Real-time High Dynamic Range Compression on A Mobile GPU

Anonymous Authours
Anonymous Affiliation

Abstract

High Dynamic Range (HDR) compression allows visualizing various scenes with wide dynamic range images. We implement a HDR compression algorithm on a mobile GPU in order to enable HDR movie playback in mobile devices.

1. HDR compression algorithm

Our system captures 960x640, 15fps movie in RAW format (from 10 to 16 bits) and applies HDR compression for each frame on GPU in order to display 8bit sRGB movie (see Figure 1). Figure 2 shows a example of our WDR compression.

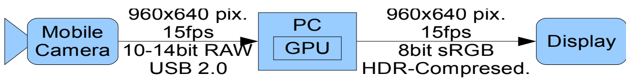

Figure 1: System Overview

Figure 2: (top) A input image (14 bit) and (bottom) the result of WDR compression (8bit).

Our HDR compression algorithm is based on Retinex theory [1], which is often used in major HDR compression algorithms. In order to estimate illumination component of the scene, our system employs illumination filtering, a modified version of bilateral filtering [2].

Despite that HDR compression and also bilateral filtering are known as time consuming algorithms, our system enables real-time HDR compression on a mobile class GPU, i.e. the GPU integrated with Intel Express 4 chipset.

2. HDR compression on GPU

Figure 3 shows the shaders and buffer objects used in our GLSL 1.2 implementation of HDR compression. The first shader converts input packed data into 16bit float texture, next two shaders employ illumination filtering and the last shader performs Retinex transform and other essential processing in imaging pipeline, i.e. demosaicking and gamma correction.

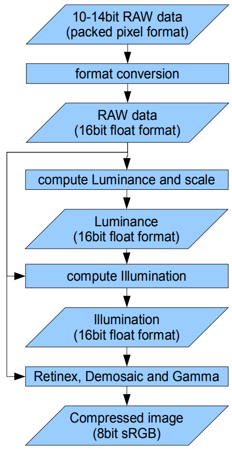

Figure 3: Shader Programs and Buffer Objects

3. Results and discussion

We implement HDR compression algorithms as GLSL 1.2 shader programs. The GPU used in Intel Express 4 chipset is able to run our algorithm at 15 fps. In our experiment, frame rate is limited by USB2.0 interface, which is used to capture RAW image data.

Our GPU implementation is more than three times faster than the CPU implementation on Core 2 Duo 1.2GHz processor, thanks to that the most time consuming instructions in Retinex transform and bilateral filtering, i.e. log, exp, float-division, are all supported by fragment shaders.

HDR compression on GPU allows us to view HDR scenes while preserving contrast and to control HDR effects in real-time so that users can give artistic effects on HDR movie by changing compression parameters and algorithms on the fly.

4. Conclusion

We implement a HDR compression algorithm on GPU and shows that a mobile class GPU is able to perform HDR compression in real-time. It means that we can utilize the dynamic range of mobile cameras with GPUs. We believe that GPU will be a key component to improve movie quality of mobile devices in near future.

References

[1] Edwin H. Land and John J. McCann, "Lightness and Retinex theory," Journal of the Optical Society of America, Vol 61, No. 1, 1971.

[2] Carlo Tomasi, Roberto Manduchi, "Bilateral filtering for gray and color images," Proceedings of the ICCV, 1998.