**Monash University FIT 3036**

**Project Proposal**

**Final Year Project:**

***Tone Mapping for High Dynamic Range Imaging***

Supervisor : Dr. Wong

Name: Liew Teik Zan

ID: 2620 1585

**Table of Contents**

1. Introduction and Literature Review ------------------------------------------------------- 3

2.0 Project Requirements

2.1 Functional Requirements ---------------------------------------------------------------- 3

2.2 Non-functional Requirements ------------------------------------------------------- 3

3.0 Project Plan

3.1 Overview ------------------------------------------------------------------------------ 4 - 6

3.2 Risk Analysis ------------------------------------------------------------------------- 7

3.3 Resource Requirements ---------------------------------------------------------------- 8

3.4 Schedule in project ---------------------------------------------------------------- 8

4.0 External Design - input, output, performance.

4.1 User Interface ------------------------------------------------------------------------- 9

4.2 Functionality ------------------------------------------------------------------------- 9

4.3 Performance ------------------------------------------------------------------------- 9

5.0 Internal Design --------------------------------------------------------------------------------- 10

6.0 Software Architecture ------------------------------------------------------------------- 11 - 12

7.0 Test Plan ------------------------------------------------------------------------------------------ 13

7.1 Test coverage ------------------------------------------------------------------------ 14

7.2 Test methods ------------------------------------------------------------------------ 14

7.3 Sample Test Cases --------------------------------------------------------------- 14

8.0 References --------------------------------------------------------------------------------- 15

**1.0 Introduction and Literature Review (problem definitions & project objectives)**

In the real world, the scene contains a large range of light intensities and different colors from a dark night to a bright sunlight. There are plenty of research topics which are based on capturing the real-world scene to high dynamic range (HDR) and reproduce it to low dynamic range (LDR) images.

Nowadays, most of the display devices are basically limited to 8 bits per color channel (LDR), and hence they cannot display a HDR image correctly. Tone-mapping for HDR imaging is the solution to display the HDR image in LDR display devices such as monitors, TV and printers. There are many different ways for tone mapping including some algorithms with global operators and local operators. All those algorithms are used to maintain the quality and contrast the HDR scenes in the common display devices.

In this project report, several methods will be discussed and tested using the inverse function to compare the image with the original image. A lot more details will be discussed in chapter 3 and final year project.

**2.0 Project Requirements**

**2.1 Functional Requirements**

* Develop an algorithm to convert the HDR image into LDR image to be displayed in common display devices.
* Reproduce the high range color channel into a smaller range color channel. For example, 10-bit to 8-bit color range.
* Minimize the differences between the images before and after the algorithm.
* Develop a testing plan to compare the images so that it will be proved that the original image has been reproduced to lower bit color range. For example, 8 bit to 6 bit and reverse back, it will lose some details.

**2.2 Non-functional Requirements**

* Understand the general concept of the tone mapping algorithms (global and local operator).
* Using the C++ programming language and OpenCV libraries to run the program.
* Calculate the luminance and the brightness for every pixel to apply the tone mapping algorithm.
* System shall be able to show the images properly before and after the operation

**3.0 Project Plan**

* Study the research paper, related book
* Research on the extra information about the sources
* Understand the light and colour
* Understand the concept of the image (HDR and LDR)
* Study on the tone mapping method
* Clearly understand the concept of global and local operator
* Apply the algorithm to prepared a LDR image (various types of them)
* Implement a testing function (inverse function)
* Compare the differences of the image

**3.1 Overview**

In this final year project, it aims to display a high dynamic range image in a low common display devices (e.g., monitor, LCD, and printer). Such tasks may scale pixel values by using tone mapping algorithm to reduce the precision errors to below humanly detectable levels. There are two main method about tone mapping which are:

* Global Operator: Compress the images using identical (non-linear) curve for every pixel, treat every pixel in the image independently, much faster and simpler, (loss of details). This includes *Reinhard’s operator (photographic tone reproduction), Tumblin-Rushmeier Brightness Operator, Ward Contrast Operator, Logarithmic and exponential mapping, and many more.*
* Local Operator: Depends on the value of the neighborhood pixels on the current mapped pixel. Modulating a non-linear curve by an adaptation level for pixels.

Explanation

HDR image is basically containing a larger range of colour channel compared to LDR image. For example, if there is a HDR image contains 10-bit color channel, meaning that it consists of 0 to 210 -1 color range for each red, blue and green in RGB format. Hence, there will be a 1024 x 1024 x 1024 color in total and there are over a billion possible colors in an image. Compared to 8 bit images which contain only around 16 million possible colors, the tone mapping algorithm will need to reduce the 10-bit image to 8-bit and minimize the differences between them. It is entirely possible to tonemap HDR to LDR image for display devices but it is not possible to reconstruct a high quality HDR image from a quantized LDR data.

Requirements

A proper image file type and a correct algorithm are needed for photographic tone reproduction.

Fully understand the project topic and look for the related research papers, books, or articles to find out more details about the tone-mapping algorithm and also the basic knowledge. Set up an environment (C++ and OpenCV) to program the algorithm and test the converted images whether they are correctly tone mapped.

Constraints

Complete all the algorithms within 12 weeks and compare the result from each of them. Project must be done with the approval from project supervisor.

Assumption

There will be some updates every week to this project. Besides, the supervisor will guide student in some aspect for the improvement in this project and student is mainly in charge of the whole project by himself.

Output image Comparison





The first image is the original image (8 bit) and followed by the tone mapped image (6 bit, 4 bit, 3 bit and 2 bit) the colour range is lower after tone mapping. (It’s obvious on the sky and the river part.)

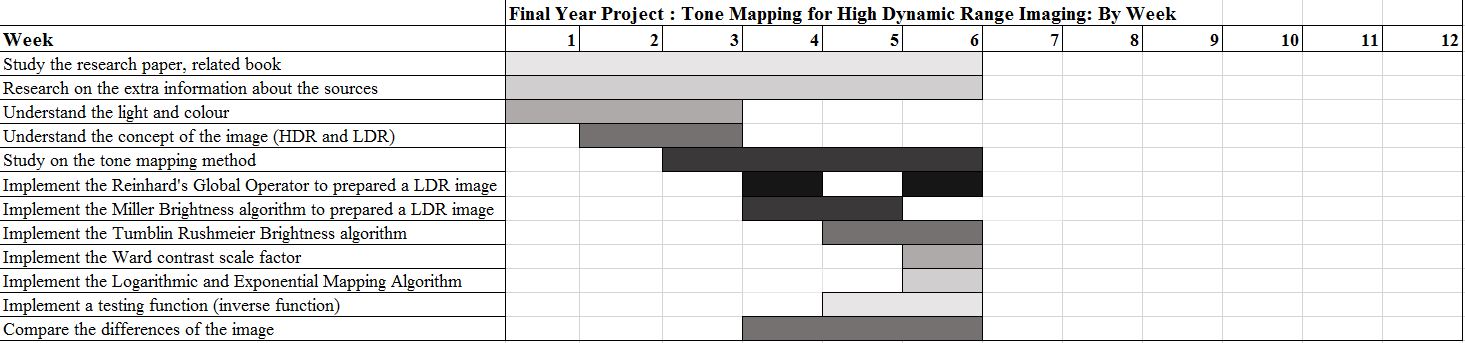
**3.2 Risk Analysis**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Project: Tone Mapping for high dynamic range (HDR) imaging    Key: H – High; M – Medium; L – Low | | | | Prepared by: Liew Teik Zan | | | | Reference: \_\_\_\_\_\_\_\_\_\_\_\_  Date: 22 August 2017 | | |
| # | Description of Risk | Probability | | | Impact | | Risk reduction Strategy | | Contingency plans | Risk owner |
| H | M | L | Pref | Time |
| 1 | Unable to display the image on some devices | X |  |  | H | M | Check the color bit range that the device able to display. | | Reproduce every HDR image to 8 bit. Reproduce it to a lower bit range if needed. | Liew |
| 2 | Some of the images are not reproduced properly to LDR, for example, the color of the image totally changed. |  |  | X | L | H | Test for every type of images by using the tone mapping algorithm | | Look for another tone mapping algorithm. | Liew |
| 3 | The quality of the image is bad after tone mapping. |  |  | X | L | H | Check and save the details of the algorithm every time before running. | | Check the details of the algorithm and the constant number inside the program. | Liew |
| 4 | The HDR image is not actually tonemapped into LDR |  |  | X | L | H | Check the program algorithm before tone mapping. | | Rewrite or reuse the program that is written before. | Liew |

**3.3 Resource Requirements**

* A setup environment for C++ programming and OpenCV
* Understanding an algorithm or a research paper includes the details of the procedures for tone mapping

**3.4 Schedule**

****

**4.0 External Design**

Input: This tone mapping program is allowed to input a number which represent bit of an image color channel range.

Output: The program will tone map the HDR image to the number of bit of an image which is the input and display the result to the user.

Performance: The program will then show the image after tone mapping to the user and hence the user is able to compare the two different images (e.g 10-bit and 8-bit or 8 bit and 6 bit). Simultaneously, the program minimizes the differences of the images before and after the algorithm.

**4.1 User Interface**

* Read the image file by using C++ and OpenCV
* Apply one of the operator to tone map the image.
* Try all different operators to get the different result.

**4.2 Functionality - Any externally available functions not covered above.**

* **-**

**4.3 Performance**

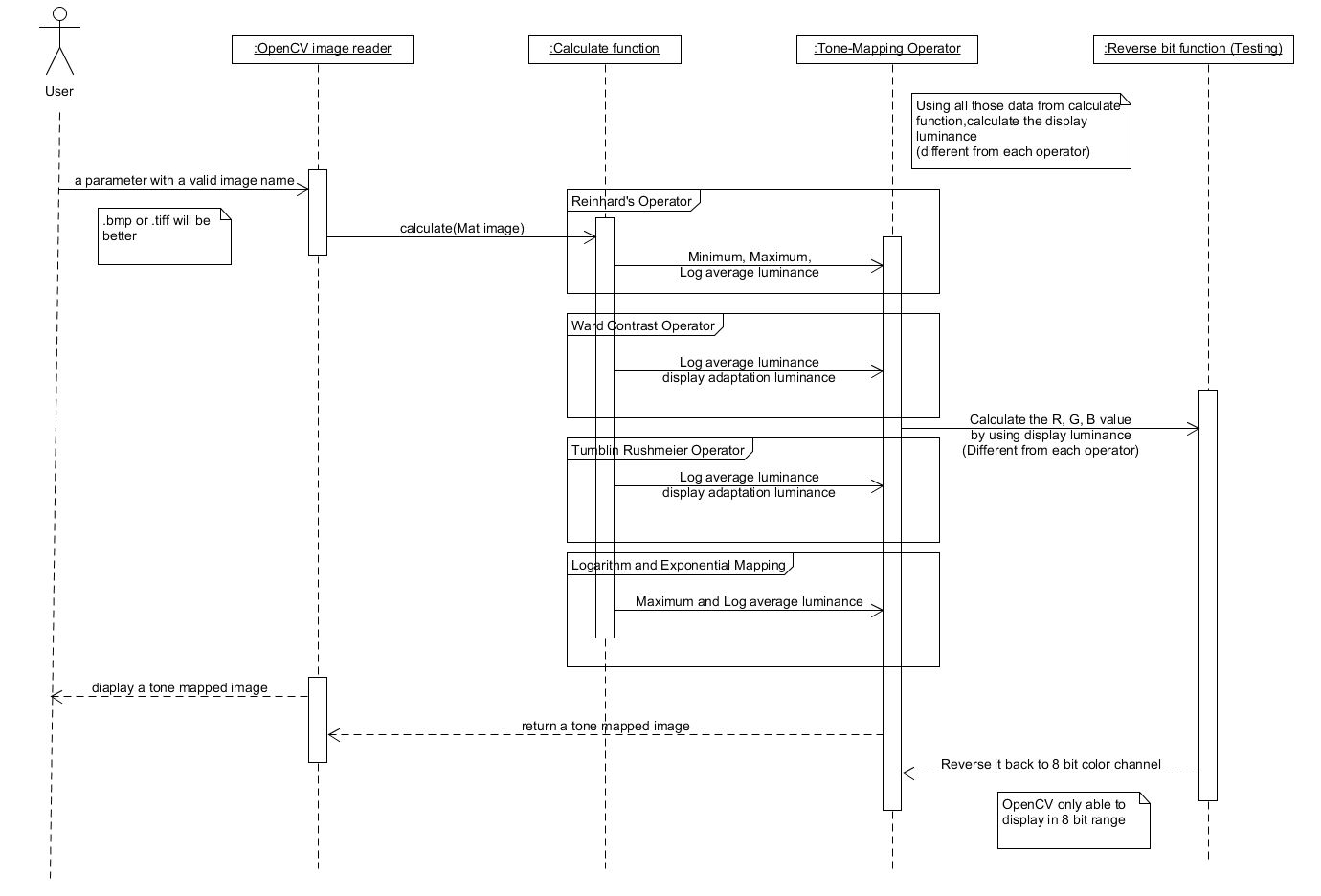
*Time Complexity*

* O(n) - n is the number of pixels of the input image.

*Space Complexity*

* O(n) - n is the number of pixels of the input image.

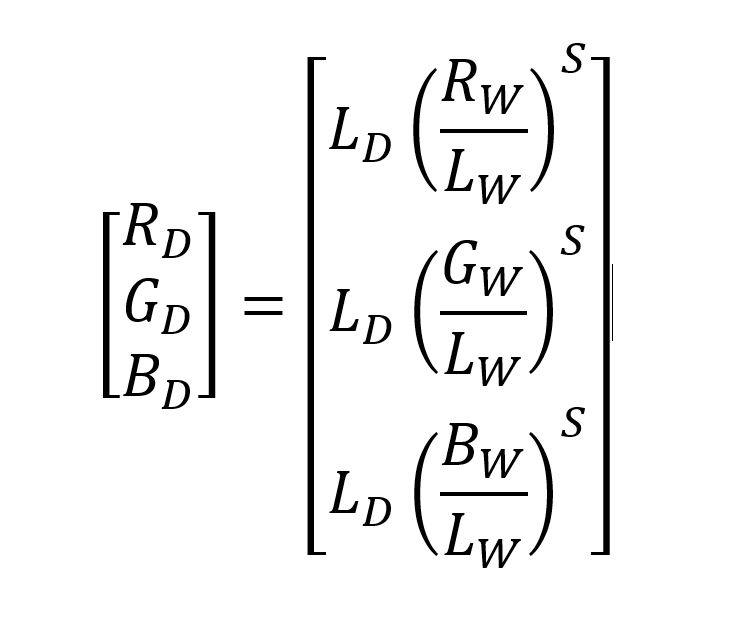
**5.0 Internal Design**

****

\* A sequence diagram of this project

**6.0 Software Architecture**

Generally, we use a simple equation to start the tone mapping algorithm:

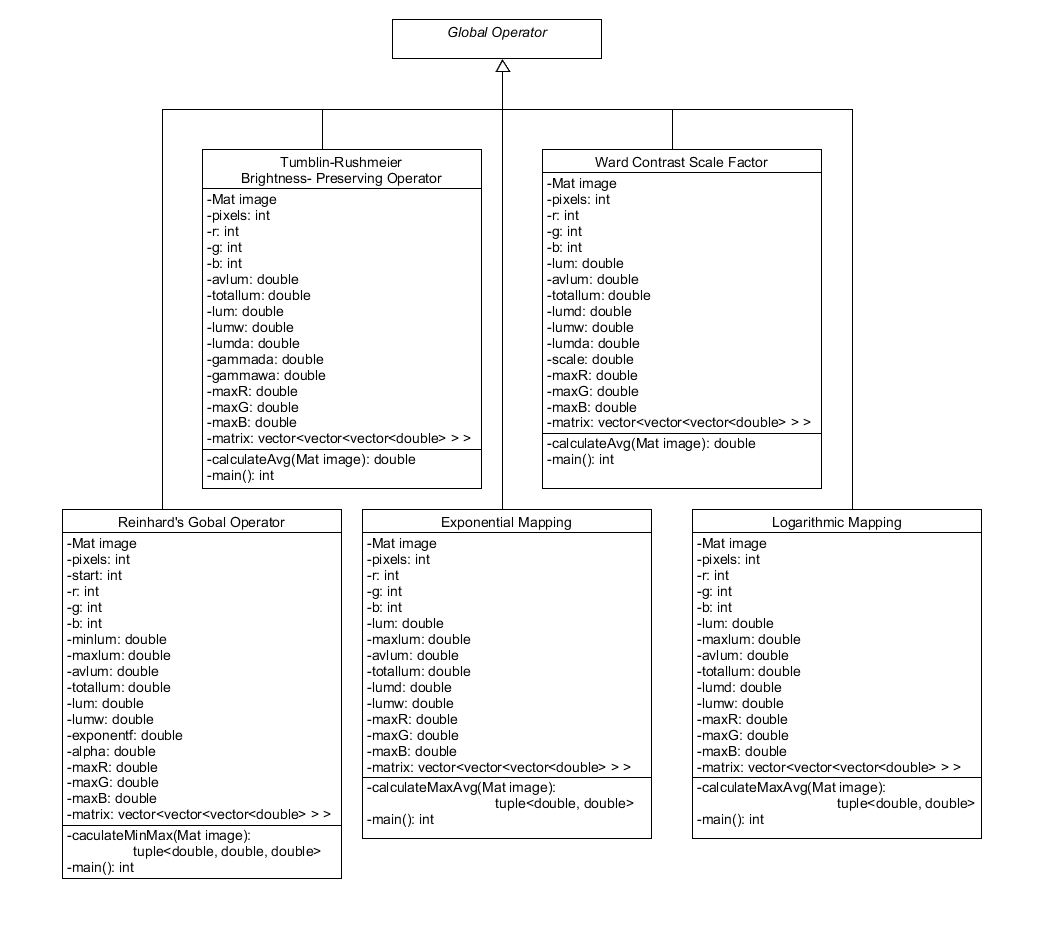


***\*Equation 1.0***

\* D = display, W = world, S = scale factor

In this tone mapping equation, RD, GD, and BD represent the value red, green and blue after tone mapping. RW, Gw, and Bw are colour values in the input image in each pixel.

LW is the luminance of that particular pixel of the input image and it is calculated by converting RGB to either Y’CbCr or Y’UV color space, and the Y’ represent luminance of the pixel. Besides, the key parameter here is LD, which is where the calculation occurs. There are plenty of ways to calculate the display luminance by using different operators (Reinhard’s Global Operator, Ward Contrast, Tumblin-Rushmeier, Logarithmic and Exponential Mapping) (Erik Reinhard, 2006)



\*UML diagram for this project

**7.0 Test Plan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Test Name** | **Brief Rationale** | **Setup** | **Input** | **Expected Output** | **Output** |
| **1.** | The openCV in C++ works correctly. | Try to read and display different type of image file | Read .bmp file | .bmp file | Display successfully | Display successfully |
| Read .tiff file | .tiff file |
| Read .jpg file | .jpg file |
| **2.** | The calculate function in each operator | Test the calculate function for maximum, minimum, and log average luminance. | Test the calculateMinMaxAvg() function | An image file | Three numbers between 0 to 255 | Three numbers between 0 to 255 |
| Test the calculateMaxAvg() function | Two numbers between 0 to 255 | Two numbers between 0 to 255 |
| Test the calculateAvg() function | A number between 0 to 255 | A number between 0 to 255 |
| **3.** | The tone mapping algorithm[\*] | There are lots of different operator and test every of them | Test Reinhard’s Global Operator | An image file | A low color range image | A low color range image |
| Test Tumblin- Rushmeier Brightness- Preserving Operator |
| Test Ward Contrast Scale Factor |
| Test Logarithmic Mapping |
| Test Exponential Mapping |

[\*] = in the test plan, we test each operator to tone map the image from 8 bit to 7/6/5/4/3 bit color range, so that we can see the obvious result by naked eyes.

**7.1 Test coverage**

First of all, it is a must to test whether the setup C++ and openCV environment and the compiler functions properly.

In order to implement the tone-mapping algorithm, most of the operators need some combinations of three attributes which are maximum, minimum and log average luminance of the image. In this step, the program will change the RGB value into Y’CbCr (Y’ means luminance) for each pixel by using the equation Y’ = 0.299R + 0.587G + 0.114B. Hence, maximum, minimum and log average luminance will always between 0 to 255.

Lastly, every global operator (tone-mapping algorithm) has its own equation to calculate the display luminance. We will test it by using operator to downscale the bit and upscale it again to 8 bit because openCV is only able to show the image in 8 bit. In this step, we will see the differences of the images.

8 bit color range image

lower bit color range image

Tone map to lower bit image

8 bit color range image

Upscale back to 8 bit in openCV

Loss of details

**7.2 Test methods**

In all those test plans, this project uses unit testing and acceptance testing. This project will use unit testing for every function (calculateAvg(), calculateMinMaxAvg(), calculateMaxAvg()), and make sure the return values are all between 0 to 255.

After that, for the next plan (testing for the tone map algorithm), we make sure it run correctly by using unit testing for the code in the C++ environment and acceptance testing for the output image. Hence, we will basically see the different if the image being reproduced to a lower bit because there will be loss of details.

**7.3 Sample Test Cases**

All the data will be collected from an image in openCV, meaning there are three values needed in this project for each pixel of the image: Red, Green, and Blue. Since the vector in the picture pixels used to store integer, so that we create a new matrix to store the modified RGB values by using double. In the last step, we will reassign the value from the matrix to the original image.

**8.0 References**

Erik Reinhard, G. W., Sumanta Pattanaik and Paul Debevec (2006). *High Dynamic Range Imaging* (Vol. 2nd). United Kingdom.

MEYLAN, L. (2006). Tone Mapping For High Dynamic Range Images.

Yuma Kinoshita, S. S. a. H. K. (2017). Fast Inverse Tone Mapping With Reinhard's Global Operator.

Dynamic Range Reduction for TOne Mapping of HDR images. (2005). Retrieved from <http://www.powershow.com/view1/c1ce5-ZDc1Z/Dynamic_Range_Reduction_for_Tone_Mapping_of_HDR_Images_powerpoint_ppt_presentation>

(1620 words)

\*excluding tables, appendix, references