DEVELOPMENT OF DIGITAL IMAGE RE-COLOURING ALGORITHM

NOORFAIRUSE BINTI MOHAMAD NOOR

UNIVERSITI SAINS MALAYSIA

2018

DEVELOPMENT OF DIGITAL IMAGE RE-COLOURING ALGORITHM

by

NOORFAIRUSE BINTI MOHAMAD

Thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering (Electronic Engineering)

JUNE 2018

ACKNOWLEDGEMENT

I am using this opportunity to express my gratitude to my final year project supervisor, Assoc. Professor Dr. Haidi Bin Ibrahim. With his patience guidance throughout the year has led me to the completion of this final year project. His advices and comment are valuable and useful which help to find the solution to this final year project. He was patient in monitoring me in order to ensure that I am on the right track along the way of completion of this project.

Thank you to the examiner, Dr Fauzi Packeer, for the involvement in my final year project and thesis assessment. His comments and advises help to improve my final year project and thesis.

Moreover, I would like to show my gratitude to the coordinators of final year project Assoc. Professor Dr. Bakhtiar Affendi and Assoc. Professor Ir. Dr. Mohamad Kamarol for their help regarding the final year project and also other lecturers who have given the guideline for the final year project and also the thesis writing.

I would like to thank both my parents who have always support me throughout the period of this project. I would sincerely thank to my friends who have given the ideas for my final year project and help me in any problems that I faces.

TABLE OF CONTENT

LIST OF FIGURE

LIST OF TABLE

LIST OF ABBREVIATION/ACRONYMS

RGB	Red Green Blue
PSNR	Peak Signal To Noise Ratio
MSE	Mean Squared Error

LIST OF SYMBOL

l	archromatic luminance channel
α	chromatic channel
β	chromatic channel
y	monochromatic luminance channel
u	chrominance channel
v	chrominance channel
$w_{\boldsymbol{rs}}$	weighting function
μ_r	mean of the intensities
σ_r	variance of the intensities
M	row of the image
N	column of the image
C	number of channel of the image

DEVELOPMENT OF DIGITAL IMAGE RE-COLOURING ALGORITHM

ABSTRACT

Re-colouring image or colorization is the process of adding colour to monochrome image or greyscale image which it is typically involve segmentation of images into regions and tracking these regions across image sequence. Colorization have been developed from time to time in order to improve its performance and to make sure that the system developed is user friendly. First colorization method was by determined the pixels region manually and added colour to the region manually. This colorization method spends a lot of cost and time consume. Another method is example-based colorization by transferring colour from a reference colour image. The artistic control of this method is quite indirect because the user is required to find a references image containing the desired colour over region. This thesis will present the simple colorization method which is scribble colorization method that based on a simple premise which is neighbouring pixels in space time that have similar intensities should have similar colour. The user only needs to annotate the image with a few colour scribbles and the indicated colours are automatically propagated in both space and time to produce a fully colorized image. Qualitative analysis and quantitative analysis are used to verify the result images. The result images are compared based on the quality of the colour and the value of PSNR. High value of PNSR indicated that the result image and the ground truth image are similar.

PEMBANGUNAN ALGORITMA PEWARNAAN SEMULA IMEJ DIGITAL

ABSTRAK

Pewarnaan semula imej atau pewarnaan adalah proses menambah warna kepada imej monokrom atau imej skala kelabu yang biasanya melibatkan peruasan imej ke kawasan-kawasan dan menjejaki kawasan ini di seluruh urutan imej. Pewarnaan telah dikembangkan dari masa ke semasa dalam usaha untuk meningkatkan prestasi dan memastikan bahawa sistem yang dibangunkan adalah mesra pengguna. Kaedah pewarnaan pertama adalah menentukan kawasan piksel secara manual dan menambah warna ke kawasan ini secara manual. Kaedah pewarnaan ini menlibatkan kos yang tinggi dan memakan masa. Kaedah lain adalah berdasarkan contoh-pewarnaan dengan memindahkan warna dari imej warna rujukan. Kawalan seni kaedah ini adalah agak tidak langsung kerana pengguna diperlukan untuk mencari imej rujukan yang mengandungi warna yang dikehendaki di rantau.Tesis ini akan membentangkan kaedah pewarnaan mudah iaitu kaedah contengan yang berdasarkan permis mudah yang jiran piksel dalam ruang masa yang mempunyai keamatan yang sama perlu mempunyai warna yang sama. Pengguna hanya perlu menganotasi imej dengan beberapa contengan warna dan warna-warna yang ditunjukkan secara automatik merebak dalam kedua-dua ruang dan masa untuk menghasilkan imej berwarna sepenuhnya. Analisis kualiti dan analisis kuantiti digunakan bagi mengenalpasti imej yang terhasil. Imej yang terhasil telah dibanding berdasarkan kualiti warna yang terhasil dan nilai PSNR. Nilai PSNR yang tinggi menunjukkan imej yang terhasil sama dengan imej yang asal.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Colorization is a process of adding colour to a monochrome image or a greyscale image [1]. Colorization-based colour image coding is the generation of a new colour image coding method that develop the colorization technique. There are many old pictures that are in monochrome image or in greyscale image, colorization will helps to re-colour the image. Monochrome image is an image consisting of a single colour against a neutral background. For example, old "green screen" monitors were called monochrome monitors because they used a single colour (green or amber) against a neutral background (black). All black and white images are monochrome images, but not all monochrome images are black and white. For example, a monochrome image might consist of black on yellow. Greyscale images are images that are represented using only 256 shades of gray rather than the full pallet of colours.

Colorization term was introduced by Wilson Markle in 1970 to describe a computer-assisted process in adding colour to black and white movies or televisions' programs [2]. Colorization is now used to describe a technique of adding colour to monochrome image or greyscale image. Markle have introduced his original colorization process which a colour mask was manually painted for at least one reference frame in a shot. Motion detection is applied and allowing colours to be automatically assigned to other frames in regions where the motion occurs.

In early years, studio features film had a theatrical run and those films that had a significant return on studio investment were likely saved. It is ironic that it took colorization, with all its controversy, to mass attention on the need to restore and preserve film heritage. With so many diverse digital distribution outlets from entertainment product and a renewed desire for proven vintage entertainment, the studio have great incentive to restore their classic black and white film libraries. Several have called Legend Films to colorize selected classic black and white titles in an effort to make them more attractive to new generations of audiences. Legend Films is the pioneer and reorganized leader in the restoration and colorization of classic black and white movie [3].

Legend Films can apply colour effects at a resolution of 48 bits of colour or 16-log bits per red, green and blue channel. In earlier digital colorization processes, single hue values were the only way to apply colour to various luminance levels. In the Legend Films process, colour is dithered in a manner that mixes hues and saturations semi-randomly in a manner that makes them appear natural, particularly for flesh colour and different lighting situations [4]. This process sounds complicated the design process.

The major difficulties of manual colorization process are the fact that it is expensive and time consuming process [1]. An artist needs to assign the colour to an image manually by begin to segmentation the image into region and then they can start to assign the colour to each region. This process required more time to colour one image. The cost on hiring an artist is also high since it required skill and patience to colorize an image manually.

In this project, an interactive colorization technique is introduced. This technique required neither precise manual segmentation nor accurate tracking. Users only need to indicate how each region should be coloured by scribbling the desired colour in the interior of the region, instead of tracing out its precise boundary. This technique will automatically propagate colour to the remaining pixels in the image sequence. The result of this project is a simple yet effective interactive colorization technique that drastically reduces the amount of input required from the user.

1.2 Research Background

Colorization was first introduced by a formed NASA engineer, Wilson Markle in 1970 [2]. Markle first task included adding colour to the original Moon landing footage. Soon, he gradually turned to commercial projects and founded Colorization Inc. in 1983. Markle's required a lot of technology, but its underlying concept was simple and elegant. His assistance would make a copy of the film and stored it in computer that would determine the precise shades of gray of every object in a scene. Then, they used a palette over 4000 shades of colour to colour the first frame of each scene. This process was not completely digitized. They had to figure it out what colour should be assigned to certain objects in each scene. Some objects would be assigned through the common sense for example the sky is commonly blue. Studios photographs of productions through the studio costume would help to determine what colour of the costume and prop really was. This process was not cheap as what people would think.

Semi-automatic technique also had been introduced for colorizing a greyscale image by transferring colour from a reference colour image [5]. This semi-automatic technique is known as example-based method. It examined the luminance values in the neighbourhood of each pixel in the target image and transferred the colour from pixels with matching neighbourhood in the references image. The technique works well on images where different colour region give rise to distinct luminance clusters. The user must direct the search for matching pixels by specifying the swatches indicating corresponding region in two images. In other word, the user need to find a reference image to colorize the input image and the reference image must contain the desired colour over region with similar textures to the input image.

1.3 Problem Statement

Most of people have old black and white picture but do not have a tool to re-colour the picture so that the picture will attract the attention of the viewer. Re-colouring is used in special effect and digital photography. In previous method [2], user will re-colour the image manually. Usually editor will edit the picture manually based on information from the picture’s owner. User needs to identify the boundary of pixels and applied the colour within the boundary. This process was expensive and consumed time.

For the semi-automatic technique [5], the artistic control over the outcome was quite indirect. It is because the user needs to find the reference image containing the desired colours and similar texture as the input image. It also had a difficulty to fine-tune the outcome of the selectively in problematic areas. Fine-tune outcome is important in order to get a quality coloured image.

This thesis will discuss on a colorization technique, scribble-based that required the user to assign colour on each region by scribbled in the interior of the image region with the colour they desired. The user is able to choose their desired colour.

1.4 Objective

The objective of this project is :

1) To re-colour old black and white picture to colour picture without edit it manually minimum and create a fully interactive colorization algorithm.
2) To compare the coloured image coloured by scribble-based method colorization with the ground truth image contain the original and create a high quality of image result from this colorization method.

1.5 Research Scope

This project is focusing on scribble-based method of colorization. This method allowed user to input their desired colour to the greyscale image. They needed to scribble desired colour to the each region in the greyscale image. They do not need to determine the boundary of the pixel regions, the program will automatically propagated to the entire region. Colours propagated to the regions are based on the scribble colour done by the user.

The proposed method required user to input two images, greyscale image and greyscale image that had been scribble with the desired colour. Greyscale image is used to calculate the differences between two neighbouring pixels. Once the differences are determined, the boundary of two neighbouring pixels is set. Greyscale image that have been scribble with colour is used to determine the colour desired by user and propagated the desired colour to the entire region.

The bit depth of the greyscale monochrome image can be in 4-bit, 8-bit, 16-bit, 24-bit or 32-bit but in this proposed method is focusing on 24-bit depth only. It is because the program is set up for only one value of colour and 24-bit depth is chose. 24-bit is commonly used for recent computer or monitor and it give more colours compared to 16-bit and below. 32-bit is same as 24-bit since it can support 16,777,215 colours but it has an alpha channel it can create more convincing gradients, shadows, and transparencies that can supports 4,294,967,296 colour combinations.

1.6 Thesis Structure

This project will be discussed about development of digital image re-colouring algorithm. In this thesis, it contain three chapter which are Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 Methodology, Chapter 4 Results and Discussion and Chapter 5 Conclusion and Future Work.

In Chapter 1, it discussed about introduction. It focused on the researched topic which is about colorization method. It also discussed about the research background of the previous method, problem statement and objective of this project.

In Chapter 2, it discussed about the literature review. Previous work on colorization methods are discussed and compared. Method discussed in this chapter is first colorization method by Wilson Markle, example-based method and scribble-based method.

In Chapter 3, it discussed about the methodology of scribble-based method. The ideas came out based on the research of the previous work by comparing those methods and find the best method that will be improved. Each of the main process was explained in details manner.

In Chapter 4, it presented the results of this project obtained from the proposed method. This chapter present the improvement of the proposed method from the first trial until it able to get a refine colour image.

In Chapter 5, it composes the conclusion and future work that could be done for the development of the proposed colorization method. Summary of the proposed method is presented to conclude the project achievement.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Colorization was first introduced by Wilson Markle in 1970. He started off by adding colour to the original Moon landing footage and later he founded Colorization Inc. in 1983. He has been adding to classic black white movie. Since then, a lot of method and algorithm have been developed based on Wilson Markle's original colorization process. The method has improved to semi-automatic technique and automatic technique that helps to reduce editing cost and time consume.

2.2 Early Colorization Process

Early colorization process is introduced by Wilson Markle, a former NASA engineer in 1940 [2]. Adding colour to the original Moon landing footage was Markle's first task in colorization. Original colorization process introduced by Markle was a colour mask is manually painted for at least one reference frame in a shot. To allow colours to be automatically assigned to other frames in regions where no motion occurs was by applying motion detection and tracking. Manual fixing by user was often required to assigned colours in the vicinity of moving edges by using optical flow. The editor would make a copy of the film and stored it in computer that would determine the precise shades of grey of every object in a scene. They had to figure it out what colour should be assigned to certain objects in each scene. Some objects would be assigned through the common sense for example the sky is commonly blue. Studios photographs of productions through the studio costume would help to determine what colour of the costume and prop really was. This process was not cheap as what people would think and it required a lot of time to adding colour to each region of the image.

Figure 2.1: First re-colour image of Moon landing footage

Figure 2.2: A re-colour monochrome old movie

2.3 Example-based Technique of Colorization

Semi-automatic technique is used for colorizing a greyscale image by transferring colour from reference colour image [5]. It involved assigning three dimensional RGB. Greyscale image is a one-dimensional distribution. Both images will be converted to $l\alpha\beta$ colour space. Jittered sampling is used to select small subset of pixels in the colour image as sample. Next, each pixel in greyscale image has been go through in scan line order and select the best matching sample in the colour image using neighbourhood statistics. The best match is determined by using a weighted average of pixel luminance and neighbourhood statistics. The chromaticity values (α, β channels) of the best matching pixel are then transferred to the greyscale image to form the final image. The $l\alpha\beta$ space was developed to minimize correlation between the three coordinate axes of the colour space. The reason the $l\alpha\beta$ colour space is selected in the current procedure is because it provides a de-correlated achromatic channel for colour images. It allowed to selectively transfer the chromatic α and β channels from the colour image to the greyscale image without cross-channel artefacts. To transfer chromaticity values from the source to the target, each pixel in the greyscale image must be matched to a pixel in the colour image. The comparison is based on the luminance value and neighbourhood statistics of that pixel. The luminance value is determined by the *l* channel in $l\alpha\beta$ colour space.

The disadvantage of this method is that the user must find reference images containing desired colour over regions with similar textures to the image the user wishes to colorize. There are some difficulties to fine tune the outcome selectively in the problematic area. Corresponding colours that do not have corresponding luminance values cause this method to fail. The example-based method does not explicitly enforce spatial continuity of the colours and in some images it may assign vastly different colours to the neighbouring pixels that have similar intensities. It also has difficulty to colour certain image that are come in varieties of colour for example a bunch of balloons that have varieties of colours.

(a) (b) (c)

Figure 2.3 : Example of example-based colorization, (a) similar example colour image, (b) target greyscale image, (c) colorized image [5].

Input image	Reference image	Output image
 (a)	 (b)	 (d)
	 (c)	 (e)

Figure 2.4 : Differences of output image if using difference reference image, (a) input greyscale image, (b) and (c) is two different target images, (d) and (e) output image based on target images [5].

Figure 2.3 shows the difference of two output images if different reference image is used in this example-based method. Figure 2.3 (a) is the greyscale image while (b) and (c) are the image used for references. The major colour in reference image (b) is brownish and it causes the output image (d) have more brownish colour compared to output image (e) since it reference image (c) have more blue colour.

2.4 Scribble-based Technique of Colorization

2.4.1 Colorization Using Optimization

Colorization using optimization an approach of scribble-based technique [1]. This technique formulated an optimization problem based on assumption that neighbouring pixel of similar intensity should have similar colour values under the limitation that the colours indicated in the scribble remain the same.

The input greyscale image will be converted to $\boldsymbol{YUV}$ colour space. Y is monochromatic luminance channel while $\boldsymbol{U}$ and $\boldsymbol{V}$ are chrominance channel [1]. Chrominance is the signal used in video system to convey the colour information of the picture and always represent two different components. The components are blue luma and red luma. Luma is the brightness in an image represents the achromatic image. The input is intensity volume $\boldsymbol{Y}(x, y, t)$ while the outputs two colours volume $\boldsymbol{U}(x, y, t)$ and $\boldsymbol{V}(x, y, t)$. To simplify the notation, **r,s** are used to denote (x, y, t) triplets. Thus, $\boldsymbol{Y}(\boldsymbol{r})$ is the intensity of a particular pixel. The constraint that two neighbouring pixels **r,s** should have similar colours if their intensities are similar need to be imposed.

The difference between the colour $\boldsymbol{U}(\boldsymbol{r})$ at pixel r and the weighted average of the colours at neighbouring pixels need to be minimized by using the equation:

$$J(U) = \sum_{r}(U(r) - \sum_{s\in N(r)} \boldsymbol{w_{rs}} \boldsymbol{U}(\boldsymbol{s}))^2 \qquad (2.1)$$

w_{rs} is a weighting function that sum to one. Its value is large when $\boldsymbol{Y}(\mathbf{r})$ is similar to $\boldsymbol{Y}(\mathbf{s})$ and small when $\boldsymbol{Y}(\mathbf{r})$ is different to $Y(\mathbf{s})$. Similar weighting functions are used extensively in image segmentation algorithms which are usually referred to as affinity function.

There are another two weighting functions that are need to be experiment. The simplest function that are commonly used by image segmentation algorithm and is based on the squared difference between the two intensities:

$$\boldsymbol{w_{rs}} \propto \boldsymbol{e}^{-(\boldsymbol{Y}(\boldsymbol{r})-\boldsymbol{Y}(\boldsymbol{s}))^2/2\sigma_r^2} \qquad (2.2)$$

The second weighting function is based on normalized correlation between the two intensities:

$$\boldsymbol{w_{rs}} \propto \mathbf{1} + \frac{\mathbf{1}}{\boldsymbol{\sigma_r^2}} (\boldsymbol{Y}(\boldsymbol{r}) - \boldsymbol{\mu_r})(\boldsymbol{Y}(\boldsymbol{s}) - \boldsymbol{\mu_r}) \qquad (2.3)$$

μ_r is the mean of the intensities while $\sigma_{\boldsymbol{r}}$ is the variance of the intensities in a windows around $\boldsymbol{r}$. By assuming a local linear relation between colour and intensity, the correlation affinity can be derived [6]. It assumes that the colour at a pixel $U(\mathbf{r})$ is a linear function of the intensity $Y(\mathbf{r})$: $U(\mathbf{r}) = a_i Y(\mathbf{r}) + b_i$ and the linear coefficients a_i, b_i are the same for all pixels in a small neighbourhood around $\mathbf{r}$. When the intensity is constant the colour should be constant, and when the intensity is an edge the colour should also be an edge (although the values on the two sides of the edge can be any two numbers). A simple elimination of the a_i ,b_i variables yields an equation equivalent to equation 1 with a correlation based affinity function, while this model adds to the system a pair of variables per each image window. The fact that **r** and **s** are neighbouring pixel is denoted by the notation **r** ∈ N(**s**). Two pixels are defined as neighbour if their image location is nearby in single frame. Let v_x(x, y), v_y(x, y) denote the optical flow calculate at time t. Then the pixel (x_0,y_0 ,t) is a neighbour of pixel (x_1,y_1 ,t +1) if:

$$\left\| \left(\boldsymbol{x_0} + \boldsymbol{v_x}(\boldsymbol{x_0}), \boldsymbol{y_0} + \boldsymbol{v_y}(\boldsymbol{y_0}) \right) - (\boldsymbol{x_1}, \boldsymbol{y_1}) \right\| < \boldsymbol{T} \tag{2.4}$$

The flow field of $v_x(x_0), v_y(y_0)$ is calculated using a standard motion estimation algorithm. The optical flow is only used to define the neighbourhood of each pixel. Given a set of locations $\boldsymbol{r}_i$where the colour are specified by the user $u(\boldsymbol{r}_i)$ =u_i,$v(\boldsymbol{r}_i)$ =$v_i$we minimize $J(U), J(V)$ subject to these constraints. Since the cost functions are quadratic and the constraints are linear, this optimization problem yields a large, sparse system of linear equations, which may be solved using a number of standard methods.

Figure 2.5. : Example of scribble-based technique [1].

This is an example of scribble-based technique as can be seen the most left image, the user scribbled the desire colour to the input greyscale image. The result of colorization is the image in the middle. The system is automatically coloured the image.

2.4.2 Propagation Paths and Chrominance Blending

Scribble-based technique by determined the propagation paths in the image. Its minimized geodesic distances from every scribble. Pixel colour is obtained by blending the scribble chrominance based on the distances from each scribble. The propagation is performed using different techniques such as probabilistic distance transform or based on random walk [7].

Distance transform isolated the scribble colours and computed it distance to all the pixels of the source greyscale image. Probabilistic distance transform is the modified distance transform based on a model of a virtual particle, which performs a random walk on the image lattice. The probability of the transition at the walking particle from a lattice point to a point belonging to its neighbourhood is determined by the Gibbs statistical distribution [8].

Monochromatic image Y is colorized based on a set of n initial scribbles $\{s_i\}, i = 1, \dots, n$, first it is necessary to determine the propagation paths from each scribble to every pixel in the image. To maps a position t in the path to the pixel coordinate, a path from a pixel x to another pixel y is defined as a discrete function $p(t) : [1, l] \rightarrow Z^2$. The position is an integer ranging from 1 for the path beginning $(p(1) = x)$ $to\ l$ for its end $(p(l) = y)$. Also, if $p(i) = a$ and $p(i + 1) = b$, then a and b are neighbouring pixels. The propagation paths from a scribble to every pixel are determined by minimizing a total path cost:

$$C(p) = \sum_{\boldsymbol{i}=1}^{l} \rho\{p(\boldsymbol{i}), p(\boldsymbol{i}+1)\} \tag{2.5}$$

ρ is a local cost between two neighbouring pixels and l > 1 is the path length. The path route depends mainly on how the local costs are computed. The local cost is obtained by projecting the luminance gradient onto a line tangent to the path direction which showed that the cost is proportional to the difference in luminance between the neighbouring pixels [9].

Based on the propagation paths from every scribble, chrominance of each pixel is determined. Its value is computed as a weighted mean of scribbles' colours with the weights obtained as a function of the total path cost. The chrominance is calculated as a weighted mean of scribbles' colours with the weights obtained as a function of the total path cost.

The final colour value $v(x)$ of a pixel x is obtained as

$$v(x) = \sum_{\boldsymbol{i}} v_{\boldsymbol{i}} w_{\boldsymbol{i}}(x) = \sum_{\boldsymbol{i}} w_{\boldsymbol{i}}(x) \qquad (2.6)$$

where $v_{\boldsymbol{i}}$ is the chrominance of $\boldsymbol{i}$-th scribble and $w_{\boldsymbol{i}}(x)$ is its weight in pixel x. We use $Y\ C_r C_b$ colour space and calculate colour values separately for C_r and C_b channels. The weights are obtained as

$$w_{\boldsymbol{i}}(x) = (C_{\boldsymbol{i}}(x) + 1)^{-2} \qquad (2.7)$$

where $C_{\boldsymbol{i}}(x)$ is the total path cost from $\boldsymbol{i}$-th scribble to pixel x.

Chrominance for every pixel is determined once the propagation path is found in order to add the colour to the greyscale image. Based on the distance transform, chrominance blending is perform using the same distance with the path have been optimized. The final pixel chrominance is calculated as a weighted mean of scribbles' colours defined by user and the weights are obtained as a function of the total path cost.

2.5 Summary

The technique chose for this project is scribble-based technique which colorization using optimization. This technique chose because it fulfils the problems face by other techniques which are manually colorization, example-based colorization and colorization using propagations path and chrominance blending. Scribble-based technique has lowest time consuming in adding the colour to greyscale image. It is also user friendly since the user is able to choose their preferred colour for the image. They are also able to reduce the cost of hiring an editor to edit their greyscale image.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In chapter 3, it discussed the methodology of this project. Based on the literature review, there two methods discussed which are example-based method and scribble-based method. These two methods are compared to find their advantages and disadvantages. Both methods have its own effectiveness but based on the comparison, example-based method does not colorize the greyscale image based on the user desired colour. It extract colour from reference image that have the same desired colour and texture. The user needs to provide the reference image while for scribble-based method, the user can easily choose their desired colour and the user only need to scribble the colour in the small part of each region. The proposed method in this project is scribble based method.

3.2 Scribble-based Method of Colorization

This project is mostly based on software, no hardware component are needed for this project. MatLab is the medium used in this project. MatLab is chose because of the problem during development of coding in Microsoft Visual Studio. Microsoft Visual Studio consumed a lot of time to run it and cause lagging.

Since the user needs to scribble the greyscale image with the desired colour, the graphic user interface is needed so that it is user friendly. User able to use other applications to scribble the colour to the image example Photoshop and paint. It is user friendly since the user able to choose their favourite paint program.

The system built only support 24 bit depth for the scribble image and 8 bit depth for the greyscale image. 24 bit depth image used for the greyscale image that have been scribble by the desired colour. 24 bit depth image is used virtually every computer and phone display. 24 bit depth able to give 16,777,216 colour variations. 8 bit depth image is used for the original greyscale image. 8 bit image only contain greyscale colour which only have 256 of colours variation.

All images are in bitmap file. Bitmap image file used because bitmap file save images which are a high quality as it have no loss from compression when it are saved. The image needs to be saved in RGB (3channels) format. Image compression files format need to be avoid because this kind of image file have some of its data.

Figure 3.1 shows the flow chart of the main system for this project. The system will start by input two same greyscale images, one is the original greyscale image and another is the same greyscale image that has been scribble. These two images are input in the Matlab code. The program does not have a graphic user interface that's allowed user to upload the images to the program.

Any paint program can be used by the user for examples are Photoshop, paint and etc. Microsoft Paint is used as a tool to scribble the colour to the greyscale image. The image files used are in bitmap file format and have 24 bit depth for the scribble image while 8 bit depth image for the original greyscale image. Bitmap file is used since it save images which are a high quality as it has no loss from compression when it is saved. The image is set with the same dimension which is 320×256.

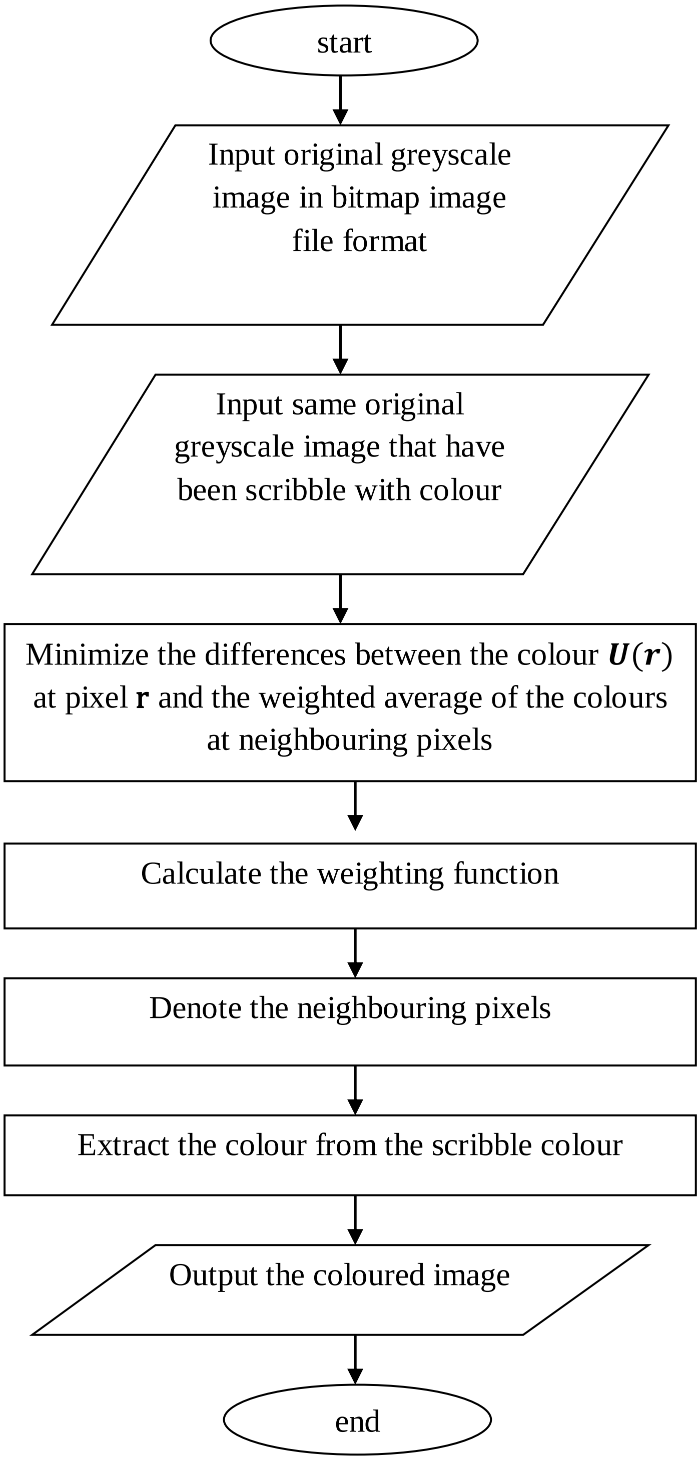

Figure 3.1 : Flow chart of the main system

The original image used is the colour image. This colour image is converted to greyscale image by using an online application. The image also converted to the same dimension, 320×256. The coloured image is used later to compare the result of the colorization with the original colour image.

Based on the original image, colour code in each region is determined by using an online application. The colour scribbled on the greyscale image has the same colour code as in the original image.

The differences between the colour U($\mathbf{r}$) at pixels $\mathbf{r}$ and weighted average of the colours at the neighbouring pixels are minimized. Then the weighting function is calculated by using two equations and the neighbouring pixels are denoted. The colours scribble in the image extracted into the pixels in each region. The output of coloured image is shown in the new windows of MatLab and it able to be saved in personal file.

3.3 Minimize the Differences

The difference between the colour $U(\boldsymbol{r})$ at pixel $\boldsymbol{r}$ and the weighted average of the colours at neighbouring pixels need to be minimized by using the equation 2.1. $w_{\boldsymbol{rs}}$ is a weighting function that sum to one. When $Y(\mathbf{r})$ is similar to $Y(\mathbf{s})$, the value of $w_{\boldsymbol{rs}}$will be large shows that $Y(\mathbf{r})$ and $Y(\mathbf{s})$ have the same intensities. When $Y(\mathbf{r})$ is different to $Y(\mathbf{s})$ the value of $w_{\boldsymbol{rs}}$will be small shows that $Y(\mathbf{r})$ and $Y(\mathbf{s})$ have different intensities. Similar weighting functions are used extensively in image segmentation algorithms which are usually referred to as affinity function.

3.4 Weighting Function

There are two weighting functions that are needed to be experimented.

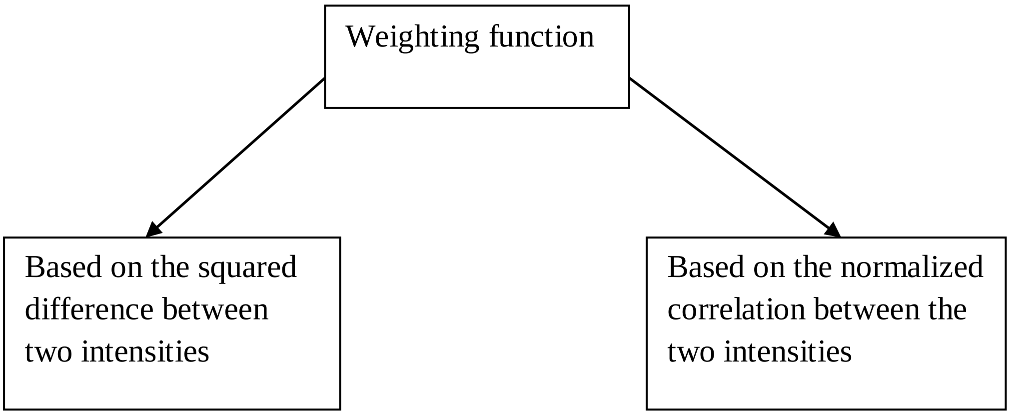

Figure 3.2 : Type of weighting function

Figure 3.2 shows type of weighting function. The simplest function that is commonly used by image segmentation algorithm and is based on the squared difference between the two intensities and this is given by equation 2.2. The second weighting function is based on normalized correlation between the two intensities by using equation 2.3. By assuming a local linear relation between colour and intensity, the correlation affinity can be derived. It assumes that the colour at a pixel $U(\mathbf{r})$ is a linear function of the intensity $Y(\boldsymbol{r})$: $U(\boldsymbol{r}) = a_i Y(\boldsymbol{r}) + b_i$ and the linear coefficients a_i, b_i are the same for all pixels in a small neighbourhood around **r**. When the intensity is constant the colour should be constant, and when the intensity is an edge the colour should also be an edge (although the values on the two sides of the edge can be any two numbers). A simple elimination of the a_i ,b_i variables yields an equation equivalent to equation 1 with a correlation based affinity function, while this model adds to the system a pair of variables per each image window.

3.5 Denote the Neighbouring Pixels

The fact that **r** and **s** are neighbouring pixel is denoted by the notation **r** ∈ N(**s**). Two pixels are defined as neighbour if their image location is nearby in single frame. Let v_x(x, y), v_y(x, y) denote the optical flow calculate at time t. Then the pixel (x_0,y_0 ,t) is a neighbour of pixel (x_1,y_1 ,t +1) if:

$$\left\|\left(\boldsymbol{x_0} + \boldsymbol{v_x}(\boldsymbol{x_0}), \boldsymbol{y_0} + \boldsymbol{v_y}(\boldsymbol{y_0})\right) - (\boldsymbol{x_1}, \boldsymbol{y_1})\right\| < \boldsymbol{T} \qquad (3.1)$$

The flow field of $v_x(x_0), v_y(y_0)$ is calculated using a standard motion estimation algorithm. The optical flow is only used to define the neighbourhood of each pixel. Given a set of locations $\boldsymbol{r}_i$where the colour are specified by the user $u(\boldsymbol{r}_i)$ =u_i,$v(\boldsymbol{r}_i)$ =$v_i$we minimize $J(U)$,$J(V)$ subject to these constraints. Since the cost functions are quadratic and the constraints are linear, this optimization problem yields a large, sparse system of linear equations, which may be solved using a number of standard methods.

Once it has set the boundary of each pixel in the image, the colour scribble is extracted into each pixel. In image segmentation algorithms is based on the normalized cuts that attempts to find the second smallest eigenvector of the matrix $D \times W$ where W is a *n pixels* × *n pixels* matrix whose elements are the pair wise affinities between pixels and D is a diagonal matrix whose diagonal elements are the sum of the affinities. These algorithm minimizes the same cost function but under different constraints.

3.6 Normalized Cut and Segmentation

There are many possible partitions of the domain I if an image into subsets and there are two aspects to be considered. The first is Bayesian view is appropriate which are several possible interpretations I the context of prior world knowledge. The difficulty is in specifying the prior world knowledge. Coherence of brightness, colour, texture, or motion are low level but equally important is mid- or high level knowledge about symmetries of objects or object models.

Second aspect is the partitioning is inherently hierarchical. Therefore, it is more appropriate to think of returning a tree structure corresponding to a hierarchical partition instead of a single "flat" partition [10]. Image segmentation based on low level cues cannot and should not aim to produce a complete final correct segmentation. The objective should instead be to use the low-level coherence of brightness, colour, texture, or motion attributes to sequentially come up with hierarchical partitions.

Grouping Algorithm :

Grouping algorithm consists of the following steps:

1. Given an image or image sequence, set up a weighted graph $G = (V, E)$ and set the weight on the edge connecting two nodes to be a measure of the similarity between the two nodes.

2. Solve $(\boldsymbol{D} - \boldsymbol{W})\, x = \lambda\, Dx$ for eigenvectors with the smallest eigen values.

3. Use the eigenvector with the second smallest eigen value to bipartition the graph.

4. Decide if the current partition should be subdivided and recursively repartition the segmented parts if necessary.