Simple Pendulum Experiment and Automatic Survey Grading using Computer Vision

Dr. Daniel Vrinceanu and Anish Patel

Dr. Daniel Vrinceanu (vrinceanud@tsu.edu) is an Associate Professor at Department of Physics at Texas Southern University and Anish Patel (a.patel6772@student.tsu.edu) is an Undergraduate Student at Texas Southern University

Abstract

The system which extract information from images and videos using intelligent application, after that manipulate and process it is computer vision. This system helps us in scene recognition, character recognition, motion detection, object recognition on the bases of shapes, and so forth. As for example, automatic vehicle number plate, bank check, face, postal-code recognition work on this idea.

During Summer Undergraduate Research Program (SURP) we created software application where first computer program is able to do physics experiment like Simple Pendulum using live camera and provides all quantities like speed, energy, angle, and tension on screen. Second, the program does Automatic Survey Grading filled by students, in the beginning it transforming original image to grayscale level, then the image goes for further in-depth processing process and after that guess the answer according to total number of pixel inside the filled and un-filled square.

Introduction

The research project aims to develop computer application which solved real world problems. In our recent project we use high level programming language like C++ (this is the computer language which has power to go inside the hardware of computer system) and C# (through which we can develop desktop and mobile native application using visual studio). Moreover, we use computer vision library like OpenCV for C++ and EmguCV for C#, which has same concept but built for two different programming language. Below we have discussed about two applications, first which do physics experiment like simple pendulum using camera and, second grading of survey paper by opening images from computer.

Abstract concepts, like velocity or force, are used in science to explain and interpret measurements and observations. An experiment is captured by a live camera and displayed on the computer screen together with some graphical representation that is calculated on the fly based on the acquired image. This methodology has been adopted by commercial products such as Google Glasses, Microsoft HoloLens and Canon Mixed Reality. The digital camera captures images that are interpreted by Computer Vision to identify objects and features in experiment's scene. Graphical objects are then placed in the scene at the corresponding positions in the captured images and then displayed on the screen superimposed on the original image. As an example, we developed a pilot application to observe a simple pendulum experiment with a camera and display it with forces, velocities and acceleration. This application has been developed in C++ using the OpenCV computer vision library.

Another application uses Optical Mark Recognition (OCR) technology which will provide institute / university to grade their survey automatically using software, rather than doing by hand and provide their desire result as a text file at default destination in their computer. The program is really robust in comparison to many applications which is available in the market which grades just if the circle is totally filled, but our application can work in any condition like if some student just tick mark, cross, half-filled bubble, and small little mark. Moreover, the main formula of this application is to detect all crosses inside page (figure 1), so we don't worry about the text, QR code, and so forth. According to the pixel density of each square we can assume that the square which has highest number of pixel in them is the answer.

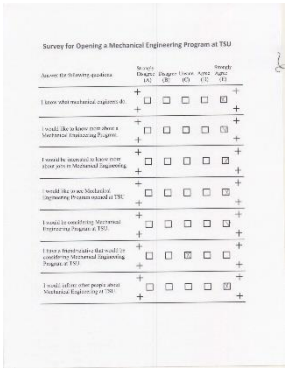


Figure 1 sample survey

Materials

- **Project #1 (Simple Pendulum)** - Microsoft LifeCam HD - 3000
- **Project #2 (Survey Grading)** - Scanner

Methods

- **Project #1 (Simple Pendulum)**

The image frame is received by live camera and saved in matrix format then image is transform into HSV format, means more brightness where the intensity of color is more and we can Adjust the color range of BGR (blue green red) from 0 to 255 using the slider provided on application. Moreover, image goes for further in-depth processing called contour detection with specific size and mark filled circle around them if it found, and after that it draws line which connect detected objects. Once the object is at normal position we can draw constant line by pressing button on application, which work as reference line when the pendulum is in motion and here is the list of formulas used in program,

- $L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
- T = g [3 * cos(d) – 2 * cos(h)]
- $V = \sqrt{2 * 9.8 * l * (\cos(d) - \cos(h))}$
- K.E = m * 9.8 * $\boldsymbol{\iota}$ * (cos(d) – cos(h))
- P.E = m * g * $\boldsymbol{\iota}$ (1-cos(d))
- Total Energy = K.E + P.E
- **Note:**

  L = distance between two point (e.g. length of string – assume 1 meter), $\boldsymbol{\iota}$ = length of string, m = mass of bob (assume 1kg), x_1 & x_2 = point on x-axis, y_1 & y_2 = point on y-axis, T = tension, m = mass of object, g = gravitational acceleration constant, h = maximum angle in radian, d = dynamic angle in radian

v = speed, K.E = kinetic energy, P.E = potential energy

Get all the data on screen.

- **Project #2 (Survey Grading)**

Open image from computer as .png or .jpeg format using button on application. Original image is transform into gray-scale, which reduce the intensity of all colors and transform them into black and white, which increase the processing power of program. Apply Gaussian blur, Morphology, and canny to gray image to detect cross with morph size 19 to cross function. Also, apply Morphology and binary threshold (means 0-black and 1 – white) to detect answer with morph size 3 to answer functions on gray image and save both process image in two different variable. Furthermore, contour can be found from above cross process image and mark filled circle around them, also get the moment of each detected contour (means co-ordinate position with respect to origin of all cross) and save it in dynamic array, which data we will used as reference and mathematical calculation in further program. Get the data of dynamic array and remove the empty element and then group it all cross by four to draw the outer rectangle, after that draw five sub-square inside that. At last, calculate the total number of non-zero pixel inside in all sub square and find average number of pixel in them and increase the value by 70 pixels (approx.). The place where answer is marked like filled box, check mark, or any other shape will have more white pixel then other unfilled square and that is the answer. Save the following data in text file (.txt).

Results & Discussion

- **Project #1 (Simple Pendulum)**

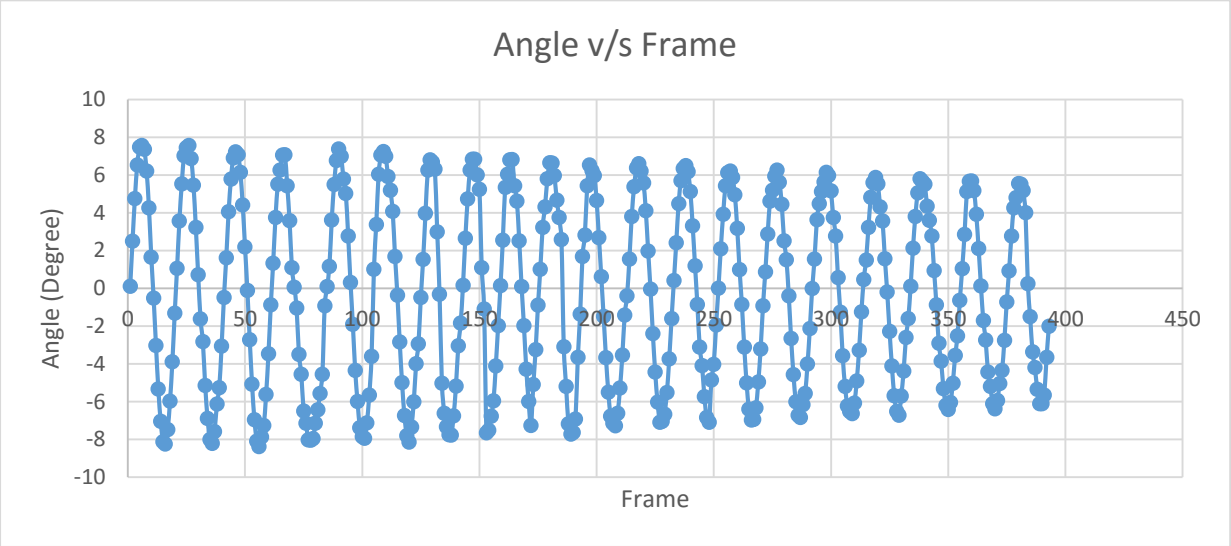


Figure 2 shows angle of pendulum

To receive the accurate angle, we have to aligned camera and simple pendulum perpendicularly during the experiment time. Moreover, the above graph (figure 2) shows angle taken by application while simple pendulum was in motion and we took five consecutive data reading with almost 700 frames of images with approx. 25-30 FPS (frame per second). To check the precision of application we can determine from graph that the maximum angle on both side of reference line (means zero) is almost same, which proves that the angle on both side of reference line is almost same. Below is the screen shot of sample application while pendulum was in motion, which is accurate in getting the angle but at this moment it fails to calculate other quantities like speed, tension, and potential/kinetic energy because we are still working on this application. In our data, we observe that as pendulum is at normal position in figure 4 the angle is almost zero and we receive 0.8735° while in figure 3 the pendulum is at 32.6863° when it is in motion.

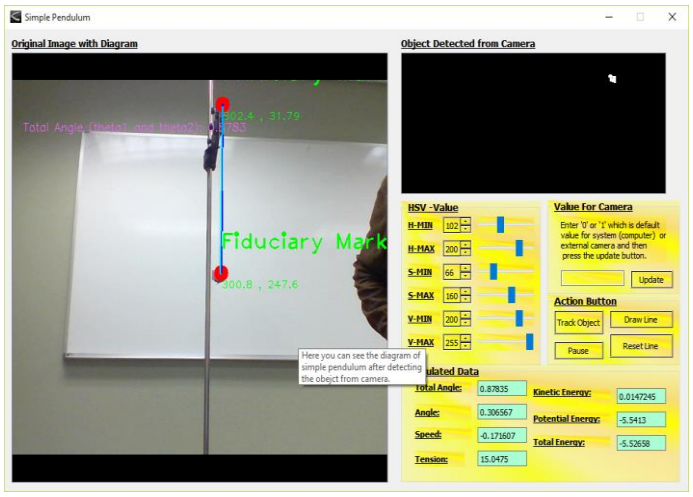


Figure 4 Pendulum at normal position

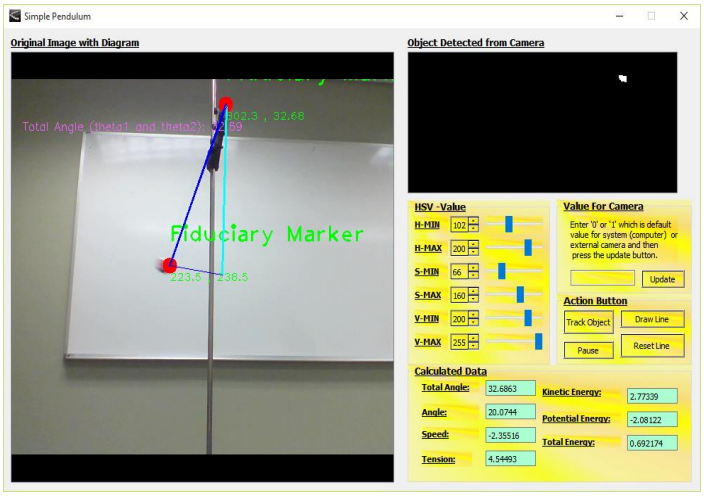


Figure 3 Pendulum in motion

- **Project #2 (Survey Grading)**

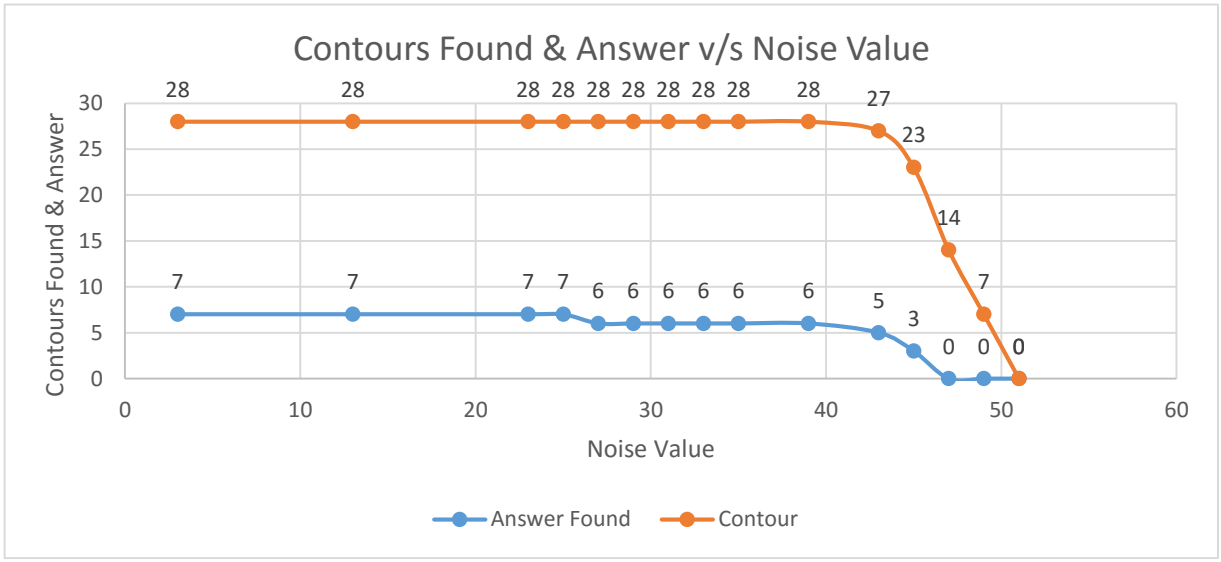


Figure 5 Compare contours and answers with the noise value

The application of automatic survey grading was test on five different kinds of survey papers marked by different students contained properties such as, 1275*1650 pixels, horizontal & vertical resolution 150 dpi and depth 24. Moreover, to test our application on different situation we increases/decreases noise value and observe that as the noise increase on image the amount of contours decreases, therefore which leads to less prediction of answer. Additionally, as you can see in above graph (figure 5) that till specific value of noise from three to twenty-five (3 to 25) there is no effect of noise on image, so the total number of contours and answer was accurate in that range of noise, but as the noise value increase the perdition of contours and answers decreases and goes to zero. Additionally, below are the sample figures of application while running using sample survey paper display in figure 6 and screenshot of application is displayed in figure 7, the top-left corner displays the binary image, while bottom one is fully process image with rectangle and sub-square marked in that. The table (figure 8) describe the total number of non-zero pixels inside the rectangle (top-right corner in figure 7) and sub-square (bottom-right corner in figure 7) and as we can see the place where answer is marked, has highest number of pixel than the average and any other sub-square of rectangle and that is the answer.

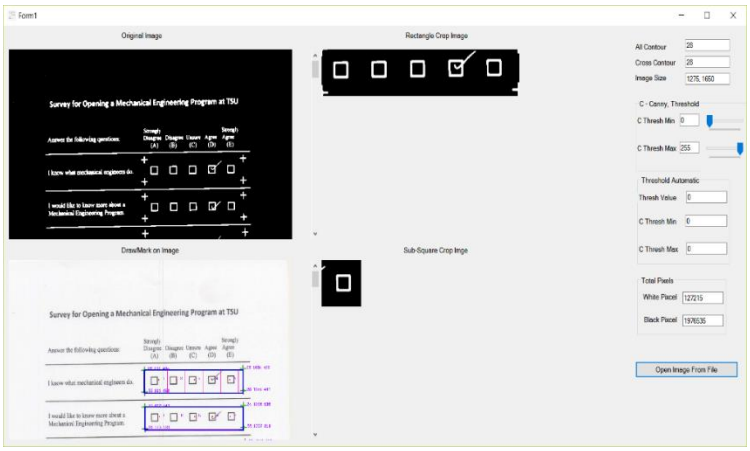


Figure 6 Application to grade survey paper

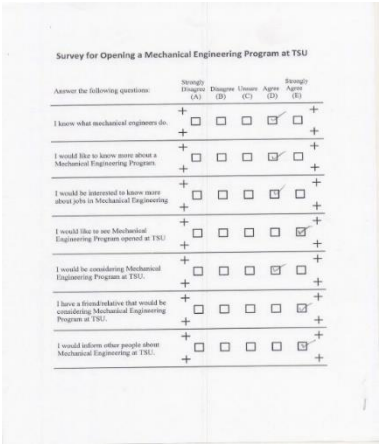


Figure 7 Sample marked survey paper

No of Questions	Total Pixel Inside Rectangle	Pixel Inside sub-five Squares					Average Number of Pixel in Square + 70
		Square 1	Square 2	Square 3	Square 4	Square 5	
		A	B	C	D	E	
1.	3580	595	602	606	859	628	728
2.	3382	602	594	565	790	567	693
3.	3424	582	591	599	758	567	689
4.	3496	572	570	574	568	872	701
5.	3795	632	627	651	875	646	756
6.	3420	580	574	592	577	755	685
7.	3808	636	631	648	645	855	753

Figure 8 shows total number of non-zero pixels inside rectangle and sub-square

Conclusion

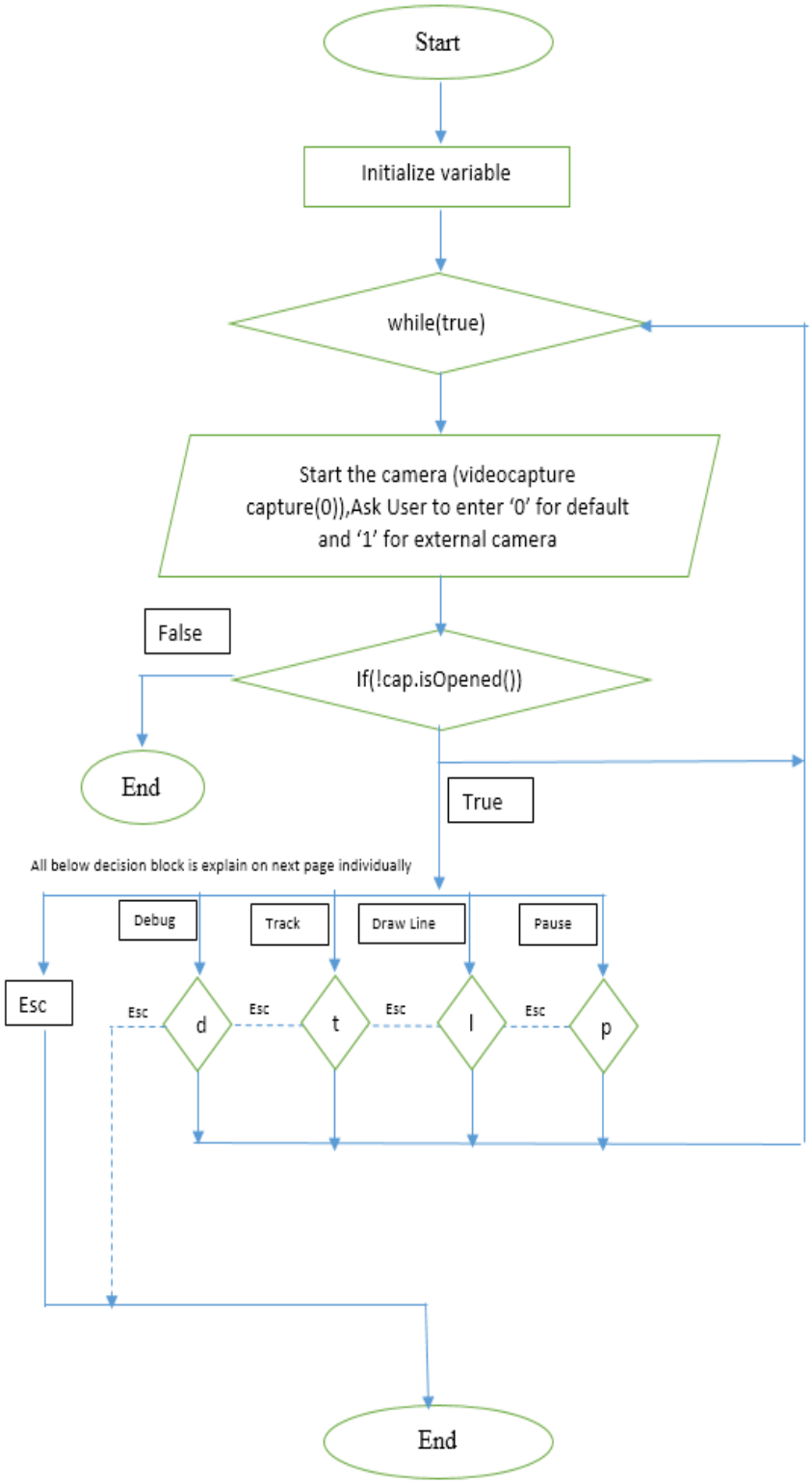
There are many more possible science experiments which can be program like collision of atom, find total number of protein inside microscopic cell, and Collecting data of patience for couples of years and predict what kind of medical prescription they need for future. Moreover, the next stage of survey application will use pdf file and transform it into image (.jpg or .png) and process each image one-by-one and save the desire result in spreadsheet and e-mail to respective person. In addition to this, automatically grading handwritten letters and words by using camera, which will reduce lots of work in the field of education, automatic address checking for post office, reading application for college – immigration office.

Reference

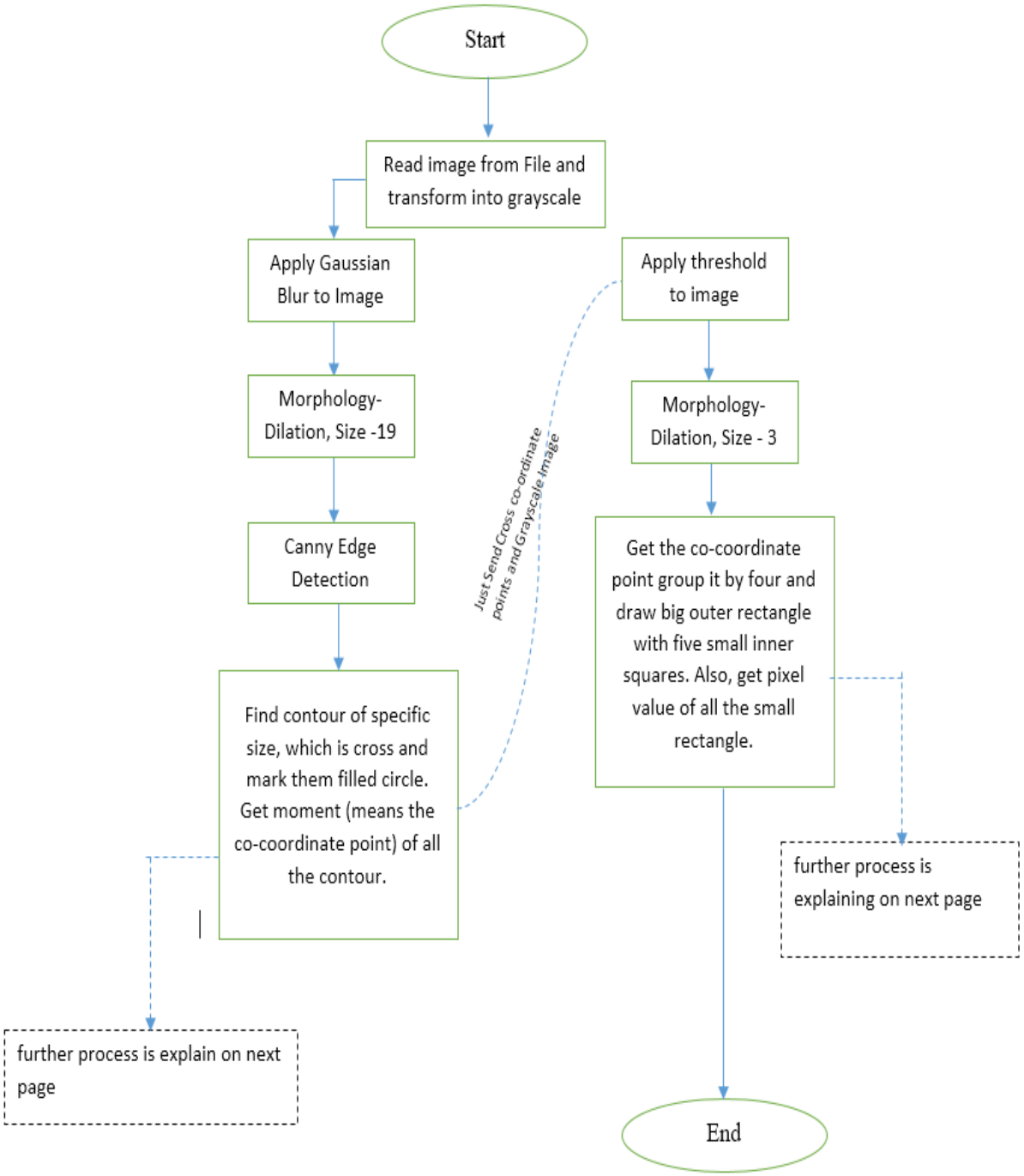
- www.opencv.org
- www.youtube.com
- http://stackoverflow.com/
- www.github.com

Programming Flowchart

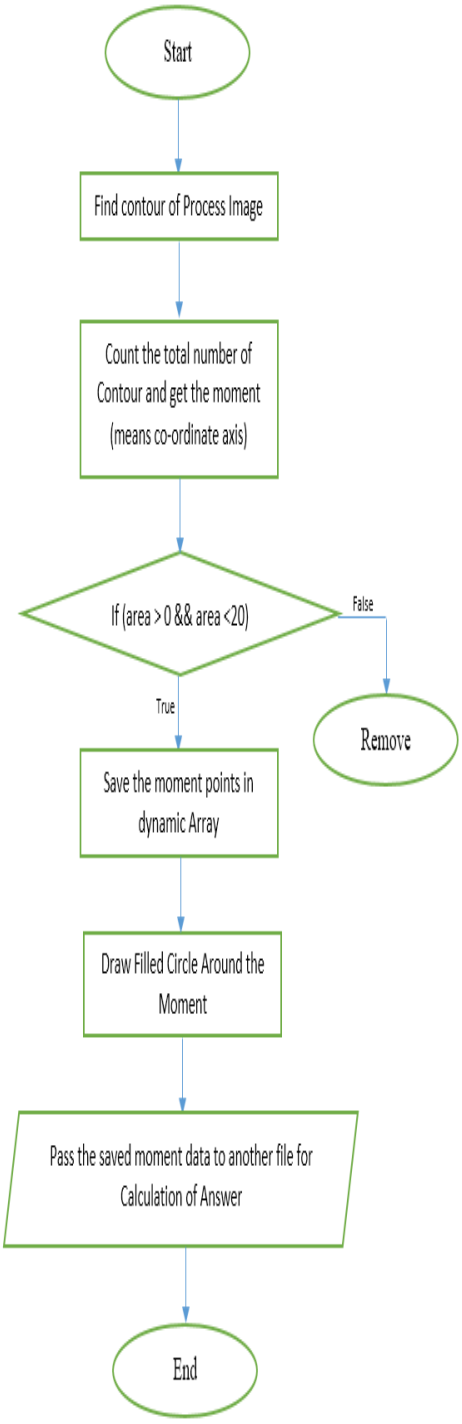
- **Project #1 (Simple Pendulum)**



- **Project #2 (Survey Grading)**



Cross Function Flow Chart



Answer Function Flow Chart