)/7$. The rest 48 bits are 16 3-bit indices, used to decide each pixel's alpha value.

3.3. Data formats of color channel

When the 16 pixels have the same alpha values, the last 48 bits are left unused in alpha data format 1. The color channel takes advantage of the free bits of alpha channel to improve visual quality. Figure 2 shows the color data formats.

Format 1 occupies extra 48 bits of the alpha channel. It stores two RGB888 colors and 16 4-bit indices. Other 14 colors are calculated through linear interpolation. The indices are then consulted to determine the color value for each pixel.

Format 2 borrows extra 32 bits from the alpha channel, and stores two RGB888 colors and 16 3-bit indices. Other 6 colors are calculated through linear interpolation. The indices are consulted to determine the color value for each pixel.

Format 3 stores two RGB565 colors and 16 2-bit indices, which is the same as in S3TC. Two other colors are calculated, such that $c_2 = (2c_0 + c_1)/3$ and $c_3 = (c_0 + 2c_1)/3$. The indices are then consulted to determine the color value for each pixel.

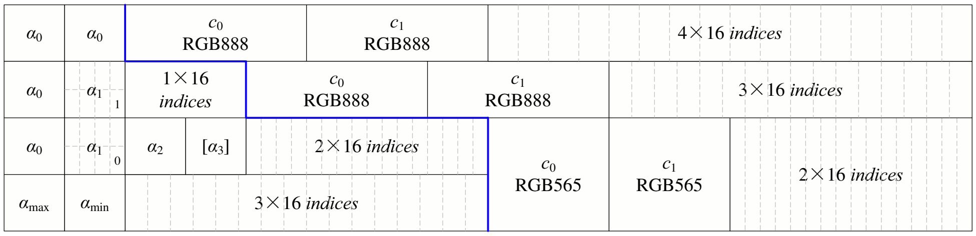

Fig. 2. ImTC color data formats, together with the adapted alpha formats. Top to bottom: format 1, format 2 and format 3.

4. EXPERIMENTS AND RESULTS

4.1. ImTC implementation

The alpha compression of ImTC employs two simple clustering control strategies besides linear interpolation method. If two alpha values are equal or their difference is below a certain threshold, such that $|\alpha_i - \alpha_k| = 0$ or $|\alpha_i - \alpha_k| \leq \delta$, they belong to the same cluster. If the number of clusters are not greater than 4, alpha channel is compressed according to its data format 1 or format 2. Otherwise alpha channel is compressed according to its data format 3. If the alpha channel can share its free bits, the color channel is compressed according to color data format 1 or format 2. Otherwise the color channel is compressed according to its data format 3.

DXT5 is also implemented for comparison. Both the alpha channel and color channel are handled through linear interpolation, as described in Section 2.

4.2. Alpha channel compression quality

Since transparency cannot be observed directly, and there are no such alpha data that are widely accepted as benchmarks, we choose the Kodak image suite. The images are transformed to YCoCg-R color space, and the Y channel data are used for test. Final decompressed luma data are stored as grayscale images to facilitate observation. *Peak Signal to Noise Ratio* (PSNR) is used to measure quality, defined as

$$PSNR = 10\log_{10}\Big(\frac{255^2}{MSE}\Big), \tag{1}$$

where MSE is the *Mean Square Error*, defined as

$$MSE = \frac{1}{w \times h}\sum_{x,y}\Delta Y_{xy}^2.$$

Here w and h are the width and the height of the image, and ΔY_{xy} is the difference of luminance.

Figure 3 shows the monochromatic images. Images for ImTC and DXT5 are provided as well as the original ones. As can be seen, the luminance changes more sharply in column 3 than in column 2, and the visual quality of ImTC is better than that of DXT5.

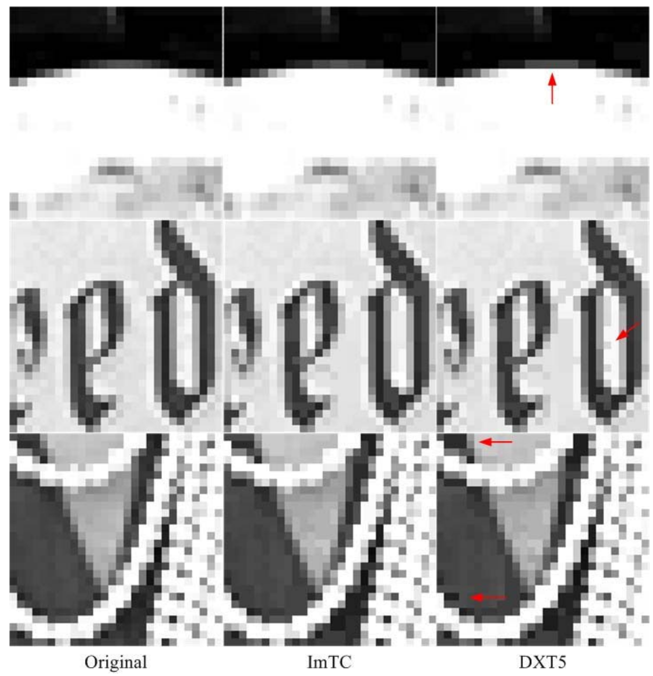

Fig. 3. Top to bottom: enlargements of kodim02, kodim08 and kodim18 (partial). Arrows point to distinguishable parts.

ImTC improves all the 24 images' PSNR scores. The average PSNR is increased by about 0.2 dB. The most significant increase is 0.667 dB from kodim08. The most subtle one is 0.007 dB from kodim02. Table 1 lists the detailed scores.

4.3. Color channel compression quality

The Kodak image suite is used for test. PSNR is defined the same way as in Equation 1, except that the three components of RGB must be calculated instead of the luminance. ImTC can increase the PSNR scores significantly. Figure 4 shows the detailed PSNR scores of 24 images. The average PSNR score of DXT5/ImTC format 3 is 33.84 dB, and the average scores of ImTC format 1 and format 2 are 40.34 dB and 38.73 dB, with improvements of 6.50 dB and 4.89 dB respectively.

Figure 5 compares the visual qualities of ImTC and DXT5. The enlargements are from kodim05, kodim08 and

DXT5	ImTC	Δ	DXT5	ImTC	Δ
40.144	40.338	0.193	37.851	38.049	0.198
46.893	46.900	0.007	42.045	42.156	0.112
47.472	47.510	0.038	45.404	45.686	0.283
46.110	46.220	0.110	45.517	45.563	0.046
39.190	39.319	0.129	44.529	44.790	0.261
41.936	42.015	0.079	41.203	41.519	0.315
44.591	44.728	0.137	42.035	42.419	0.383
37.791	38.458	0.667	44.260	44.481	0.221
45.031	45.180	0.148	41.855	41.920	0.065
45.202	45.481	0.279	44.054	44.211	0.157
42.878	42.956	0.078	46.773	46.806	0.033
46.215	46.409	0.195	40.544	41.073	0.529
Average increase = 0.194 dB					

Table 1. PSNR scores (alpha channel). Left: kodim01 to kodim12. Right: kodim13 to kodim24. Unit is dB.

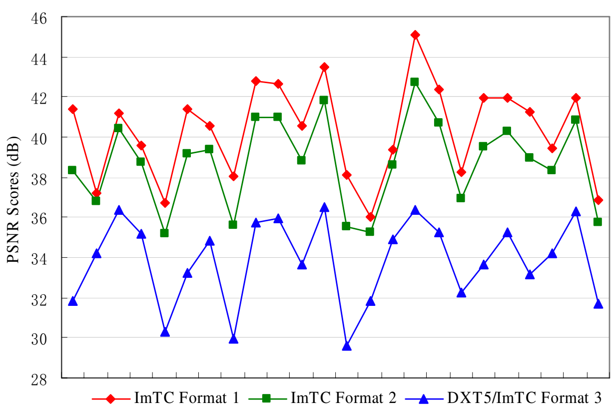

Fig. 4. PSNR scores of 24 images (color channel).

kodim09. It shows that DXT5/ImTC format 3 generates unpleasant visual artifacts, and ImTC format 2 alleviates these artifacts. The differences between images compressed with ImTC format 1 and the original ones are too trivial to perceive at their original size.

5. CONCLUSION

We have presented a new texture compression scheme called ImTC based on the insight into the essential difference between transparency and color. The flexible data formats of ImTC make it superior than DXT5 in both alpha channel and color channel compression qualities. Thus we conclude that ImTC is a better alternative to DXT5.

The luma data are used for test in this paper due to the absence of alpha channel in common images. As the diversity of transparency is generally not as rich as that of natural images' luminance, the improvements of alpha channel are supposed to be more significant when ImTC is applied to real textures. It must be pointed out that the improvements of color channel

Fig. 5. Left to right: enlargements of kodim05, kodim08 and kodim09 (partial).

are predominated by the the size of compressed alpha block, and the final improvements will be between that achieved by DXT5/ImTC format 3 and ImTC format 1.

6. REFERENCES

[1] Konstantine I. Iourcha, Krishna S. Nayak, and Zhou Hong, "System and method for fixed-rate block-based image compression with inferred pixel values," *US Patent 5,956,431*, September 1999.

[2] Jacob Ström and Martin Pettersson, "ETC2: texture compression using invalid combinatios," in *SIGGRAPH/EUROGRAPHICS Conference On Graphics Hardware*. 2007, pp. 49–54, Eurographics Association.

[3] Jacob Ström and Tomas Akenine-Möller, "iPACKMAN: high-quality, low-complexity texture compression for mobile phones," in *SIGGRAPH/EUROGRAPHICS Conference On Graphics Hardware*. 2005, pp. 63–70, ACM.

[4] Per Wennersten and Jacob Ström, "Table-based alpha compression," *Eurographics 2009 Computer Graphics Forum*, vol. 28, no. 2, pp. 687–695, 2009.