Real-Time Multi-view Rendering Architecture for Autostereoscopic Displays

Hsin-Jung Chen, Feng-Hsiang Lo, Fu-Chiang Jan, and Sheng-Dong Wu
Electronics and Optoelectronics Research Laboratories, Industrial Technology Research Institute
Hsinchu, Taiwan, 31040, R.O.C.
{hjchen0328, fhLo, FCJan, and antoniowu}@itri.org.tw

***Abstract*—Three-dimensional television (3D-TV) has attracted significant attention because of 3D immersive feeling for advanced TV development. Multi-view rendering by depth image based rendering (DIBR) and interlacing are the key technologies to realize 3D-TV system from content to display. In this paper, a new multi-view rendering hardware architecture consisting of hybrid parallel DBIR and pipeline interlacing are proposed to improve the performance. Experimental results show that the proposed architecture can achieve 60 frames per second for processing full HD (1920x1080) video in real-time processing system. Only 3% logic elements of ALTERA Cyclone III FPGA are used.**

I. INTRODUCTION

3D-TV is expected to be the next generation of television because the autostereoscopic 3D-TV can provide the feeling of reality and increase visual immersion. Recently, DIBR has been defined as the key technology for virtual view synthesis in the 3DTV system [1]-[2]. There is also a great interest in the use of DIBR with respect to the development of a practical 3D-TV system because of efficient transmission and storage [3]. In DIBR, a depth map is essentially to be a two-dimensional (2D) function that generates depth value. And the depth value is produced by the reference camera position from a point on an object in a visual scene as a function of the image coordinates [4].

After giving an original image and the associated depth map, a virtual image of nearby viewpoints can be generated by re-projecting the pixels onto the virtual image plane. Therefore, DIBR is used to generate multi-view for autostereoscopic displays. However, effectively calculating occlusion holes in the virtual images will be a problem in DIBR [5]. Holes appear due to the accreted portions of objects and visible background on the new viewpoint. Afterwards, the multi-view interlacing has to be applied after DIBR to fit in the data format of the specified autostereoscopic display [6]. Considering the loading of the whole computation, a better hardware is necessary to achieve the real-time performance.

In this paper, a hybrid-parallel hardware architecture for DIBR system is proposed to generate multi-view image. The proposed scheme utilizes the feature of data processing by the parallel rendering to improve the processing performance. Furthermore, the pipeline interlacing is designed to achieve the image format of multi-view for autostereoscopic display by real-time transferring.

This paper will firstly describe multi-view rendering and it will explain the hardware architecture of DIBR and multi-view interlacing later. Finally, experiment results will be shown and a conclusion will be given.

II. MULTI-VIEW RENDERING SYSTEM

As shown in Fig. 1, the multi-view rendering system contains two components, DIBR and Interlacing respectively. In DIBR, the function of 2D image rendering is to warp the original image into multiple views according to depth information. Then the function of hole-filling is applied to inpaint the holes that still occur on the rendered views. Finally, the function of 3D image interlacing integrates the multiple views into interlaced multi-view image.

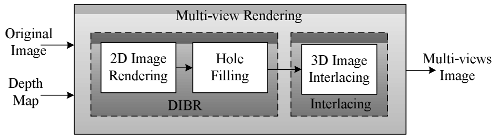


Fig. 1. Block diagram of the proposed multi-view rendering system.

The 2D image rendering generates the new multiple views according to the associated depth and the original image. After new images are generated, there are some holes appearing due to occlusions, so that the hole-filling is designed to fill holes by interpolation or inpainting.

In DIBR, the associated depth information is transformed into the disparity for projecting the original image into the multiple views. As shown in Fig. 2, for example, the viewpoint of the virtual views at the first and second virtual camera positions are c_1 and c_2 can be created based on a specific baseline distance as the total virtual cameras distance t divided by

978-1-4244-5309-2/10/$26.00 ©2010 IEEE

N. Moreover, p is a point of interest in the scene, the focal length of cameras is f, and the depth value, Z.

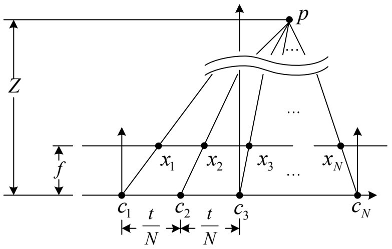


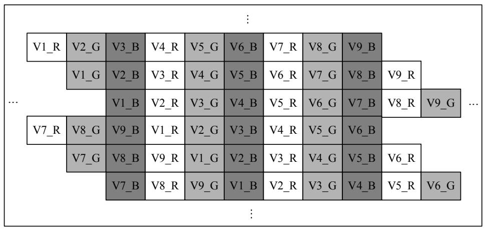
Fig. 2. Top-view of camera configuration for generating the virtual multi-views.

According to the definition of the previous camera configuration, the geometrical relation can be expressed mathematically in equation (1) as following:

$$x_m = x_{m-1} + \left(\frac{t}{N} \frac{f}{Z} \right) \quad m = 2, \cdots, N \tag{1}$$

The x_1 and x_2 are the horizontal coordinate of the first and the second view respectively. The set can be assembled to generate the multiple views.

On the other hand, the multi-view interlacing generates interlaced 3D image by using the data format for the target display. The configuration is for a slanted barrier with a slope of 3, for example. Within the interlaced 3D image as shown in Fig. 3, the subpixels are selected from RGB subpixels of multi-view images corresponding to the subpixel's position in the final interlaced view.



						⋮						
	V1_R	V2_G	V3_B	V4_R	V5_G	V6_B	V7_R	V8_G	V9_B			
		V1_G	V2_B	V3_R	V4_G	V5_B	V6_R	V7_G	V8_B	V9_R		
…			V1_B	V2_R	V3_G	V4_B	V5_R	V6_G	V7_B	V8_R	V9_G	…
	V7_R	V8_G	V9_B	V1_R	V2_G	V3_B	V4_R	V5_G	V6_B			
		V7_G	V8_B	V9_R	V1_G	V2_B	V3_R	V4_G	V5_B	V6_R		
			V7_B	V8_R	V9_G	V1_B	V2_R	V3_G	V4_B	V5_R	V6_G	
						⋮						

Fig. 3. The relation of multi-view interleacing.

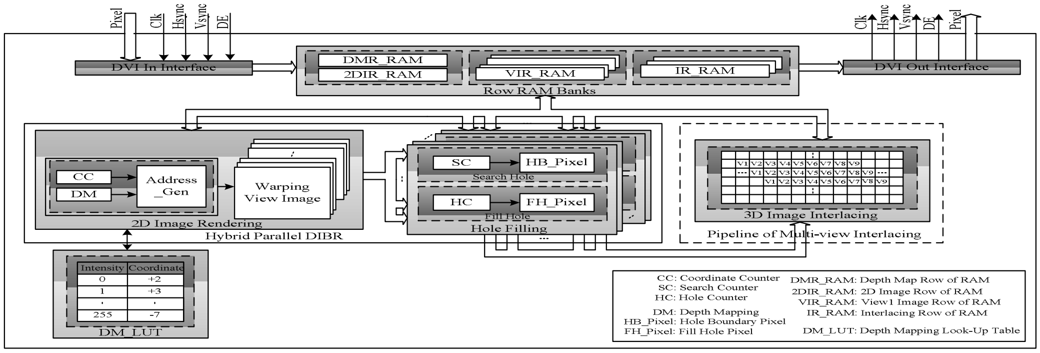
III. HARDWARE ARCHITECTURE

In 3D-TV applications, the multi-view rendering system is described in section II, and it has to be in real-time. Therefore, the hardware acceleration is implemented by using an FPGA. As shown in Fig. 4, the proposed architecture includes two major designs. One is the hybrid parallel of DIBR, and another is the pipeline of multi-view interlacing.

The entire flow is started from loading the original image and the depth map into a row of RAM by the DVI protocol. The hybrid parallel architecture of DIBR and hole-filling are computed continuously. The function of DIBR is to render multiple novel views by taking DM_LUT which is the disparity information transformed from the depth map. Furthermore, the holes due to the foreground and background occlusion in multiple views will be inpainted by the function of hole-filling. Afterwards, the function of multi-view interlacing depending on the data format required for the autostereoscopic display is optimized by using pipeline architecture. Finally, the DVI interface outputs the interlaced multi-view to show 3D on the autostereoscopic display.

A. Hybrid Parallel Architecture for DIBR

For autostereoscopic 3D displays, it requires multiple views in the different viewpoints, therefore, a larger memory is needed to store them. Also, it costs a lot of time to read and write pixels in memory. Consequently, in the proposed hybrid parallel architecture, the conventional DIBR flow path is re-arranged to improve the processing performance. The architecture design is based on the independent relation of multiple views, so that the original image and depth only have to be loaded once and used to render multiple views in parallel. As shown in Fig. 5, the other parallel architecture is for hole-filling to inpaint multiple views at the same time.



Intensity	Coordinate
0	+2
1	+3
·	·
255	-7

CC: Coordinate Counter
SC: Search Counter
HC: Hole Counter
DM: Depth Mapping
HB_Pixel: Hole Boundary Pixel
FH_Pixel: Fill Hole Pixel
DMR_RAM: Depth Map Row of RAM
2DIR_RAM: 2D Image Row of RAM
VIR_RAM: View1 Image Row of RAM
IR_RAM: Interlacing Row of RAM
DM_LUT: Depth Mapping Look-Up Table

Fig. 4. Overall hardware architecture of the proposed system.

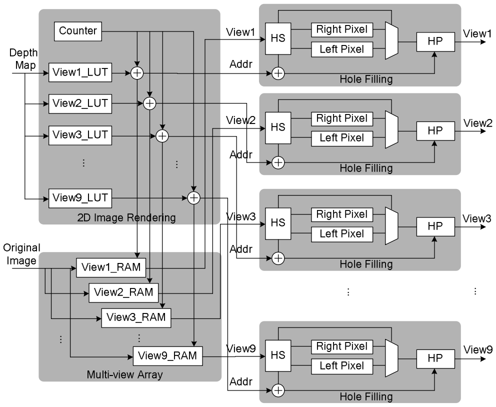


Fig. 5. Hardware architecture of hybrid parallel DIBR.

In the first module, the original image is loaded into a row of RAM. The depth map is transferred to the address of RAM by using the counter and the view LUT that calculates the horizontal coordination to represent the disparity of pixels. Next, the original image is scaled down based on the counter and then each view is rendered at the same time and stored in a row of RAM to reduce the memory, followed by a row of views in RAM is transferred into an array of multiple images. The multi-view array is allocated since the view number is determined in the first stage.

The parallel modules of holes filling receive the row pixels from the first module and then all the views are inpainted to fill the occlusion holes concurrently. In the holes filling process, it is the bottle neck to find out the occlusion holes in real time. Therefore, the function of hole-searching (HS) is designed to search the occlusion holes, meanwhile, the position of view is located, and then the right-side and left-side pixels are recorded. After positions of occlusion holes are determined, the function of hole-inpainting (HP) will fill the holes for all multiple views at the same time. Afterwards, the rendered and holes-filled multiple views are carried to the multi-view interlacing stages.

B. Pipeline Architecture for Multi-view Interlacing

For accelerating the multi-view interlacing architecture and fully using logic elements, the pipeline framework of interlacing is divided into three stages, scaling up (SU), subpixel interlacing (SI), and three row of 3D pixel (3R3P) respectively. The whole concept of pipeline interlacing is illustrated in Fig. 6.

The framework starts at processing cycle 1 where the SU stage is to compute subpixels for high resolution image. Continuing at the cycle 2, the scaled data, data-1, is processed by the SI stage and the SU stage calculates the next data group, data-2, followed by the results from the SI stage of Data-1 is converted into 3D pixel at processing cycle 3 and stage 3, the 3R3P stage.

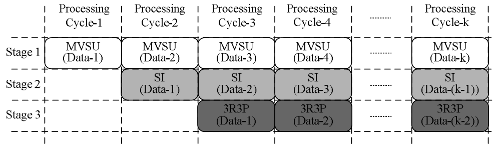


Fig. 6. Pipeline framework of interlacing.

The pipeline architecture of multi-view interlacing is shown in Fig. 7. In the SU stage, the right-side and left-side pixels are record to scaling up the resolution of views by interpolation. The new pixels of views are divided into RGB subpixels in the SI stage. Furthermore, the RGB subpixels of views are arranged to convert into the new pixels of 3D image according to the target autostereoscopic display. In the end, the new pixel of 3D is transferred into three different row of multi-view image in the 3R3P stage.

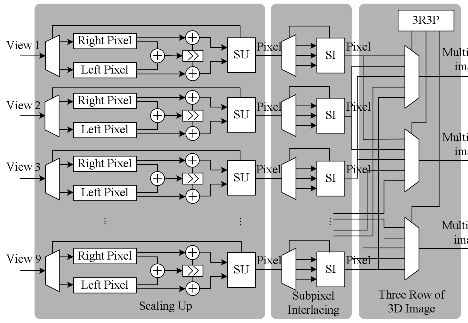


Fig. 7. Hardware architecture of pipeline interlacing.

IV. EXPERIMENTAL RESULTS

The proposed hardware architecture of multi-view rendering was implemented by using Verilog. The implementation was verified with RTL simulations using ModelSim ALTERA. The Verilog code was synthesized to an EP3C120F780C7 ALTERA Cyclone III FPGA using ALTERA Quartus. The resulting netlist was placed and routed to the same FPGA at 153.85MHz using Quartus II 9.0. Table 1 shows the result from the ALTERA synthesis tool of what specifications is used for a device. The number of logic elements was 3,079 (about 3% of the device), and the maximum frequency was 153.85 MHz. The number of on-chip block RAMs was 17 with 470 Kbs in total. The simulation results show that the proposed system can achieve real-time requirement for full

HD(1920x1080) stereo video with a 60fps at 148.5MHz. Hence the performance can still achieve real-time when the resolution required is increasing.

Table 1. FPGA device utilization specifications

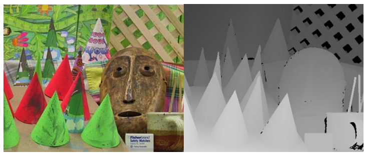
Altera Cyclone III EP3C120F780C7	
Logic Utilization	
Number of Logic Elements	3079
Number of 4 input LUTs	1397
Number of Registers	1284
Logic Distribution	
Number of occupied Logic elements	
Hybrid parallel DIBR	1301
Rendering	240
Hole-Filling	1061
Pipeline Interlacing	547
Number of occupied BRAM	
Memory bits	470,016
Number of multi-views:	**9**
Maximum frequency:	**157.85 MHz**
Processing capability:	**60 Full HD(1920x1080) frames/s**

Software implementation of the multi-view rendering was programmed by using Visual C++ 2008 in order to compare the performance and quality with the proposed hardware architecture. Intel core2 duo 3.0GHz with 2GB RAM was used. Besides, the parallel architecture based on CUDA by using GPU NVIDIA GeForce GTX 280 with 1GB RAM was also implemented to compare the performance. Fifteen thread blocks contains 192 threads were used in DIBR and 24.3K thread blocks includes 256 threads were used in multi-view interlacing. As shown in the Table 2, comparing the performance of software implementation, using the proposed hardware FPGA was faster than CPU by 88% frame rate (about 8.57x) and was faster than GPU by 22% frame rate (about 1.27x).

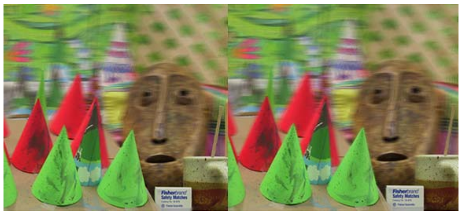
Table 2. The computation performance

		DIBR (per frame)	**Multi-view Interlacing (per frame)**	**Frame Rate**
Software	C	0.05 s	0.08 s	7 fps
	CUDA	0.011 s	0.008 s	47 fps
Hardware	FPGA	0.006 s	0.01 s	60 fps

The results of interlaced multi-view image are shown in Fig. 8, where Fig. 8(a) is the original image and depth map in the 640x360 resolution. The 1920x1080 resolution of interlaced multi-view with autostereoscopic 9-views was generated by the software and the hardware methods are shown in Fig. 8(b). The proposed hardware architecture can have the similar quality as the software implementation.



(a)



(b)

Fig. 8. Simulation of the proposed system,
(a) original image and depth map,
(b) software and hardware method.

V. CONCLUSIONS

In this paper, the novel hardware architecture is proposed for multi-view rendering in 3D-TV system. Comparing with the CPU-C and GPU-CUDA based rendering platform, the performance of the proposed architecture can achieve 60fps HD rendering which is 8.57x and 1.27x faster respectively. Based on the design of hybrid parallel DIBR and pipeline interlacing, the proposed architecture can greatly increase hardware capability to achieve real-time and reduce the cost. In the future, the IC chip integrating real-time depth estimation will be developed so that the 2D image/video/show can be experienced in any 3D applications.

REFERENCES

[1] A.Redert, M. Op de Beeck, C. Fehn, W.IJsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, I. Sexton, and P. Surman, “ATTEST – advanced three-dimensional television system techniques,” in *Proc. of 3DPVT’ 02*, pp.313-319, Jun. 2002.

[2] C. Fehn, “A 3D-TV approach using depth-image-based rendering (DIBR),” in *Proc. of Visualization, Image, and Image Processing*, pp. 482-487, Sep. 2003.

[3] C. Fehn, “Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV,” in *Proc. SPIE Conf. Stereoscopic Displays and Virtual Reality Systems XI*, vol. 5291, pp. 93-104, Jan. 2004.

[4] L. Zhang, J. Tam, and D. Wand, “Stereoscopic image generation based on depth images,” in *Proc. IEEE Conf. Image Processing*, pp. 2993-2996, Oct. 2004.

[5] Wan-Yu Chen, Yu-Lin Chang, Hsu-Kuang Chiu, Shao-Yi Chien, and Liang-Gee Chen, “Real-time depth image based rendering hardware accelerator for advanced three dimensional television system,” in *Proc. IEEE Conf. Multimedia and Expo*, pp. 2069-2072, Jul. 2006.

[6] R-P. M. Berretty, F. J. Peters, and G. T. G. Volleberg, “Real time rendering for multiview autostereoscopic displays,” in *Proc. SPIE*, vol. 6055, Jan. 2006.