**CONTENTS**

1. Introduction
2. Literature Survey
3. Algorithm
   1. About the algorithm
   2. Existing algorithms
   3. Feature Point Detection model
   4. Steps of the algorithm

3.4.1. Apply the Difference mask with Threshold parameter P1

3.4.2. Apply the Pseudo Gaussian Mask to the Points Satisfying the First

Difference Operator

* + 1. Compute Information Content of Feature Point

1. Prototype Description and Requirement
   1. Prototype
   2. Requirements
2. Summary and Conclusion
   1. Summary
   2. Conclusion
3. **Introduction**

Colorization is a computer aided process of adding colors to a grayscale image or videos.

There was a time when all images were solely grayscale due to limitation in technology. Color images always provide more clear information than grayscale images. Coloring of old Black and White movies and rare images of monuments, celebrities is one of the interesting applications of colorization of gray scale images. The color details in the images can be utilized for analysis and study of particular image in the applications like medical tomography, information security, image segmentation, etc.

Many techniques have been proposed to perform the task of coloring grayscale image. But all of these techniques have inherent drawback of needing certain amount of human interaction such as selecting a color from color palette, choosing a seed pixel and segmenting the regions of image for colorization. The main purpose of this algorithm is to reduce human interaction and achieve the effect of colorization of grayscale images. All that is needed is a source image of similar feature as of input grayscale image to be colorized. Also the hindrance of needing source color image to be bigger than the target to be colored grayscale image is removed by use of Vector Quantization based on colorization process discussed here. Colors perceived in an object are determined.

The steps of the algorithm are:

1. Input images

We input the image grey scale image to be colored and a reference image for the color transfer.

1. Vector Quantization

Used for the data compression i.e. mapping of k-dimensional vector space to finite set.

1. Fast codebook generation

Codebook refers to the mapping of the color pixels to the target grayscale image from the source color image. To generate a good codebook we use Kekre’s Fast codebook generation algorithm.

1. Colorization

Transfer of the colors from source refenece image to target image.

1. **Literature Survey**

2.1- Dr. H. B. Kekre, Dr. Tanuja K. Sarode, Sudeep D. Thepade, Ms. Supriya Kamoji, “Analysing Assorted Window Sizes with LBG and KPE Codebook Generation Techniques for Grayscale Image Colorization”, *(IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 6 June 2011*

Abstract-

This paper presents use of assorted window sizes and their impact on colorization of grayscale images using Vector Quantization (VQ) Code Book generation techniques. The problem of coloring grayscale image has no exact solution. Attempt is made to minimize the human efforts needed in manually coloring grayscale images. Here human interaction is only to find reference image of similar type. The job of transferring color from reference image to grayscale is done by proposed techniques.

2.2- H. B. Kekre, Tanuja Sarode, Sudeep D. Thepade, Supriya Kamoji, ”Performance Comparison of Various Pixel Window Sizes for Colorization of Grayscale Images using LBG, KPE, KFCG and KEVR in Kekre’s LUV Color Space”,International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-2, December 2011

2.3- Vivek George Jacob, Sumana Gupta” COLORIZATION OF GRAYSCALE IMAGES AND VIDEOS USING A SEMIAUTOMATIC

APPROACH”, *978-1-4244-5654-3/09/$26.00 ©2009 IEEE*

Abstract-

Colorization is a computer-aided process of adding color to a grayscale image or video. The task of colorizing a grayscale image involves assigning three dimensional (RGB) pixel values to an image which varies along only one dimension (luminance or intensity). Since different colors may have the same luminance value but vary in hue or saturation, mapping between intensity and color is not unique, and colorization is ambiguous in nature, requiring some amount of human interaction or external information.

This paper proposes a semi-automatic process for colorization where the user indicates how each region should be colored by putting the desired color marker in the interior of the region. The algorithm based on the position and color of the markers, segments the image and colors it. In order to colorize videos, few reference frames are chosen manually from a set of automatically generated key frames and colorized using the above marker approach and their chrominance information is then transferred to the other frames in the video using a color transfer technique making use of motion estimation.

2.4-Yogesh Rathore, Avinash Dhole, Ramnivas Giri, Umesh Agrawal, “Colorization of Gray Scale Images using Fully Automated Approach”,IJECT Vol. 1, Issue 1, December 2010, I S S N : 2 2 3 0 - 7 1 0 9 (Online)

Abstract-

This paper proposes a method for adding colors to grayscale images without human interference. In contrast to many previous computer-aided colorizing methods, which require intensive and perfect human interference, this method needs only the user to provide a target gray level image for the process of ‘colorization’, a colorful image of the similar content as the grayscale image is automatically retrieve from the database of images, as an input source image. Then, the best matching source pixel is determined using luminance and texture matching procedure, for each pixel of the target image into a perceptually de-correlated color space. Once a best matching source pixel is found, its chromaticity values are assigned to the target pixel while the original luminance value of the target pixel is retained.

**3. Algorithms**

**3.1 Introduction.**

Under this seminar we present three main approaches for colorization of grayscale image. The input, way of processing etc., considerably differ for these algorithms. These algorithms are:

1) Colorization using Scribbling.

2) Colorization using color transfer from source color image.

3) Colorization using source color image and fast codebook generation.

In the upcoming text, the brief description of these algorithms is given.

**3.2 Previously used approaches.**

Before going into depths of these algorithms, let us first know about previous methods used for this problem. Some of these methods are:

* Hand coloring : In which colorization was done using manual painting and software tools were used for coloring. Here no specific algorithm was used. Almost all work was done by user, although use of software gives more sophisticated results. But colorizing long videos was exhaustive work.

**3.3 Colorization by using scribbles.**

**3.4 Colorization using color transfer.**

Although previous method gives accurate results, there are some drawbacks from computation point of view.

Main drawbacks:

1. Very high quality user interface is required: The input is given graphically from user in the form of scribbles. It is essential and difficult to determine exact orientation of such random form.
2. More user interaction is required.

This approach tries to reduce these drawbacks. Following is the brief description of this algorithm.

Input: Grayscale image or target image, a source color image or source image.

Output: Colored version of target image.

Following steps are then taken:

* Convert source image to decorrelated lαβ color space

– l: luminance

– α, β: chromatic channels (yellow/blue and red/green).

* Perform luminance remapping (histogram matching)
* Take ~200 color samples from the source image
* For each pixel in the target image (in scan-line order):

– Find best matching source pixel (compare luminance and std.dev. of luminance values in neighborhood.)

– Transfer color from source pixel to target pixel.

These steps can be explained as follows:

1. First, both images are converted to decorrelated color space. The color space used is lαβ color space. This color space is called as decorrelated, because changes made in one channel minimally affect other channels.
2. Then luminance values of source image are linearly shifted to bring them into range of target image. This is essential because color transfer is done on the basis of luminance values.
3. Then various samples of source image are taken. Many sampling methods are developed of which grid sampling is generally used.
4. Then for each pixel of target image:

-It’s luminance value is compared with samples of source image.

-Matching is done based on luminance and std. deviation values.

-For most appropriate match, the color info of source pixel is transferred to target pixel.

In this way colorization is performed.

**3.5 Colorization using fast codebook generation algorithm.**

The disadvantage of above algorithm is more processing time and selectively good performance. To overcome these, following algorithm is developed. The main concepts related to algorithm are as follows.

**3.5.1 Vector quantization.**

Vector Quantization is an efficient technique for data compression. VQ can be defined as the mapping function that maps k-dimensional vector space to the finite set CB = {C1, C2, C3,...,CN}. The set CB is called codebook consisting of N number of code-vectors and each code-vector, Ci = {ci1, ci2, ci3,……,cik} is of dimension k. The key to VQ is the good codebook.

**3.5.2 Fast codebook generation.**

Now how to generate the codebook? For this purpose the following algorithm is used, which is proved to be of better performance than other ones.

Following are the steps:

1) Initially we have one cluster with the entire training vectors and the code vector C1 which is centroid.

2) In the first iteration of the algorithm, the clusters are formed by comparing first element of training vector with first element of code vector C1. The vector Xi is grouped into the cluster 1 if xi1< c11 otherwise vector Xi is grouped into cluster 2 as shown in Figure 1.a. where code vector dimension space is 2.

3) In second iteration, the cluster 1 is split into two by comparing second element xi2 of vector Xi belonging to cluster 1 with that of the second element of the code vector. Cluster 2 is split into two by comparing the second element xi2 of vector Xi belonging to cluster 2 with that of the second element of the code vector as shown in Figure 1.b.

4) This procedure is repeated till the codebook size is reached to the size specified by user.

It is observed that this algorithm gives less error as compared to LBG and requires least time to generate codebook as compared to other algorithms, as it does not require any computation of Euclidean distance. The algorithm shown in Figure 1.a. and Figure 1.b. for two dimensional case it is easily extended to higher dimensions.

****

****

**3.5.3 Colorization technique:**

The actual colorization is performed in two steps. Firstly the color palette is generated for source color image using codebook generation explained above. A color palette is nothing but compressed version of source image. The purpose of this step is to obtain this compression or color palette for later comparisons. Now it is clear that comparisons can be performed at faster rate. However, the color palette generated must be a perfect compression of source color image.

In the next step we have color palette and target image. Now we find match for every pixel of target image into color palette. Once the best match is obtained, the color info is transferred to target image. Following is the brief outline of these steps:

**A) Color palette generation using FCG.**

The steps generates color palette as the VQ codebook of source color image.

1) This source color image is divided into pixel windows of size 2x2 (each pixel consisting of red, green and blue components).

2) These are put in a row to get 12 values per vector (as 4 sets of Y, Crg and Crb values in YCrgCrb color space or 4 sets of R, G and B values in RGB color space or 4 sets of red, green and blue values in RGB color space ). Collection of these vectors is a training set (initial cluster).

3) The Fast codebook generation algorithm is applied on this initial training set to obtain the codebook of specific size (here four sizes are considered 64, 128, 256).

**B) Colorization.**

The target grayscale image is divided into pixel windows of size 2x2. These 4 values are put into the row and are compared with GR component of all the code-vectors in RGB color space, with Y component of the all the code-vectors in YCrgCrb color space and with average of RGB for each of the four pixels of the code-vector in RGB color space.

The closest match in the color palette is determined by calculating the Euclidean distance between Y or Average RGB of four values in color palette (Codebook) and grayscale pixel window values from the grey image.

Then the color transfer is done from source pixel to target pixel.

These are the main algorithms/approaches presented in the seminar.

**4. Prototype Description.**

**4.1 Brief idea.**

The prototype which is presented under this seminar is small code written in Matlab that performs colorization using concepts of above algorithms. Input is source color image and target grayscale image. The code uses brute-force approach. Initially it converts source color image into grayscale version. Then it considers every pixel of target image in scan-line order. For every target pixel, it scans entire grayscale source image until match is found. Once match is found it uses color info of that pixel.

**4.2 General outline.**

Prototype works in following way.

1. First source and target images are read.
2. The grayscale form of source color image is obtained (say, x).
3. For each pixel of target image:

- Check every pixel of ‘x’ until best match is found.

- Once match is obtained, the color info of that pixel in source image is transferred to target image. The go to next pixel of target image.

**4.3 Proposed improvements.**

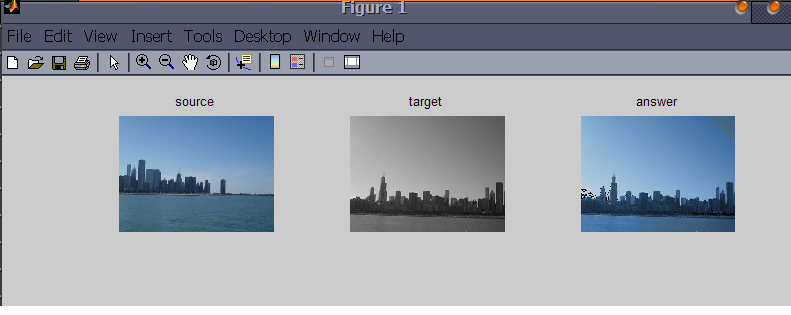
1. Sampling may be used, in order to improve the speed of finding best match.
2. Preprocessing may be used to improve the results.
3. Introduction of user interaction.

Even if the prototype can be improvised in many ways, it was enough to provide the basic idea of project and various details of the concerned subject.

The working of the prototype is as shown below:

In each of the following screenshots, the source and target are provided as inputs and answer is the colored version of the target image generated by program.

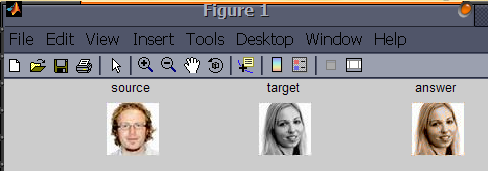
**1)**

****

**2)**

****

**3)**

****

**4)**

****