Comp Photography
Fall 2015
Final Project

Jacob Kilver
jkilver3@gatech.edu

https://cs6475.wordpress.com/2015-fall/

Video Stabilization

There are many instances where video is captured but is not of the quality (in terms of stability) that is desirable. While many well-developed tools for solving this problem already exist, this project seeks to understand some of the nuances of producing stable video by re-implementing this functionality.

Project Goal

Video is becoming increasingly a part of our lives and increasingly easier to capture. However, with the convenience comes a decrease with the quality of the video, particularly with capturing stable video. Fortunately, post-processing can render shaky video much smoother. To better understand and appreciate the nuances of video stabilization this project intends to re-implement the functionality found in more refined tools, such as the ones built into YouTube.

Project Results

Input

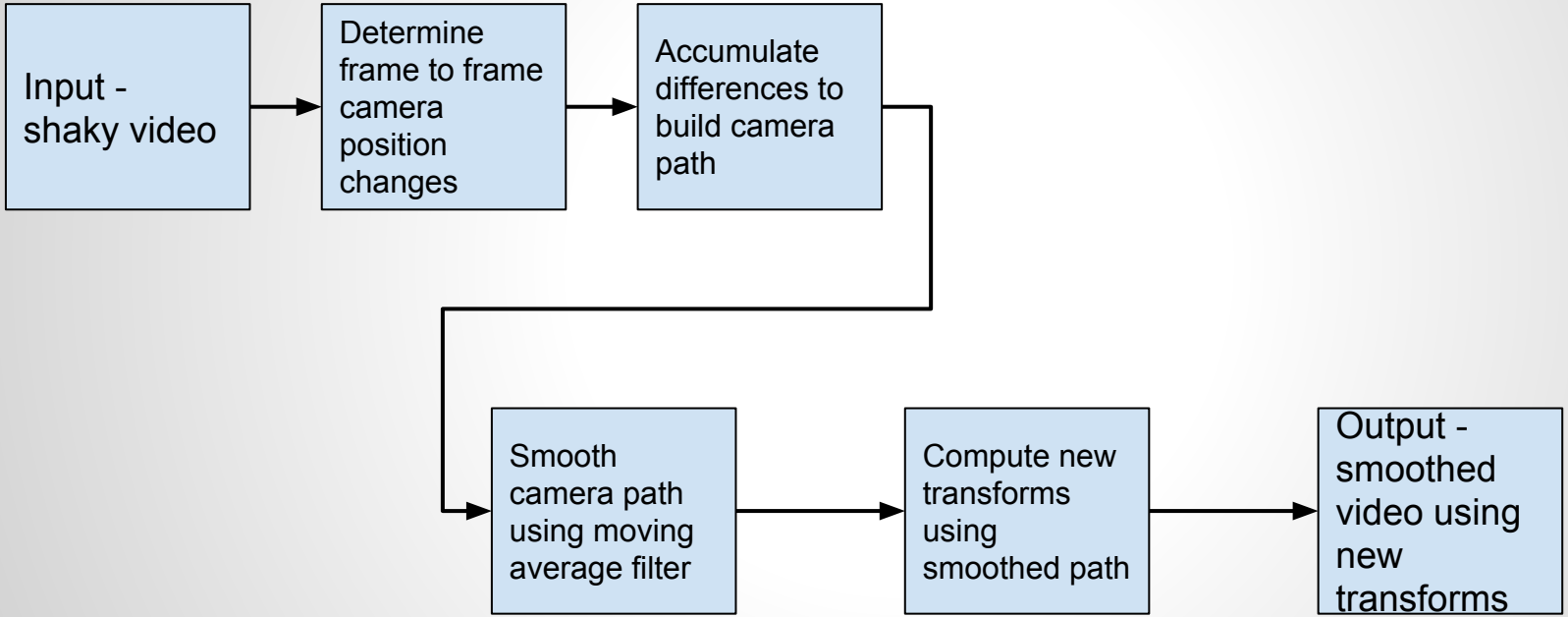
Output

Project Results

Input

Output

Computational Pipeline for Stabilization



Theory of Operation

- Python/OpenCV was used for this project
- Step 1 - Input video
  - The user passes the file path via the command line
- Step 2 - Determine pairwise path differences
  - Looping over every frame in the video sequence, find points to track.
  - Calculate optical flow using Lucas-Kanade
  - Estimate a transform (restricted to rigid transformations)

Theory of Operation

- Step 3 - Build Camera Path
  - With the frame-to-frame transformations from Step 2, keeping running sum of transforms to build camera path
- Step 4 - Smooth Camera Path
  - Use moving average filter to smooth camera path locally

Theory of Operation

- Step 5 - Compute new transformation sequence using smoothed path
  - Find the difference between the smoothed path and the actual path
  - Add this difference to the original transformation sequence to create a transformation sequence that adds up to the smoothed path

Theory of Operation

- Step 6 - Build smoothed video
  - For each frame, apply the newly computed transformations
  - Crop the frame
  - Append frame to new video

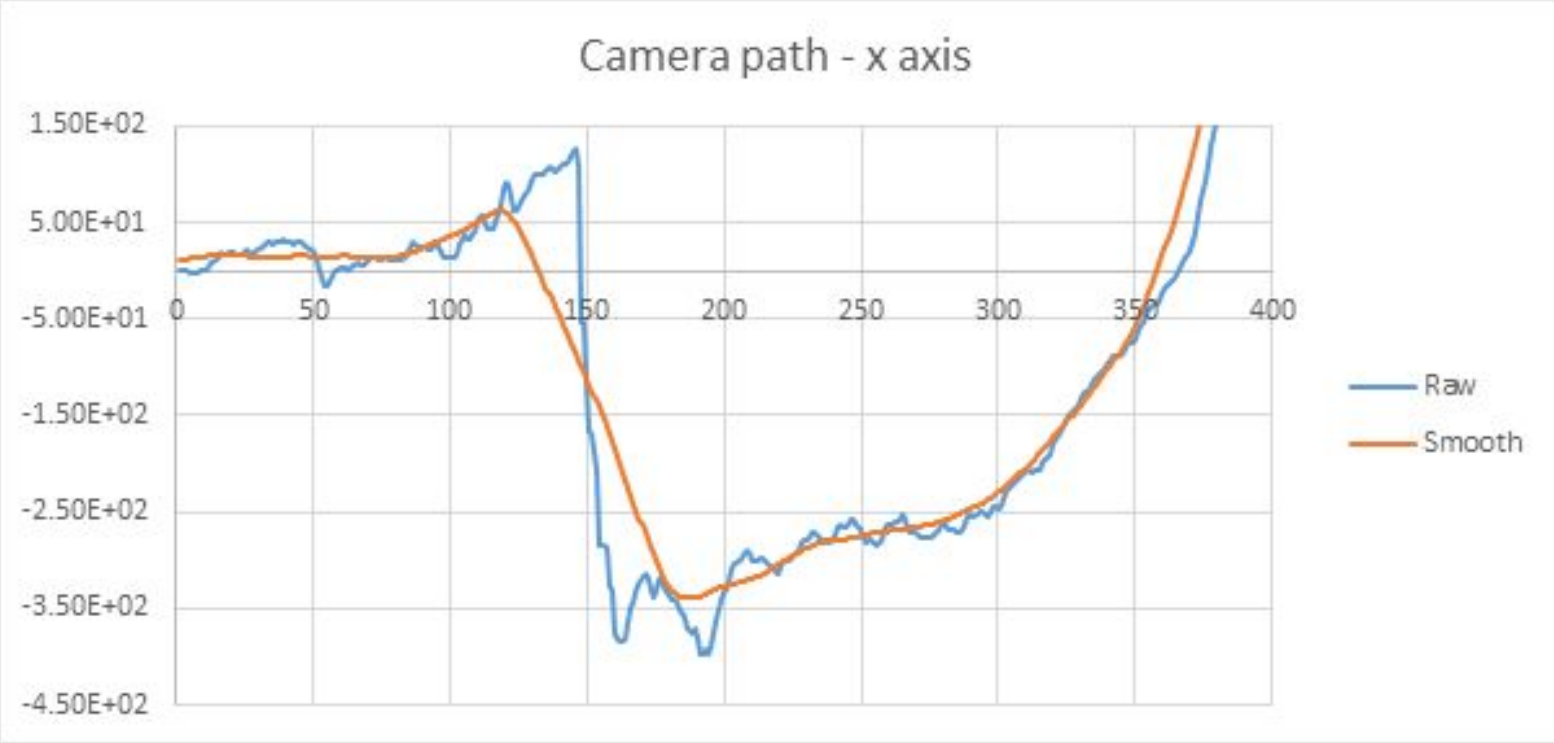
Where to see results

- The best way to view my project is via the YouTube links below
  - Simple example
    - Original: https://youtu.be/GdfwyoTvi-w
    - Stabilized: https://youtu.be/OsuuwgqYysk
  - Real life example
    - Original: https://youtu.be/Ww953wOwPGg
    - Stabilized: https://youtu.be/g2mowfhrnSI

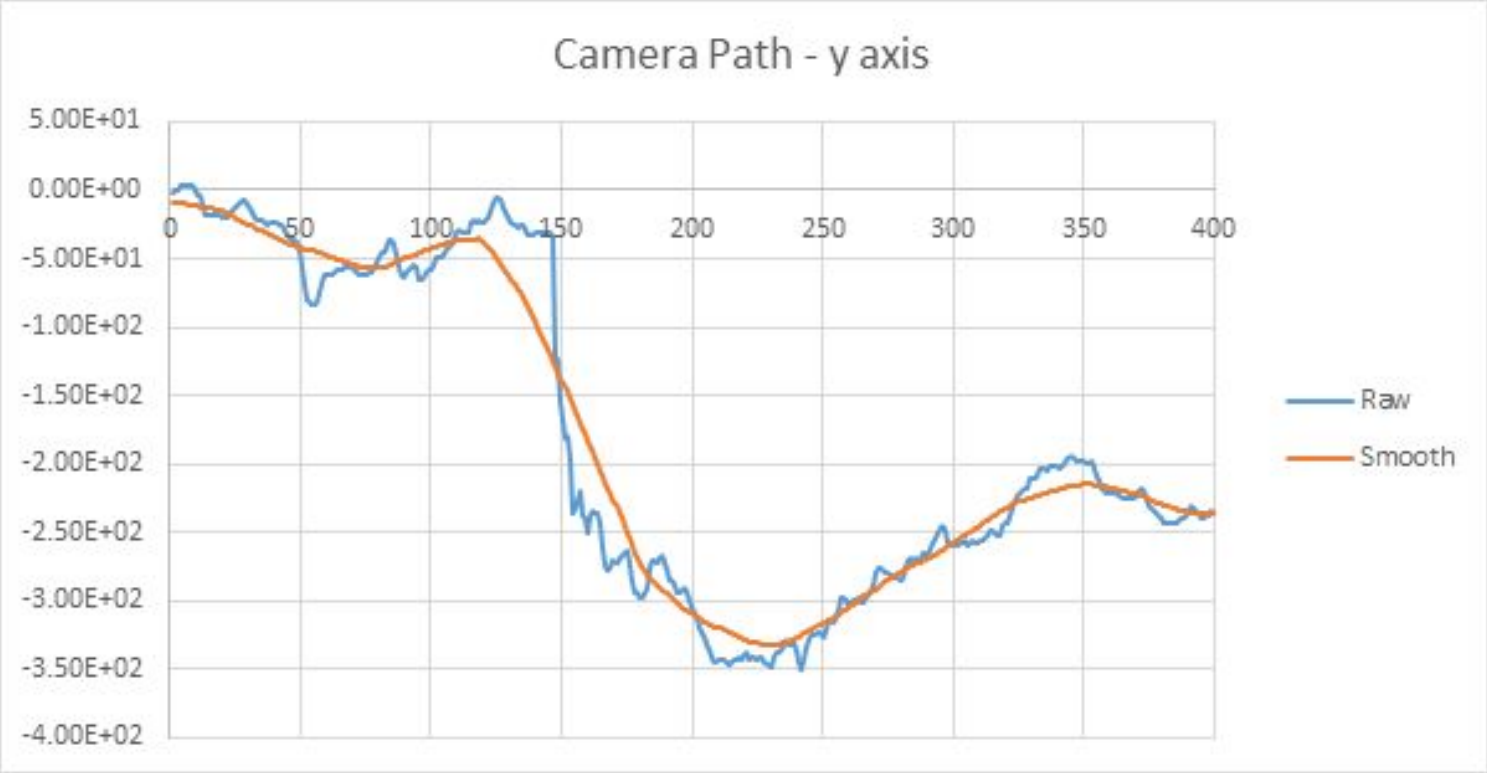
What worked

- The video stabilization implemented here did stabilize the camera path.
- See the charts on the following slides
  - The charts show the original camera paths and smoothed camera paths
  - The x- and y-axis paths are much smoother than the original
- The cropping did reduce the amount of “noise” along the video borders

Camera path


Camera path - x axis
1.50E+02
5.00E+01
-5.00E+01
-1.50E+02
-2.50E+02
-3.50E+02
-4.50E+02
0
50
100
150
200
250
300
350
400
Raw
Smooth

Camera path


Camera Path - y axis
5.00E+01
0.00E+00
-5.00E+01
-1.00E+02
-1.50E+02
-2.00E+02
-2.50E+02
-3.00E+02
-3.50E+02
-4.00E+02
0
50
100
150
200
250
300
350
400
Raw
Smooth

Deficiencies

- Only rigid transformations were used for stabilization
- The camera path was not entirely a rigid transformation
- Thus, while the video is more stable, some distortions are added
- This is most visible in the living room scene.
  - The couch doesn’t shake so much as it warps.
- The cropping was very rudimentary. This could be improved, especially to preserve the aspect ratio

Deficiencies

- An attempt at applying a full homographic transformation was made
- However, it was not successful in smoothing the video
- The reason this did not work was because the path was smoothed over the homography matrix directly
- The transformations are actually linear combinations of the homography matrix

Comparison to existing tools

- The real-life video (horse scene) was tested using existing video stabilization software
- YouTube suite
  - The results can be viewed here: https://youtu.be/FpXs4LAMoo4
  - The YouTube software is definitely more refined than the software developed here.

Future Work

- Use a full homography transformation instead of a simple rigid transformation
- Implement a better cropping method, particularly to maintain the aspect ratio
- Design a way to include sound in video output

References / Pointers

- Lecture videos from class 06-03
- http://nghiaho.com/?p=2093