BLOCK-BASED GRADIENT DOMAIN HIGH DYNAMIC RANGE COMPRESSION DESIGN FOR REAL-TIME APPLICATIONS

Tsun-Hsien Wang[1], Wei-Ming Ke[2], Ding-Chuang Zwao[2], Fang-Chu Chen[3], and Ching-Te Chiu[2]

[1]SoC Technology Center, Industrial Technology Research Institute, Hsin-Chu, Taiwan, R.O.C.
[2] Department of Computer Science, National Tsing Hua University, Hsin-Chu, Taiwan, R.O.C
[3]Information & Communications Research Laboratories, Industrial Technology Research Institute, Hsin-Chu, Taiwan, R.O.C.
Email: [1]thwang@itri.org.tw, [3]fcchen@itri.org.tw, and [2]ctchiu@cs.nthu.edu.tw

ABSTRACT

Due to progress in high dynamic range (HDR) capture technologies, the HDR image or video display on conventional LCD devices has become an important topic. Many tone mapping algorithms are proposed for rendering HDR images on conventional displays, but intensive computation time makes them impractical for video applications. In this paper, we present a real-time block-based gradient domain HDR compression for image or video applications. The gradient domain HDR compression is selected as our tone mapping scheme for its ability to compress and preserve details. We divide one HDR image/frame into several equal blocks and process each by the modified gradient domain HDR compression. The gradients of smaller magnitudes are attenuated less in each block to maintain local contrast and thus expose details. By solving the Poisson equation on the attenuated gradient field block by block, we are able to reconstruct a low dynamic range image. A real-time Discrete Sine Transform (DST) architecture is proposed and developed to solve the Poisson equation. Our synthesis results show that our DST Poisson solver can run at 50MHz clock and consume area of 9 mm^2 under TSMC 0.18um technology.

Keywords: High Dynamic Range (HDR), gradient, tone mapping, Discrete Sine Transform (DST), Inverse Discrete Sine Transform (IDST),

1. INTRODUCTION

The luminance ratio of the natural scenes is about 100,000,000:1 while traditional displays can only show images in dynamic range of 100~1000:1. Due to the technological advances in the HDR capture, high dynamic range images or video become available. Compared with the low dynamic range images, the HDR images show the real scene information much better [2]. However, we are confronted by two challenges. The first is how to display high dynamic range images on the low dynamic range devices, for example, monitors, TV displays, and printers. The second is how to maintain the details of images and to show the complete information in the scene.

Tone mapping or tone reproduction is an image processing technique to render the high dynamic range images on conventional displays. Over the past few years, a considerable number of studies have been made on tone mapping. They are commonly classified into global and local tone mappings. Chiu, et al., discover that mapping each pixel of images by global tone operators results in common artifacts because human vision system is nonlinear [3]. They believe that tone mapping based on the feature of pixels results in better effects. However, there is no ideal tone mapping curve that can be applied to every pixel. A method that emphasizes the variety in local areas of images is needed. Because their algorithm is based on local operators, it consumes much computing resource. Furthermore, it is developed based on experimental results rather than theoretical derivations.

In 1998, Pattanail, et al., developed a technique to represent patterns, luminance and color processing in human visual systems. They proposed a real scene tone mapping computing model. Their model could process HDR images with perception of scenes at threshold and supra-threshold [4]. Because this algorithm took the adaptability of colors into account, it consumed large amount of computing resource. Nevertheless, it had better sensitivity compared with other tone mapping algorithms.

In 2002, Fattal, et al., proposed a simple and effective method to render high dynamic rage images [1]. Their approach manipulated the gradient field of luminance (illumination differences) and attenuated the magnitudes of large gradients. A new low dynamic range image was obtained by solving a Poisson equation on the modified gradient field. It achieved drastic dynamic range compression and well preserved the fine details.

Human eyes are more sensitive to the luminance than to colors. Therefore, most HDR compressions process the images on the luminance channel. We adopt the gradient domain high dynamic range compression proposed by Fattal, etc. The reason is that human visual system is more sensitive to illumination differences than to absolute luminance reaching the retina, based on the retinex theory by Land and McCann in 1971[5]. However, the gradient domain HDR compression consumes significant computational time. In [6], it shows that 45.5 sec is required to process a 1600x1200 image on Apple iBook with a G3 processor running at 800MHz. This kind of processing speed cannot meet the requirement for real-time HDR video display.

In this paper, we present a block-based gradient domain high dynamic range compression scheme for real-time applications. Instead of processing the whole image at a time, we manipulate the gradient computations block by block. This enhances the processing speed significantly. A fully pipelined fast discrete sine transform (DST) is also presented to solve the Poisson equation. The block-based gradient compression is presented in section II. The hardware implementation and the DST-based Poisson solver are described in section III. The implementation results are shown in section IV. Finally, section V gives a brief conclusion.

2. BLOCK-BASED GRADIENT COMPRESSION

2.1 Gradient Domain HDR Compression

The gradient domain high dynamic range compression is shown in Figure 1. The human visual system is not sensitive to absolute luminance; it is sensitive to luminance changes of both local and global contrasts. The algorithm is based on the simple observation that any drastic change in the luminance in a high dynamic range image results in the rise of the magnitude of luminance gradients. On the contrary, the magnitude of gradients of fine texture in the image is smaller. Therefore, the idea is to identify large gradients, and then attenuate their magnitudes at various scales [7] while keeping their direction unchanged. The attenuation must shrink larger gradients more than smaller ones. Hence, it can compress drastic luminance changes and preserve the fine details. The LDR image is reconstructed by solving a partial differential equation from the attenuated gradient fields.

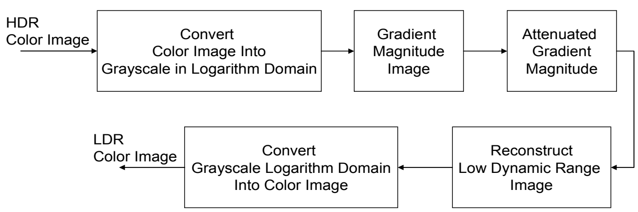

Figure 1: Gradient Domain High Dynamic Range Compression

We compute the gradients in the logarithm domain because the logarithm of the luminance is a better approximation of the perceived brightness [1]. Denote the logarithm of the luminance in the HDR image as H(x,y). To avoid spatial distortions in the image, we change only the magnitudes of the gradients and keep their directions unaltered. This goal is achieved by applying a spatially variant attenuating function Φ, we compute

$$G(x,y) = \nabla H(x,y)\Phi(x,y) \quad (1)$$

Here, $\nabla H(x,y)$ is the gradient of the logarithm image and $G(x,y)$ is the gradient image after attenuation. The gradient is approximated by the forward difference values.

$$\nabla H(x,y) \approx (H(x+1,y) - H(x,y), H(x,y+1) - H(x.y)) \quad (2)$$

The attenuation function will be discussed in section 2.2. After attenuating the gradient, we compute the differential of attenuated gradient (G_x, G_y) to get the divergence. We use the following backward difference to get the approximations of the divergence.

$$divG \approx G_x(x,y) - G_x(x-1,y) + G_y(x,y) - G_y(x,y-1) \quad (3)$$

According to [8], we can reconstruct a LDR image ***I*** by solving the following Poisson Equation

$$\nabla^2 I = div\ G \quad (4)$$

where

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

and

$$divG = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y}$$

The ∇^2 is the Laplacian operator. The $\nabla^2 I$ is obtained from the following standard finite difference approximation.

$$\nabla^2 I(x,y) \approx I(x+1,y) + I(x-1,y) + I(x,y+1) + I(x,y-1) - 4I(x,y) \quad (5)$$

To solve the partial differential equation (PDE), we must assign boundary conditions, and then find the integration constant in general solutions. At the boundaries, we assume that the derivatives around the original image grid are zeros.
The numerical solution of the PDE can be obtained through the Discrete Sine Transform [9]. Finally, we compute the luminance by the exponential operation, and transform the processed luminance to RGB colored image by the following equation [1].

$$C_{out} = \left(\frac{C_{in}}{L_{in}}\right)^s L_{out} \quad (6)$$

In this equation, the **C** presents the R, G, B color component. The C_{in} and C_{out} are the R, G, B values before and after the HDR compression. The L_{in} and L_{out} are the gray-scaled luminance before and after HDR compression. The exponent s controls the color saturation of the resulting image, and it is assigned as 0.5.

2.2 Gradient Attenuation Function

The HDR compression can be achieved by attenuating the gradients of each pixel in the image by the attenuated factor. The attenuation factor is obtained from the modified gradient as

$$\nabla H_m = \left(\frac{H(x+1,y) - H(x-1,y)}{2}, \frac{H(x,y+1) - H(x,y-1)}{2}\right) \quad (7)$$

The attenuation factor for each pixel is determined by the magnitude of the modified gradient, as the following equation

$$\Phi(x,y) = \frac{\alpha}{\|\nabla H_m(x,y)\|}\left(\frac{\|\nabla H_m(x,y)\|}{\alpha}\right)^\beta \quad (8)$$

The function has two parameters α and β which determine how to attenuate the gradient of each pixel. The relation between gradient magnitude and attenuation is shown in Figure 2 and Figure 3.

Figure 2 shows the relation between gradient magnitude and attenuation factor for different α with a fixed β. We attenuate the gradient of a pixel if its magnitude is bigger than α, preserve it if its magnitude is equal to α, and magnify it if its magnitude is smaller than α. In order to achieve attenuation, the value of β must be smaller than one. Figure 3 shows that the attenuation factor curve is shaper if β is smaller.

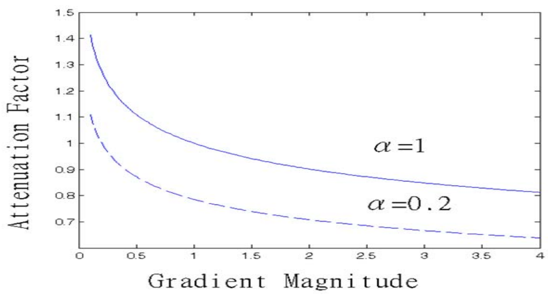

Figure 2: Relation between gradient magnitude and attenuation factor for different α and β=0.85.

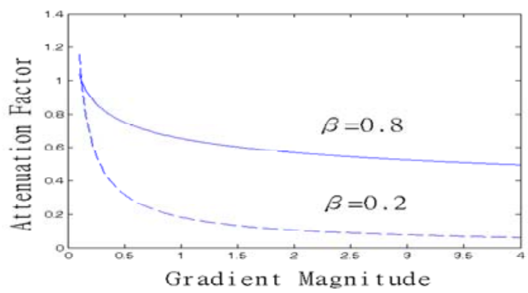

Figure 3: Relation between gradient magnitude and attenuation factor for different β and α= 0.12

2.3 Block-Based Gradient Compression Algorithm

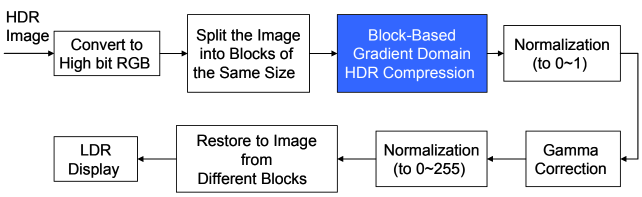

Figure 4: Block-Based Gradient Domain HDR Compression Procedure

The procedure of the block-based gradient domain HDR compression is shown in Figure 4. The HDR input can be in any HDR format such as RGBE, XYZE or LogLuv etc.. In this paper, we use the 32-bit per pixel RGBE format with 76 orders of dynamic range. The RGBE HDR image is converted into three 32-bit floating point RGB channels. The image is further divided into blocks of the same size *N*x*N*. Then we extend the block by copying pixels from four boundaries as shown in Figure 5. The extended block size becomes (*N*+2)x(*N*+2). The extended block is for the approximation of the divergence and Laplacian. Then we apply the gradient domain HDR compression scheme described in section 2.1 to each block. Finally the compressed RBG channels are normalized, passed through the Gamma correction and mapped to the range of 0~255.

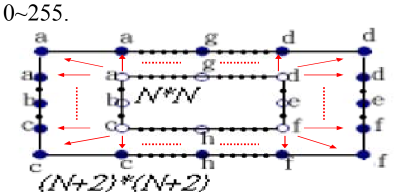

Figure 5: Extended block with boundary points

3. HARDWARE IMPLEMENTATIONS

Figure 6 shows the architecture of the block-based gradient domain HDR compression. The input is the extended block (size (*N*+2)*(*N*+2)) obtained from the high dynamic range image. The detail data flow includes the logarithm luminance calculation, the one-level attenuation, and reconstruction through the PDE. The output is the compressed LDR block of size *N***N*. Among all the operations, the PDE block consumes most of the computation time. We propose the DST-based Poisson solver. The algorithm of the DST-based Poisson solver is described below.

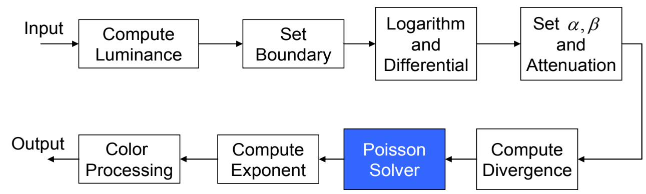

Figure 6: Block-based Gradient Domain HDR Compression Prcцedure

3.1 Solving Poisson Equation by DST

The discrete model of the Poisson equation (4) can be written as the following linear system.

$$T\ U = F \qquad (9)$$

In the equation, vector *U* is an approximation of the reconstructed LDR image I(x,y). The matrix T is a '1 1 -4 1 1'tridiagonal matrix, and the F is a Right Hand Side Matrix which includes boundary conditions. We can solve the discrete Poisson equation by the Discrete Sine Transform (DST) via the eigensystem [9].

The data flow of solving the DST-based Poisson solver is shown in Figure 7. We apply the 2-D DST on the right hand side matrix F and call the result as the matrix B. The 2-D DST/IDST is done through two 1-D DST/IDST and the matrix transposition.

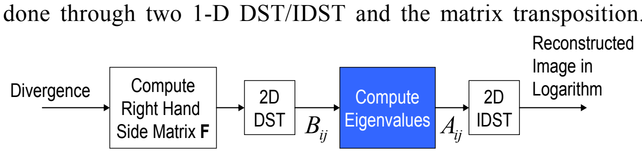

Figure 7: Solving the Poisson Equation by the DST

Then we divide the result B_{ij} by the system eigenvalues to get A_{ij}, as follows:

$$A_{ij} = \frac{B_{ij}}{2\cos(\frac{j\pi}{N})-2 + 2\cos(\frac{i\pi}{N})-2} \qquad (10)$$

Note that the denominator in the equation above is the eigenvalue. The *j* and *i* are the column and row indexes. Next, we apply the 2-D IDST to the matrix A to get the reconstructed image in the logarithm domain. The definition and implementation of the 1-D DST are described below.

3.2 DST Hardware Architecture

Given $\mathbf{v}=(v_0,.....,v_{N-1}) \in R^N$,we say that the vector **w**, where $\mathbf{w}=(w_0,....,w_{N-1})^T$, is the Discrete Sine Transform of **v**, as

$$w_k = \sum_{n=0}^{N-1} v_n \sin[\frac{\pi(n+1)(k+1)}{N+1}] \qquad (11)$$

We can rewrite the 1-D DST above into matrix form **w**=**Sv**, where **S** is the DST transform matrix. The IDST is the DST in Eq. (11) multiplied by 2/(*N*+1).

We design a new hardware architecture to implement the *N*-point 1-D DST. The architecture is very simple and contains only few additions and multiplications. We decompose the w_k into the even and odd indexes. The *N* point DST can be decomposed into the sum of two *N/2* point DST[10]. Figure 8(a) is the 8-point DST architecture with 4-stage pipeline. Figure 8(b) ~ 8(e) shows the architectures for stage 1, stage 2, stage3 and stage 4 respectively. The C_b^a denotes $\cos(\frac{a}{b}\pi)$, and S_b^a denotes $\sin(\frac{a}{b}\pi)$ in the Figure 8(b) ~ 8(e) .

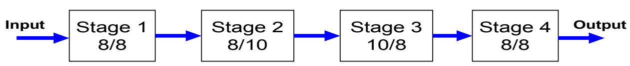

Figure 8(a): 8-point DST architecture with 4-stage pipeline, where 8/10 indicates 8 pixels input and 10 pixels output and so on.

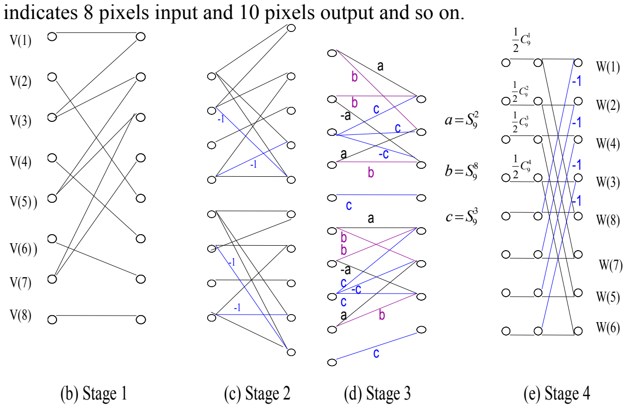

Figure 8(b)~(e): The architecture for every stage in 8-point DST/IDST

4. IMPLEMENTATION RESULTS

The software simulation of the block-based gradient domain HDR compression with block size of 8x8 is shown in Figure 9(a). The input is a HDR image with size of 512x768. The dynamic range of RGB components of the HDR image are 600000:1, 550000:1, and 800000:1 respectively. The fine details of the compressed LDR image are still well preserved. Furthermore, there is no blocking effect in the reconstructed image.

We implement the Poisson Solver by taking the consideration of speed and area. The data bus for each pixel used in the Poisson block is 24-bit wide. The eigenvalues are stored in the memory and obtained through the look up table. We implement the transpose buffer and controller to read the pixels from memory in parallel for the matrix transposition. The reconstructed image obtained from the hardware implementation is shown in Figure 9(b). The image quality is close to that of the software realization. The synthesis result shows that the whole Poisson Solver can run at 50 MHz and consume area of 9 mm^2 under TSMC 0.18um technology.

5. CONCLUSION

In this paper, we present a real-time block-based gradient domain HDR compression architecture. The implementation results show that real-time gradient domain HDR compression is possible. Moreover, the local details of pixels are well preserved. Since our approach is block-based so we can reduce the computation time and save the chip area. We propose a fully-pipeline DST-based Poisson solver for the PDE equation. Our synthesis results show that our DST Poisson solver can run at 50 MHz clock and consume 9 mm^2 area under TSMC 0.18um technology.

(a) (b)

Figure 9: (a) and (b) are software simulation result and hardware implementation result respectively

ACKNOWLEDGEMENTS

CTC acknowledges support from National Science Council project No. NSF-95-2220-E-007-033. We also acknowledge support from Information and Communication Research Laboratories, and SoC Technology Center, ITRI.

REFERENCES

[1] R.Fattal, D.Lischinski and M. Werman, "Gradient Domain High Dynamic Range Compression", ACM SIGGRAPH, 2002.

[2] P. E. Debevec and J. Malik,"Recovering High Dynamic Range Radiance Maps from Photographs", ACM SIGGRAPH,1997.

[3] K.Chiu, M.Herf, P.Shirley, S.Swamy, C.Wang, and K. Zimmerman, "Spatially Nonuniform Scaling Functions for High Contrast Images", In Graphics Interface '93, pp.245-244, May 1993.

[4] S.N. Pattanaik, J.A. Ferwerda, M.D. Fairchild, and D.P. Greenberg,"A Multiscale Model of Adaptation and Spatial Vision for Realistic Image Display", Proc. of SIGGRAPH 98, pp.287-298,1998.

[5] E.Land and J.McCann, "Lightness and Retinex Theory", Journal of the Optical Society of America, vol. 61, issue 1, pp.1-11,1971.

[6] E.Reinhard, G.Ward, S.Pattanaik and P.Debevec, "High Dynamic Range Imaging",MORGAN KAUFMANN PUBLISHERS ,2006.

[7] P.J.Burt and E.H.Adelson," The Laplacian Pyramid as a Compact Image Code", IEEE Trans. on Comm. ,vol.31,pp.532-540, April 1983.

[8] David R. Kincaid, "Celebrating Fifty Years of David M. Young's Successive Overrelaxation Iterative Method", *Numerical Mathematics and Advanced Applications*, M. Feistauer, V. Dolejsi, P. Knobloch, K. Najzar (Eds.), Springer-Verlag, 2004, pp. 549-558.

[9] S.N. Wansundara, "Solving The Discrete Poisson Equation Using the Fast Fourier Transform Technique", Memorial University of Newfoundland ,2002.

[10] J.C. Yao and C. Y. Hsu , "New Approach For Fast Sine Transform", ELECTRONICS LETTERS 16th July 1992 vol.28 NO.15 , 1992.