Assignment 1

Group Members: Shubham Sarda(2018TT10958), Rohan Sharma(2018TT10909)

PART A

In this part we try to quantize the image using various techniques:

Popularity Algorithm

Here we first create a dictionary with different pixels (stored as tuple) as the key and number of pixels of that type in the image as values. Then we sort the pixels in decreasing order of number of times they appear on the image. We take the first k colours where k is a parameter given to the function for quantizing using popularity algorithm. Next, for every pixel in the image we replace that with pixel it is closest (Euclidean distance used as the measure of distance) to in the colour palette of k colours obtained. In our example we used k=64.

Median Cut Algorithm

In this algorithm we also, we chose a parameter k for the number of colours to be used in the quantized image. Here k must be a multiple of 2 as we recursively divide the colour space into 2 regions. First we find the colour channel (R or G or B) with maximum range and then sort the colour space with respect to that channel. We then divide the colour space in two regions based on the median of the sorted colour space. The pixels in the colour subset to left of the median are assigned a pixel value which is average of all the pixels in that subset. Same is done for the right subset. So now we have 2 subsets. Depending on value of k keep on recursively dividing the left and right subsets into smaller subsets. In our example we used k=16.

Floyd Stein Algorithm

We try to improve the quality of image by using Floyd stein algorithm. For every pixel in the image, we find the error between the pixel and its closest counterpart in the colour palette (found by median cut or popularity algorithm). We then propagate these errors in the neighbourhood using appropriate weights

(Error)e = f(i,j)-C(i,j)	f(i,j+1)+=3/8e
f(i+1,j)+=3/8e	f(i+1,j+1)+=1/4e

Results

We get better results with median cut algorithm compared to popularity algorithm even with lesser number of colours in the median cut algorithm. We also find that median cut algorithm takes lesser to complete than the popularity cut. Further on dithering, we see a significant improvement when using the colour palette obtained by median cut algorithm. When dithering is combined with the popularity algorithm, there is a deterioration in the image.

Original Image

Median Cut Dithered (k=16)

Median Cut (k=16)

Popularity Dithered (k=64)

Popularity Algorithm (k=64)

PART B

In this part we try to transfer colour to target grayscale image from a source (coloured) image.

1) Global Transfer

We have the grayscale and the coloured image. We convert coloured image to (L, alpha, beta) space from rgb. For finding L (luminance) value of grayscale image, we simply do a rescaling of intensity (0 to 255).

L=I*100/255 [L lies from 0 to 100]

We then do a luminance remapping of source image with respect to target image so as to shift and scale the luminance histogram of source to match with that of target. We then use jittered samples (600-700) to transfer colour values i.e. the alpha and beta values from coloured to target. We find

the best matching pixel using equally weighted average of luminance and standard deviation across a 5*5 neighbourhood. We assign alpha and beta values of best matching pixel from the jittered samples to every pixel in the grayscale image. The luminance values are kept intact. Following this the coloured target image is again converted to rgb space.

2) Swatch Transfer

This is a more user involved process. We have selected 3 appropriately placed swatches (to cover all colour shades) of size 50*50 in source and target swatch image. We transfer colour to the target swatches using the above-mentioned global transfer algorithm. Now colour values (alpha and beta) are assigned from these target swatches to the rest of the image. To colour the remaining pixels, we use texture matching as a similarity metric. This is done by taking the mean of pixel-by-pixel L2 distance between the luminance values of 5*5 neighbourhood of swatch region and uncoloured pixels. We find this for all pixels across the swatches and choose the pixel with least error to assign alpha and beta value for the uncoloured pixels. The image is then converted to rgb space again. We've optimized the results by optimizing swatch size, location, jittered sample length and the neighbourhood size for various operations.

Source Swatches:

1)  2)  3) 

Corresponding Target Swatches

1)  2)  3) 

Results

Swatch transfer gives much better results compared to the global transfer algorithm.

Source Image

Target Image

Coloured Target using Global Transfer

Coloured Target using Swatch Transfer