Finger Cursor

-Smart Paddle in the ballgame-

Intro to Computer Vision Final Project Report

Group member: Kaixuan Zhang, Ji Lin
Report Author: Ji Lin
Student ID:3071712

Northwestern University
Electrical Engineering and Computer Science
December/05/2017

Table of Contents

Chapter 1 Introduction and Personal View

Computer Vision is a captivating field of computer science, with broad, interesting applications. It is also an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do. The field allows for conveniences in our daily life – such as face detection on Facebook and digital cameras, image manipulation in programs like GIMP or Adobe Photoshop, recognition of common image paradigms like Chase check upload and QR code scanning, and easy searching of huge databases of images using services like Google Image Search. Analysis also provides for critical functions, such as military intelligence and observation, face detection for security purposes, and much, much more. In fact, the aforementioned applications barely scrape the surface of what is possible, and often already practiced, in this field. Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. Understanding in this context means the transformation of visual images (the input of the retina) into descriptions of the world that can interface with other thought processes and elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.

Personally, I was enthralled by the power and ubiquity of computer vision techniques surveyed in this course. I had failed to realize the importance and prevalence of some of the key concepts, like edge detection, Gaussian smoothing, and color segmentation, but this course allowed a glimpse into this exciting world.

Despite the fact that this is merely an introductory course on the subject, I felt like I learned a huge breadth of topics. In fact, I would feel comfortable tackling all but the most advanced techniques in the field of computer vision, armed with the knowledge instilled by this course. Among the things I picked up are: fast iteration techniques for processing images, design and application of structuring elements, Gaussian averaging and normalization, four-neighbor operations, matrix and vector operations for fast image analysis, and basic image processing techniques divulged in MPs one through six. Personally, I enjoyed this course and its associated projects very much. Using the backbone of knowledge learned in the course and textbook, as well as generally basic coding skills, I was able to implement advanced image processing functions normally witnessed only in expensive, proprietary software like Adobe Photoshop.

I thought most of the topics were covered clearly, and at a good level of detail. However, there were a few key areas that I would have liked to explore a little more. Primarily, these are a more in-depth discussion (especially regarding possible techniques) of face detection, an exploration of the applications of edge detection, and a close look at techniques for comparing images against one another. I cannot wait to learn much deeper in such field in the next quarter's course.

Chapter 2 Project Description

My project group consisted of Kaixuan Zhang and myself. As for our final project, we decided to spare no effort to implement the Finger Cursor Assignment. The goal of Finger Cursor is to allow a user to control a cursor using their fingertip, rather than a conventional mouse. The key point of Finger Cursor is to detect, locate and track a fingertip via a video sequence accurately and robustly.

Put it into broader terms, given a video with a recognizable human hand, it should be able to identify where the tip of the finger is in any given video frame, and place the cursor at this position. This identification should function despite poor video quality, unpredictable lighting and series of other potential suboptimal conditions.

In addition, an optional goal for this project is to implement gesture recognition, which allows more than one gestures made by the hand in the video sequence. In our project, we are able to detect two gestures in a sensitive and robust way.

To make it more interesting, we also design a ballgame based on the implementation of Finger Cursor. It is alike a popular classical game named Brick Breaker. In the game, we define two gestures to control the paddle. The first one can only move the paddle while the other one can not only move the paddle but also rebound the moving ball, which is show in the figure 1 and figure 2. And we also define a gesture zero, which is the original status, that the paddle is able to rebound the moving ball at the very beginning. Through controlling paddle, we could rebound the moving ball to eat other still balls. We will win the game if we could eat all balls in the limit time without running out all energy. More interestingly, the speed of the moving ball will increase during eating balls. And it will consume 20 energies for one rebound. But you could earn extra energy back by eating balls or gold bonus.

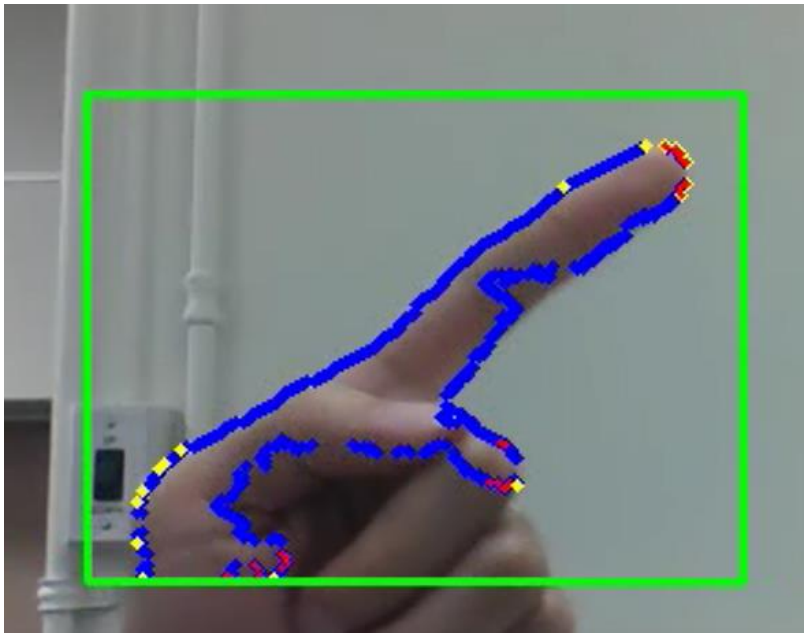

Figure 1 Gesture one: move the paddle

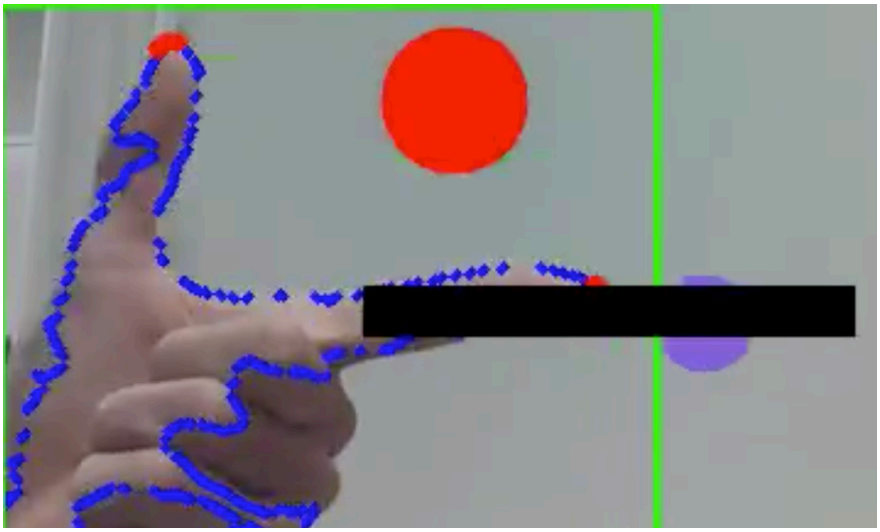

Figure 2 Gesture Two: move the paddle and rebound the moving ball

Through many tests upon the ballgame, we could prove that how sensitive and robust our finger cursor is. However, our original goal is to draw curve paddle based on the fingertip location. We are unable to implement draw paddle anytime anywhere we want to due to time constraints. A further discussion of this project will be shown in Chapter 5.

Chapter 3 Design and implementation

3.1 Design Process

Our Finger Cursor project was designed, developed, implemented and tested over the course of roughly half a dozen meetings. We set out four weeks before the deadline. In the first meeting, the general format for the project was drawn up on a whiteboard. After thorough discussion, the ideal steps of our project are shown as follows.

1. Video input, decoding and frame sampling
2. Finger Detection
3. Gesture Recognition
4. Fingertip location output
5. Game design
6. Combine Finger Cursor with designed game
7. Demo Output

3.2 Personal Contribution

What I have done in this project focus mainly on Game design, Combination of Finger Cursor and ballgame and a small part of Gesture Recognition.

First of all, at the circumstance of Python 3 and OpenCV, I tried to analyze hand gestures using method of contour detection and convex hull of palm. At the begin, I capture frames and convert it into grayscale. Our POI is the hand region, so we capture the images of the hand and convert them to grayscale. The reason why I chose to convert images from RGB

to grayscale is that through the binary scale we could find the portion of image I am further interested for image processing. The grayscale image is shown as Figure 3.

Figure 3 grayscale image

Next, I used Gaussian Model to blur the original image. I blur the image for smoothing and to reduce noise and details from the image. I am not interested in the details of the image but in the shape of the object to track. By blurring, we create smooth transition from one color to another and reduce the edge content. We use thresholding for image segmentation, to create binary images from grayscale images. The blurred image is shown as figure 4.

Figure 4 blurred image

In very basic terms, thresholding is like a Low Pass Filter by allowing only particular color ranges to be highlighted as white while the other colors are suppressed by showing them as black. I've used Otsu's Binarization method. In this method, OpenCV automatically calculates/approximates the threshold value of a bimodal image from its image histogram. But for optimal results, we may need a clear background in front of the webcam which sometimes may not be possible. The image after thresholding is shown as figure 5.

And then, I drew the contour, found convex hull and detected convexity defects. The convex points are generally, the tip of the fingers. But there are other convex point too. So, we find convexity defects, which is the deepest point of deviation on the contour. By this we can find the number of fingers extended and then we can perform different functions

according to the number of fingers extended. The detected convex hull and defects are shown as Figure 6.

Figure 5 Image after thresholding

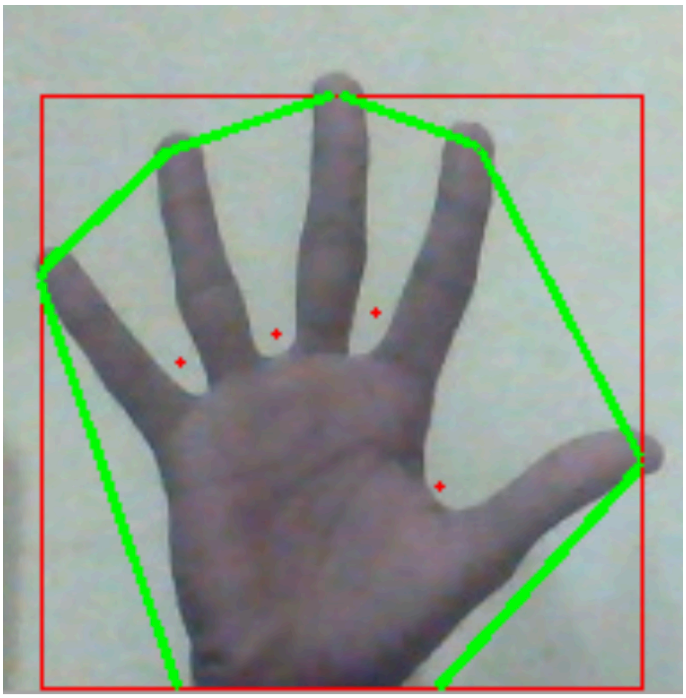

Figure 6 Convex hull and convexity defects

Using the method of finding convex hull and convexity defects, I succeeded in recognizing 5 different gestures in a relatively sensitive way. However, it still existed failure when the light is pool or the background is too dark. However, my partner combined this method with another method, which turned out to be rather sensitively and robustly.

When it comes to the ballgame design, I first put up a simple platform containing just one moving ball and one controllable paddle. Based on Tkinter pattern, I first built up a canvas and set its parameters. Then I defined a class of moving ball, where I set two original parameters, one is canvas, the other is color. And in the original function, set an original location and let ball in the canvas. In addition, I also defined a draw function, which aims to make ball move. More importantly, I also add two variables (self.x self.y) to let ball rebound when it runs into the boundary or paddle. And I also added a judgement that game will be over once the ball touch the bottom of the canvas. And then we set up another class of paddle. In its original function, I firstly set two keys to control the paddle to move. By the time, a simple ballgame platform was successfully constructed, which is shown as Figure 7.

In the next step, I again defined a stillball class to improve the ballgame. In its original function, I let still balls exist on the canvas randomly with different colors. Besides, I also needed to let it disappear when the moving ball hit anyone of them. The improved ballgame is shown as the figure 8.

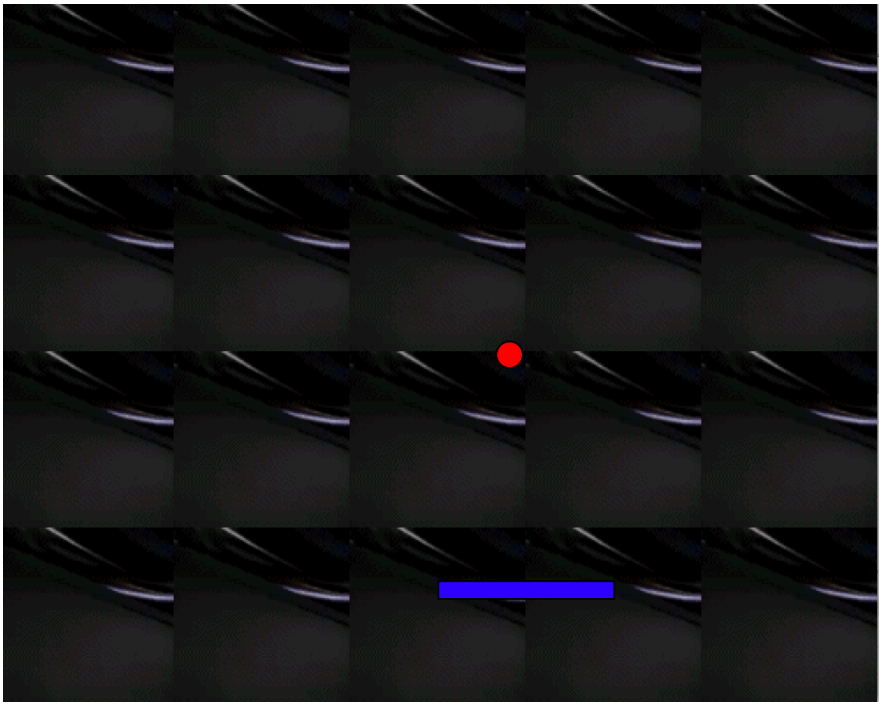

Figure 7 original simple ballgame

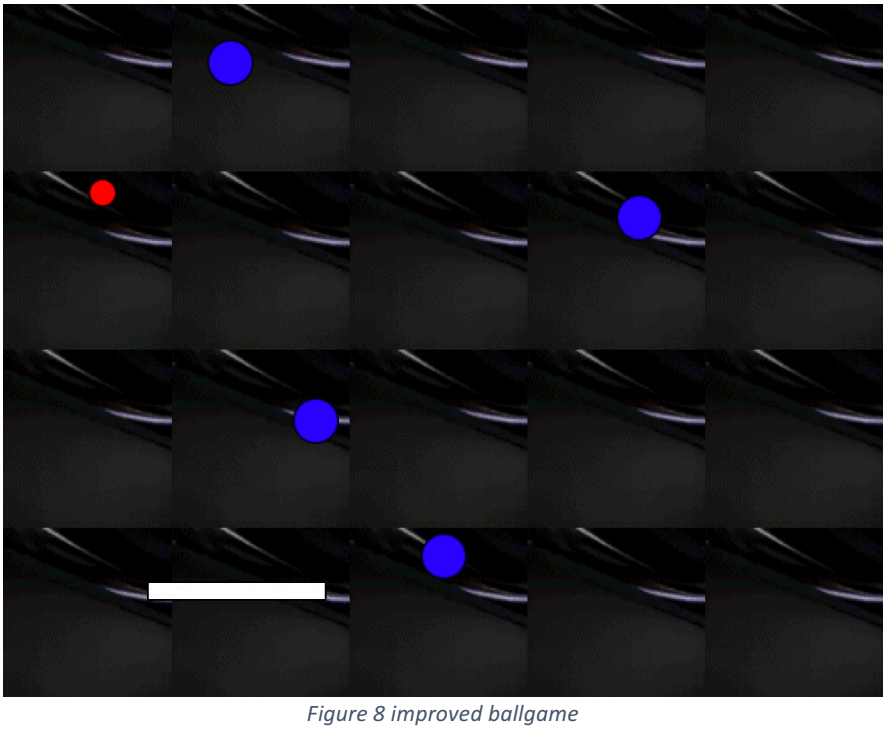

Figure 8 improved ballgame

And then I worked on how to combine finger cursor with the ballgame. I tried to put my ballgame platform into the finger cursor frame and transformed my draw function of

paddle. I made the location of the fingertip as the location of the paddle. That is, once detecting relative gesture, the paddle will move with the fingertip. To make it more fun, I also define a function of draw gold, which aims to drop gold bonus in some likelihood when the moving ball eats anyone of the still balls. The transformed ballgame platform is shown as figure 9.

Figure 9 Transformed ballgame platform

Finally, I added some other functions, including energy, timer and remaining ball calculator. All the functions are displayed in the same frame. And I, together with my partner, ran the code under different circumstance, and it turned out to be good. The whole demo output platform is shown as figure 10.

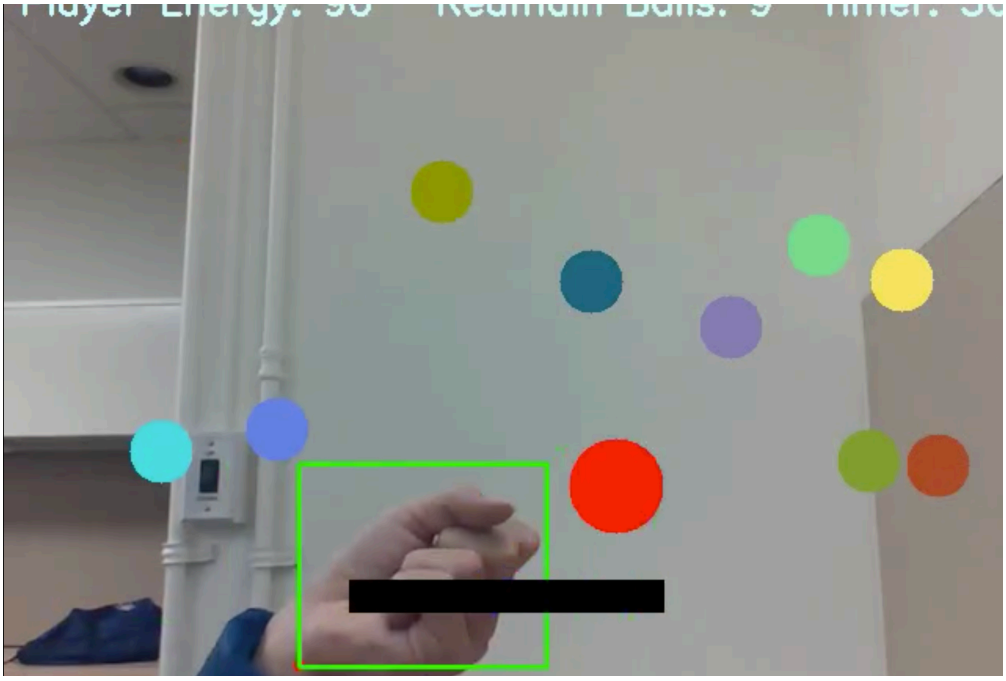

Figure 10 The whole demo output platform

Chapter 4 Results and Expectation

In general, our results were quite impressive. In the demo of success as well as failure, our detection algorithm is 100% accurate, displaying a point on the tip of the index finger in every frame under the positive circumstance. This result is pleasing that the paddle controlled by the fingertip is sensitive and robust.

However, we did not get a chance to implement our algorithm under a bad environment. Time constraints prohibited a deeper analysis of the algorithm under different environments. Therefore, it will become our future job to enhance our algorithm.

Overall, I am quite satisfied with our design. The algorithm is accurate and robust, as requested in the problem description. Fingertips are identified with great accuracy across many tests. Also, I am confident with the ballgame platform that I designed in this project,

which is successfully combined with the finger cursor and reveals how well that our algorithm performs.

The main areas for improvement and future work that could be applied to our project are shown as follows:

- More suitable circumstance: our current algorithm cannot be implemented well in the poor light. Thus, we should make it applicable to more suitable circumstance.
- More complex gestures recognition. As aforementioned, our model currently can only detect two simple gesture sensitively and robustly. When it comes to the relatively complex gesture, it will fail. Therefore, it is our next goal.
- More independent paddle design. In our model, the paddle is fixed and cannot be changed as we want. If we could implement creating the paddle anytime and anywhere as we want, I believe that the game will be more interesting.

With regard to the course, I thought the course material was covered in an appropriate level of depth. All of the relevant topics were covered through the textbook and relevant examples in class. One area on which I would have enjoyed a more in-depth analysis is face detection. Since we only spent part of one lecture on it, I felt that we didn't fully explore the possibilities and ramifications of this particular sub-field. In addition, more information on face detection could provide for more interesting term projects, as well as projects that have real-world value, as face detection is a valuable, emerging field.

Chapter 5 Course Feedback and Suggestions

Generally speaking, I very much enjoyed this course and its respective content. Learning how to implement important image analysis techniques was very interesting, and a very welcome addition to my skillset as a programmer.

As for suggestion, I just have one little suggestion for such course. The slides of each lectures is a bit simple that it did not cover what the instructor shares in the class. Thus, I recommend that you could make slides more details, especially including ways to resolve confusion and to reinforce key concepts. And it would be better for us get a better understanding of the content of each lecture if lecturer could provide more interesting demos of impress us.

All in all, EECS 332 – Intro to Computer Vision is a thought-provoking and interesting class, which I learn a lot basic knowledge of the field of CV. And I would recommend to any undergraduate student or other graduate students who are interested in Computer Vision.