RV College of Engineering®

Go, change the world

PADP(18CS73)

Department of CSE

Contrast Enhancement Using Histogram Equalization Phase 1

Sanjana Reddy - 1RV18CS145

Introduction

- In our daily life, if the images are captured in back-lighting and night-time cases, the objects in digital images will be **very dark and have poor contrast**.
- To improve the visual quality, **colour image enhancement methods** need to be developed to enhance the contrast.
- **Histogram Equalization (HE)** is a simple and effective technique for image contrast enhancement. HE is a computer image processing technique used to improve contrast in images.
- It accomplishes this by effectively **spreading out the most frequent intensity values**, i.e. stretching out the intensity range of the image.
- This method usually increases the global contrast of images when its usable data is represented by close contrast values.
- This allows for areas of lower local contrast to gain a higher contrast.
- However, if the image is first converted to another colour space, like **HSL (hue, saturation, lightness)/ YUV (Luminance,Chrominance)**, then the algorithm can be applied to the luminance or value channel without resulting in changes to the hue and saturation of the image.

Literature Survey

A Comprehensive Review of Image Enhancement Techniques.

- According to the authors, the **point processing methods** are the most primitive, yet essential image processing operations and are used primarily for contrast enhancement.
- **Image Negative** is suited for enhancing white detail embedded in dark regions and has applications in medical imaging.
- For **a dark image**, an expansion of grey levels is accomplished using a **power-law transformation** with a fractional exponent.
- **Log Transformation** is useful for enhancing details in the darker regions of the image at the expense of detail in the brighter regions the higher-level values.
- The **histogram** of an image (i.e., a plot of the grey level frequencies) provides valuable information regarding the contrast of an image.

Literature Survey

Implementation of Image Enhancement Algorithms and Recursive Ray Tracing using CUDA

- Implement various image enhancement algorithms.
- They found an increase in performance with an increase in the number of threads in applications that involve high **inter-thread communication**.
- They also noted that while implementing **image filters**, the product of the number of blocks and number of thread/blocks should be equal to the number of elements to be processed.
- Since recursive ray tracing is a highly computation requiring task, they were able to achieve better results by **parallelizing it via a GPU**.

Literature Survey

Image contrast and colour enhancement using adaptive gamma correction and histogram equalization

- Propose an effective image contrast enhancement method called an **Adaptive Gamma Correction with Weighted Histogram Distribution** (AGCWHD) method to **improve contrast** while preserving natural colour and richer details in images.
- In the proposed method, the new adaptive gamma correction method is implemented to enhance the contrast, while a **weighted histogram distribution** is employed for natural colour and detail preservation.
- Experiment results showed that the proposed technique performs well for a wide variety of **low-quality images** and achieves the better or comparable objective and subjective comparisons with the state-of-the-art methods.

Literature Survey

A Dynamic Histogram Equalization for Image Contrast Enhancement

- Propose a smart contrast enhancement technique based on conventional histogram equalization (HE) algorithm.
- This **dynamic histogram equalization** (DHE) technique takes control over the effect of traditional HE so that it performs the enhancement of an image **without making any loss of details in it**.
- DHE partitions the image histogram based on **local minima** and assigns specific grey level ranges for each partition before equalizing them separately.
- These partitions further go through a **repartitioning test** to ensure the absence of any dominating portions.

Methodology

1. **Installing NVIDIA and other dependencies**

   A driver is software that enables the operating system to communicate with hardware or a device.

2. **C Compiler**

   NVIDIA CUDA Compiler (NVCC) is a proprietary compiler by NVIDIA intended for use with CUDA. CUDA code runs on both the CPU and GPU. NVCC separates these two parts and sends host code (the part of code which will be run on the CPU) to a C compiler like GCC or Intel C++ Compiler (ICC) and sends the device code (the part which will run on the GPU) to the GPU. The device code is further compiled by NVCC.

3. **Installing CUDA and CUDA toolkit**

   CUDA is a parallel computing platform and programming model developed by NVIDIA for general computing on graphical processing units (GPUs). With CUDA, developers can dramatically speed up computing applications by harnessing the power of GPUs.

Methodology

4. **Linking Compiler to CUDA**

   Managing complexity in large programs requires breaking them down into components that are responsible for small, well-defined portions of the overall program. Separate compilation is an integral part of the C and C++ programming languages which allows portions of a program to be compiled into separate objects and then linked together to form an executable or library.

5. **CMake**

   In software development, CMake is cross-platform free and open-source software for build automation, testing, packaging and installation of software by using a compiler-independent method. CMake is not a build system but rather it generates another system's build files.

6. **Make Command**

   The Linux make command is used to build and maintain groups of programs and files from the source code. In Linux, it is one of the most frequently used commands by the developers. It assists developers to install and compile many utilities from the terminal. Further, it handles the compilation process of the sizable projects. It saves the compilation time.

7. **Running the executable file**

   The executable file is executed and the input files are provided as input.

Implementation

1. NVIDIA GPU

The CUDA NVIDIA GPU is used.

2. CMakeLists.txt

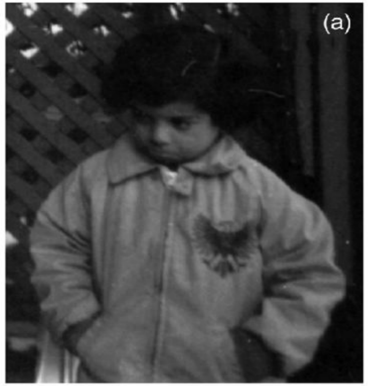
CMakeLists.txt file contains a set of directives and instructions describing the project's source files and targets (executable, library, or both).

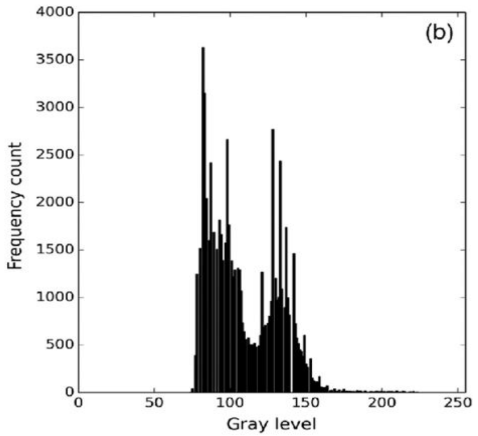
The project name and the required packages are imported. Paths for required directories are set. The files with the C++ and CUDA code are set. Executables are built from the files which contain the CUDA code.

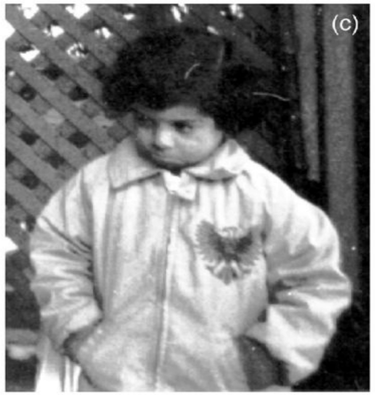
Implementation

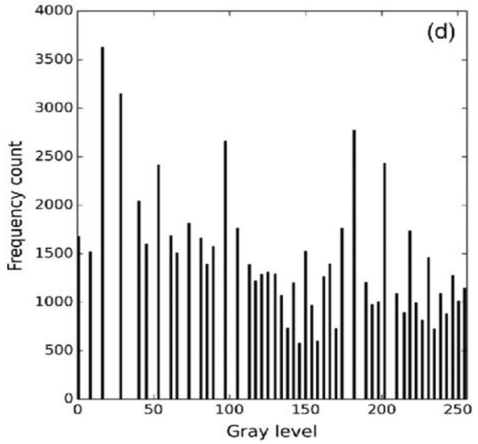
3. Histogram equalization

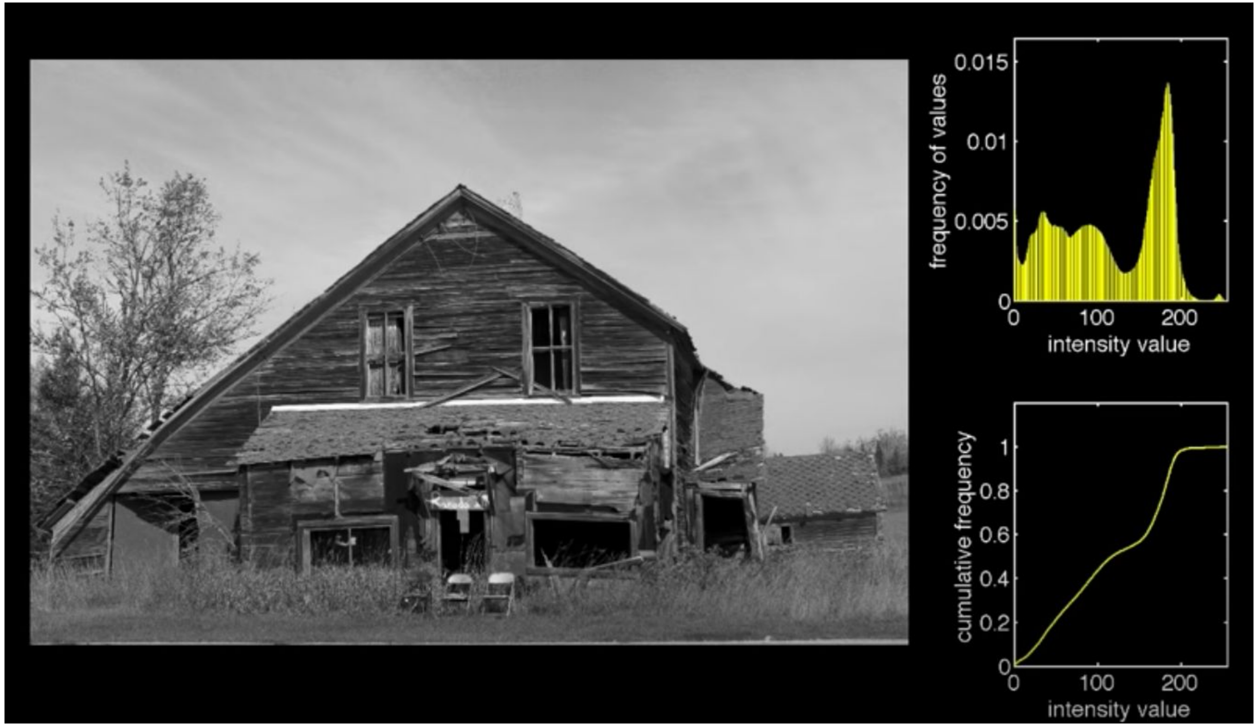
- This method usually increases the global contrast of many images, especially when the image is represented by a narrow range of intensity values.
- This allows for areas of lower local contrast to gain a higher contrast.
- Histogram equalization accomplishes this by effectively spreading out the highly populated intensity values which degrade the image contrast.
- The method is useful in images with backgrounds and foregrounds that are both bright or both dark.

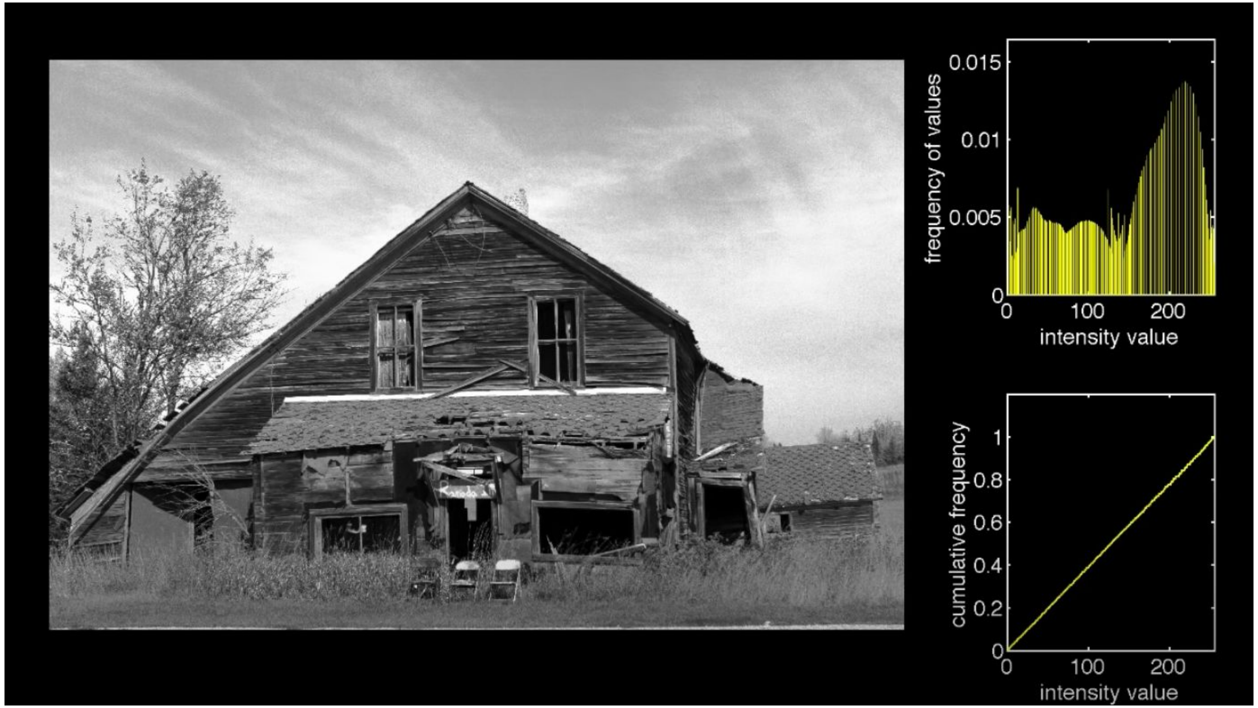

(a)


(b)
Frequency count
Gray level
0
500
1000
1500
2000
2500
3000
3500
4000
0
50
100
150
200
250


(c)

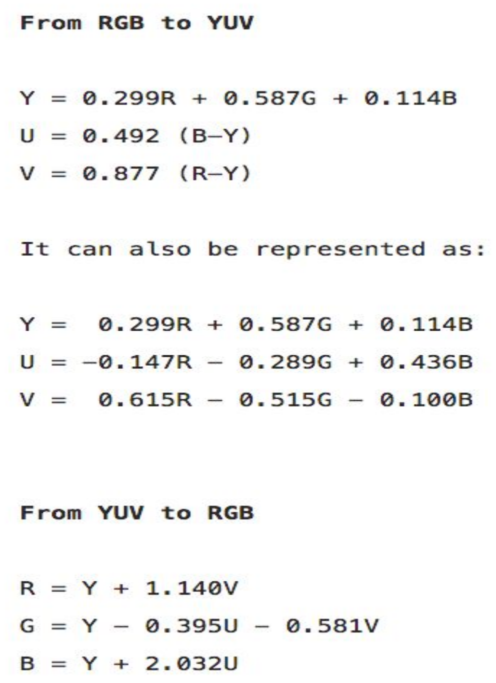

(d)
Frequency count
Gray level
0
500
1000
1500
2000
2500
3000
3500
4000
0
50
100
150
200
250


frequency of values
0.015
0.01
0.005
0
0
100
200
intensity value
cumulative frequency
1
0.8
0.6
0.4
0.2
0
0
100
200
intensity value



$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - 1} \times (L-1)\right)$$

RGB & YUV conversions



From RGB to YUV

$Y = 0.299R + 0.587G + 0.114B$

$U = 0.492\ (B-Y)$

$V = 0.877\ (R-Y)$

It can also be represented as:

$Y = 0.299R + 0.587G + 0.114B$

$U = -0.147R - 0.289G + 0.436B$

$V = 0.615R - 0.515G - 0.100B$

From YUV to RGB

$R = Y + 1.140V$

$G = Y - 0.395U - 0.581V$

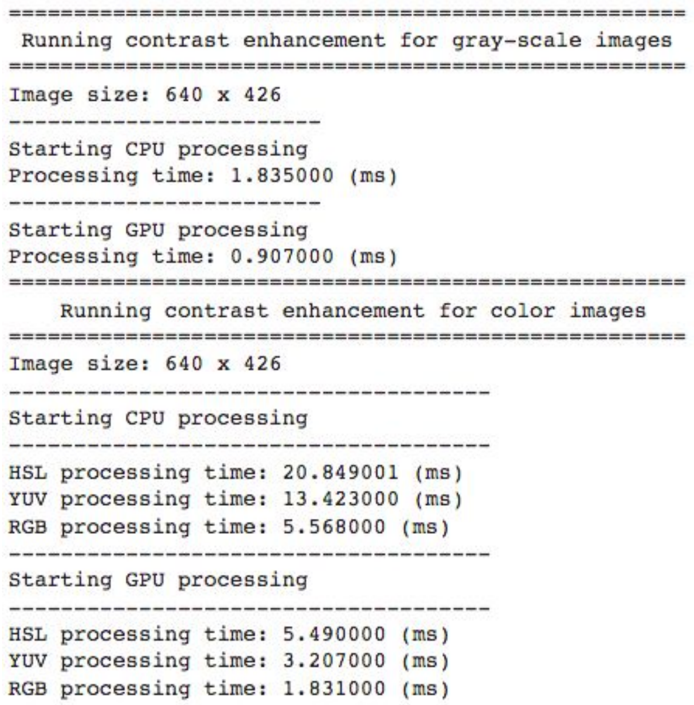
$B = Y + 2.032U$

RGB & HSL conversions

$$H = \begin{cases} 0 & \text{if} \quad \max = \min \\ 60^0 \times \dfrac{G - B}{\max - \min} & \text{if} \quad \max = R \\ 60^0 \times \dfrac{B - R}{\max - \min} + 120^0 & \text{if} \quad \max = G \\ 60^0 \times \dfrac{R - G}{\max - \min} + 240^0 & \text{if} \quad \max = B \end{cases} , \qquad L = \frac{1}{2}(\max + \min)$$

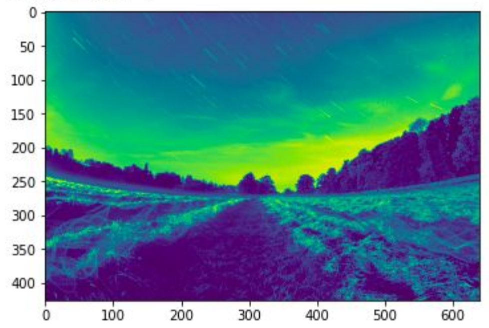
$$S = \begin{cases} 0 & \text{if} \quad \max = \min \\ \dfrac{\max - \min}{\max + \min} & \text{if} \quad L \le \dfrac{1}{2} \\ \dfrac{\max - \min}{2 - (\max + \min)} & \text{if} \quad L > \dfrac{1}{2} \end{cases}$$

Results



```
=====================================================
 Running contrast enhancement for gray-scale images
=====================================================
Image size: 640 x 426
------------------------
Starting CPU processing
Processing time: 1.835000 (ms)
------------------------
Starting GPU processing
Processing time: 0.907000 (ms)
=====================================================
   Running contrast enhancement for color images
=====================================================
Image size: 640 x 426
-------------------------------------
Starting CPU processing
-------------------------------------
HSL processing time: 20.849001 (ms)
YUV processing time: 13.423000 (ms)
RGB processing time: 5.568000 (ms)
-------------------------------------
Starting GPU processing
-------------------------------------
HSL processing time: 5.490000 (ms)
YUV processing time: 3.207000 (ms)
RGB processing time: 1.831000 (ms)
```

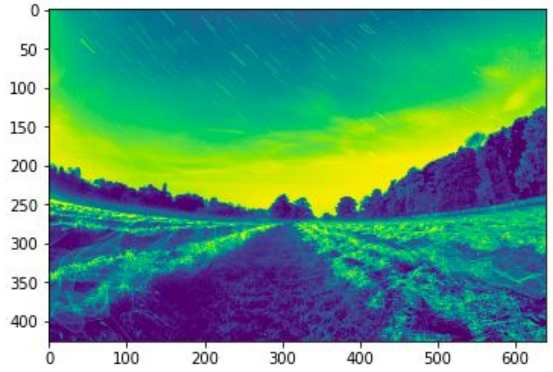
img_in_gray :



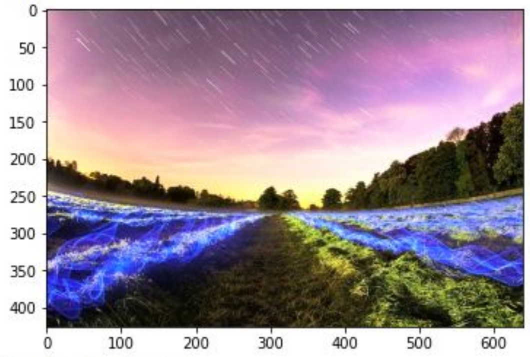
img_in_color:

Results

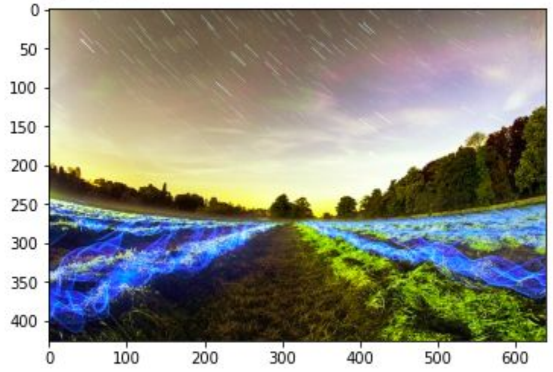
img_out:



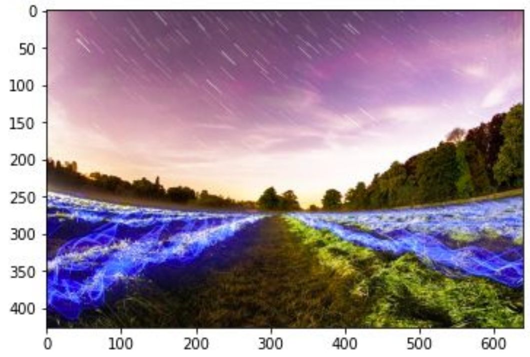
img_in_out_yuv:



img_out_rgb:



img_in_out_hsl:



References

[1]Mr Diptarup Saha, Mr Karan Darji, Narendra Patel, Darshak Thakore, Implementation of Image Enhancement Algorithms and Recursive Ray Tracing using CUDA, Procedia Computer Science, Volume 79, 2016, Pages 516-524, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2016.03.066.

[2]Maini, R. (2010, March 22). A Comprehensive Review of Image Enhancement Techniques. ArXiv.Org. https://arxiv.org/abs/1003.4053

[3]M. Abdullah-Al-Wadud, M. H. Kabir, M. A. Akber Dewan and O. Chae, "A Dynamic Histogram Equalization for Image Contrast Enhancement," in IEEE Transactions on Consumer Electronics, vol. 53, no. 2, pp. 593-600, May 2007, doi: 10.1109/TCE.2007.381734.

[4]Magudeeswaran Veluchamy, Bharath Subramani, Image contrast and color enhancement using adaptive gamma correction and histogram equalization, Optik, Volume 183, 2019, Pages 329-337, ISSN 0030-4026, https://doi.org/10.1016/j.ijleo.2019.02.054.