

REAL-TIME DEPTH IMAGE BASED RENDERING HARDWARE ACCELERATOR FOR ADVANCED THREE DIMENSIONAL TELEVISION SYSTEM

Wan-Yu Chen, Yu-Lin Chang, Hsu-Kuang Chiu, Shao-Yi Chien, and Liang-Gee Chen

DSP/IC Design Lab,
Graduate Institute of Electronics Engineering,
National Taiwan University, Taipei, Taiwan

ABSTRACT

3D TV will become a prominent technology in the next generation. In this paper, a depth image based rendering system is proposed from algorithm level to hardware architecture level. We propose a novel depth image based rendering algorithm with edge-dependent gaussian filter and interpolation to improve the rendered stereo image quality. Based on our proposed algorithm, a fully-pipelined depth image based rendering hardware accelerator is proposed to support real-time rendering. The proposed hardware accelerator is optimized in three steps. First, we analyze the effect of fixed point operation and choose the optimal wordlength to keep the stereo image quality. Second, a three-parallel edge-dependent gaussian filter architecture is proposed to solve the critical problem of memory bandwidth. Finally, we optimize the hardware cost by the proposed hardware architecture. Only 1/21 amounts of vertical PEs and 1/11 amounts of horizontal PEs is needed by the proposed folded edge-dependent gaussian filter architecture. Furthermore, by the proposed check mode, the whole Z-buffer can be eliminated during 3D image warping. In additions, the on-chip SRAMs can be reduced to 66.7 percent compared with direct implementation by global and local disparity separation scheme. A prototype chip can achieve real-time requirement under the operating frequency of 80 MHz for 25 SDTV frames per second (fps) in left and right channel simultaneously. The simulation result also shows the hardware cost is quite small compared with the conventional rendering architecture.

1. INTRODUCTION

Depth-Image-Based-Rendering (DIBR) is a key technology in advanced three dimensional television system (ATTEST 3D TV System) [1] [2]. Traditional 3D TV system requires the transmission of two video streams, the left and right view, to construct 3D vision. Unlike the traditional method, the advanced three dimensional television system proposed a novel technology "DIBR" to provide 3D vision. DIBR uses intermediate view plus intermediate depth map to render left and right view at display end. In this way, broadcast content provider only has to transmit the video and gray level depth map of the intermediate view. It has been testified the coding efficiency is better than the transmission of stereo video stream [3]. Another advantage is the 2D/3D selectivity. Users can change 3D vision into 2D vision just by displaying intermediate view. Depth image based rendering contains three steps [4]. First of all, pre-processing of depth map is applied for reducing the sharp horizontal transitions on depth map. Second, 3D image warping renders left and right images based on the pre-processed depth map and intermediate color image. Finally, if there are still occlusion holes on the rendered view, hole-filling is applied for interpolating the occlusion holes. Our previous work [4] is proposed to reduce the nonnecessary computation cycles while main-

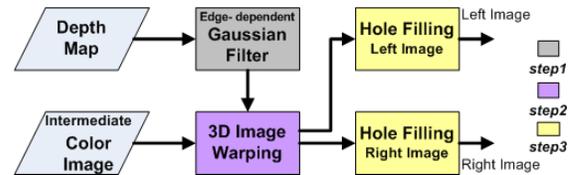


Fig. 1. Block Diagram of the Proposed DIBR System

taining good rendering quality. We can reduce the computation cycles from 190GIPS(Giga Instructions Per Second) to 7.08GIPS by our DIBR algorithm [4]. However, the computation load is still too heavy to implement by the current CPU system. Therefore, a hardware accelerator solution is required urgently. Under real-time constraint, it is preferred that edge-dependent gaussian filter and 3D image warping, which is called computation core in the DIBR system, should be realized by a hardware accelerator due to its heavy computational load, which dominates the DIBR system by 99 percent. In this paper, a fully-pipelined hardware architecture for DIBR hardware accelerator with edge-dependent gaussian filter and 3D image warping is proposed. The algorithm is designed for hardware implementation. It is modified from our prior successful algorithm [4]. We exploit edge-dependent gaussian filter to achieve good stereo image quality in the DIBR system. In addition to that, we reduce the external memory bandwidth to reach the practical bus requirement by the proposed three parallel edge-dependent gaussian filter. Furthermore, the proposed hardware architecture is optimized for the reduction of hardware cost. The hardware-oriented algorithm for depth image based rendering system is first described in the next section. Then the hardware architecture and implementation results are shown in Section III and IV, respectively. Finally, in Section V gives the conclusion.

2. HARDWARE-ORIENTED ALGORITHM

The block diagram of the proposed DIBR system is shown in Fig. 1. The proposed DIBR system contains three steps. First of all, edge-dependent gaussian filtering of depth map is needed for reducing the horizontal transitions on depth map. Second, 3D image warping renders left and right images according to the pre-processed depth map and intermediate color image. Finally, hole-filling will be applied to interpolate the occlusion holes if there are still occlusion holes on the rendered view.

2.1. Complexity and Run-time Analysis

In our proposed DIBR system, the computation complexity can be lowered to 7.08 GIPS under ATTEST real-time spec(frame size 720x576 @25fps)[1]. However, the computation complexity is

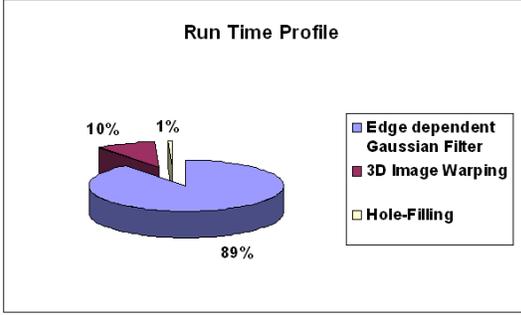


Fig. 2. Run Time Profile

still too heavy for the current CPU system. The current Intel P4 CPU system can only afford 3.8 GIPS [5] for PC application. To this end, we provide a real-time hardware accelerator for the proposed DIBR system. According to the run-time profile in Fig. 2, the computation core in our DIBR system is constituted by edge-dependent gaussian filter and 3D image warping. If we implement the heavy computation core by hardware, the whole DIBR system can be runned at the real-time spec.

2.2. Edge-dependent gaussian filter

The edge-dependent gaussian filter is used to reduce the sharp horizontal transitions and thus reduce the number of big hole after 3D image warping. The formula of 2D gaussian filter is listed as follows:

$$d'(x, y) = \frac{\sum_{i \in W_H} (\sum_{j \in W_V} (g(j) * d(x+i, y+j)) * g(i))}{\sum_{i \in W_H} (\sum_{j \in W_V} (g(j) * g(i)))}$$

$d'(x, y)$ means the filtered depth value. $d(x, y)$ is the original depth value. $g(i)$ and $g(j)$ means the weighting parameter of 2D gaussian filter. W_H and W_V are the horizontal and vertical window size of the above 2D gaussian filter. The 2D gaussian filter can be decomposed into two pass. In the first pass, we compute the vertical one dimensional gaussian filter result. In the second pass, we compute the horizontal gaussian filter result according to the vertical gaussian filter result. By the two pass decomposition, we can reuse the vertical gaussian filter result and thus save the external memory bandwidth. When we implement the 2D gaussian filter by hardware, there are two variables we have to determine. First, the window size have to be determined. Second, the word length of the weighting parameter have to be made up. The decision criterion of the above two variants is the image quality of the rendered view and hardware cost. As we know, the larger the window size is, the more the amount of external memory bandwidth and multiplication unit is. In addition to that, longer wordlength of weighting parameter would cause larger multiplication hardware cost and longer the critical path. Thus we optimize the window size and wordlength of the two dimensional filter by the trade-offs between image quality and hardware cost. The optimized window size is $5(W_H) \times 30(W_V)$. The optimized fixed wordlength is 14bit.

2.3. The Proposed 3D Image Warping

The 3D Image Warping algorithm can be expressed by the following pseudo codes:

$$DV_{local}(x, y) = Transform(depth(x, y));$$

$$DV_{global} = -CCDSHIFT;$$

```

DV(x, y) = DVlocal(x, y) + DVglobal;
SL = x + DV(x, y); SR = x - DV(x, y);
if (FlagL(SL, y) == false || DV(x, y) > DVL(SL, y))
begin
L(SL, y) = I(x, y);
DVL(SL, y) = DV(x, y);
FlagL(SL, y) = true;
end
if (FlagR(SR, y) == false || DV(x, y) > DVR(SR, y))
begin
R(SR, y) = I(x, y);
DVR(SR, y) = DV(x, y);
FlagR(SR, y) = true;
end

```

DV means disparity vector. DV_{local} and DV_{global} means local and global disparity vector. $I, L,$ and R means intermediate view, left view and right view respectively. S_R and S_L means the mapped address of the current pixel on right and left image respectively. $Flag_L$ and $Flag_R$ is represented by true or false. When the mapped color buffer is occupied, $Flag_L$ and $Flag_R$ is marked by true. Otherwise the $Flag_L$ and $Flag_R$ maintains false. DV_R and DV_L means the disparity vector of the mapped left and right location. As shown in above pseudo codes, the intermediate color view maps to left and right view according to its depth value. When doing the 3D image warping, disparity vector is first computed. The disparity vector is constituted by local disparity vector DV_{local} and global disparity vector DV_{global} . The global disparity vector is invariant on the whole image and determined by the camera CCD shift. Instead of that, local disparity varies with depth value. Thus we can divide the 3D image warping into two pass. First we can do 3D image warping just according to local disparity vector only. And then shift the whole rendered image in horizontal direction according to global disparity vector. Therefore, we can reduce the range of disparity vector and thus reduce the size of color buffer during 3D image warping. The range of mapping location is related to the range of disparity vector. The mapping range will be confined by 96 pixels, considering both global disparity vector and local disparity vector. By the proposed global and local disparity separation scheme, the mapping range is reduced to 64 pixels. In addition to that, we can discover that the original check mode needs Z-Buffer when over-mapped condition occurs. The concept of the check mode is that the pixel value of the rendered view is decided by the pixel value with maximum disparity. For the reason that near object with larger disparity vector is visible and the farther object with smaller disparity vector is invisible. However, the Z-buffer can be eliminated by the proposed check mode. Because we do 3D image warping in raster scan, $I(x, y)$ is former mapping pixel and $I(x+1, y)$ is later mapping pixel. Suppose $I(x, y)$ and $I(x-2, y)$ map to the same location $L(a, y)$ on left image. That means $DV(x, y) = a - x$ and $DV(x-2, y) = a - x + 2$ according to the above pseudo codes. It represents that the former color value $I(x-2, y)$ would determined the color of $L(a, y)$. On the other hand, suppose $I(x-2, y)$ and $I(x, y)$ map to the same location $R(a, y)$. The corresponding disparity vector is $DV(x-2, y) = x-2-a$ and $DV(x, y) = x-a$. It means that the color of $R(a, y)$ is determined by $I(x, y)$, which is the later mapping pixel. Thus we can decide the final mapped color of the rendering color buffer by the following scheme.

```

begin
L(SL, y) = I(x, y);
FlagL(SL, y) = true;
end
if (FlagR(SR, y) == false)
begin
R(SR, y) = I(x, y);

```

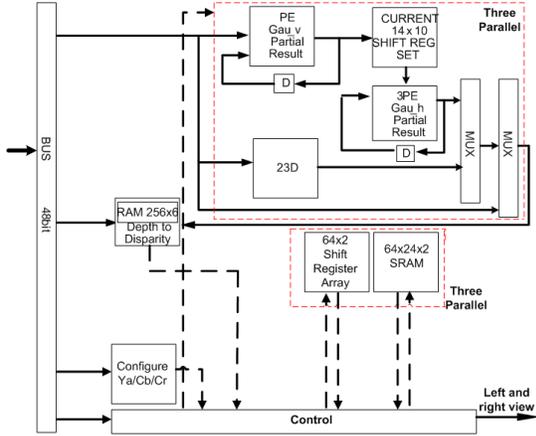


Fig. 3. Hardware Architecture of the proposed DIBR accelerator

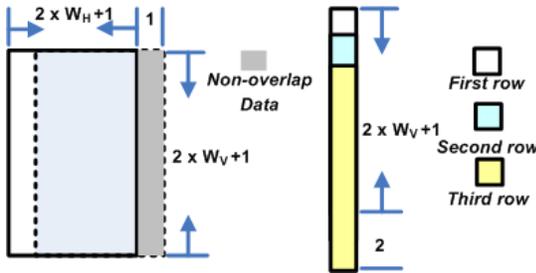


Fig. 4. Data Reuse and Data Sharing Scheme

$Flag_R(S_R, y) = true;$

end

By using the proposed check mode, we can check if we have to write the mapped color buffer again without Z-buffers.

3. HARDWARE ARCHITECTURE

As shown in Fig. 3, the proposed DIBR accelerator includes two parts. One is edge-dependent gaussian filter and another is 3D image warping. First the depth-to-disparity transform table is loaded into the 256x6 depth-to-disparity SRAM Unit. Second, the edge information is transmitted to the Control Unit. At the same time, the three-parallel edge-dependent gaussian filter computes the filtered depth value according to the control signal. When the current pixel is edge, the horizontal and vertical PEs will operate to compute the filtered result. Otherwise, the original depth value will be passed to the 3D image warping unit without filtering. After edge-dependent gaussian filtering, the filtered depth value would be transformed into local disparity by the depth-to-disparity transform SRAM. Then, the transformed local disparity is used to compute the mapped address by the control unit. Then, the control unit will decide whether the color buffer need to be written again and write the mapped color buffer if necessary. Finally, the control unit output the synthesized stereo image with the corresponding flag by reading the color buffer. The corresponding flag represents whether the pixel is a hole or not.

3.1. Trade-offs between External Memory Bandwidth and Hardware Cost

$$B = \left(\frac{720 * 576 * 25}{N} \right) * \left(\frac{61 + (N - 1)}{N} \right)$$

$$B < 133MHz$$

In the above equation, B means external bus speed and N means the number of parallelism. The external memory bandwidth is the critical problem when we design the proposed DIBR hardware accelerator. That is because the large window size of gaussian filter, which is 5x30. Although we can reuse the vertical gaussian filtered result in the previous stage, we also need to input the 61 depth value when doing 2D gaussian filter of one pixel. That is to say, the bus speed have to be runned at 632.448MHz according the above equation with N equals to one. It is not practical in current hardware system. The current AMBA bus speed can only reach 133MHz in common case [6]. It is shown in Fig. 4 that if we applied gaussian filter on several horizontal lines at the same time, we can share the depth data when computing the vertical gaussian filter result. Besides, we can also reuse the vertical gaussian filtered result by saving them in the CURRENT SHIFT REG SET in Fig. 3. We have to find the minimum number of parallelism to reduce the hardware cost caused by parallelism. The minimum number of parallelism is three. Thus we adopt three-parallel architecture in our DIBR hardware accelerator.

3.2. Folded architecture for Two-dimensional gaussian filter

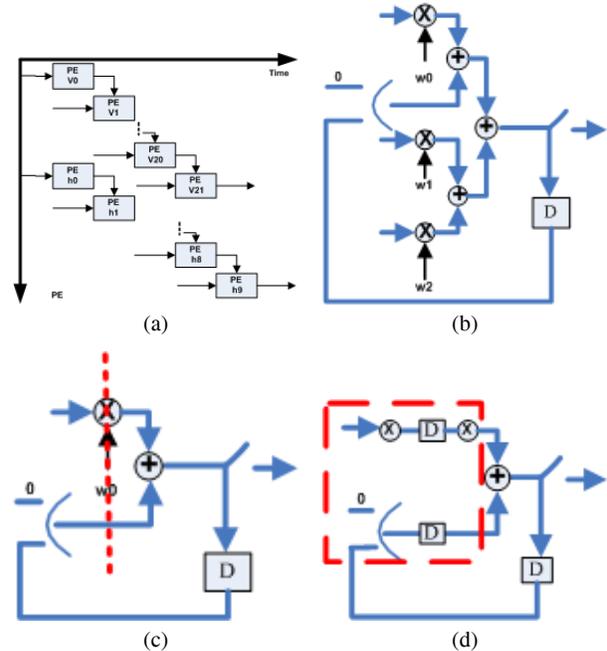


Fig. 5. Subjective View Results of flag (a)Original Scheduling without folding (b) Folded Vertical PE (c) Folded Horizontal PE (d) Pipelined of the Folded Horizontal PE

The two-dimensionnall gaussian filter have to be processed in 23 cycles to reach the real-time constraint. As shown in Fig. 5, the three-parallel depth data will be inputted cycle by cycle during the first 21 cycles. During the 23 cycles, each cycle we only

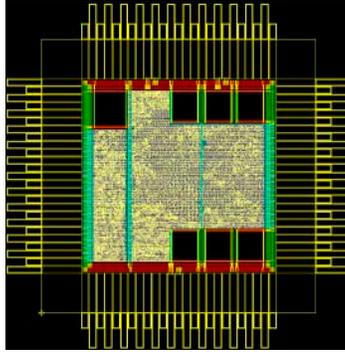


Fig. 6. Chip Photo

need one horizontal PE and one vertical PE to implement the 2D gaussian filter. Thus we can implement the vertical gaussian filter and horizontal gaussian filter by a folded architecture in Fig. 5 to reduce the number of PEs. By the proposed folding architecture, 1/21 amounts of horizontal PEs and 1/11 amounts of vertical PEs can be reduced.

3.3. Pipeline of The Horizontal Gaussian Filter

The horizontal gaussian filter result is also computed during 23 processing cycles. Because of the asymmetrical two dimensional gaussian filter in the proposed DIBR system, the processing time of horizontal gaussian filter is just 11 cycles. Besides, the critical path in the proposed architecture is due to the computation in the PE of horizontal gaussian filter. It is practical to pipeline the horizontal gaussian filter PE and reduce the critical path of the proposed architecture. Gate level synthesis results also shows the critical path is reduced from 12.5ns to 10.4ns.

4. IMPLEMENTATION

The design goal of our chip implementation is listed as follows: 720x576 frame size and 25 frames per second both in the left and right channels simultaneously. The chip is designed in cell-based design flow with Artisan 0.18um 1P6M standard cell library and Artisan RAM compiler. The rendering core chip is currently under fabrication by TSMC. The chip layout is shown in Fig. 6. There are seven groups of on-chip single-port SRAM on the chip. The functionality of these SRAMs are described in the previous section. The technology is TSMC 0.18um 1P6M, and the chip size is $2.03 \times 2.03 \text{ mm}^2$. The detailed features of the chip is shown in Table. 1. Simulation results show that this chip can achieve real-time requirement for 720x576 stereo video system at 80MHz. Note that the chip can perform depth image based rendering of the left and right channel in 1/25 second. Compared with the direct implementation of edge dependent gaussian filter, only 1/21 amount of vertical PEs and 1/11 amount of horizontal PEs are needed in this chip. Besides, the whole Z-Buffer which exists in common rendering system [7] is eliminated by the proposed checking mode. Furthermore, 1/3 amount of color buffer is reduced by the proposed global and local disparity separation scheme. It is quite area-efficient. Algorithm analysis also shows that it maintains good stereo image quality [4].

5. CONCLUSION

A hardware-oriented depth image based rendering algorithm and its associated hardware architecture is first proposed in this paper.

Table 1. CHIP SPECIFICATIONS.

Technology	TSMC 1P6M 0.18um
Chip size	2.03mmx2.03mm
Package	128 CQFP
On chip memory	74, 072bits
Logic gate count	162,410 (SRAM Included)
Maximum frequency	80 MHz
Power supply	1.8V
Power consumption	52.56 mW @ 80 MHz
Range of DV	[0:63]
Processing capability	25 SDTV(720x576)frames per sec in both channels

Compared with the common rendering architecture, the proposed depth image based rendering hardware can greatly reduces hardware resource requirement (on-chip SRAMs and PEs), while it still maintains good video quality. With the proposed data sharing and data reuse scheme, the problem of critical memory bandwidth requirement can be solved by sharing the vertical data while doing vertical gaussian filter in parallel and reusing the previous vertical gaussian filtered result while doing gaussian filter in the horizontal direction. We also reduce the critical path by the pipelining in the horizontal gaussian PE to satisfy the timing constraint in real-time DIBR system. After satisfying the real-time constraint, our design target turn into cost reduction. The folded two dimensional gaussian filter architecture reduce the combinational circuit to 1/21 in the amounts of vertical PEs and 1/11 in the amounts of the horizontal PEs. In the algorithm level, global and local disparity vector separation reduces the size of color buffer at the rate of 33%. In addition to that, the proposed check mode which determines the pixel value on left and right image when over-mapped condition occurs can totally eliminate the Z-buffer in common rendering system. A prototype chip is currently under fabrication with 0.18um 1P6M technology by TSMC. It shows the chip size is small and can be easily applied into advanced three dimensional television system.

6. REFERENCES

- [1] A. Redert, M. Op de Beeck, C. Fehn, W. IJsselstein, M. Pollefeys, L. Van Gool, E. Ofek, I. Sexton, and P. Surman, "Attest advanced three-dimensional television system," *Proc. of 3DPVT*, pp. 313–319, 2002.
- [2] Chung J. Kuo, Ching Liao, and Ching C. Jin, "Stereoscopic image generation based on depth images," *ICIP*, pp. 2993–2996, 2002.
- [3] C. Fehn, K. Hopf, and Q. Quante, "Key technologies for an advanced 3d-tv system," *In Proceedings of SPIE Three-Dimensional TV, Video and Display III*, pp. 66–80, 2004.
- [4] Wan-Yu Chen, Yu-Lin Chang, Shyh-Feng Lin, Li-Fu Ding, , and Liang-Gee Chen, "Efficient depth image based rendering with edge dependent depth filter and interpolation," *ICME*, 2005.
- [5] "http://www.intel.com," *Intel Cooperation*.
- [6] "http://www.arm.com," *ARM Cooperation*.
- [7] Voicu Popescu, John Eyles, Anselmo Lastra, Joshua Steinhurst, Nick England, and Lars Nyland, "The warpengine: An architecture for the post-polygonal age," *Proceedings of SIG-GRAPH 2000*, pp. 433–442, 2000.