Image Stabilization – Jon

I am working on image stabilization in order to increase the accuracy of tracking. At the moment I have a tool which can track points in a video, and I began attempting to implement warping the video so that it is more stable throughout.

My first attempt at stabilizing a video used OpenCV's *goodFeaturesToTrack*, *calcOpticalFlowPyrLK*, *findHomography*, and *warpPerspective* functions. Unfortunately, the results were significantly more unstable than the original video. Because of these results I began to do more research on methods of image stabilization. I may attempt to implement this version later, as I believe there were a few steps I was missing.

I found one resource which includes very general steps on how to do stabilization. This source only uses rigid Euclidean transforms, followed by trajectory calculation and smoothing, and calculating a final transform using the trajectories. I implemented calculating rigid euclidean transforms and simply applying those transformations frame by frame, and the results were significantly better. I will add the additional steps next.

There may be a problem in the future with the videos of the monkeys where some features are tracked that are on a monkey, which would cause a disparity when calculating the transformation between frames. I may have to implement a way to exclude some features when tracking, but I will leave that to be completed later. For now I am testing on smaller videos taken on my phone with a significant amount of instability.

Dan's section:

My part of the project is trying to track one (or hopefully all) monkeys as they move through the video.

From research, meanshift (specifically CamShift) looks like it will yield the best results for tracking a monkey in the videos. Blob testing has the ability of finding every monkey in the image, but requires a lot of fine tuning to find all monkeys, and seems to find a lot of false matches in rocks and bushes and the like.

CamShift requires you to input a box beforehand, which is a box around the object you want to track. My current plan of action is to use blob tracking (cv2.simpleBlobDetector) to find a bunch keypoints that are (hopefully) all monkeys, then try to extract correctly sized boxes around those monkeys based on their keypoint size. Then, I will attempt to track all of those found object throughout the video. The end output will most likely be a video with boxes inscribed over each tracked object.

Background Extraction - Avi

My initial goal was to work with Dan on object tracking to monitor the movements of the monkeys in a video. Currently, I am looking at finding movement in a video with the CV functions goodFeaturesToTrack and calcOpticalFlowPyrLK to see if it will provide substantial information to go along with Dan’s implementation. I have copied around 6 or 7 movies from the larger data bank to test this method.

While researching motion tracking, I came across OpenCV functions for separating a background from objects in motion, classified under background subtraction. I found these to be pertinent to our project so I have decided to explore the BackgroundSubtractorMOG function as well as the BackgroundSubtractorMOG2 and BackgroundSubtractorGMG functions. I aim to apply these to the videos to distinguish the background from the monkeys in motion. It might be interesting to count the number of monkeys that are present in the tested frames, perhaps better for videos where generally the monkeys do not move too much.