

Visualizing a Dynamic Color Table

In previous exercise, we have implemented the Oct Tree algorithm for indexed color. The C++ program will generate a reduced color table for the colors in an image.

I will represent the color table as a matrix, with those indexed colors as its elements. I will traverse the color tree not only at the end, but also for all the steps. After getting all the color tables for each moment, I will be able to generate an animation using Movie Maker. I'd like to show you how colors are added or merged during this process. The final animation can also be a vivid demonstration for the Oct Tree algorithm class.

The image files I use are the ones for indexed color project. Their pixel dimension is 400 x 300. So one image has 120000 pixels in total. If we use a 256-color table, there will be 120000 adding operations and at most $120000 - 256 = 110744$ merging operations (Pigeonhole Principle).

If we record the color table for each step, the data file will be very large. I tried to print all the color tables to separate files, but got some fstream problems when generating so many small files. Therefore, I printed them to a single file.

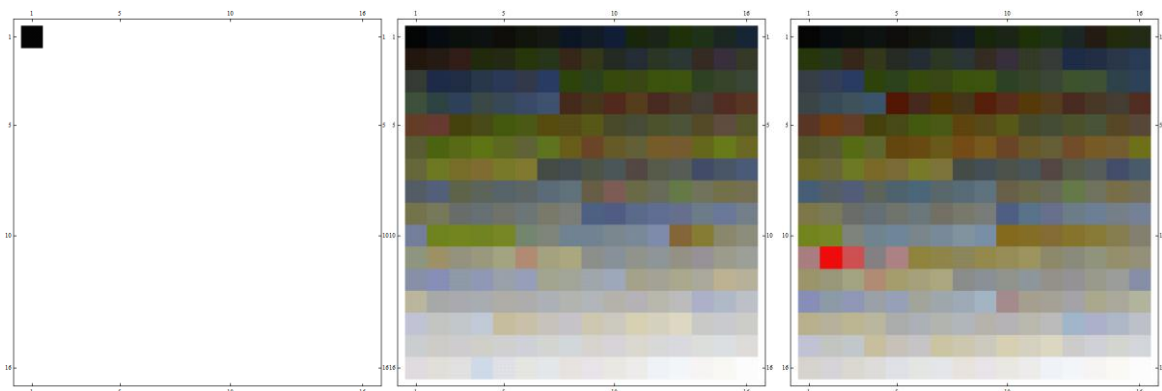
The structure of the data file is:

{List of {List of {R, G, B} for all the indexes} for all the steps}

So the dimension is steps x indexes x channels.

I have to skip some steps to reduce the file size. I coarse grained the time step by updating the table every 5 steps before first filled up, and update every 100 steps afterwards. So I got a 5M data file in the end.

I imported the data into Mathematica, and plotted out the color table. The first and final samples are:



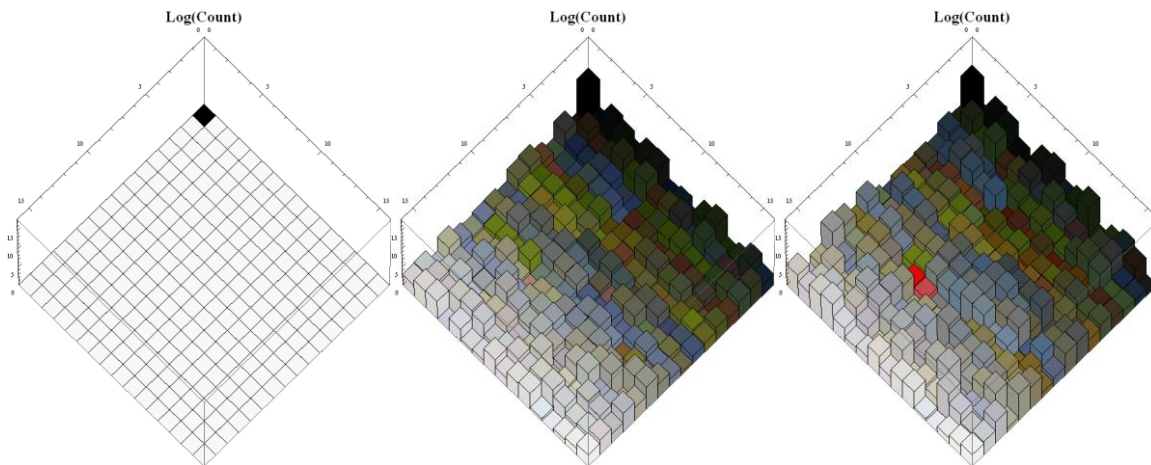
for the WhiteRose. The animation is attached in the link:

<https://dl.dropboxusercontent.com/u/35302248/work/rose2D.wmv>

Downloading the file will take some time. We can see how the color table changes when adding new colors or merging existed ones.

Later I decided to show the statistics with the color table. The only change is to add one channel after the RGB for the counts, which is the numvisits of that node. Then use colored cuboids for the matrix elements, with height representing the counts.

Because the colors distribute over a broad range, I took the logarithm of the counts to balance both the large and small counts. The first and final samples are:



for the WhiteRose. The animation is attached in the link:

<https://dl.dropboxusercontent.com/u/35302248/work/rose3D.wmv>

In this 3D representation, the color table looks like hills or waves changing its shape dynamically.

I also made the animation for the two boys. The links are:

2D <https://dl.dropboxusercontent.com/u/35302248/work/boys2D.wmv>

3D <https://dl.dropboxusercontent.com/u/35302248/work/boys3D.wmv>

I use Mathematica only because I'm more experienced in it. It can be written in Matlab or C++ for compatibility. I may try to write the color table in binary to reduce file size and allow more steps to smoothen the animation. An animation with an actual tree structure may help to show the process even better, but it requires more efforts.