Comp Photography (Summer 2015) Final Project

Christopher Owens
chris.owens@gatech.edu
Jake Ritter
RitterGT@gatech.edu

Moving Video Panorama

(SelfiePanorama)

Create a panoramic video from a video that pans across a space, with only a specified object moving. The result will be a video with static panoramic background with a single object moving through it.

The Goal of Your Project

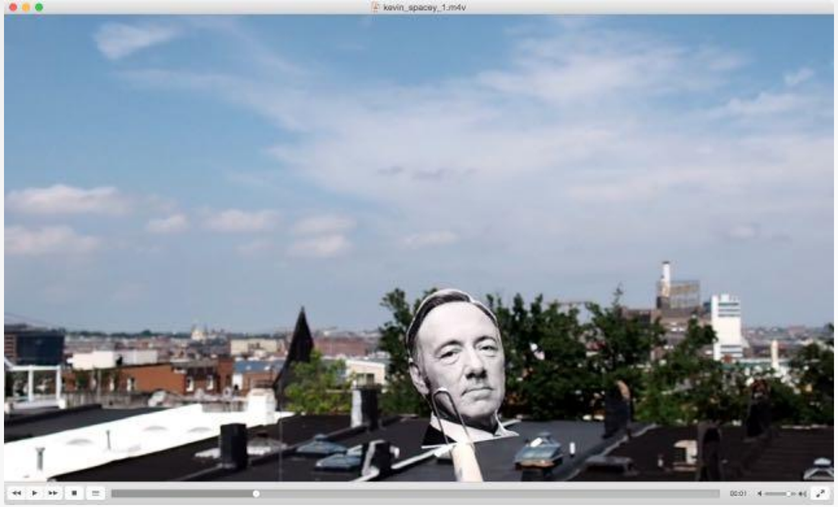
The goal of this project is to explore video textures, panoramas and feature detection to create a novel video using image processing.

We thought this would be an interesting end result and leverage a lot of what we learned throughout the class.

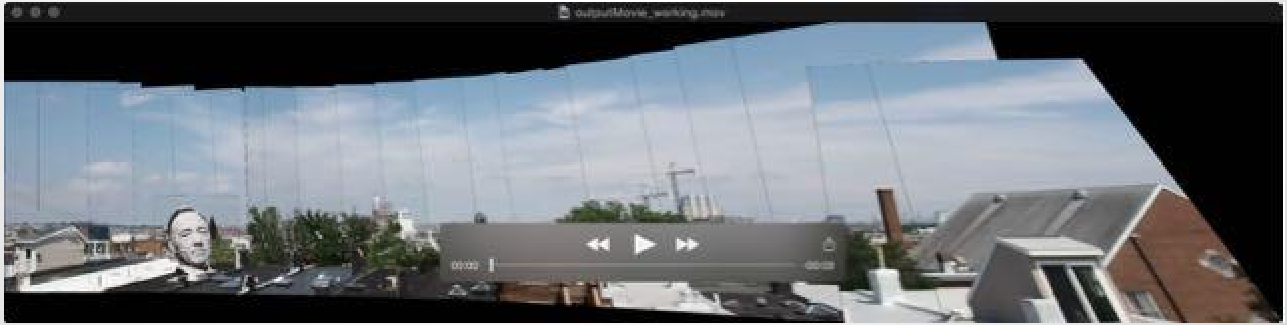
We thought this:http://giant.gfycat.com/TatteredEveryGordonsetter.gif is a really interesting way to view this: https://www.jukinmedia.com/videos/view/895986 video

Showcase

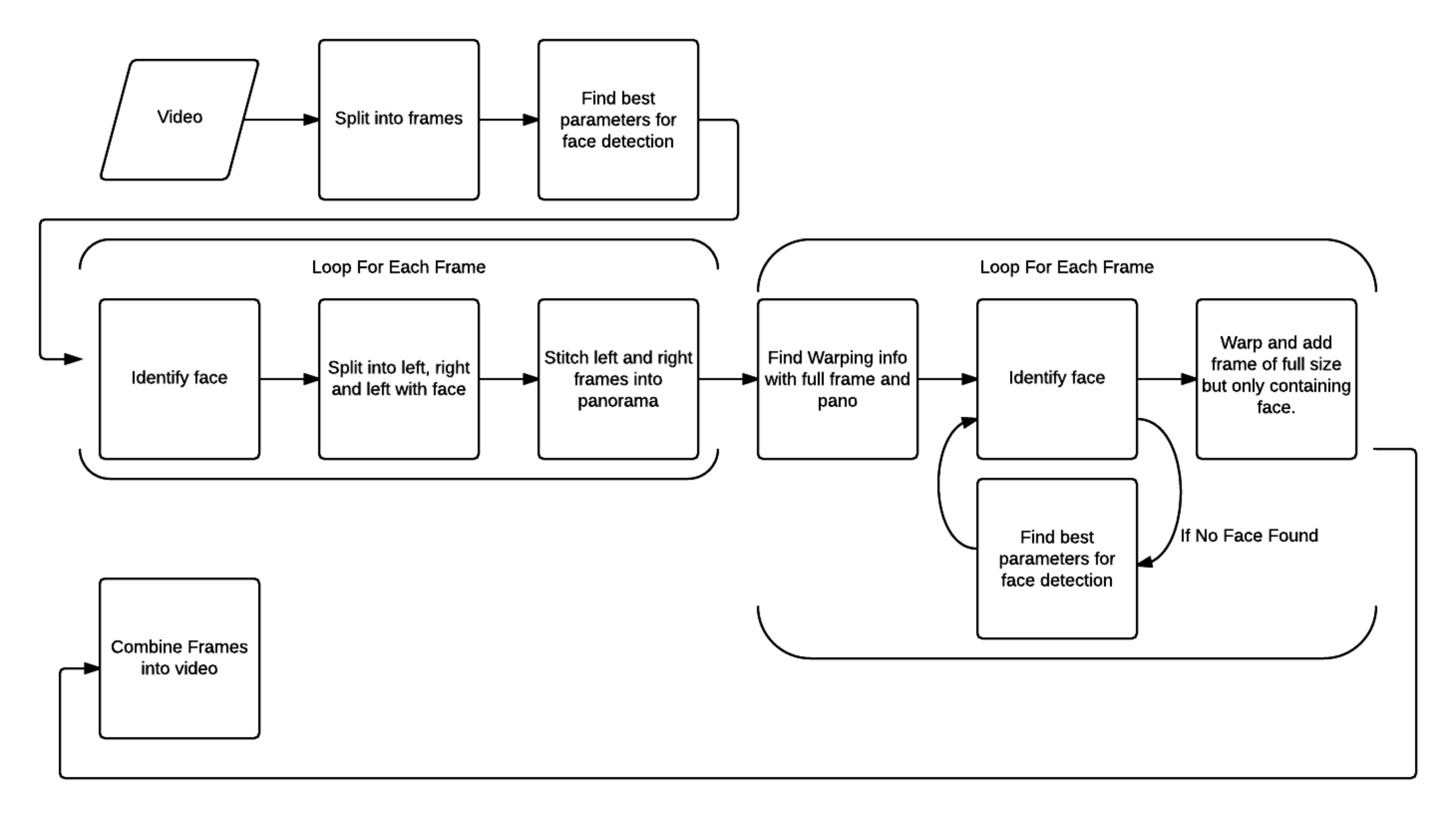
Input
(video)



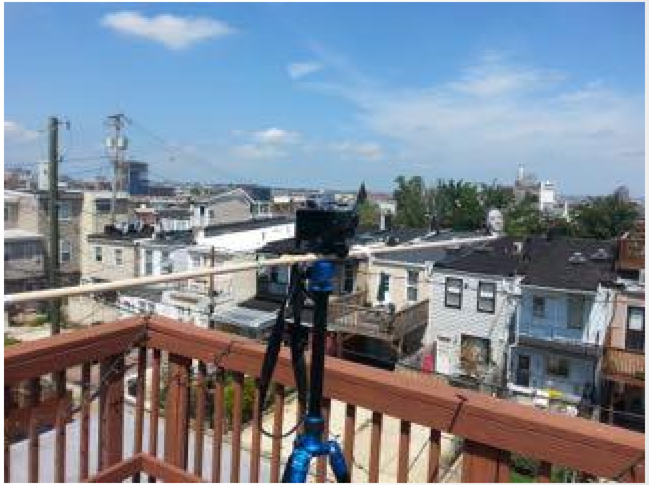
Output
(panorama Video)



Our pipeline


Video
Split into frames
Find best parameters for face detection
Loop For Each Frame
Identify face
Split into left, right and left with face
Stitch left and right frames into panorama
Loop For Each Frame
Find Warping info with full frame and pano
Identify face
Warp and add frame of full size but only containing face.
If No Face Found
Find best parameters for face detection
Combine Frames into video

Our Setup



We wanted to make sure Frank stayed in the center of the frame (or close to it) This allowed us to make a better panorama.

What is the best way to see your project?

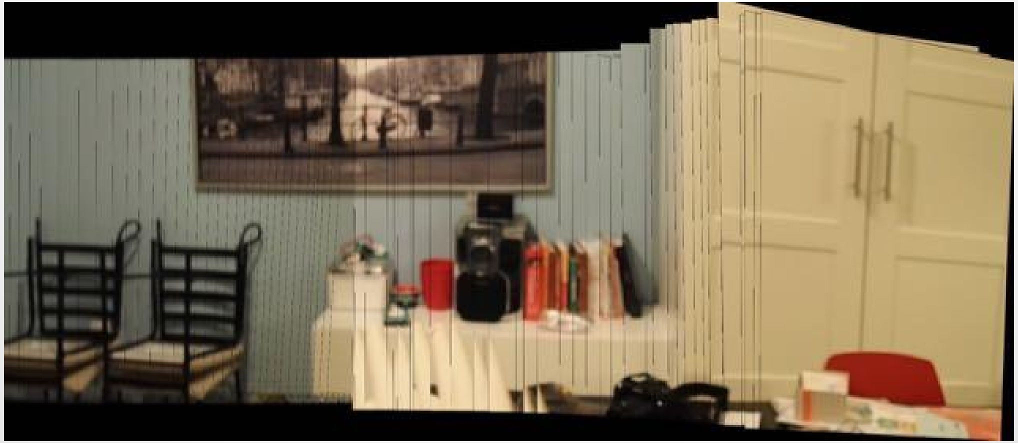
- Input: A video panning across the baltimore skyline with Frank’s (Kevin Spacey) face in the center of every frame. https://drive.google.com/file/d/0B8hm1u0E2CNeMW5sWkRJbkVWdTg/view?usp=sharing
- Output: A panoramic video with Frank’s face moving across the panorama. https://drive.google.com/file/d/0B8hm1u0E2CNeV0F3R1hoQ09TY3c/view?usp=sharing

What worked

- Classification of faces
    - OpenCV comes with the ability to identify faces and it works very well.
    - We created an optimizer to identify what parameters to pass into the classifier. To do this we run the classifier with a range of inputs, counting how many frames in which it found just one face. We then use these values to find the face throughout the video.
    - This works well because we control the scene to ensure there is only one real face within the shot
- The panorama
    - We were able to create a panorama without the identified face in it.
    - This took some modification of the panorama code so it would skip images that didn’t work. This was needed because we create the panorama from small images and sometimes it can’t find enough matches.

Had Troubles With

- The panorama
  - The panorama ended up being hard to create. Given that we want to identify a face and create a panorama without it, we need to cut the face out of the image before adding it to the panorama. To do this, we identify where the face is and split the frame into two images, everything to the left of the face and everything to the right of the face. When running this process on the banana, we didn’t have enough of an image to find the matches needed to create the panorama.
  - Another issue we had was that we haven’t done any blending. Since we were using every single frame in the video, we got lots of artifacts from every stitch. To decrease this we sampled every 15th frame when generating the panorama. This had the added benefit of speeding up our processing



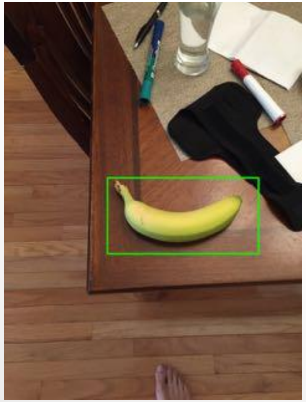
many artifacts from stitching

Had Troubles With Cont.

- The moving image
  - To place the face in the panorama correctly, we need to find where it goes and warp it correctly.
    - We began by warping the entire image and placing it on the panorama for each frame. This resulted in a video that had a flickering picture moving through it. One of the issues with this was how we created the panorama in the first place. It was created using only the left and right sides of each frame. This meant that trying to warp the entire image to the panorama ended up with neither the left nor right side being warped correctly. https://drive.google.com/file/d/0B8hm1u0E2CNeZWtLVmJJVEVRenM/view?usp=sharing
    - Our second try was to warp using the face and everything left of it. Then find the face and add it to the panorama. This worked fairly well until we progressed to where the panorama was created using right slices almost exclusively. This is it working decently on our first video: https://drive.google.com/file/d/0B8hm1u0E2CNeZzctbGtmb2t2bGc/view?usp=sharing
    - The solution was to warp with the entire image and then only place the face into the panorama. We did this by calculating the homography and the move points from the entire image and then applying them to an image containing only the face and 0 padding to make it the same size as the original image. The key here was to do all processing on the original image, including the facial recognition. When we tried to run the facial recognition on the warped image, we got multiple frames in which no face was found.

What did not work? Why?

- Classification of a banana
    - OpenCV comes with a face classifier which works very well, but we wanted to be able to create a classifier for any image.
    - Thorsten Ball wrote up a process for creating your own classifier (http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html). We decided to use his banana classifier as a proof of concept that you could create your own classifier.
    - This ended up working well for very simple images, but failed on more complicated video frames. We spent a long time trying to modfiy the shots to match the classifier. We believe more training could have fixed this, but Ball said it takes days to train and we didn't have the time to burn.
        - The banana classifier only found real bananas.
        - It found them well in static images, but fails in our movies: https://drive.google.com/file/d/0B8hm1u0E2CNedVIuZXI3OGhQaHc/view?usp=sharing



Future Improvements

- Blending
  - We decided to do blending only if we had extra time due to our blending code taking a very long time in the panorama assignment and only improving the panorama a minimal amount. With our processing times running more than 15 min, we didn’t want to add anymore time.
- Object detection
  - We would like to remove the constraint of only one face in the frame. To do this we would need to find all the faces in every frame and then track only the mode. This would require identifying each face similar to our feature detection assignment, but we would use found faces as the original image.

References / Pointers

- Stitching: https://github.com/cbuntain/stitcher
- Face Recognition: https://realpython.com/blog/python/face-recognition-with-python/
- Face Recognition: http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html
- Our Code: https://github.com/cowens85/CompPhotoFinalProj

Team

- Jake Ritter: Classifier
- Chris Owens: Panorama, Inputs (experimental setup)
- Both: Pipeline, Research

We collaborated very closely on the code, using GitHub and Hangouts when we weren’t able to meet.