1 Introduction

Through a combination of image thresholding, applicaiton of morphological components, and connected component analysis, we created a general user interface for general motion tracking, and applied it successfully to two videos.

2 Approach

As described above, our process involved four major steps:

- Temporal image creation
- Thresholding
- Application of morphological operators
- Connected components analysis

Since the objects in our videos took up a small percentage of the image space, we took the color average of each frame and used it to subtract the background. With the background subtracted, we thresholded the image to detect the moving objects. Afterward, we applied a combination of erosion and dilation to ensure that each object's internals were completely white, and to suppress background noise.

With a good mask created where each object is visually segmented from the others in the image, we perfored a connected components analysis. This returned a list of contours in the current frame. The most difficult problem we struggled with was how to relate contours in the current image to contours in previous images. Essentially, we wanted to plot trajectories of the objects. This is trivial task if the number of objects is small, but when there are many objects potentially moving very close to one another, they can change the number of connected components that are detected.

We devised a fairly simple algorithm to draw the trajectories on the image. For each frame that was not the first frame, we mapped each contour to the closest contour in the previous frame and gave it the same color. We then saved the list of points and their associated colors for use with the next set of contours. For the first frame of the image or when we detected that a new object had entered the scene, we gave that object a random color. We had to prevent new objects entering from trying to associate themselves with trajectories from other objects far away because they didn't have any points recorded.

3 Fly Video

This video file was provided by the instructor. Given the steady video feed, the temporal average made thresholding and obtaining fully defined components fairly easy. Additionally, the speed the flies moved was slow enough to minimize the acceptable distance for two components to be the same object Even when the flies moved close to each other, we were able to distinguish them.

As far as limitations, we still see potential issues with our motion tracking algorithm if two flies were to get close and stay close, possibly causing one fly to change to the same color or swap colors. We suggest values of 30+/-10 for thresholding, 3 for eroding, and 3 for dilation to get a good result for this video.

4 Lacrosse Ball Video

This video was shot on iPhone 5S at 120 fps. Initially we struggled with getting a good temporal average because of camera shakiness + light effects. We ended up changing the background a few times and holding the camera against a rigid surface to ensure a good temporal average.

Additionally, we had issues distinguishing the ball from the floor. Originally we converted each frame to black and white, which worked well for the fly video since it was already mostly black and white. We found that averaging over color allowed us to distinguish the differences in colors between the ball and the floor, even though both were bright objects.

We also struggled finding a suitable threshold that would not cause discontinuities between image frames. Every time we had a discontinuity, the ball would reappear as a new color since it had no reference data point. Sometimes if there were multiple balls in the same image, and one of them had a discontinuation, it would reference the other ball's location after reappearing and draw a line from it. To prevent this, we made sure that a line is no longer than a reasonably large length. For the purpose of these clips, this is reasonable since the objects are not moving fast, and the image files are around the same size, but for the future, using a distance as a percentage of the image frame would be ideal.

Once we found a good threshold however, the motion tracking turned out nicely. We suggest values of 60+/-5 for thresholding, 4 for eroding, and 5 for dilation to get a good result for this video.

5 Conclusions

5.1 General Issues

We had an issue with the a video frame cutoff on the last 20 pixels on the right side. This ended up not affecting our motion tracking though because that component just stayed stationary. We had a few instances of the divide by zero error if we didn't dilate our objects enough because there were components of zero are that were identified.

5.2 Future Work

We would like to improve our motion tracking in a few ways. First, we would change the distance limit for line drawing based on the size of the input video. We hard-coded a sufficient value given our inputs directly into the source code, but should probably get a value as a function of the image shape in the future. Second, we would like to devise an algorithm that maintain points across multiple image frames in the case there is a discontinuity between components. Currently, we only reference the points from the most recent image. We also didn't implement closing on our videos because our objects were both white against dark backgrounds. In the future, we could experiment with different colors on different backgrounds.

In general, our system works well with steady backgrounds and discrete moving objects that aren't moving too fast for a camera to pick up. Our system does manage to distinguish a moving object (lacrosse ball) near a "bright" surface (gray carpet). Our system would not work well with changing brightness conditions or any scene without a static background. We did not implement any adaptive thresholding capabilities.