Henry Wandover

Lab Report

**COMPRESSION VIA DEPTH**

The quantitative way I came up with to compare data savings with the compressed version of the tree versus the original is that the compressed ColorGrids or images are based on region. Rather than the original image which would have to be drawn pixel by pixel. By compressing the image, there are now a bunch of little regions of pixels that all share an average color which is significantly better than that of a difference pixel to pixel.

I used the picture of Chris Cuomo holding the barbell because it was on my desktop and seemed remotely complex with having a person. I saw at 16 n that the image that I was drawing was now satisfactory, and on my scale of quantitative data saving, if I was going to compare redrawing this with the assistance of the regions of the K-d tree versus the original—it will be much faster. As stated earlier, the original image that is read has to be drawn to the ColorGrid, cg.set(row, col), one pixel at a time read through RGB. While with this satisfactory tree representation, there are a bunch of regions which have to be cg.set with the average color but isn’t going one at a time like previously.

As we get a clearer it will take longer but still should be faster than the original representation as explained already.

**SAVE & LOAD METHOD**

In my testing I was able to write a copy method that could copy all the partitions in a KDTree and then when inputed into a version of the BuildImageTree method would be able to perfectly make the same image. That was my idea going into making the Save and Load methods, what save does is basically copy all the partitions values in level based order, i.e. level by level in order. I was able to successfully in binary write these into my .bin file, the problem I ran into was loading said file because we never discussed serializing at all in class so I went in blind. I got close a couple times but ending up sinking hours into this and nothing that produced a positive result. However, I know that my conclusion on how to save and load does work because of my testing with the copy method which I can provide, I just need help with serializing files. But by passing partitioned lines we can replicate all the same regions and create the same image without needing to remember all data from the KDTrees.