Motion Tracking of Hands

Albert Chu

University of California, San Diego
alchu@ucsd.edu

Abstract

Hand motion tracking is a difficult problem. Usually, this area of interest also deals with hand gesture recognition, taking into consideration various parameters such as the placement and location of each finger, illumination, shading, and depth cues. A hand has at least 27 degrees of movement including wrist motion. Taking all of the variability into account is beyond the scope of this project. Here, this project will only deal with the direction of movement of a square target printed on a hand, ignoring finger tracking. Many other factors will also be held constant such as lighting, occlusion, and hand gestures. I will be experimenting with multiple recognition and tracking algorithms and researching basic computer vision and image processing techniques to implement my project.

1. Introduction

Since only tracking of a square target is necessary, I am planning on implementing object recognition of the square target, either by correlation-based algorithms and template matching or some other recognition techniques. Another approach is that the square can be segmented out based on its color via k-means clustering. The only problem I can think of right now is if the background has the same color as the color of the printed target. In addition to template matching, using Canny edge detection and comparing edges images will reduce the effects that lighting and color have on the images. However, we must consider a large range of possible template positions. Yet another approach involves blob detection, which uses differential methods to detect regions that are darker or brighter than the surrounding. Given that I will be using a black square with a white square center target, this method might benefit me the most.

Next, using the Lucas-Kanade optical flow technique, I can track motion of the hand between frames. If the window size for calculating optical flow for each pixel is sufficiently small, many pixels might suffer from the aperture problem. Part of the Lucas-Kanade method is applying a threshold on the eigenvalues, which can suppress these flow vectors.

The hand pose will have to stay constant so as to not occlude the printed target on the hand. For example, if the target is printed on the palm, the hand will have the palm face the camera the entire time. Possible solution to any problems presented by rotating the printed target uses affine transformation, which provides a linear transformation from one vector space to another. This will preserve a one-to-one mapping as well as a collinear relationship between points.

1.1. Qualifications

Here are the related courses I've taken at UCSD: CSE 150, CSE 151, and CSE 166. I am also taking CSE 152 next quarter. CSE 151 required us to work on a quarter-long project as well, similar to a research project type setting with our own implementation decisions. We implemented skin classification as a side project. Our project was detecting and classifying objects in GroZi, the algorithm was inspired by the Viola and Jones algorithm.

1.2. Milestones and deadlines

Milestone 1: finish setting up environment and get used to the development software by the end of week 1.

Milestone 2: finish gathering datasets or training data by the end of week 2.

Milestone 3: finish detection of printed square target using any of the proposed algorithms by the end of week 5.

Milestone 4: implement motion tracking of the printed square target using optical flow or some other technique and account for any rotation or scaling of the target via affine transformation. The scaling factor in affine transform might also take care of any depth ambiguities if the perceived size is smaller or larger. Finish by the end of week 8.

Milestone 5: finish preparation of presentation and final report by beginning of week 10.

1.3. List of questions

One of the key problems in monocular hand tracking concerns the existence of depth ambiguities. If I'm using

correlation-based algorithm and template matching, would the template be able to match the printed target at various depths, or would I need to provide various templates and perform iterative matching?

How is a learning and classification algorithm different from object recognition and would a learning algorithm such as the perceptron algorithm be more efficient than classical object detection methods such as the Viola-Jones method in the detection of the target?

1.4. Software

I will be using Matlab to implement my project. To extract individual frames from video files, I will be using the FFmpeg software. If I decide to switch to using C, I will most likely be using the OpenCV library for functionality reference.

1.5. Datasets

I will be using and obtaining datasets of my own. Using my own camera, I can record videos of various individuals holding the printed target, comprising my own test and training set.

References

[1] T. Lindeberg. Feature Detection with Automatic Scale Selection. International Journal of Computer Vision, Vol.30, No.2, pp. 77-116, 1998.

[2] Luis Ferraz, and Xavier Binefa. A Scale Invariant Interest Point Detector for Discriminative Blob Detection. Pattern Recognition and Image Analysis, 2009.

[3] Ming Anlong, and Ma Huadong. A Hessian-Laplace Based Blob Detector for Stamp Image Classification. Chinese Journal of Electronics, Vol.19, No.4, 2010.

[4] Z. Wei, D. Lee, B. Nelson, and J. Archibald. Real-Time Accurate Optical Flow-Based Motion Sensor. International Conference on Pattern Recognition (ICPR), December 2008.

[5] Carlo Tomasi, and Takeo Kanade. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[6] B.K.P. Horn, and B.G. Schunck. Determining Optical Flow. Artificial Intelligence, vol 17, pp 185-203, 1981.

[7] Q. Zheng, and R. Chellappa. Motion Detection in Image Sequences Acquired from a Moving Platform. ICASSP, Vol.5, pp. 201-204, Acoustics, Speech, and Signal Processing, 1993. ICASSP-93 Vol.5, 1993 IEEE International Conference, 1993.