

Color Mapping to a Limited Palette and Generating Palettes with Median Cut

Description:

This project is a program that can map an image to a user-provided color palette in the form of an image or a text file of hex codes. In addition to mapping colors to palettes provided by the user, it can also generate palettes of its own. The program uses a method called median cut to generate a limited palette which can then be mapped to the image.

Methodology:

Color Mapping:

Mapping colors to an image is a fairly simple task. For each pixel in an image, its color is checked against each color in the palette and the “closest” one will be substituted. The distance between colors is defined by the following equation:

$$D = \sqrt{(C_r - P_r)^2 + (C_g - P_g)^2 + (C_b - P_b)^2}.$$

Palette Generation:

This program uses the median cut algorithm to generate palettes based on an image. This algorithm takes the pixels in an image and places them into a bucket. It then finds the range of each color channel and sorts the bucket in ascending order by the largest range found. Finally it splits the bucket in half down the middle (hence the “median cut”) and repeats the process for each smaller bucket. This is repeated until the desired amount of buckets is achieved. Each bucket then has its colors averaged into one, these average colors are the generated palette.

This program in particular uses selection sort for the sorting of each bucket, this unfortunately can lead to lengthy generation times. In the example below a roughly 600 by 600 image took roughly 8 minutes to have an 8 color palette generated. Even a smaller palette will take around 5 minutes with an image of this size.

Examples:

Base Image:



Simple 8 color palette:



2 color generated palette



4 color generated palette



8 color generated palette



Extra:

Originally I wanted this project to also include dithering. I attempted to implement Floyd-Steinberg dithering, which shifts the difference between old and new colors to neighboring pixels, giving the illusion of a larger color palette. Despite the code not working I left in, as it can produce some interesting images on its own. The examples below are from various stages of debugging the dithering function.

