SOFTWARE REQUIREMENTS SPECIFICATIONS

Image Editor

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

The purpose of this document is provide a detailed overview of our software product, its functionality and usage.

1.2 Scope

The Image Editor is intended to work as software for browsing, editing and saving an image. It provides the user with a multiple tools to alter the image properties, to name some - crop, blur, flip, rotate, negative, sepia, grayscale, halftone and finally save the edited image.
As part of the project, the Image Editor works efficiently with only bmp files and in Linux operating system.

1.3 Definitions, Acronyms and Abbreviations

(a) Linux

Linux is an open source operating system, with extensive documentation and user support.

(b) GTK

gtkmm is the official C++ interface for the popular GUI library GTK+. gtkmm is free software distributed under the GNU Lesser General Public License (LGPL).

(c) Pixel

In digital imaging, a pixel is the smallest addressable element in an all points addressable display device; so it is the smallest controllable element of a picture represented on the screen. The address of a pixel corresponds to its physical coordinates.

(d) The Class Image

This class contains all the functions which are provided by the 'Image Editor' software. It also contains an array called 'process', which keeps track of all the operations which are perform on the pixel matrix of the image, declarations of **'R','G','B'** - the red, green, blue arrays .

(e) R, G and B

R, G and B refer to pixel values of red, blue and green colours respectively. In the class definition, however, they are pointers to the R,G,B matrices.

(f) newpixelvalue[][], oldpixelvalue[][]

These are generic names uses to refer to the R,G and B matrices collectively. newpixelvalue[][], oldpixelvalue[][],are 2 Dimensional arrays of containing pixel values indexed[height][width]. 'height' and 'width' are the height and width of the image in pixels.

1.4 References

The general references used in developing this software are:

1. Tutorial on "Creating a bitmap file with hexadecimal editor"
http://turrier.fr/tutoriels/form_02/create-a-bitmap-file-with-an-hexadecimal-editor.html

2. Book: "Image Processing in C, second edition" by Dwayne Phillips
http://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/cips2ed.pdf

3. Online Tutorial on "C++gtkmm"
http://blog.mpshouse.com/ - a online 6 tutorial series for gtkmm

4. Websites that came to general aid:
(1) Wikipedia: http://en.wikipedia.org/
(2) Google: https://www.google.co.in/
(3) StackOverflow: https://www.stackoverflow.com

5. https://www.cambridge.color.in

6.For conversion of images,
*image.online-**convert**.com/**convert-to-bmp***

7.http://docs.knobbits.org/local/gtkmm-3.0/tutorial/html/sec-dialogs-filechooserdialog.html – used for making the file chooser dialog box

8.Google images for sample images for the program

Other references have been listed in the product description, along with the individual function description.

1.5 Overview

The rest of the SRS examines the specifications of the Image Editor in detail. Section 2 of the SRS presents the general factors that affect the Image Editor and its requirements, such as user characteristics and project constraints. Section 3 outlines the detailed, specific functional, performance, system and other related requirements of the Image Editor. Supporting information about appendices is provided in Section 3.

2. OVERALL DESCRIPTION

2.1 Product Description:

Image Editor is a software designed to carry out image processing tasks as per the user's choice.

2.2 Product Functions:

The functional features in the Image Editor software are:

1. Copy
2. Grayscale
3. Brightness
4. Negative
5. Contrast
6. Blur
7. Flip Horizontal
8. Flip Vertical
9. Rotate
10. Sepia
11. Salt and Pepper Effect
12. Halftone effect
13. Edge Detection
14. Dithering
15. Crop
16. Posterize
17. Undo

2.3 Assumptions and Dependencies:

For the Image Editor to run successfully, the GTK library must be installed in the system and linked to the compiler. Further, the Editor works only on Bitmap Images.

3. SPECIFIC REQUIREMENTS

3.1 Functional Requirements

A brief overview of all the functions used by the software is given below:

(1) Copy

Name	**Copy**
Summary	The bitmap image is copied.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the Copy button in the display window. 2. The software copies the pixel values of the Image object and these are stored in another Image object. 3. The createimage() function is called which creates the output image. 4. The software displays the copied image.
Post-conditions	The image is copied and the output file is saved.

(2) Grayscale

Name	**Grayscale**
[1]Summary	The bitmap image is converted to grayscale.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the grayscale button in the display window. 2. The software updates the pixel values of the Image object as 0.21 R + 0.72 G + 0.09 B in the same Image object.[1] 3. The createimage() function is called which creates the output image. 4. The software displays the grayscale image.
Post-conditions	The image is converted to grayscale and the output file is saved.

(3) Brightness

Name	**Brightness**
Summary	The bitmap image is brightened.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the brightness button in the display window, a dialog box opens asking for

1 Image Editor uses the Luminosity algorithm: http://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/

Name	Brightness
	the users choice for the brightening factor. 2. The software updates the pixel values of the Image object according to brightening algorithm.[2] 3. The createimage() function is called which creates the output image. 4. The software displays the brightened image.
Post-conditions	The image is brightened and the output file is saved.

(4) Negative

Name	Negative
Summary	The bitmap image is converted to grayscale.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the negative button in the display window. 2. The software updates the pixel values of the Image object as 255- R, 255- B and 255- G. 3. The createimage() function is called which creates the output image. 4. The software displays the negative image.
Post-conditions	The negative of the image is displayed and the output file is saved.

(5) Contrast

Name	Contrast
Summary	The contrast of the bitmap image is improved.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the contrast button in the display window. 2. The software updates the pixel values of the image object according to the standard histogram equalization except that the RGB matrix is first converted to YCbCr and equalization is applied on the Y part and Cb and Cr are left unchanged and YCbCr is back converted to RGB. 3. The createimage() function is called which

[2]The Brightness Algorithm: Each pixel value is increased by the brightening factor. If it exceeds 255, it is set to 255. If it is less than 0, it is set to 0. Positive values increase the brightness and negative values decrease the brightness. 3. http://en.wikipedia.org/wiki/Histogram_equalization

3 http://stackoverflow.com/questions/15007304/histogram-equalization-not-working-on-color-image-opencv

[3] cs 101 Lecture slides on histogram equalization

Name	**Contrast**
	creates the output image. 4. The software displays the image with improved contrast.
Post-conditions	The image is displayed with improved contrast and the output file is saved.

(6) Blurring

Name	**Blurring**
Summary	The bitmap image is blurred.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the blur button in the display window. 2. The software updates the pixel values of the Image object by applying a Gaussian blur[3]. 3. The createimage() function is called which creates the output image. 4. The software displays the blurred image.
Post-conditions	The blurred image is displayed and the output file is saved.

(7) Flipping Horizontally

Name	**Flip Horizontal**
Summary	The bitmap image is flipped horizontally.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the Flip Horizontal button in the display window. 2. The software updates the pixel values of the Image object as **newpixelvalue[x][y]=oldpixelvalue[height-1-x][y]** for effecting horizontal flip. 3. The createimage() function is called which creates the output image. 4. The software displays the horizontally flipped image.
Post-conditions	The horizontally flipped image is displayed and the output file is saved.

(8) Flipping Vertically

Name	**Flip Vertical**
Summary	The bitmap image is flipped vertically.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the Flip Vertical button in the display window. 2. The software updates the pixel values of the

3 Description on Gaussian blur: http://docs.gimp.org/en/plug-in-gauss.html

Name	**Flip Vertical**
	Image object as **newpixelvalue[x][y]=oldpixelvalue[x][width-1-y]** for effecting vertical flip. 3. The createimage() function is called which creates the output image. 4. The software displays the vertically flipped image.
Post-conditions	The flipped image is displayed and the output file is saved.

(9) Rotate

Name	**Rotate**
Summary	The bitmap image is rotated by an angle of the user's choice.
Pre-conditions	A bitmap image must be opened and its R, G, B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the Rotate button in the display window, which prompts the user to enter an angle or click one of the two buttons: 90 clockwise and 90 anticlockwise. 2. The pixel values of the Image object are modified using the rotation formula[4]. The output image remains in the same boundary as that of the earlier image. 3. The createimage() function is called which creates the output image. 4. The software displays the rotated image.
Post-conditions	The rotated image is displayed and the output file is saved.

(10) Sepia

Name	**Sepia**
Summary	The bitmap image is converted to sepia tone.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.

4 With i1, j1 are indices of array and **temph, tempw** are new height and new width of the image arrays. Rotation Formula: **i1=(y-n)*sin(theta)+(x-m)*cos(theta)+m+ 1/2*(temph-height)** **j1=(y-n)*cos(theta)-(x-m)*sin(theta)+n+ 1/2*(tempw-width)**
if -1<i1 and i1<height and -1<j1 and j1<width
then: newpixelvalue [i][j]= oldpixelvalue [i1][j1]
else newpixelvalue [i][j]=255,

Name	**Sepia**
Basic Course of Events	1. The user clicks the sepia button on the display window. 2. The software updates the pixel values of the Image object according to Sepia standard set by Microsoft for Sepia image.[5] 3. The createimage() function is called which creates the output image. 4. The software displays the image in sepia tone.
Post-conditions	The sepia image is displayed and the output file is saved.

(11) Salt and Pepper Effect

Name	**Salt and Pepper Effect**
Summary	A salt and pepper effect is given to the bitmap image.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the Salt and Pepper Effect button on the display window. 2. The software generates random numbers in the range 0-255 for each pixel value of the Image object and sets it to 0 or 255 if the random number is 0 or 255. 3. The createimage() function is called which creates the output image. 4. The software displays the image with the salt and paper effect.
Post-conditions	The salt and paper image is displayed and the output file is saved.

(12)Halftone Effect

Name	**Halftone**
Summary	The color bitmap image is halftoned.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the Halftoning button on the display window. 2. The software traverses the pixel array of the Image object and calculates the generated and propagated errors at each position. If the sum of this error and the gray-value of the pixel exceeds a certain threshold, it is set on (white), else it set off (black). 3. The createimage() function is called which

5The Microsoft standard recommends the following values for Sepia tone:NEWR = R * 0.393 + G * 0.769 + B * 0.189 NEWG = R * 0.349 + G * 0.686 + B * 0.168
NEWB = R * 0.272 + G * 0.534 + B * 0.131
http://stackoverflow.com/questions/1061093/how-is-a-sepia-tone-created

Name	Halftone
	creates the output image. 4. The software displays the halftoned image created using the error diffusion technique.[6]
Post-conditions	The halftoned image is displayed and the output file is saved.

(13)Edge Detection

Name	Edge Detection
Summary	The edges of the bitmap image are detected.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the Edge Detection button on the display window. 2. The software sections the pixel matrix of the Image object into 3X3 sub-matrices and it is multiplied point-by-point by a Sobel mask and the resulting sum is placed at the center of the sub-matrix.[7] 3. The createimage() function is called which creates the output image. 4. The software displays the image having the edges of the original image.
Post-conditions	The edge-image is displayed and the output file is saved.

(14)Dithering

Name	Dithering
Summary	The dithering operation which uses a 16-color palette is performed on the bitmap image.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the Dither button on the display window. 2. The software traverses the pixel matrix of the Image object and computes the color in the palette that is closest to the original color. The color is updated and the error is then dispersed in the neighboring pixels.[8] 3. The createimage() function is called which creates the output image.

6 Section 1.4.1 Chapter 3.
7 Section 1.4.1 Chapter 5.

8REFERENCES:http://waset.org/publications/10396/improved-posterized-color-images-based-on-color-quantization-and-contrast-enhancement 10. Algorithm from http://en.wikipedia.org/wiki/Floyd%E2%80%93Steinberg_dithering

Name	**Dithering**
	4. The software displays the image using the 16-color palette.
Post-conditions	The dithered image is displayed and the output file is saved.

(15) Crop

Name	**Crop**
Summary	The bitmap image is cropped.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the Crop button on the display window and is asked to enter the x and y co-ordinates of the bounding rectangle which will bound the cropped image. 2. The software calculates the new height and width and copies the elements in the bounding rectangle to new array of appropriate size.The deleteRGB() and createRGB() with appropriate dimensions are called. 3. The createimage() function is called which creates the output image. 4. The software displays the cropped image.
Post-conditions	The cropped image is displayed and the output file is saved.

(16) Posterize

Name	**Posterize**
Summary	The bitmap image is posterized converted into a poster like image.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1. The user clicks the Posterize button on the display window and is asked to enter the no. of colors(n) to be used to create the poster image. 2. The software traverses through the pixel matrix and updates the values of the matrix accordingly as it lies which range of values (of width 256/n) it lies to a appropriate value in that range .The number of color in the output image is usually very small which makes the user to perceive it as a poster. 3. The createimage() function is called which creates the output image. 4. The software displays the posterized image.
Post-conditions	The cropped image is displayed and the output file is saved.

(17)Undo

Name	**Undo**
Summary	The last operation is reversed.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1.The user clicks the Undo button on the display window 2.An array stores every process by a unique no. along with its arguments (if any).This create a new RGB array by calling copyimage function and apply all the processes except the last one stored in the above array. 3.The createimage() function is called which creates the output image. 4.The software displays the previous image.
Post-conditions	The previous image is displayed and the output file is saved.

(18) Redo

Name	**Redo**
Summary	The previously applied undo operation is nullified.
Pre-conditions	A bitmap image must be opened and its R,G,B values must be stored in an object of the class Image.
Basic Course of Events	1.The user clicks the Redo button on the display window 2.An array stores every process by a unique no. along with its arguments (if any).The redo just applies the last process stored in the array on the image again if that process was previouly applied undo 3.The createimage() function is called which creates the output image. 4.The software displays the previous image.
Post-conditions	The previous image is displayed and the output file is saved.

3.2 Reliability

The program will display an error message in case of file input of a type other than bmp. It will not crash.

3.3 Performance Requirements

The modified file will be created, stored and displayed within the time frame of 5s.

3.4 Design Constraints

All coding will be done in C++.

3.5 Interfaces

(a) User Interfaces

The user interface is the display window created by GTK, with the coding done in C++.

(b) Hardware Interfaces:

1. CPU usage
2. Screen usage
3. Memory usage
4. Mouse usage
5. Keyboard usage

(c) Software Interfaces

Image Editor interfaces with the operating system. For the execution of image editing functions, the Gtkmm library will be used.