

15.4 A 36fps SXGA 3D Display Processor with a Programmable 3D Graphics Rendering Engine

Seok-Hoon Kim, Jae-Sung Yoon, Chang-Hyo Yu, Donghyun Kim, Kyusik Chung, Han Shin Lim, HyunWook Park, Lee-Sup Kim

KAIST, Daejeon, Korea

There has been tremendous progress in 3D graphics hardware for various multimedia devices in recent years, but true 3D realism is not being provided to users due to insufficient computing power and the lack of depth-perception in 2D displays. A probable next generation display, 3D display technologies, realize stereoscopy [6] and make viewers perceive realistic 3D effects. However, only pre-processed static pictures and video data have been visualized in 3D displays to date, because complex image processing is required to synthesize output images from two view-images (left and right eye view-images) and a disparity-map [10]. An SXGA (1280×1024) 3D display processor is presented with a programmable 3D graphics rendering engine that can synthesize output images at interactive rate of 36fps. The synergetic coupling of the 3D display processor and the rendering engine enables users to experience true realism in real-time interactive 3D applications like games and GUIs.

Figure 15.4.1 shows the entire system architecture and operation. The system is divided into two parts: 3D graphics rendering engine (RE) and 3D image synthesis engine (SE). The RE performs 3D graphics-related pixel processing such as rasterization, texture mapping, etc., and a pixel shader in the RE provides various types of functionality. For a 3D display, the RE generates two different view-images and a depth-map in two passes. The SE generates 7 intermediate view-images for in-between viewpoints from the outputs of the RE and spatially multiplexes 9 view-images including two original view-images into an output image for a target 3D display, a 9-view slanted lenticular display [7]. 3D graphics applications inherently produce a depth-map in the RE, and the depth-map is linearly converted to a disparity-map while static picture and video data require the complex extraction of the disparity-map. The SE is designed to use internal memory instead of external memory for storing intermediates so that output images can be synthesized in real-time.

Figure 15.4.2 shows the block diagram of the RE. The RE consists of a rasterizer, pixel shader, and raster operation unit (ROP). The rasterizer generates pixels with triangle data fetched from an external geometry engine (GE). The pixel shader is a main block of the RE and processes state-of-the-art rendering effects by supporting the latest 3D graphics API, Pixel Shader 3.0 [3] and OpenGL ES 2.0 [4]. The enhanced features such as 32b floating point arithmetic and dynamic branching remarkably improve the functionality compared with the previous works on RE [1,11], which support only fixed function pipelines. The texture unit of the pixel shader supports up to tri-linear filtering, which needs to fetch eight texels concurrently. So it requires the most intensive memory accesses in an overall 3D graphics system. A texture cache is adopted to alleviate the frequent memory accesses and optimized based on the access pattern of the texture unit. In the case of tri-linear filtering, two 2×2 texel blocks are fetched from adjacent texture levels. The cache is composed of four banks, and texels of a 2×2 texel block are distributed to different banks. Since a bank is a 2-port SRAM, the cache provides 8 texels simultaneously for a maximum performance of 400Mtexels/s. The 8KB cache is composed of four 2KB banks. The line size is selected as 64B, and a 2-way set associative structure is used for the best efficiency [9]. Furthermore, the pixel shader adopts a multi-threaded 4-way SIMD architecture to fully utilize data parallelism [2].

The synthesis process for the 3D display is divided into view interpolation [5] and multiplexing [7,8]. View interpolation creates 7 intermediate view-images from two view-images and a depth-map, and multiplexing merges all 9 view-images to generate an output image. Normally, 9 view-images are stored in an external memory after the view interpolation, and they are loaded and spatially multiplexed into an output image during the multiplexing. Since these processes require the memory accesses to store and to load all the view-images, output images cannot be synthesized in real-time due to the frequent accesses to external memory.

To solve this problem, the SE processes the interpolation and the multiplexing operation line by line. Since the view interpolation generates intermediate view-images by moving pixels of a left eye view-image or a right eye view-image horizontally, the process can be executed line by line. The size of a line is only 0.3% of a view-image, so each line data can be stored in internal memory. Therefore, the view interpolation generates 9 view-lines instead of 9 view-images and stores them in line-buffers which are internal memory. The multiplexing unit loads and multiplexes these lines into three lines of an output image. This technique can synthesize output images at an interactive-rate by reducing 73.3% of external memory accesses but requires internal memory for storing the 9 view-lines.

In order to reduce the internal memory size, a combined architecture is developed that processes view interpolation and multiplexing simultaneously, as shown in Fig. 15.4.3. The architecture removes the line-buffers of 7 intermediate view-lines. In the view interpolation process, a pixel loaded from the left line-buffer is moved to the left as far as disparity: the horizontal offset of the same pixel due to the different view position. During this process, some areas cannot be filled due to the occluded area or the boundary. This is detected when the disparity of the moved-pixel is less than that of the left pixel in the left line-buffer. It is removed by taking the proper pixels from the right line-buffer. The interpolated pixel is not stored in a line-buffer but directly multiplexed, which reduces the internal memory size to 49.5%, as shown in Fig. 15.4.4. In the multiplexing process, a 3×3 pixel box where the interpolated pixel can be mapped is calculated, and the linear mapping equation [7] evaluates view-numbers for the sub-pixels in the box. If the evaluated view-number matches the current view-number generated by the view position counter, the sub-pixels of the interpolated pixel are filled to the sub-pixel position in the box. An output image is fully synthesized by the repetition of these processes.

To exploit data parallelism, the SE consists of three independent sub-pixel units, and one of them is shown in Fig. 15.4.5. Since multiplexing is directly performed after view interpolation, a sub-pixel unit does not contain 9 line-buffers but only the source and the output line-buffers. Although the equation of disparity and view-number is calculated by floating-point, the error of changing floating-point to fixed-point is negligible. It enables all operators in a sub-pixel unit to be designed as fixed-point, which can save area and power.

The processor integrates approximately 1.7M gates and 40KB SRAM in 5.0×5.0mm² using a 0.18μm CMOS technology and runs at 50MHz. At this frequency, the maximum 3D image synthesis rate is 36fps. A detailed specification is shown in Fig. 15.4.6, and the chip micrograph is shown in Fig. 15.4.7.

Acknowledgements:

This work was supported by the university IT research center program and ETRI, Korea.

References:

- [1] Donghyun Kim, et al., "An SoC with 1.3Gtexels/s 3D Graphics Full Pipeline Engine for Consumer Applications," *ISSCC Dig. Tech. Papers*, pp.190-191, Feb., 2005.
- [2] Chang-Hyo Yu, et al., "A 120Mvertices/s Multi-threaded VLIW Vertex Processor for Mobile Multimedia Applications," *ISSCC Dig. Tech. Papers*, pp. 408-409, Feb., 2006.
- [3] Shader model 3.0, <http://www.microsoft.com>.
- [4] OpenGL ES 2.0, <http://www.khronos.org>.
- [5] E. Chen, et al., "View Interpolation for Image Synthesis," *SIGGRAPH*, pp.279-288, Aug., 1993.
- [6] D. Scharstein, "View Synthesis Using Stereo Vision," vol. 1583 of LNCS, Springer-Verlag, 1999.
- [7] C. Van Berkel, "Image Preparation for 3D-LCD," *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, vol. 3639, pp. 84-91, May 1999.
- [8] Yun-Gu Lee, et al., "Reduction of the Distortion Due to Non-Ideal Lens Alignment in Lenticular 3D Displays," *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, vol. 5664, pp. 506-516, Jan., 2005.
- [9] Z. Hakura, and A. Gupta, "The Design and Analysis of a Cache Architecture for Texture Mapping," *Proc. 24th Int. Symp. on Computer Architecture*, vol. 25, pp. 108-120, May 1997.
- [10] Goshtasby, "A. Stereo Correspondence by Selective Search," *Proc. Japan Computer Vision Conf.*, pp. 1-10, July, 1989.
- [11] Masatoshi Imai, et al., "A 109.5mW 1.2V 600M texels/s 3-D Graphics Engine," *ISSCC Dig. Tech. Papers*, pp. 332-333, Feb., 2004.

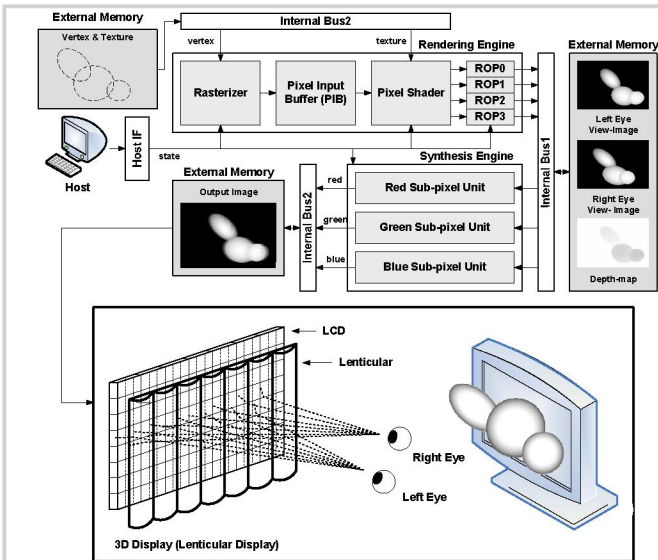


Figure 15.4.1: Block diagram of the proposed processor.

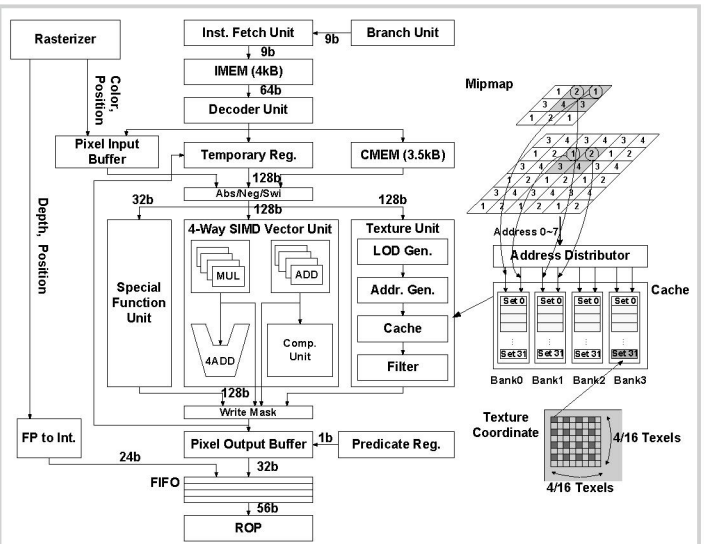


Figure 15.4.2: Block diagram of RE.

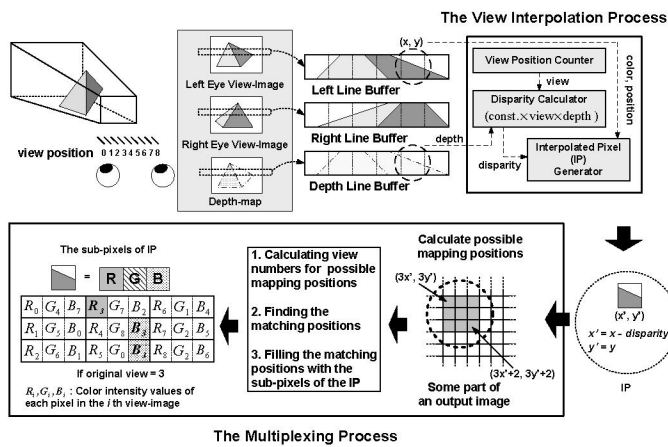


Figure 15.4.3: View interpolation and multiplexing techniques.

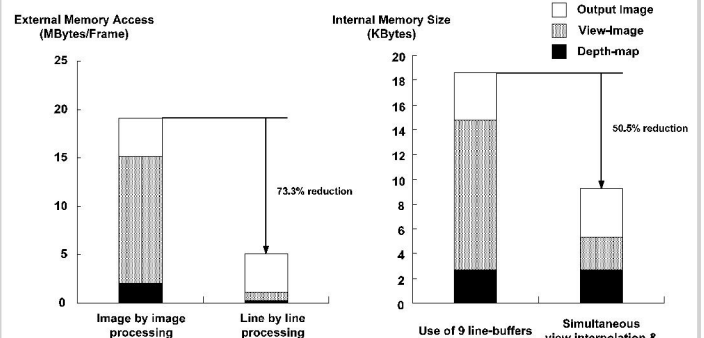


Figure 15.4.4: Results of the proposed techniques.

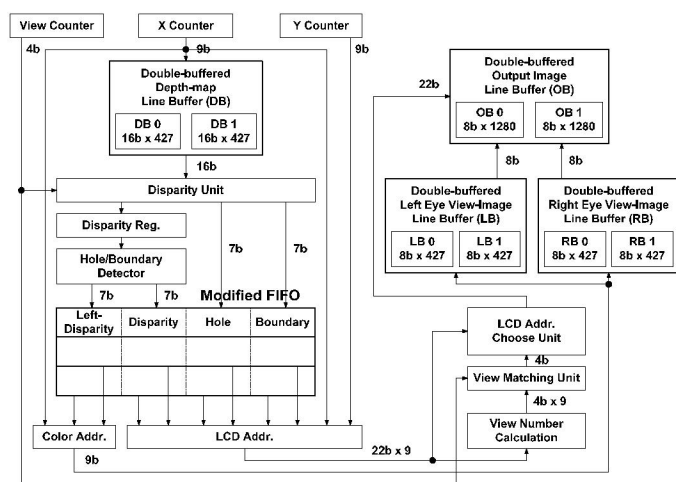


Figure 15.4.5: Block diagram of a sub-pixel unit.

Target 3D Display	9-view Slanted Lenticular Display
Process Technology	0.18μm CMOS
Power Supply	1.8V
Chip Size	5x5mm ²
Gate Counts	1,743,659
Frequency of Engine	50MHz
3D Image Synthesis Rate	Max 36fps
Texturing Performance	400Mtexels/s
Power Consumption	379mW
3D Graphics Function	All the function in OpenGL ES specification2.0 All the function in PS specification3.0

Figure 15.4.6: Chip specification and characteristics.

Continued on Page 602

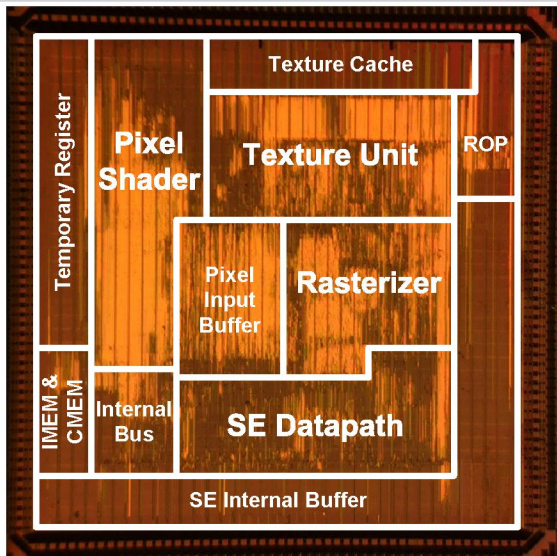


Figure 15.4.7: Chip micrograph.