Conversion of Virtual Writing on a 2D Plane into Real Writing using LED Pen and Web Camera

Abdul Lathief Jubailah Bagum

Department of Computing,
British College of Applied Studies, Sri Lanka
jubailah.lathief@gmail.com

Abstract. The use of conversion of virtual writing on a 2D plane into Real writing is becoming a very important need for this contemporary developing community as far as we are looking for. As we see, writing on a vast screen in big letters in front of a huge crowd is essentially needed in a conference for instance. But, because of the lack of facilities for that it becomes impossible unfortunately. So, they just explain the fact to the audience without writing anything which makes the crowd feel uncomfortable. Therefore, we have created a tool to compensate the very essential need that most of the people are looking forward.

Key words: Virtual writing, Web camera, LED pointing pen, Image Processing, C#.Net

1 Introduction

In this project we are going to convert writing on a virtual 2D plane to real writing using an LED pen and camera. The 2D plane is not light sensitive but it acts as a light sensitive plane with a help of a camera. The camera will monitor the movements of the pointing device – a LED pen (an Infrared or normal LED pen/torch). See Fig. 1.

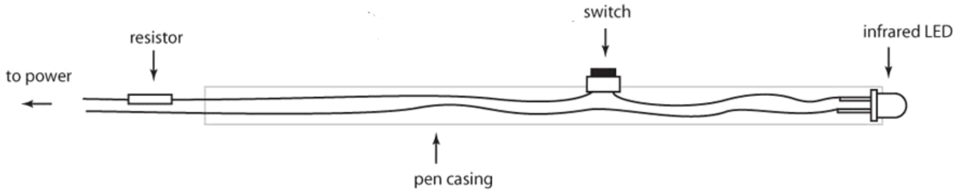


Fig. 1. LED pointing pen

Here we are going to use a LED light pointer. While we are writing on a 2D plane using the light spot of the LED pen virtually, the camera captures the movements of the light spot and those images will be sent to a computer. After that the tool that we are going to develop will recognize the final shape of the written figure and make it visible on the monitor screen. By using a multimedia projector, the written figure can be made visible at the place where we wrote the figure virtually.

Received: 09th Dec, 2020
Revised and Accepted: 29th Dec, 2020

2 Motivation

Today many devices and systems in the marketplace provide plenty of digital information handling capabilities to allow user to enjoy different digital services and consume information with easiness. For example, HD monitor can be used as a television or as a display for a personal computer. A smart phone allows the user to make phone calls, watch video clips, browse webpages, play games and use all kinds of applications.

The mode of digital information consumption is raised to a new level with the improvement of new technologies, such as the touchscreen and motion sensor. An effective user interface design together with a proper control device can reduce the complexity occurs in these technologies.

When we talk about a user interface which is designed to match the functionality that the device intents to support, we can consider remote control which has an array of buttons each of which is associated with a command or action for the target system to perform.

The touch-based pointing device is widely used nowadays. The multi-touch technology further broadens the gesture methods to enable more control actions and functions. The touchscreen technology is a perfect example of combining the display and single or multi-touch tracking to build a very efficient control interface. And also it is essential to consider about the touch-based writing on a digital screen which we can see on tablets and notes that are very popular most recently.

Touchscreen works very well on handheld devices or display panels within arm’s reach. When the display is fairly large or far from the user, the touch control becomes infeasible and a new type of control interface is needed.

Therefore, here arises an unfulfilled demand for above mentioned effective user interface which we are going to create and implement hopefully.

3 Literature Survey

3.1 A Virtual 3D Blackboard - 3D Finger Tracking using a Single Camera

It is a method for tracking the 3D position of a finger, using a single camera placed several meters away from the user. After skin detection, use motion to identify the gesticulating arm. The finger point is found by analyzing the arm's outline.

To derive a 3D trajectory, we first track 2D positions of the user's elbow and shoulder. Given that a human's upper arm and lower arm have consistent length, we observe that the possible locations of a finger and elbow form two spheres with constant radii. From the previously tracked body points, we can reconstruct these spheres, computing the 3D position of the elbow and finger. These steps are fully automated and do not require human intervention. The system presented can be used as a visualization tool, or as a user input interface, in cases when the user would rather not be constrained by the camera system [4] [6].

3.2 Sketching in the Air - A Vision-Based System for 3D Object Design

3D object design has many applications including flexible 3D sketch input in CAD, computer game, webpage content design, image based object modeling, and 3D object retrieval. Most current 3D object design tools work on a 2D drawing plane such as computer screen or tablet, which is often inflexible with one dimension lost.

On the other hand, virtual reality based methods have the drawbacks that there are awkward devices worn by the user and the virtual environment systems are expensive. In this paper, we propose a novel vision-based approach to 3D object design. Our system consists of a PC, a camera, and a mirror. We use the camera and mirror to track a wand so that the user can design 3D objects by sketching in 3D free space directly without having to wear any cumbersome devices.

A number of new techniques are developed for working in this system, including input of object wireframes, gestures for editing and drawing objects, and optimization-based planar and curved surface generation. Our system provides designers a new user interface for designing 3D objects conveniently [3].

3.3 Face Detection in Digital Imagery Using Computer Vision and Image Processing

The aim of this thesis was to add an increased detection rate for profile/partial faces while increasing the stability and run-time of the system. A main goal of the project was to increase throughput in detected faces at any given image, if the image contains faces. If the image does not contain any faces, then none should be reported as detected.

Augmenting detection of faces in non-vertical positions has multiple angles in the same image and detecting a high percentage of existing faces. Also, the system should be able to detect faces in profile, not excluding already given result and not generating false-positive detections.

Additionally, I strive to create a modern implementation that uses system resources in an economical, and optimal, way. Another goal is that the construction of the new implementation has to be integrated in a currently used system, and using the current API for integration and delivering the same standard reply format [5].

3.4 Virtual Touch Screen

The Virtual Touch Screen is a simulation of a Touch Screen using a webcam, a mouse pointer simulation which is actually the user's finger or any other object used by the user as pointer. It is done by motion detection - subtraction of the images captured at time t and t+1. Because of the great complexity involved, it is extremely hard to detect the user's finger at any possible camera input.

4 Project Objective

The objective of our project is to develop a user input tool which converts a figure which is virtually written using a LED pen on a 2D plan into really written one using a camera.
It is going to be an integrated Image Processing framework that supports all general interface functionalities in a single design.

5 Methodology

The method is depicted in the flow chart of Fig. 2. As it can be seen the whole algorithm is comprised of following three steps

1) Detecting Difference :- which is used to detect difference between images
2) Finding Coordinates :- which is used to find coordinates of points
3) Result Display :- which is used to draw the motion in software tool

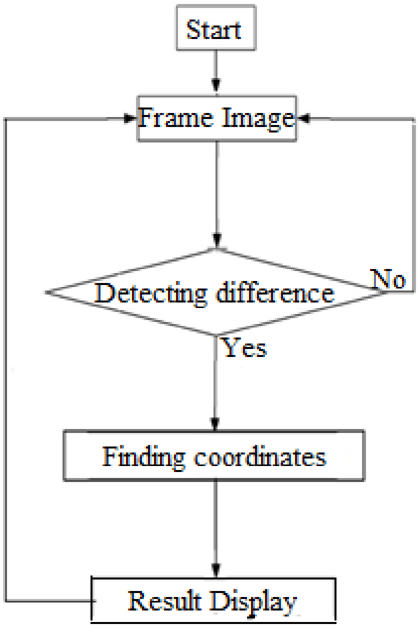


Fig. 2. Flow chart for the methodology

The implementation is in C# language.

5.1 Step 01: Detecting Difference between images

For this part we use the *AForge.NET Framework-2.2.5*. AForge.NET Framework is an open source which supports for webcam. In this part we are going to take photographs of the movement of the LED pen in every 100ms using webcam.

Then we apply a filter called *Euclidean filter* to filter out particular color (red) of an image which we capture by webcam. Euclidean filter is a filter which filters out a particular color. Fig. 3. shows the output instance we got using that filter.

Next to that we apply a filter called *Threshold filter* on two images which is taken by Euclidean filter. Threshold filter is a filter which takes two images and detects difference between two images. If there is any difference between two images then it will give a black and white image.

Here white color represents the difference. If there is no difference between two images then it will give a black color image. Then using that resultant image calculates the number of white pixels of resultant image. Fig. 4. shows an output instance of usage of Threshold filter.

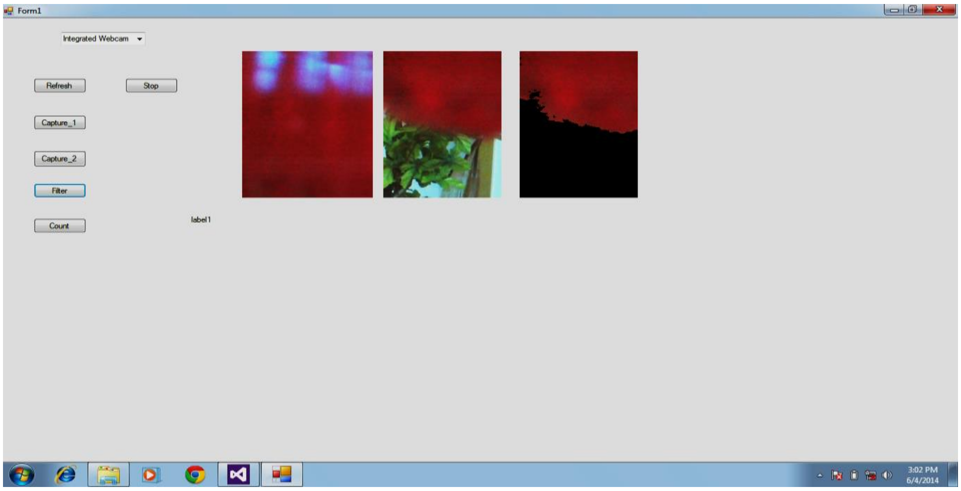


Fig. 3. Resultant of Euclidean Filter

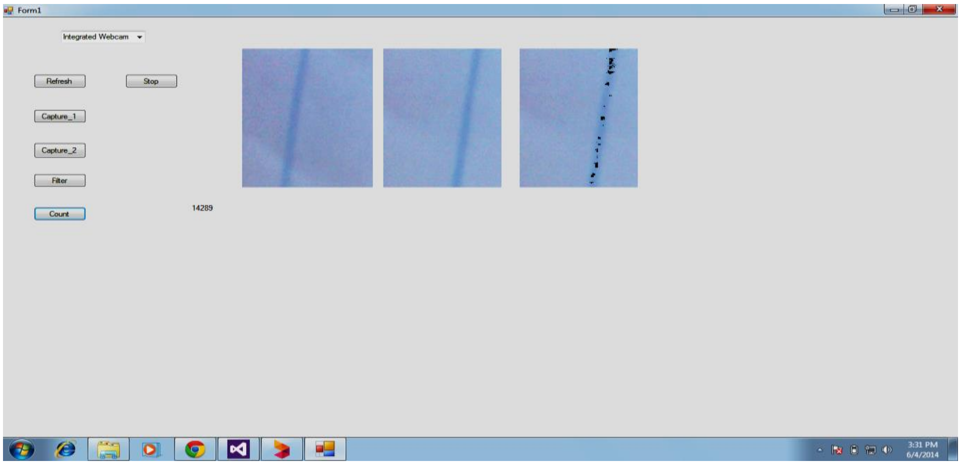


Fig. 4. Resultant of Threshold Filter

If there is any difference between two images it means threshold filter give a black & white image then those two images are the input to next task.

Problems. Problem what we faced in this part so far is we have done this part using webcam_capture.dll reference. But it couldn’t work more than one time. It is operating system dependent reference.

Solution. We found a new solution that is AForge.NET Framework-2.2.5. It is an open source. It provides all the support for web camera.

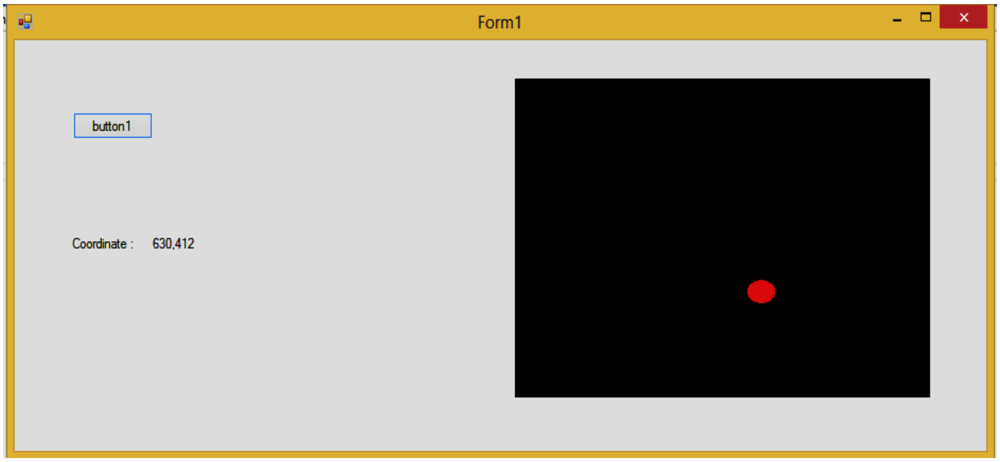
5.2 Step 02: Finding the Coordinates of a point in the image

After getting the output images of the previous task as the input for this task we develop the solution for the problem one more step.

The specific input for this part (program) is an image which has got a point that is different in color from the background (E.g.: Red point on a Black background). See Fig. 5.

After that the coordinate of the point in the image is going to be displayed. This is also done in the IDE called Microsoft Visual Studio with C#.net.

Here we are using the getpixel() method in C#.net. Using this method every pixel in the image is tested. When the traversal is going on it detects the pixel for the red color. Eventually it displays the coordinate of the final pixel corresponds to the Red color.



Coordinate : 630,412

Fig. 5. Screen shot of the output

The problem arises in this method. Here we assume that the shape of the point which comes from the spot light (LED pen) is a circle. But it cannot be an accurate circle. So finding the coordinate of the last Red pixel becomes as an issue. So as to solve this we are going to convert the shape of the point as a rectangle by taking the whole area of the visible point, using our program.

By taking the length and the height of the rectangle we can calculate the center of the point. See Fig. 6.

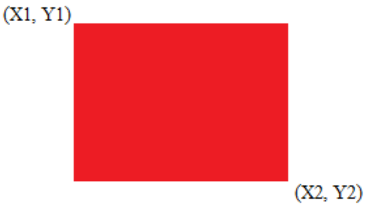


Fig. 6. Converted rectangle

Length L = X2 – X1 . (1)

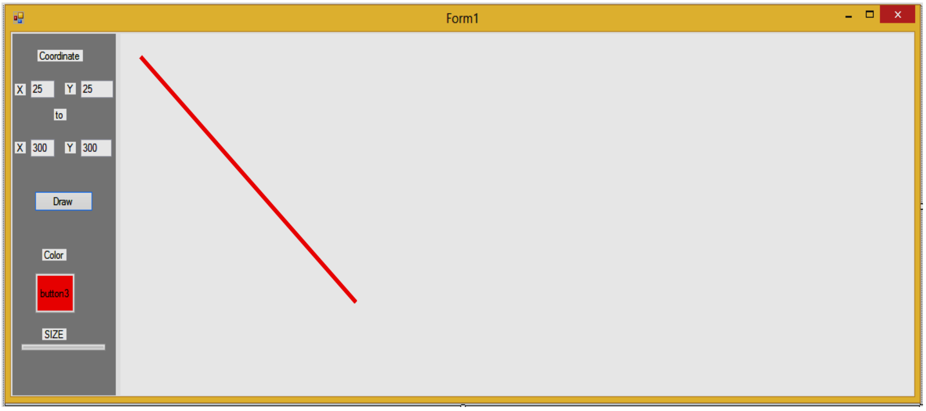
Height H = Y2 – Y1 . (2)

Center coordinate = ((X2-X1)/2, (Y2-Y1)/2) . (3)

5.3 Step 03: Result Display

In this final part of the task the major work done is joining two coordinates of points which are the output of the second part and given as the input for this part.

Once the coordinates of two points are received they are joined with a colored line using the program we have developed. Fig. 7 shows the output we got as the result display. When a null coordinate is obtained it represents the break (discontinuity in line) between two points. For instance it can be a gap between two letters.



X 25 Y 25

X 300 Y 300

Fig. 7. A line joining two coordinates

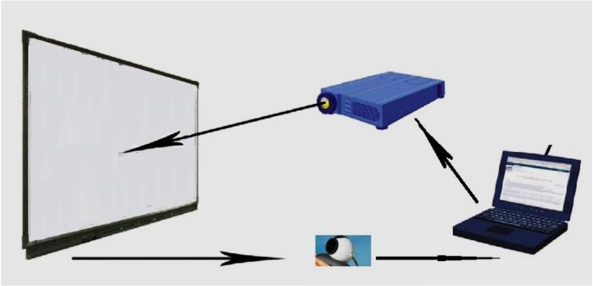


Fig. 8. System Design

6 Conclusion

The tool has been created in C# programming language and can be executed in IDE platform like Microsoft Visual Studio. People who use this tool will find their works such as explaining something graphically in front of a big crowded people, easier and comfortable. Rather using software applications such as Paint and Microsoft Word where we use mouse or touch pad to draw which make us inconvenient, while using this tool it will be more convenient in drawing on our own with the help of LED pen on a 2D plane.

7 Future Direction

As far as our research project is concerned, it is a C# program that has to be run and executed in an applicable IDE platform such as Microsoft Visual Studio. It is not advisable to expect everyone to handle the program whenever needed. It is not practically possible as well.

Therefore, hereafter we have been planning to develop this program as an application which can be installed and utilized in the devices like personal computers, smart phones, tablet PCs and etc. Anyhow, the web camera and the projector must be there at the place of implementation.

8 Acknowledgement

The whole hearted acknowledgement is expressed to my batch mates who helped me in developing the program code in C# and correcting the errors which had arisen when developing the program for this project.

Especially, I would like to responsibly thank my project supervisor Dr. Kamalanath Samarakoon who is a senior Lecturer at Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, who supervised me throughout the development process of the project.

References

[1] Autodesk Inc. *AutoCAD*. http://www.autodesk.com/.

[2] SolidWorks Corporation. *SolidWorks*. http://www.solidworks.com/.

[3] A. Alexe, L. Barthe, M. Cani, and V. Gaildrat. Shape modeling by sketching using convolution surfaces. *Proc. Pacific Graphics*, 2005.

[4] T. Baudel and M. Beaudouin-Lafon, \Charade: Remote control of objects using free-hand gestures," CACM, vol. 36, no. 7, pp. 28-35, 1993.

[5] R. Kjeldsen and J. Kender, \Towards the use of Gesture in Traditional User Interfaces," Proc. International Conference on Automatic Face and Gesture Recognition, pp. 151-156. October 1996.

[6] Goebl, W.; Palmer, C. (2013). "Temporal Control and Hand Movement Efficiency in Skilled Music Performance". In Balasubramaniam, Ramesh.

[7] David Noonan, Peter Mountney, Daniel Elson, Ara Darzi and Guang-Zhong Yang. A Stereoscopic Fibroscope for Camera Motion and 3D Depth Recovery During Minimally Invasive Surgery.