Date : 11/26/2015

Oregon State University

**Research Paper Summary**

**Title**  **Accelerating Vector Graphics Rendering using**

**the Graphic Hardware Pipeline**

**Student Name Yongbin Gu**

**Student ID 932694954**

**Class Instructor Mike Bailey**

**The summary of the paper titled “Accelerating Vector Graphics Rendering using the Graphics Hardware Pipeline”**

1. **General information of the paper**

The paper named *Accelerating Vector Graphics Rendering using the Graphics Hardware Pipeline* mainly explains that they use graphic hardware (GPU) to accelerate the Adobe illustrator instead of using CPU to do so. Typically, the illustrator artwork is authored in the CMYK color space. However, the conventional GPU blending mode cannot blend appropriately for CMYK. In addition, the CMYK rendering needs 5 framebuffer components. These problems will reduce the quality of the rendering if they are not dealt properly. In their research process, they realize repurposing of the RGBA render buffers in GPU for NChannel color space rendering of CMYK process colors and additional spot colors with full blending support. And they also develop the GPU algorithms to deal with the isolated and non-isolated transparency properly; they employ the GPU hardware tessellation to tessellate gradient meshes to shade paths; to support the arbitrary path clipping, pattern shading, and knockout groups, they harness “stencil, then cover” path rendering. As display technology developed, the traditional way (CPU-based) rendering method seems not to satisfy designers’ needs. Poor rendering performance probably frustrates the designers’ creativity. Therefore, these authors decide to develop GPU-accelerate illustrator for improving the rendering speed and interactivity, which could help improve the users’ productivity.

1. **The information of authors**

The first author of this paper is Vineet Batra, who is the principal Computer Scientist at Adobe Systems. He got his bachelor of Technology, majoring in Computer Science in 1998, and has worked in Adobe Systems since 1998.

The following author’s name is Mark J. Kilgard, a graphics software engineer working at Nvidia. He graduated from Rice University. And he has published two books: *OpenGL for the X Window System* (1996), and *The Cg Tutorial* (2003). Prior to joining Nvidia, he worked at Compaq and Silicon Graphics. While at Silicon Graphics, he developed the OpenGL Utility Toolkit, better known as GLUT, to make it very easy to write OpenGL-based 3D examples and demos. The primary reason for this was the lack of a windowing and input API with OpenGL using GLX. Mark has published many OpenGL technical sample programs during the pushback against Microsoft’s FUD against the API. The GLUT toolkit developed by Mark allowed the samples to run in different platforms, such as the Windows system and the SGI workstation. In Nvdia, he also makes a great contribution. He helped design important parts of 3D graphic APIs. He has written the key whitepapers, including “Cg in Two Pages”. He is also the lead author of the NV\_path\_rendering extension – a GPU-accelerated method for rendering vector graphics.

The third author is Harish Kumar. He is also a computer scientist at Abobe, India. He graduate from National Institute of Technology Kurukshetra in 2011. And then he joined the Winshuttle Softwares Pvt. Ltd as a software Engineer. After 2013, he is employed by Adobe.

The last author of this paper named Tristan Lorach, a faculty of Nvdia. He got his Bachelor of Science in EFREI, France, and continued his master degree in University Paris 7 &8 in 1994 and 1995. When he joined the Nvdia, he mainly focused on the technology Relations for Games (performance analysis, new GPU/Driver features and how to take advantage of them, new visual techniques and how to implement them into a game). And then he became a senior Engineer in GPU workstation. His technical consulting area mainly include: engage technical discussions with partners in CAD/DCC; help them to solve issues and improve their use of the GPU; assist partners in taking the right technical decisions for their product, according to their strategy; R&D: papers; presentations; collaborate with NVIDIA architects, driver engineers and partners to implement new ideas and concepts around real-time rendering

1. **The conclusion of the paper**

To realize the GPU-accelerated rendering, the research mainly include: Introducing a new method to deal with CMYK colors using existing RGB framebuffer; developing proper algorithm to composite both isolated and non-isolated transparency groups; employing the GPU hardware tessellation to tessellate the gradient mesh; harnessing the stencil, then cover path-rendering to deal with shading problems.After being tested by different types of benchmark, their research on GPU-accelerated rendering has been proved effective. The results of benchmarks showed that their work has a significant speedup, particular at 4K resolution, compared with the CPU-based one. They also introduce the novel techniques for supporting CMYK color space rendering on existing GPUs, proper transparency group support including non-isolated groups and knock-out, and mapping PDF’s gradient mesh shading to GPU tessellation. Their work has also explored ways to support a broader range of GPUs. They put forward to the standardization of NV\_path\_rendering will make this much easier. They also make a comparison between their own work and the recent work. These works are both used GPU to do the rendering works. However, the performance of this paper’s work seems better than the Ganacim et al. work. They attribute this to forcing use of Nvdia’s dual core driver and Illustrator’s tuned OpenGL usage and scene traversal.

1. **The insight of the paper**

Through reading this paper, I found that vector graphic rendering is usually completed in the CPU rather than in the GPU in previous work. Before reading this paper, I always think that the graphic processing things are all processed in the GPU. However, this paper explained that there is still graphic work being done by the CPU. Due to their different roles in computer running, CPU and GPU actually have big difference in the graphic processing. From my perspective, the researchers and the engineers should make their efforts to let GPU do increasing graphic processing things in the future.

And also, I always hold the view that the pictures are all stored in the RGB color mode. Nevertheless, this paper showed that there is a quite number of graphs use CMYK, which is also known as the subtractive color. The subtractive color can best facilitate high-quality color printing. And before being displayed on the screen, the colors of CMYK mode will be transformed into the RGB mode.

As for the transparency, I only know a superficial knowledge of the transparency. This paper describes that the transparency groups are divided into 2 groups: Isolated and non-isolated groups. For an isolated group, the backdrop is fully transparent so it has nether color, shape, nor opacity. For a non-isolated group, the backdrop will be inherited from whatever has already been rendered prior in the objects “beneath” the group. So when the researchers use the GPU rendering, they need to deal with the isolated and non-isolated group separately.

1. **The flaws of the paper**

The first shortcoming is the memory requirement when they use RGB buffer to store the CMYK colors. The worst case is that a document in CMYK color mode without spot colors requires double the framebuffer memory as an RGBA document. This paper does not have any better solutions. They only say that the GPU hardware innovation could provide less wasteful memory organizations in future GPUs.

After reading this paper, I think the comparison between the CPU and GPU is not convincing. This paper just simply gives the configuration of their computer: Xeon E3-

1240 V2 CPU @ 3.40GHz (4 cores), 8 GB RAM, and NVIDIA GeForce GTX 780 Ti GPU. They did not explain why they choose Xeon E3 and GTX 780 Ti. I think GPU and CPU are totally different types of processor, they have different utilizations. Let us assume that if they compare the Xeon E3 with the GPU from ten years ago, the result probably turns out that the CPU-based rendering is better than the GPU-based rendering. Therefore, I think the gain of the speedup cannot be used to judge the improvement of the performance. They should build a standard that explains how to make the comparison.

In addition, they made a comparison with recent work. The performance of their work is better than the recent work. They give the reason. I think before they give these reasons, they should verify these whether they are accurate by the experiments.

1. **The outlook of the next-step work**

This research mainly use the GPU to accelerate the adobe illustrator. However, most user interface assumes re-rendering the scene is slow and expensive. Some user interface also avoid rendering the complete scene by encouraging user to isolate just a portion of the scene. Editing is also not *what-you-see-is-what-you-get* interaction mode due to avoidance of the slow rendering performance of the complete scene. So the next-step work of this research should update the user interface to realize a more powerful intuitive, and fluid user interface for vector graphics editing. Many other research efforts related to vector graphics editing are also employing the GPU-acceleration. The research of this paper also make possible bringing other research achievements into the GPU, and supporting by the illustrator. As mentioned in the section 5, the framebuffer memory usage is very substantial. The future work of this research must focus on the reduction of the memory usage without decreasing the rasterization quality of path rendering. The adoption of the Nvdia’s *NV\_Framebuffer\_mixed\_samples* OpenGL extention could have fewer color samples. And also, the improvement of the GPU architecture can also help them reduce the memory bandwidth, memory footprint, and greatly simplify their work of dealing with CMYK rendering.