Spatial-Temporal Image (STI) Builder

CMPT365 | SPRING 2020 | TERM PROJECT

SAHIL MODAK | 301365056 | smodak@sfu.ca
ALI NANJI | 301361228 | amnanji@sfu.ca

Introduction

The purpose of this program is to recognize transitions within a video, specifically wipes, cuts, and dissolves, and create an "spatio-temporal" image (STI). The STIs have been created using the following two methods.

The first method creates an STI by copying pixels. From every frame in the video, the program copies the center column or row into a new image. The image's columns equal the number of frames in the video and the number of rows equals to the number of rows or columns each frame in the video contains. The second method creates an STI by using a histogram of color instead of just raw pixel data. This method allows for the creation of an STI that is not affected by excessive movement or occlusion.

Libraries Used

The following are all the libraries that have been used in the program:

- OS/Shutil
  - These two libraries allow for operating system dependent or functions on files. In our program, we use these libraries to create a new directory to store frames of the video temporarily, and later delete that directory at the end of the program. It also allows us to ask the user to choose a video file of their choice.
- Open CV
  - Open CV is a library of functions used for image and video processing. In our program, we use Open CV to open the video file and extract all the frames in the video.
- Tkinter
  - Tkinter is the GUI package for python which is layered on top of Tk. This allows for the creation of the GUI used in our program.
- Python Imaging Library (PIL)
  - PIL is a library that allows for image processing of multiple image file formats and allows for opening, saving , or manipulating images.
- Numpy
  - Numpy is a library that allows for the use of multidimensional arrays and apply function on such arrays.
- Matplotlib
  - Matplotlib is a plotting library for numpy which allows to plot arrays created in numpy.

GUI Explained

When the user runs the file (main.py), a GUI interface appears. The first popup asks the user to provide a name for a temporary file. This can be any name the user wishes to use. This folder will be used to save every frame from the video as an image and save the various STIs created from the video. The folder will also be deleted when the program ends. Moving on, the GUI asks the user to choose a file, which should be a video file. This video file is then analyzed to create the various STIs. The functions used to create the STIs will be explained below. Once the STIs are created, the four buttons show up on the GUI indicating that the STIs are ready to be viewed. The user can click each button to see the labelled STI. Once the user has finished looking at the various STIs, they can close the GUI, which would delete the temporary directory that was created initially. The GUI functionality can be seen in a video on the following link ___________.

Functions

Column STI

This function takes in a parameter for the total number of frames, calculates image height and width from the first frame, and makes a 3 dimensional array with dimensions (height, totalFrames, 3). These are the dimensions for the STI. Then, using a loop to access all the frames and another loop within it to process each pixel in the middle column of each frame, these pixels are assigned to the i^{th} column representing the current frame being processed.

Row STI

This function takes in a parameter for the total number of frames, calculates image height and width from the first frame, and makes a 3 dimensional array with dimensions (width, totalFrames, 3). These are the dimensions for the STI. Then, using a loop to access all the frames and another loop within it to process each pixel in the middle row of each frame, these pixels are assigned to the i^{th} column representing the current frame being processed.

Column Histogram STI

This function, like previous ones, creates a 3 dimensional array by using values from the first frame. For each frame within the video, the program chromatizes the frame and creates a histogram column by column. It then calculates the difference between the histograms and then stores them to create the STI.

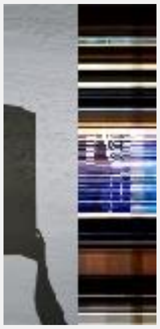
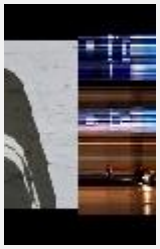
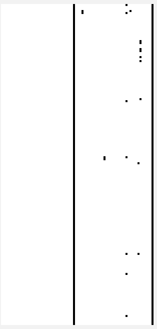
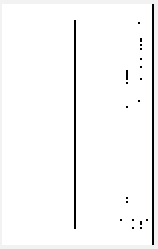
Row Histogram STI

This function, like previous ones, creates a 3 dimensional array by using values from the first frame. For each frame within the video, the program chromatizes the frame and creates a histogram row by row. It then calculates the difference between the histograms and then stores them to create the STI.

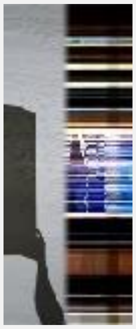
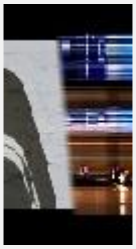
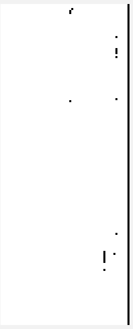
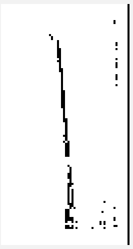
Results

Below are the results from various test case videos with various transitions we created. Each video was reduced to 120 x 160 frames to allow faster processing.

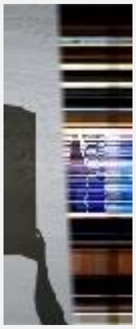
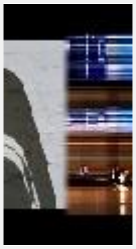
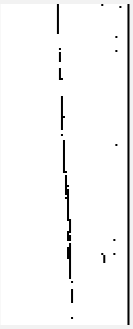
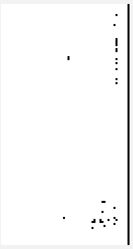
Cut

Row STI	Column STI	Column Histogram STI	Row Histogram STI
			

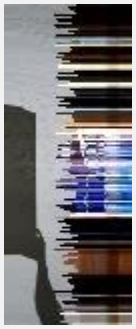
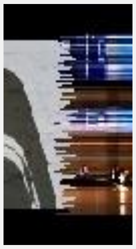
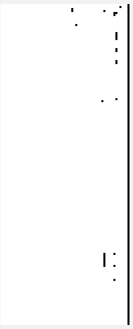
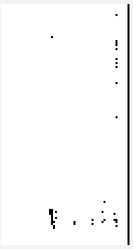
Horizontal Wipe

Row STI	Column STI	Column Histogram STI	Row Histogram STI
			

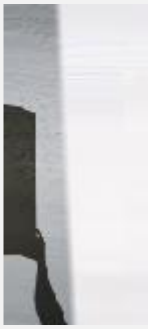
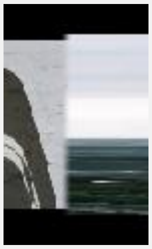
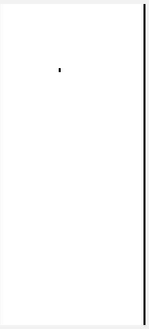
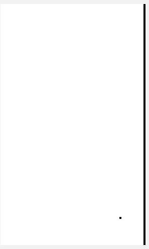
Vertical Wipe

Row STI	Column STI	Column Histogram STI	Row Histogram STI
			

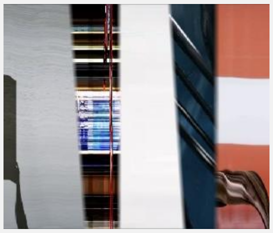
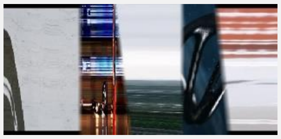
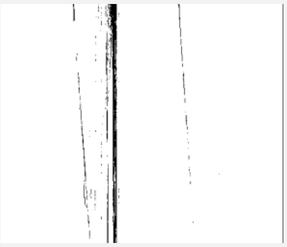
Dissolve

Row STI	Column STI	Column Histogram STI	Row Histogram STI
			

Low Contrast Videos

Row STI	Column STI	Column Histogram STI	Row Histogram STI
			

Video with Multiple Transitions

Row STI	Column STI	Column Histogram STI	Row Histogram STI
			

Conclusion

Most of the part of our code worked as expected, however, some of the results were not as expected. Such as the video that had two low contrast videos had produced good column and row STIs, but the column and row histogram STIs were not as expected. All the other results allowed us to recognize transitions in the video, which was the purpose of the program.

If the code were to be extended, we would upgrade the GUI extensively. This would include adding various snippets of code for exceptions. Furthermore, we would make the GUI look more visually appealing and user friendly. Lastly, we would optimize the code to run the program on videos with a larger ratio and more frames per second faster and more efficiently.

Final Thoughts

The project was not too hard but just slightly challenging. A few ideas to go above ad beyond the basic requirements would make the project better. Also, a slightly more challenging bonus component would make this project more fun.