Week 4

This week, we have gotten a lot done. We researched how to get more colors into our program. We want more colors so customers can see a more detailed analysis. The reason we I say, customers, is because we are trying to sell this idea, not just created because it's a project that we will get graded on. If we treated like this project was to get a good grade on the final then we would have low expectations. If we treated it like we were selling this, then we would be able to put more effort. So to make it a product worth selling, we included more colors to get a more detailed analysis. So far what we did was, we researched the RGB values of different colors, then we looked at the proportions of each RGB value to get it to display specific colors. Yao did a very nice job while researching the different color values. All we had to do essentially was to find more colors to add and then use the proportions. Wasn't too difficult, but was time-consuming as we had to make sure the values were correct. Next, We contacted the creator of Disney Animation. He was able to show us how he got the color line to show up for his video. It wasn't easy to get a response from someone who is very popular. However, with an extensive 2 hours just to come up with what to write on the email body, and subject, we were finally able to get his attention and give us part of his code! Here is the email we got back from him:

I think this blog post will answer all or most of your questions:

<http://home.theodoregray.com/blog/2013/08/13/a-visit-to-disneys-magic-kingdom?rq=lion>

(If you’re seen it already, go back and look again, because I just fixed all the broken links and videos.)

I used Mathematica, so my code would not work for you in Matlab, but in case it’s useful, this is a stripped-down version of the core function, which takes frames from the movie and identifies the color clusters, then plots them in 3D. The blog post explains how this is used to create the overall color maps. As you will see from the blog post, this is all about clustering, which is done by magic in the FindClusters command. I have no idea how that function works, because it’s built in to Mathematica.

data = Flatten[  
   ImageData[Import["Lion\_King.SceneColorMaps/077750-078003.png"]], 1];  
  
dataCut = data[[1 ;; -1 ;; 10]];  
  
clusters = FindClusters[data];  
  
colorGroups = Map[Mean, clusters];  
  
Graphics3D[{  
  Point[dataCut, VertexColors -> dataCut],   
  Map[{Apply[RGBColor, #], Sphere[#, 0.05]} &, colorGroups]}

Hope that helps,

Theodore

Thirdly, we had to get the video processing part done. Trush had the video manipulated since week 1, and had the video displayed on the figure. All we had to do now was to add the data analysis part to the program. Shane was able to get the original video to show up on the top left and the manipulated video to show up on the right. Below both video is a bar graph that shows the color proportions for every frame until the video was over. Always it calculated the differences live! Below the bar graphs, we had the 3D cluster to show up. It would show the 3D cluster being updated every frame for the original video and manipulated video. Also we included the color line variation in order to have it look similar to the Disney Animated app created by Theodore Grey. We struggled with this, but the man himself was able to help us create it! Here are the links to the Manipulated video Data Analysis: <https://www.youtube.com/watch?v=2DLCQYCy_aw> <https://www.youtube.com/watch?v=oD7uPBSo3xM>

Currently, we are working with Arduino in order to display color proportions received from MATLAB. Unfortunately, MATLAB requires us to purchase a subscription in order to let MATLAB communicate with Arduino. Therefore, we thought outside the box and got MATLAB to send a text file of the output, and have it sent to JAVA. Now that we got the proportions in JAVA, we can now allow JAVA and Arduino to communicate with each other. So far we have got the servo to spin at certain angles. All that remains is just to have the Arduino get actual data from JAVA and have it applied to Arduino. Trush got the Arduino to move at a certain degree, and Shane got the color proportions from MATLAB to Java. We sort of switched roles. Trush was good with Arduino and Shane was good with JAVA. So Trush was basically the lead programmer and Shane was the documenter.