A STUDY ON VQ BASED COMPRESSION OF HIGH DYNAMIC RANGE IMAGES

Nagisa Sugiyama and Masahiro Okuda

Faculty of Environmental Engineering, The University of Kitakyshu

ABSTRACT

An encoding method of the high dynamic range (HDR) images based on Vector Quantization is discussed. The HDR images have much higher dynamic ranges compared with the low dynamic range (LDR) images. Due to its high dynamic range, the size is often huge, thus development for functional compression is needed. On the other hand, we often need to convert them into the LDR images because the existing output devices cannot directly output the dynamic ranges of HDR images. Our goal is to develop a HDRI compression method that minimizes the error of the tone-mapped LDR images. Therefore we applied the adjustment of scale and the evaluation of the error in the log domain. Our method improves a compression performance, compared to methods that directly apply the conventional vector quantization to the HDR images.

Index Terms— High Dynamic Range Images, Vector quantization, log domain, tone-mapping

1. INTRODUCTION

The HDR (High Dynamic Range) images often have huge dynamic ranges compared with Low Dynamic Range (LDR) images (e.g. JPEG and BMP image), where the dynamic range is defined as a ratio of lowest and highest intensities of light in pixels. The existing output devices such as displays and printers have generally lower dynamic range, and hence cannot directly output the dynamic ranges of HDR images. For rendering the HDR images, we often have to convert them into the LDR images. This conversion from HDR to LDR images is called "tone-mapping"[2], [3]. In practice, most of the applications that use HDR images needs the tone-mapping operations. Therefore, the quality of the tone-mapped LDR image is important when compressing the HDR image.

Although many methods on HDRI compression have been proposed[4]-[7], there exist few approaches based on Vector Quantization. In this report, we investigate the efficiency of the Vector Quantization for the purposed of HDRI compression. Vector Quantization that collectively quantize the multiple sample values is the typical method in the data compression. The VQ encodes it by replacing a set of data in a code block with the codebook. The encoder and decoder have a common codebook. We have to be able to use the codebook for wide variety of the HDR image, so the designing codebook is very important for the vector quantization. We introduce two modifications: (1) scaling to the HDR images and (2) the calculation of error in the log domain to obtain the optimal codebook and to realize the VQ for the HDR images.

In the following section, we briefly review the VQ algorithm. In Section 3, we introduce our method for HDR image compression. Some experimental results are shown in Section 4 to verify the validity of our method.

2. VECTOR QUANTIZATION

In this paper we adopt a VQ based coding method. The vector quantization is used in many applications such as image and audio compression. It is a lossy data compression methods based on the principle of classifying data with similar features. The vector quantization method quantizes multiple sampling values to one vector collectively. It can efficiently reduce the redundancy of sampling values in image compression. In this paper, we suggest that the vector quantization method for the HDR images. Our method improves a compression performance, compared to methods that directly apply the conventional vector quantization to the HDR images.

In the encoder of the VQ, the input image is divided into the code blocks (ex. the block of 2×2pixels) and replace them by the most similar codebook. The indices are only sent to the decoder. The decoder has the common codebook. Images are reconstructed by mapping the received index to a set of pixels. As mentioned above, the VQ uses a common codebook in an encoder and a decoder. And it encodes by replacing a set of data in a code block with the codebook, so the training algorithm for acquiring the codebook is very important. Splitting algorithm and LBG algorithm are often used for the training.

The conventional image compression methods with this LBG algorithm mainly consist of four steps. Firstly, we decide initial settings such as a codebook size, a dimension, an initial codebook and thresholds. Secondly, the above mentioned splitting algorithm is used. Thirdly, the errors of mean squares are calculated by training data and codebook. Finally, the regions are divided with the codebook fixed and the errors are discriminated by the threshold. In addition, the codebooks

We are grateful for the support of a Grant-in-Aid for Young Sciences (#14750305) of Japan Society for the Promotion of Science

are divided into some fixed regions. The optimal codebooks are created by repeating these processes. Detailed explanation of this algorithm is given below.

2.1. LBG algorithm

The LBG algorithm has been used for quantizer design. Fig.1 depicts the process of the LBG algorithm. It's an iterative numerical method and may converge to local optimal codebook by repeatedly applying the region dividing and the update of representative points. In the region dividing, the error between the input vector and the quantizer representative vector has to be minimized. In the update step the conventional representative points are replaced by the centroid to each regions. It means the new codebook is made from the average of training order. The training vector can belong to the optimal region by repeating the processes in Fig.1.

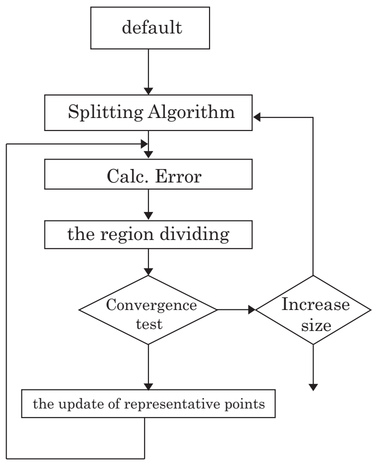


Fig. 1. LBG Algorithm

The error is typically calculated using the squared error. It is given by

$$d(i,j) = \mid T(i,j) - C(i,j) \mid^2 \quad (1)$$

where T(i,j) is the training data and $C(i,j)$ is codebook. On a step of the convergence test, $d(i,j)$ is used to compare with a threshold ϵ.

$$(d^{(m-1)} - d^{(m)})/d^{(m)} < \epsilon \quad (2)$$

When the codebook size correspond to the number of the codebook, then this algorithm exit from the loop.

2.2. Splitting algorithm

The algorithm requires an initial codebook. This initial codebook is obtained by the splitting algorithm. In this method, an initial code vector is set as the average of the all training data. This code vector is then split into two by the minimal vector, that is,

$$\vec{y}_i' = \vec{y}_i - \vec{\delta}, \vec{y}_{i+N}' = \vec{y}_i + \vec{\delta} \quad (3)$$

where $\vec{\delta}$ is a small vector. The iterative algorithm is run with these two vectors as the initial codebook. The final two code vectors are split into four and the process is repeated until the desired number of code vectors is obtained. It can produce codebook of multiple levels (N=2,4,8$\cdots$) by using with the LBG algorithm.

Minimizing the error of the LDR images tone-mapped from the HDR images is our goal for the compression. The above compression process is not suitable for our purpose, since the codebook is optimized for HDR images. Of course the optimized quality of the HDR images are not necessarily be the same as that of the LDR images. In the training step of VQ, the method for calculating error corresponds to optimizing the PSNR of the HDR images. Consequently, we need some modification to optimize the LDR images.

3. PROPOSED METHOD

3.1. Summary

As mentioned above, our goal is minimizing the error of the LDR images tone-mapped by the HDR images. To realize it, our method introduces two modifications to realize the VQ for the HDR images: scalling and nonlinear transform. In general, the unit of pixel values in the HDR images is not standardized. Some HDR images are stored in the unit of luminance (cd/m^2), some may be normalized to different scales. To unify the various scales, the normalization is necessary.

In the second step, we transform the scaled luminance by applying nonlinearity to approximate the tone-mapping operations. Fig.2 depicts the process of our method.

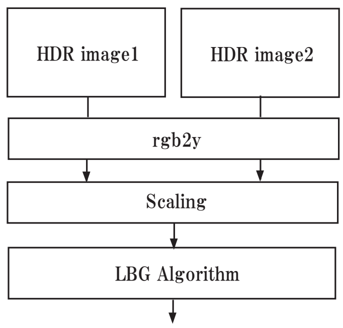


Fig. 2. the build process of codebook

3.2. the transform to the luminance image

We use only luminance image, but it is easily applied to color image encoding. the transform to the luminance image is

$$y(i,j) = 0.213x_R(i,j)+0.715x_G(i,j)+0.072x_B(i,j) \quad (4)$$

where $x_R(i,j)$, $x_G(i,j)$ and $x_B(i,j)$ are the red, green and blue elements in the image x.

3.3. Scaling

The HDR images have huge dynamic ranges and have different scales because the ways for creating the HDR images are various. Some HDR images have pixel values in the unit of cd/m^2, some are in a different unit. we need to normalize the HDR images to a same scale. In order to adjust the scale, we calculate the mean of the HDR image in log domain,

$$m_{logave} = \exp\left(\frac{\sum_{i,j} \log T(i,j)}{N}\right) \quad (5)$$

where $T(i,j)$ is training data and N is the number of pixels. Then, the training data $T(i,j)$ is normalized by the mean such that the mean value are mapped to 0.18 as follows.

$$T' = 0.18 \cdot \frac{T(i,j)}{m_{logave}} \quad (6)$$

We apply this scaling to the input and training set of HDR images.

3.4. Nonlinearity

It is widely agreed that there is a nonlinear relationship between an amount of sensation and intensity of lights, or in other words, brightness human perceives and actual luminance are not linearly related. There are many experimental results for the approximation of the nonlinearity. The well known Weber-Fechner's Law indicates that the relationship between them is modeled by

$$y = k\log(x) + C, \quad (7)$$

where y and x are brightness that human retina perceives and input luminance, respectively. The many tone-mapping operators allow for this kind of nonlinearity.

Thus it is reasonable to calculate the error in log domain when we optimize it by LBG Algorithm. Although the HDR images have huge dynamic ranges, most energy is concentrated in lower luminance regions. Thus, this evaluation in the log domain is suitable for the HDR image, since the log function implicitly weights the lower luminance. We calculate the error between training data

$$d(i,j) = \mid log(T(i,j)) - log(C(i,j)) \mid^2 \quad (8)$$

The codebook created by our method is used at the encoding process. The scaled original image is encoded with designed codebook.

4. EXPERIMENTAL RESULTS

For training data we use some HDR images that are often used for research purpose. We group 2x2 pixels into a vector and the dimension of code book is set to 4. The codebook size is set from 8 to 1024. We have tested dozens of the HDR images collected from some web sites, many of which are frequently used as sample images. We used only luminance image, but it is easily applied to color image encoding. Fig.3 depicts the HDR images that we actually used as training data. (1016 $\times$ 760 pixels) Fig.4 indicates that we used the HDR images as input image.

(a) training data(1) (b) training data(2)

Fig. 3. The training data for codebook

(a) input(1) (b) input(2)

Fig. 4. Input images

We compared a conventional VQ with our method by PSNR of tone mapped LDR images, instead of comparing HDR images directly. Fig.5 shows that the decoding process and the comparison by PSNR.We use Reinhard et al.'s global tone mapping operator [3] to make the LDR images.

Fig.6 shows that the PSNR correspond to each entropy to compare conventional method with our method.

Most of HDR images have higher PSNR than conventional methods in same bit rates. As the diagram indicates, our algorithm performs better than the conventional method.

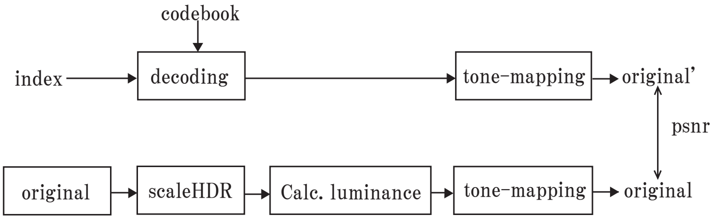


Fig. 5. decoding process and psnr

size	8	16	32	64	128	256	512	1024
entropy	0.1449	0.33848	0.55648	0.7837	1.01255	0.10425	1.44515	1.6999
PSNR	18.14	20.81	24.21	26.78	29.08	30.97	32.21	33.58

conventional method

size	8	16	32	64	128	256	512	1024
entropy	0.4459	0.67578	0.8946	1.07173	1.27873	1.484	1.70025	1.9283
PSNR	23.21	26.43	28.96	30.42	31.28	32.46	33.57	34.67

Our proposal method

(a)input(1)

size	8	16	32	64	128	256	512	1024
entropy	0.4175	0.585	0.76	0.9475	1.1575	1.385	1.63	1.8625
PSNR	17.8	20.2	22.7	24.6	26.8	29	30.7	32.2

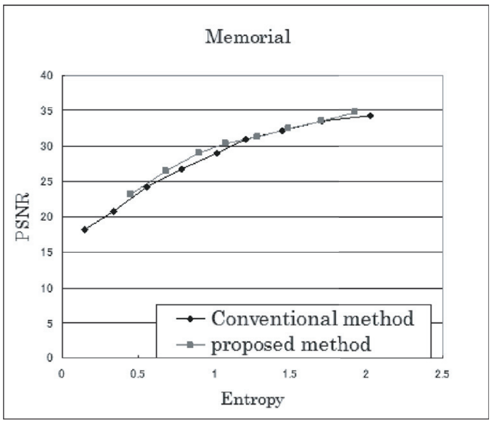
conventional method

size	8	16	32	64	128	256	512	1024
entropy	0.585	0.777	0.98	1.1925	1.415	1.62	1.817	2.00
PSNR	21.6	24.1	26.2	28.4	29.7	31.2	32.28	33.26

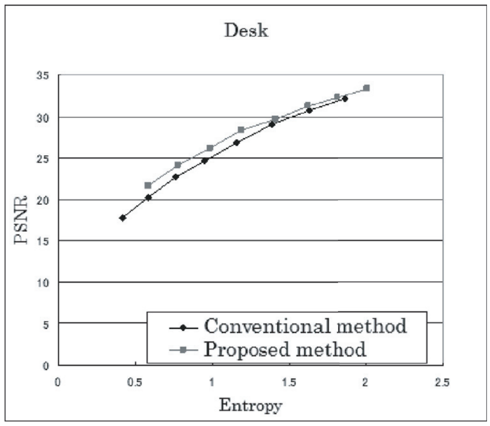
Our proposal method

(b)input(2)

Table 1. Comparing conventional method and our proposed method.



(a) Result(input(1))



(b) Results(input(2))

Fig. 6. Comparing conventional method and our proposed method.

5. CONCLUSION

We succeeded at designing the codebook suitable for the HDR images by our proposal method. Our method realized a high compression performance, compared to methods that directly apply the conventional vector quantization to the HDR images. From now on, the designing of codebook which make it possible that the higher PSNR will be necessary.

6. REFERENCES

[1] A.Gersho, and R.M. Gray, "Vector Quantization and Signal Compression", Springer, 1991.

[2] Erik Reinhard, Sumanta Pattanaik, Greg Ward and Paul Debevec, " High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting (Morgan Kaufmann Series in Computer Graphics and Geometric Modeling), Morgan Kaufmann Publisher 2005.

[3] Erik Reinhard, Mike Stark, Peter Shirley and Jim Ferwerda, " Photographic Tone Reproduction for Digital Images ", ACM Trans on Graphics. 21, 3, 267–276.

[4] Ruifeng Xu, Sumanta N. Pattanaik, Charles E. Hughes: High-Dynamic-Range Still-Image Encoding in JPEG 2000. IEEE Computer Graphics and Applications 25(6): 57-64 (2005)

[5] R. Mantiuk, G. Krawczyk, K. Myszkowski and H. P. Siedel, "Perception-Motivated High-Dynamic Range Video Encoding, ACM Trans. on Graphics, col23, no.3 2004, pp.773-741.

[6] Ward, Greg, and Maryann Simmons, "JPEG-HDR: A Backwards-Compatible, High Dynamic Range Extension to JPEG," Proceedings of the Thirteenth Color Imaging Conference, November 2005.

[7] Masahiro Okuda, and Nicola Adami, "Two-Layer Coding Algorithm for High Dynamic Range Images based on Luminance Compensation," Journal of Visual Communication and Image Representation, Elsevier, Vol.18, Issue 5, pp.377-386, Oct., 2007