Parallel Architecture & Distributed Architecture (18CS73)

Atreya Bain (1RV18CS030)
Chirag Bapat (1RV18CS048)

Submitted To: Dr. Minal Moharir
Associate Professor

Self Study Assignment

January 13, 2022

HDR Tonemapping

Problem Statement

Implement HDR Tonemapping to display HDR images, using the Luminance remapping technique on a GPU using CUDA.

Introduction - Continuation

- HDR Images contain high precision and high dynamic range image pixel data.
- HDR displays can show these by changing the brightness of that section of the screen.
- Showing HDR Image on a standard image? Eg. A Sunrise photo
  - Linearly scale - Bright parts of the image will make the image mostly dark and unviewable.
  - Tonemap - Scale the brightness so that the image is still viewable.

Introduction

Tone mapping is a technique used in image processing and computer graphics to map one set of colors to another to approximate the appearance of high-dynamic-range images in a medium that has a more limited dynamic range.

Methodology

- There are two ways to bring about tonemapping:
  1. Global operators: Non-linear functions based on the luminance and other global variables of the image.
  2. Local operators: the parameters of the non-linear function change in each pixel, according to features extracted from the surrounding parameters. Those algorithms are more complicated; they can show artifacts, but they can (if used correctly) provide the best performance, since human vision is mainly sensitive to local contrast.

Methodology - Luminance Model

A good way to begin with

- The YC_bC_r color space is a popular color space used in TVs, as it splits out a luminance component (the black and white) and 2 color components.
  - Traditionally, the color TVs would use this as a way of displaying color, and the B&W TVs would simply display only the luminance component, saving on hardware and bandwidth costs.
- The Y component can be scaled here accordingly to change the brightness of that region of the image.

Methodology (Continued)

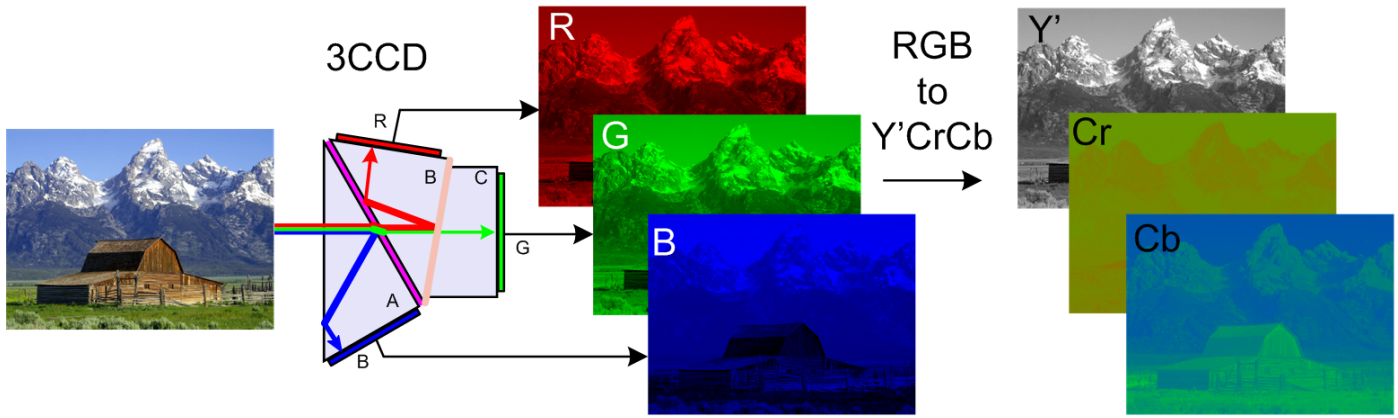

Figure: RGB to YC_bC_r

Methodology

Here is the general description of the procedure:

1. Convert the image the the YC_bC_r space.

$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1146 & -0.3854 & 0.5 \\ 0.5 & -0.4542 & -0.0458 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{1}$$

2. Construct a histogram of brightness values, constructing a CDF of it.
3. Taking the log of the luminance, scale it according to the CDF.
4. Normalize the CDF, scaling the values of luminance.
5. Process and convert back to RGB model as required.

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5748 \\ 1 & -0.1873 & -0.4681 \\ 1 & 1.8556 & 0 \end{bmatrix} \begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} \tag{2}$$

CUDA - Parallelizing

- CUDA (Compute Unified Device Architecture) is a parallel computing platform and programming model created by NVIDIA and implemented by the graphics processing units (GPUs) that they produce.
- The CUDA platform is accessible to software developers through CUDA accelerated libraries, compiler directives and extensions to industry-standard programming languages, including C, C++ and Fortran.
- CUDA gives developers access to the virtual instruction set and memory of the parallel computational elements in CUDA GPUs. Unlike CPUs, GPUs have a parallel throughput architecture that emphasizes executing many concurrent threads slowly, rather than executing a single thread very quickly.
- This approach of solving general-purpose problems on GPUs is known as GPGPU (General Purpose Graphics Processing Unit)

CUDA - Issues for parallelizability

- No recursive functions
- Limited CPU $<->$ GPU bandwidth (Has to pass through PCI Bus
- CUDA is a proprietary Nvidia product
- Applications which depend upon previous computed data cannot be implemented trivially on CUDA.

Why Parallelize?

- Tone-mapping is done for images, where there are millions of pixels present.
- Processing HDR images and videos can become a repetitive and taxing job for the CPU, and this is best offloaded to the GPU.
- This is the perfect opportunity for GPU Acceleration.

Results

- Prefix sum calculation is a bottleneck - This can be done efficiently with the Hillis Steele prefix calculation algorithm.
- Serial and parallel implementation can be proven to be similar - Use differences to measure.
- 8-10x improvements in performance over naive implementations

Results

P	Serial	Parallel	Acceleration factor
GoldenGate	16.509000	1.228768	13.435409
Memorial	1.235000	0.200224	6.168092
Rec709(Flower)	3.198000	0.333056	9.601989
je_gray_park_4k	114.597000	10.039872	11.414189

References

- Hunt, R. 2004. "The Reproduction of Colour in Photography, Printing and Television: 6th Edition." John Wiley & Sons.
- Charles Poynton, Digital Video and HDTV, Chapter 24, pp. 291–292, Morgan Kaufmann, 2003.
- "High dynamic range television for production and international programme exchange". www.itu.int. Retrieved 2021-01-16.
- Florian Kainz; Rod Bogart; Piotr Stanczyk; Peter Hillman (5 November 2013). "Technical Introduction to OpenEXR" (PDF). Industrial Light & Magic. Accessed 2015-11-09.