**CMPT365 Project Report**

Yubo Hu yuboh@sfu.ca

Chauncey Liu cla284@sfu.ca

**1. Introduction**

The project is about finding and characterizing video transition. We tried to find out if there is a wipe in a short clip of video by producing its STIs(“spatio-temporal” image) and look for edges in them.

Sometimes it’s very hard to see the edge in a raw STI when the two clips of video have frames with similar color. So we tried to calculate the histogram difference of color to plot the STIs rather than using just the raw pixel data. It turns out to be cleaner especially after the threshold is added. After that we use Hough Transform method to find out the line in the histogram difference STI and then determine what kind of a wipe it is.

**2. Histogram Differences**

We implement both methods introduced in the project description: histogram intersection method and IBM histogram

difference **.** Methods can be switched by choosing ‘IBM model’ or ‘Minimum method’ in pop-over the menu in GUI we provide before clicking ‘Generate Result’ button to run the code.

The performances of both methods are quite good, and they produce similar results and conclusions too.

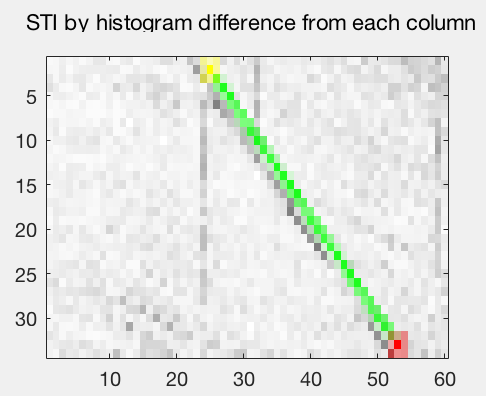
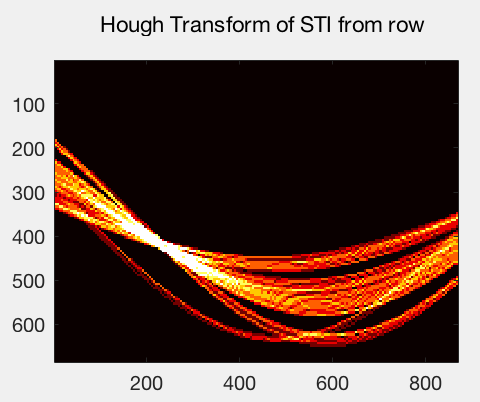
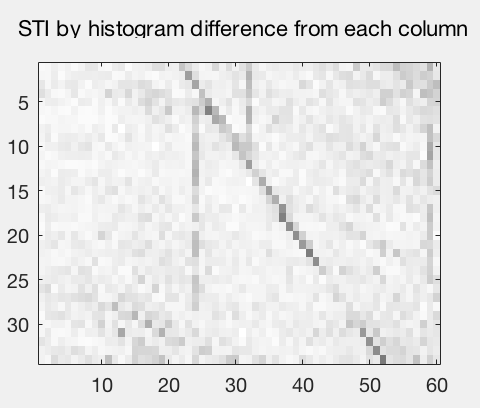
**3. Using Hough transform to find edge automatically**

After applying IBM model or histogram intersection, we can get an STI by Histogram Differences, in which an edge can be clearly found. Then we use Hough transform to find that edge automatically.

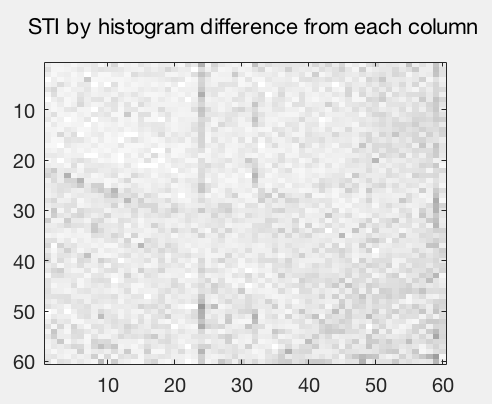
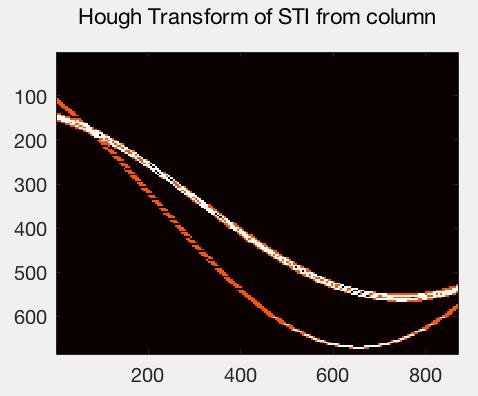
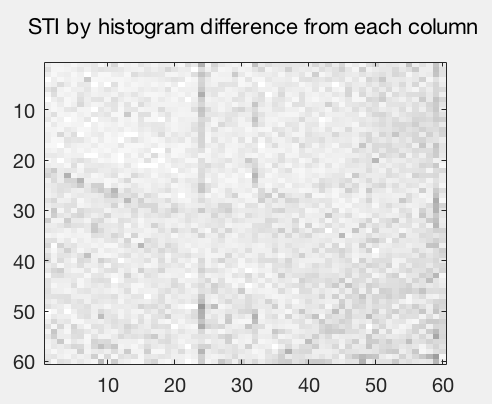
The Hough transform is a feature extraction technique to find edges in an image. In brief, the straight line y = k0\*x + k1 is represented as a point (k0, k1) in the image generated by Hough transform. If there are more curves go through a point, the straight line that point represents is more likely to be in the original image.

In general, the Hough transform can find multiple edges in the image. However, in this project we expect to get only one edge which is most clear to find. And we also set the minimum length of edge so as to get a better result.

The first image below is the STI by Histogram Differences, in which we need to find the position of the edge. The second one is the image generated by the Hough Transform, in which we can find a clear intersection point (k0’, k1’). Then we draw the straight line y = k0’\*x + k1’ on the image and that perfectly finds the edge.



If there is no clear edge in the image, the Hough transform will not be able to find an edge.(As below)

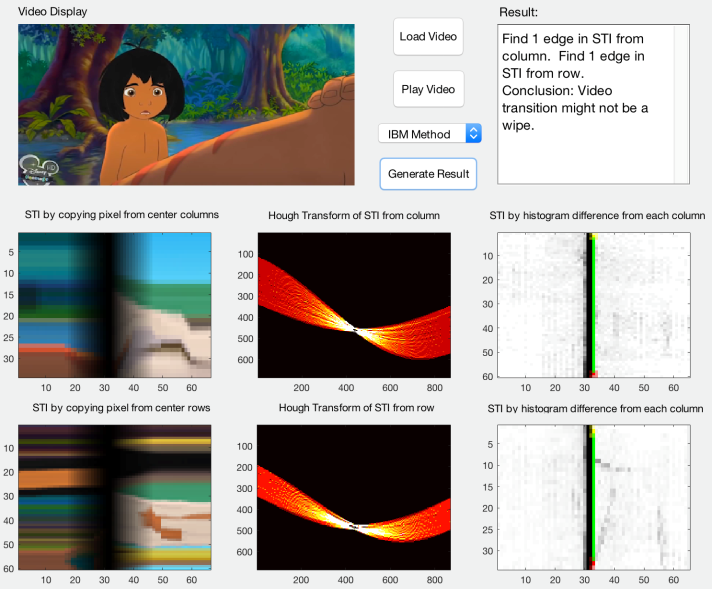
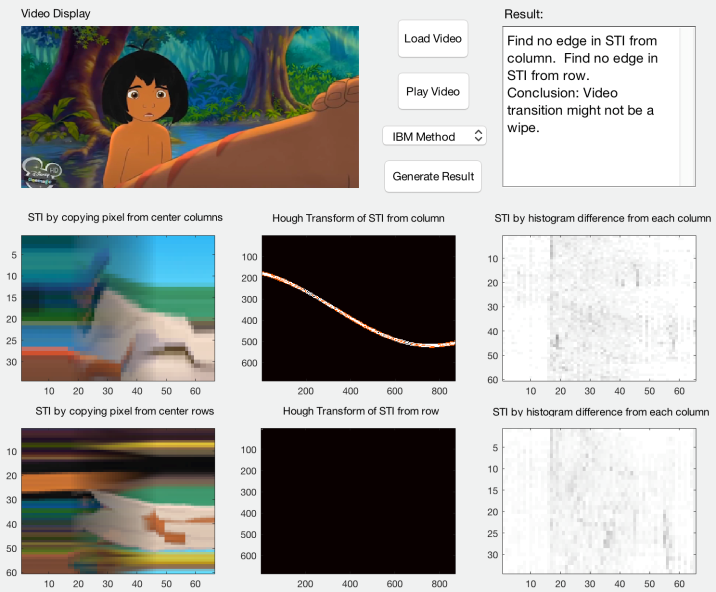


**4. Experiment on video dissolves**

We have tried the code on the video dissolves to find what will happen.

Firstly, we use the video with a cross dissolve as the video transition. The result is in the first image below. It turns out that no edges are found in both images. So the video transition might not be a wipe.

Secondly, we use the video with a dip dissolve as video transition. The result is in the second image below. It turns out that both images contain a thick edge, which is abnormal if the video transition is a wipe. Thus the video transition might not be a wipe.



**5. Programming language**

We use Matlab as our programming language on the latest version of Matlab IDE. We use the Processing Toolbox that is already built in Matlab. We also use Matlab GUI to implement the interface of our application.

In the GUI, it’s able to load any video from a path, play it and generate those STIs and hough transform image depends on the histogram difference method selected. Finally it will give out a conclusion related to the result images in the right side dialog box.

**6. Possible Extensions**

This application we built can be extended in many ways to be mature useful tools. For example, it may be extended to deal with longer video and give results to multiple transitions. It may also be able to deal with faster transition where the edge is tends to be a horizontal line. Furthermore there could be possibilities to make it able to recognize different kinds of dissolving as well.