Proceedings of IEEE International Conference on Image Processing (ICIP), Oct., 2014

OPTIMIZED QUANTIZATION OF WAVELET SUBBANDS FOR HIGH QUALITY REAL-TIME TEXTURE COMPRESSION

Bob Andries, Jan Lemeire, Adrian Munteanu

Department of Electronics and Informatics, Vrije Universiteit Brussel - iMinds
Pleinlaan 2, 1050 Brussels, Belgium

ABSTRACT

This paper proposes a new wavelet-based system for fixed-rate texture compression in 3D graphics applications. An analysis of the optimized quantization of the wavelet subbands is carried out, focusing on Lloyd-Max quantization of subband blocks as well as on quantization techniques employed in existing texture compression formats on GPUs. The results demonstrate that the proposed compression technique yields higher performance compared to existing GPU texture compression schemes and previously proposed transformed-based systems. Moreover, compared to conventional schemes, the wide range of quantization schemes results in a much wider range of available bitrates. Additionally, the proposed scheme offers real-time execution, being suitable for real time rendering applications.

***Index Terms*—** GPU, compression, real-time, quantization

1. INTRODUCTION

One of the main data sources in real time 3D engines is 2D texture data. As monitor resolutions increase, higher resolution textures are required, introducing memory and bandwidth bottlenecks. Texture compression is a useful tool to lower both the memory and bandwidth usage during texture sampling, an essential stage of real time rendering. These compression systems necessarily employ a fixed number of bits per pixel (fixed rate compression), as this guarantees constant time for random read access on the compressed texture, which is required for texture sampling. A 2D texture representation requires a hierarchical set of downsampled versions in order to perform trilinear filtering. Block truncation techniques [1] have been used for more than ten years to independently compress each resolution level. However, such texture representation formats are not efficient, due to oversampling resulting from constructing the multiresolution pyramids and for not exploiting the dependencies between the different spatial layers. Block truncation techniques still form the core of the currently used compression formats, commonly known as DXT texture compression [2]. Low power platforms often feature PVRTC [3], a technique which blends two different low pass versions to reconstruct the original data. Similar to DXT, PVRTC requires separate compression of the resolution levels. Transform-based techniques on the other hand do construct critically sampled multiresolution representations, hence no additional storage cost or memory are needed compared to the original spatial-domain texture representation.

Previously, transform based texture compression systems have been proposed, usually requiring dedicated hardware implementations [4, 5]. Other solutions were limited to suboptimal transforms for constructing multiresolution texture representations, such as Haar [6] and DCT [7]. In contrast with these techniques, we propose a texture compression system based on the 2D discrete wavelet transform [8]. This transform is particularly suitable to produce multiple lower resolution versions of the input textures and for decorrelating natural image data [9]. By optimized quantization of the subband data, we lower the total bitrate of the texture data, resulting in a superior fixed rate codec.

The remainder of the paper is structured as follows. In section 2 we will provide an overview of the proposed texture compression system. Section 3 focuses on the optimized quantization of the wavelet subbands. In section 4 we report the experimental results and in section 5 we draw the conclusions of our work.

2. COMPRESSION SYSTEM ARCHITECTURE

The overall architecture of the proposed compression system is shown in figure 1. In order to achieve high compression ratios, we must exploit the available spatial and color correlation in the input images as much as possible. As a first step, the color channels are decorrelated using the YCoCg-R transform [10], a simple and reversible integer color transform. Each of the resulting three channels is then spatially decorrelated by the 2D discrete wavelet transform. Rate allocation is subsequently performed, based on which the quantizer configuration yielding the desired quality is selected. Next, the subband data is quantized and packed into the appropriate texture formats, suitable for real time texture sampling. The rate allocation and quantization are detailed in section 3.

This work was supported by the Flemish Institute for the Promotion of Innovation by Science and Technology (IWT) - PhD grant of Bob Andries.

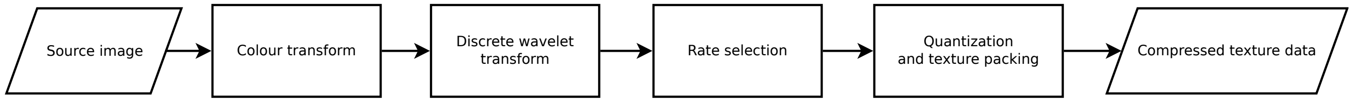


Fig. 1. Compression system flowchart.

3. OPTIMIZED SUBBAND QUANTIZATION

As entropy coding techniques cannot be used when designing a fixed rate GPU codec, the quantizers should be fully optimized for minimal distortion under constraints on the number of bins. Specifically, given a certain number of bins N, an optimal quantizer is constructed for each wavelet subband or block of coefficients within any given subband. When optimizing for minimal distortion, it can be shown that the reconstruction levels y_i for such a quantizer must satisfy [11]:

$$\int_{x_i}^{y_i} |x - y_i|^{j-1} p_X(x)\,\mathrm{d}x = \int_{y_i}^{x_{i+1}} |x - y_i|^{j-1} p_X(x)\,\mathrm{d}x \quad (1)$$

where x_i denote the partition boundaries and $p_X(x)$ denotes the probability density function of X. In general, quantizers minimizing the error $|x - y_i|^j$ for a given number of bins N and positive integer j are also known as Lloyd-Max quantizers [11]. Setting the partial derivatives in equation 1 with respect to x_k to zero with $j = 2$ yields the following condition for the partition boundaries [11]:

$$x_k = \frac{y_{k-1} + y_k}{2}, k = 1, 2, ..., N - 1 \quad (2)$$

Differentiating with respect to y_k results into the following condition for the reconstruction levels:

$$y_k = \frac{\int_{x_k}^{x_{k+1}} x p_X(x)\mathrm{d}x}{\int_{x_k}^{x_{k+1}} p_X(x)\mathrm{d}x}, k = 0, 1, ..., N - 1 \quad (3)$$

Equations 2 and 3 are necessary conditions for these quantizers to be optimal quantizers. In the case in which $p_X(x)$ is a logarithmic concave probability distribution, these are also sufficient conditions.

For any distribution other than the uniform distribution, quantizers satisfying equations 2, 3 will be non linear quantizers, which are less than ideal for a real time GPU application. As GPU shaders are executed per-pixel, inverse Lloyd-Max quantization involves loading the correct reconstruction value from a table, corresponding to the quantized pixel value. This implies indirection in the pixel shader, which is detrimental for shader performance. Additionally, due to the non linearity of Lloyd-Max quantization, hardware accelerated texel filtering becomes impossible, and at high bitrates, the reconstruction tables will start to occupy a relevant amount of memory. A practical alternative is to adopt existing high quality texture formats, mimicking Lloyd-Max quantization. Based on these formats, we will apply a rate allocation algorithm and compare those results to theoretical cases where Lloyd-Max quantization is applied.

The analysis of conventional texture compression formats reveals that most techniques are targeting multi-channel texture compression, exploiting the correlation between the channels of the raw texture data. As our system decorrelates the RGB channels of the input data, these formats are of no use to us. The existing single channel texture formats [12] offer a wide range of high rate uncompressed formats, of which the format with the lowest rate is an 8 bpp texture format; this is used in our approach as the format providing the highest available bitrate for a given subband. Additionally, we also employ LATC [13], which is a 4 bits per pixel compressed texture format. Hence, the rates available to our rate allocation algorithm are respectively 0, 4 and 8 bits per subband pixel.

LATC (used as 4 bpp texture format) is a single channel texture compression algorithm which works on 4x4 blocks. Per block it stores an eight bit minimum value, an eight bit maximum value and 16 three bit indices, each referring to one of the eight linearly interpolated values defined by the minimum-maximum pair. As texture compression is a one time offline process, a large search space can be used to find the best possible minimum-maximum pair for each 4x4 block.

Eight bit quantization is performed by using a regular eight bit texture format to store a uniformly quantized wavelet subband. We point out that in contrast to conventional uniform quantization operating on the entire dynamic range of the coefficients, in our approach the original subband data is optimally clipped to avoid wasting precision on outliers. Specifically, given a signal with probability distribution $p_X(x)$, one determines the optimal clipping bounds which minimize the quantization error:

$$\begin{aligned} \underset{c_{min}, c_{max}}{\text{argmin}} & \int_{-\infty}^{c_{min}} p_X(x)(x - (c_{min} + \frac{c_{max} - c_{min}}{2N}))^2 \mathrm{d}x \\ & + \int_{c_{min}}^{c_{max}} p_X(x)(\frac{c_{max} - c_{min}}{N\sqrt{12}})^2 \mathrm{d}x \\ & + \int_{c_{max}}^{+\infty} p_X(x)(x - (c_{max} - \frac{c_{max} - c_{min}}{2N}))^2 \mathrm{d}x \end{aligned} \quad (4)$$

where c_{min} and c_{max} are the lower and upper clipping bounds and N is the number of bins. We assume the quantization error in the non clipped area is uniformly distributed, as is the case when the quantization rate is sufficiently high [14].

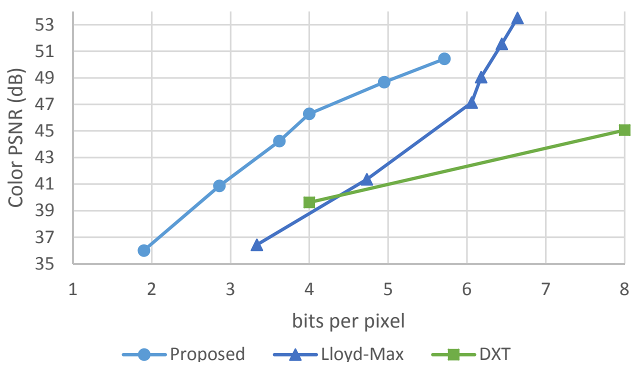


Fig. 2. Lena color PSNR results

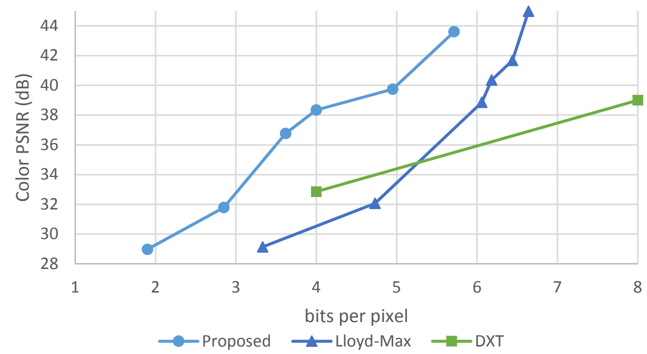


Fig. 3. Baboon color PSNR results

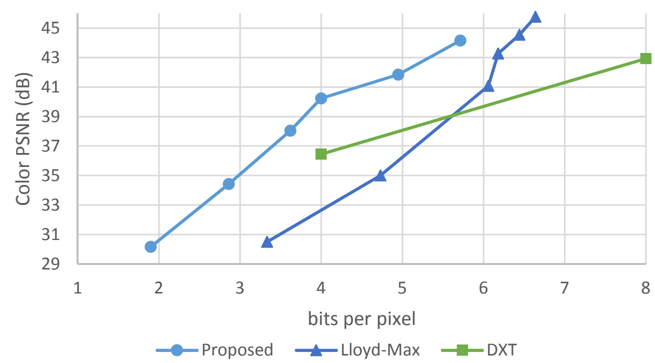


Fig. 4. Pzero color PSNR results

4. RESULTS

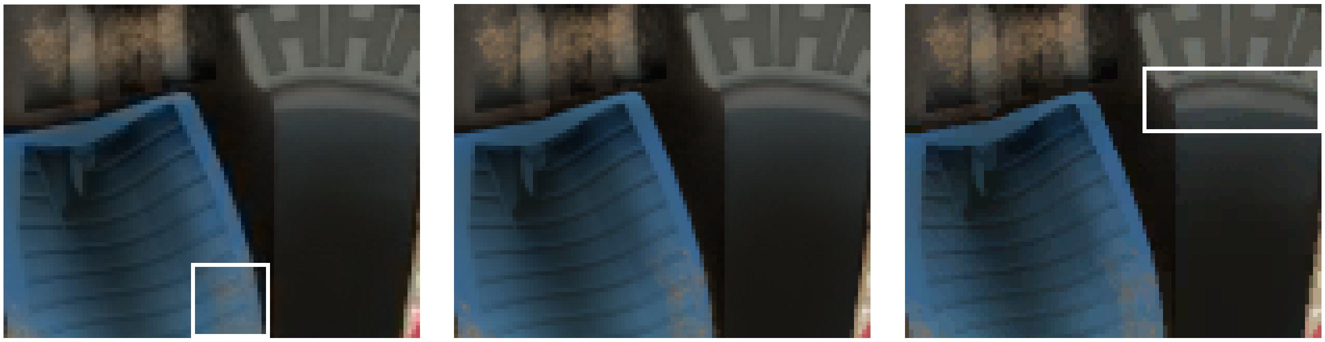
In our experiments, we have used a 2-level DWT implemented using the biorthogonal (2,2) set of filters, where the (2,2) indicates the number of vanishing moments for the analysis and synthesis wavelets respectively. These filters have a higher number of vanishing moments when compared to the conventional Haar wavelets used in previous wavelet-based GPU texture codecs [6]. This leads to sparser representations, improves the compression efficiency and yields better visual results compared to the Haar wavelets. However, increasing the number of vanishing moments increases also complexity. Experimentally, we found that the biorthogonal (2,2) pair yields a good performance-complexity trade-off.

To generate a set of quantization modes, rate allocation using the three available rates per subband was applied on a set of three images, representing both low frequency textures (Lena) and high frequency textures (Baboon and Pzero). The Pzero texture is also used in the reference DCT-based codec of [7].

For each of the generated bitrates, a comparison between the proposed system and a theoretical Lloyd-Max based system is shown in table 1. The alternative for 4 bpp LATC subband quantization is a three bit Lloyd-Max quantizer which is applied to 4x4 blocks, similar to LATC quantization which also performs three bit quantization internally. Eight bit clamped uniform quantization is replaced by an eight bit Lloyd-Max quantizer optimized for the full subband in question. The overhead needed to store the Lloyd-Max reconstruction values is not included. As can be noticed from the table, especially at higher bitrates, the difference between the proposed quantization and Lloyd-Max quantization is very clear. At lower rates, most subbands are either discarded or quantized at 4 bpp (LATC). Although performing sub-optimal quantization, LATC manages to capture nearly the same level of detail as an equivalent optimal Lloyd-Max quantizer optimized for each 4x4 block of coefficients, resulting in a much smaller difference in quality between both.

In the subsequent set of experiments, we also account for the overhead associated with optimal Lloyd-Max quantization. Each image was compressed using the proposed solution to quantize the wavelet subbands and compared against Lloyd-Max alternative using the corresponding optimal quantizers. Additionally, we compare our results with DXT1 (4 bpp) and a modified version of DXT5 (8 bpp), which applies color decorrelation and is suitable for high quality texture compression [15]. The quality metric used was weighted PSNR, applied on the decorrelated color channels with weight $\frac{4}{6}$ for the luminance channel and $\frac{1}{6}$ for both color channels. As can be seen from table 1, the proposed quantization modes are sub-optimal compared to their theoretical best alternative. However, when accounting for the overhead caused by Lloyd-Max quantization (especially on the 4x4 blocks), the proposed solution easily outperforms the Lloyd-Max quantizers because of the much lower overhead of LATC texture compression. This overhead consists of just 2 eight bit values per 4x4 block (representing the lower and upper bound of the values in the block), compared to 8 eight bit reconstruction values for its Lloyd-Max alternatives.

A visual comparison is shown in figure 5, where we show a cutout of the Pzero texture, compressed with our proposed codec at 4 bpp and with DXT1, also 4 bpp. Note the banding artifacts on the smooth regions in DXT1, which are not visible in the proposed codec. Conversely, high frequency color



(a) Proposed, 4 bpp (b) Original, 24 bpp (c) DXT1, 4 bpp

Fig. 5. A comparison of the four bits per pixel version of our proposed codec and DXT1. One notes the blocking and banding artifacts near edges in the DXT1 image, but also some loss of high pass color information in our proposed version.

Table 1. Average color PSNR over the three images.

bpp	1.9	2.86	3.62	4	4.95	5.71
Proposed	31.7	35.7	39.7	41.6	43.4	46.1
Lloyd-Max	32.0	36.1	42.4	44.2	45.9	48.0

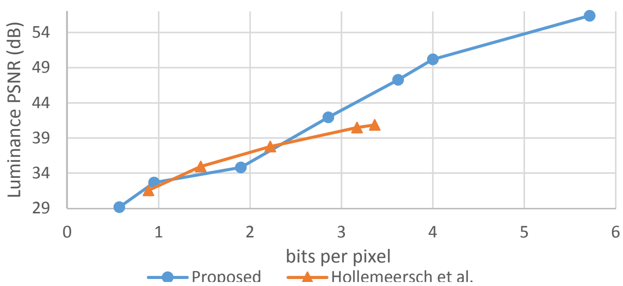


Fig. 6. Lena luminance PSNR comparison

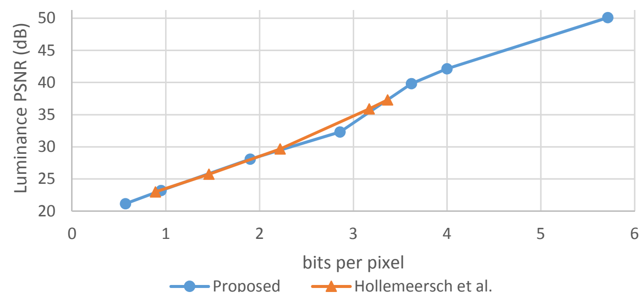


Fig. 7. Baboon luminance PSNR comparison

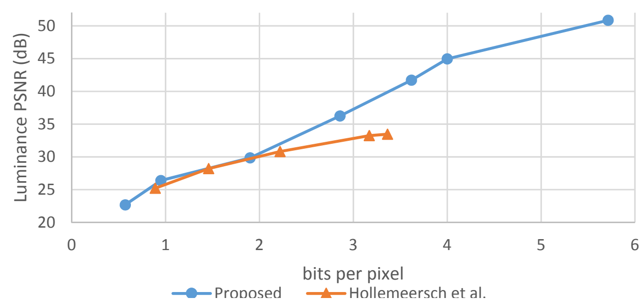


Fig. 8. Pzero luminance PSNR comparison

details might be blurred with the proposed codec because of the downsampling on the two color channels.

We also compare the proposed codec with the current state of the art, which is the codec proposed in [7]. As the technique of [7] uses luminance PSNR as a metric, we generated separate charts for this comparison. Figures 6, 7 and 8 show how the proposed codec compares to the work of [7]. Note that our proposed codec spends double the rate on color data compared to the codec of [7] as our rate allocation was performed using the color PSNR metric. Still, our solution easily outperforms the DCT-based system, especially at higher rates, whereas at lower rates the performance is similar.

In terms of execution time, the proposed system achieves real-time decoding on a high end GPU of HD resolution textures at more than 1200 frames per second. A Lloyd-max based system would be less performing due to the additional complexity of the non linear inverse quantization. A thorough analysis of the complexity is a topic of further investigation.

5. CONCLUSION

This work shows that texture compression can be significantly improved by making use of color decorrelation and the discrete wavelet transform, combined with existing texture compression formats. We successfully applied rate allocation using a limited set of subband bitrates, aiming for optimized quantization while keeping the overhead considerably below what would be required when applying optimal Lloyd-Max quantization. Paired with these texture formats, a compression system was developed featuring high compression performance and applicability on existing consumer GPUs.

6. REFERENCES

[1] E. Delp and O. Mitchell, "Image compression using block truncation coding," *IEEE Transactions on Communications*, vol. 27, no. 9, pp. 1335–1342, 1979.

[2] K. I. Iourcha, K. S. Nayak, and Z. Hong, "Fixed-rate block-based image compression with inferred pixel values," Dec. 2 2003, US Patent 6,658,146.

[3] S. Fenney, "Texture compression using low-frequency signal modulation," in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*. Eurographics Association, 2003, pp. 84–91.

[4] A. V. Pereberin et al., "Hierarchical approach for texture compression," in *Proceedings of GraphiCon 99*, 1999, pp. 195–199.

[5] C.-H. Sun, Y.-M. Tsao, and S.-Y. Chien, "High-quality mipmapping texture compression with alpha maps for graphics processing units," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 589–599, 2009.

[6] S. Diverdi, N. Candussi, and T. Höllerer, "Real-time rendering with wavelet-compressed multi-dimensional textures on the GPU," 2005.

[7] C.-F. Hollemeersch, B. Pieters, P. Lambert, and R. Van de Walle, "A new approach to combine texture compression and filtering," *The Visual Computer*, vol. 28, no. 4, pp. 371–385, 2012.

[8] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[9] A. van der Schaaf and J.H. van Hateren, "Modelling the power spectra of natural images: Statistics and information," *Vision research*, vol. 36, no. 17, pp. 2759–2770, 1996.

[10] H. Malvar and G. Sullivan, "YCoCg-R: A color space with RGB reversibility and low dynamic range," *ISO/IEC JTC1/SC29/WG11 and ITU*, 2003.

[11] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[12] The Khronos Group, "OpenGL texture formats," `http://www.opengl.org/wiki/Image_Format`, 2014.

[13] M. J. Kilgard, P. Brown, Y. Zhang, and A. Barsi, "LATC OpenGL extension," `http://developer.download.nvidia.com/opengl/specs/GL_EXT_texture_compression_latc.txt`, 2008.

[14] W. R. Bennett, "Spectra of quantized signals," *Bell syst. tech. J*, vol. 27, no. 3, pp. 446–472, 1948.

[15] J. van Waveren and I. Castaño, "Real-time YCoCg-DXT compression," Tech. Rep., id Software, Inc. and NVIDIA Corp., 2007. 2, 3, 2007.