

Tracking and Recognizing Gestures using TLD for Camera based Multi-touch

Veeramalai Sankaradass^{1*}, Z. Faizal Khan² and G. Suresh¹

¹Department of Computer Science and Engineering, Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai – 600062, India; veera2000uk@gmail.com, sureshwisdomedu@gmail.com

²Department of Computer and Network Engineering, College of Engineering, Shaqra University, Kingdom of Saudi Arabia; faizalkhan@su.edu.sa

Abstract

This research work discusses about the tracking system and also recognizing gestures using Tracking-Learning-Detection (TLD) for camera based multi-touch technology. The tracked fingers are assigned unique IDs, and the information about the finger movements are passed to the TUIO protocol. This enables us to have a communication channel between the screen elements and the touch input. A novel and low-cost approach for tracking objects is discussed. This proposed works facilitate to remove the noise by image processing technique. In this research concluded that, the issue of the touch input technologies is large and the implementation has been fulfilled, and also relatively cheap device was built successfully.

Keywords: Gestures Recognizing, Image Processing, Multi Touch, Tracking System

1. Introduction

Interaction with objects on a multi-touch platform is often limited to the type of display technology used and common marker-based techniques for object tracking, it typically provides little more than position and orientation information for the objects. In our approach, we track the various gestures made by humans on a large on a large touch area. We also used Diffused Illumination (DI) as the lighting condition. Camera-based multi-touch is the technology used building the prototype in this project. Camera-based multi-touch will capture the noise also¹. But it can be removed easily through image processing techniques. This proposed technique will help sense the finger movement and also identify the pointer. Most of the multi-touch systems make use of gesture databases for all the dimensions². This often imparts a condition on the user's gestures. Microsoft offers a multi-touch solution for companies. Touch Screen and Gesture Recognition technologies came into existence nearly two decades back and many innovative ideas have come to light since then³. Some

of them are Resistive Touch, Capacitive Touch, Flux based touch, etc. But to announce a new method to create low cost touch screen technology, one has to consider the previous research works carried out by in the domain experts. This gesture set has been recorder for six Dimensional coordinates. Not only the human hand but other devices can also be used to aid humans o provide gesture inputs to the system⁴. In that category, many new devices have come into play and the paper discusses about some of the easy-to-use devices. Wii is a game console developed by Nintendo and its core operations depend on the movements of the user⁵. A new method based on a C++ library submitted at the Google Summer of Code, helps to turn any PC monitor into a Touch Screen monitor by using the Wiimote. A revolutionary tracking technique was introduced, about which a detailed description is given in this proposed paper⁶. This tracking mechanism serves as the basis to track the finger movements. It focuses on the possible gesture sets that can be used on Mobile Platforms⁷. Fitts' law is inherently 1-dimensional with strong 2D extensions, but it does not extend well to 3D movements⁸.

*Author for correspondence

The conventional methods do not provide the flexibility for implementation in SDR platform and mitigation solution for dynamically updating the model⁹.

2. Methodology

2.1 Object Tracking

For a camera based touch solution, object tracking is the most important feature. Tracking-Learning-Detecting⁶ provides a better solution for object tracking. The components of the framework are characterized as follows: Tracker estimates the object's motion between consecutive frames under the assumption that the frame-to-frame motion is limited and the object is visible. The tracker is likely to fail if the object moves out of the frame but can recover if the object moves into the camera view. Detector treats every frame as independent and performs full scanning of the image to localize all appearances that have been observed and learned in the past. As any other detector, the detector makes two types of errors: false positives and false negative. The learning component assumes that both the tracker and the detector can fail. By the virtue of the learning, the detector generalizes to more object appearances and discriminates against the background.

2.1.1 Positive Analyser (P-Expert)

The goal of P-expert is to discover new appearances of the object and thus increase generalization of the object detector. P-expert can exploit the fact that the object moves on a trajectory and add positive examples extracted from such a trajectory. However, in this system, the object trajectory is generated by a combination of a tracker, detector and the integrator. This combined process traces a discontinuous trajectory, which is not correct all the time. The challenge of the P-expert is to identify reliable parts of the trajectory and use it to generate positive training examples.

2.1.2 Negative Analyser (N-Expert)

N-expert generates negative training examples. Its goal is to discover clutter in the background against which the detector should discriminate. The key assumption of the N-expert is that the object can occupy at most one location in the image. Therefore, if the object location is known, the surrounding of the location is labelled as negative. The N-expert is applied at the same time as P-expert, i.e. if the trajectory is reliable. In that case, patches that are far from current bounding box (overlap < 0.2) are all labelled as negative.

3. Tracker Implementation

The algorithm continuously frames the real-time video feed into logical frames as per the FPS (Frames per Second) rate. The results of the tracking process are written to a log file in the pixel co-ordinates format, [Frame id, Left column, Top row, Right column, Bottom row]. The detected Object is shown in Figure 1. The movement of the detected object is found also shown in Figure 2.

3.1 Diffused Illumination

With help of camera, the finger print will be scanned and the same was captured for further process. The

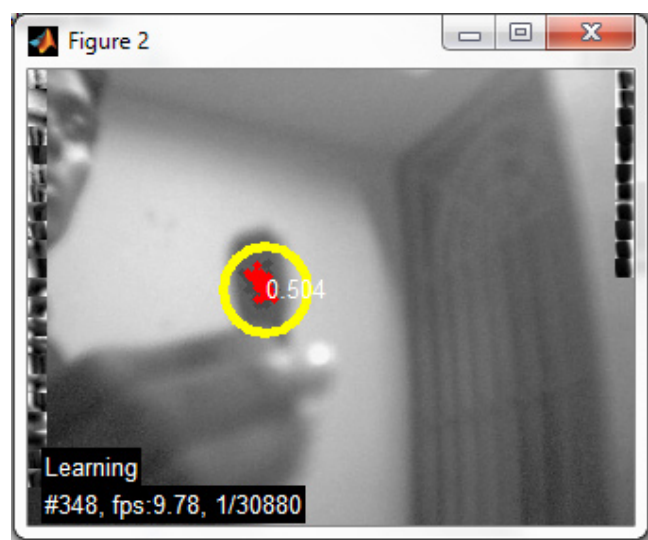


Figure 1. Object is detected.



Figure 2. Object moves.

shadow image itself enough for further process. Diffused illumination method will be suitable for this suitable proposed work analysis. The Diffused Illumination of the object is shown in Figure 3.

3.2 Finger Recognition

The Raw Image from camera and Static Background are subtracted of images in Figure 4 and Figure 5 show how the image is enhanced during the pre-processing. During the pre-processing of the image, the unwanted darkness and shadow will be filtered and the same way the image will be enhanced. Even if it is necessary, it will be amplified to the required level. The sample camera captured non filtered and filtered images are given below. These finger spots are individually called as blobs. Separate blobs are assigned unique IDs for further use.

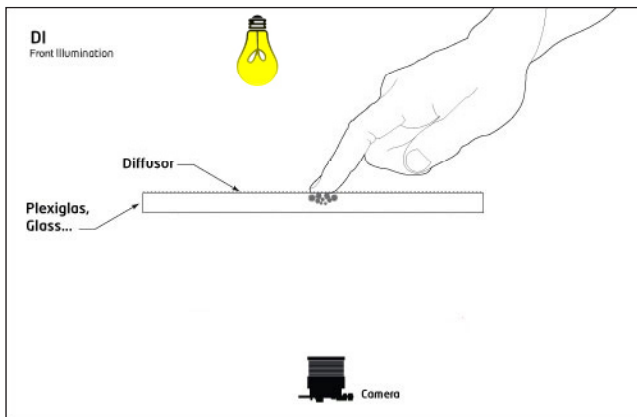


Figure 3. Diffused illumination.

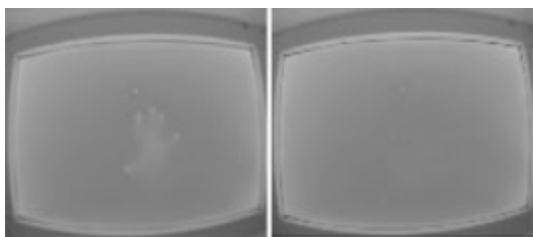


Figure 4. Raw Image from camera.



Figure 5. Static Background is subtracted.

3.3 Moving the Mouse using the Tracking Data

When we move the mouse using the tracking data, this will be continuous monitoring and it will be tracked based on some set of rules called protocol. As a result, an application running on a computer can be manipulated by directly using a touch motion. The protocol will send the message about the image like orientation, size, direction and others to learn about the images.

3.4 System Design Stack

A web camera is used to provide real-time feed of the user's gestures, which is in turn tracked by the tracking algorithm. The tracking data is fed to the TUIO protocol and the process continues towards the top layer of the system design stack. The Stack for native windows applications is shown in Figure 6. A general USB driver for the web camera is used for Windows and Linux based Operating Systems and the FireWire driver is used for implementation on Mac based PCs. The Stack for Flash and C# applications is shown in Figure 7. The Stack for Python and a C++ application is shown in Figure 8.

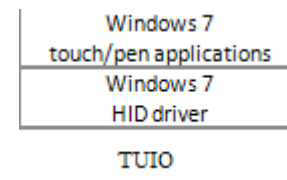


Figure 6. Stack for native windows applications.

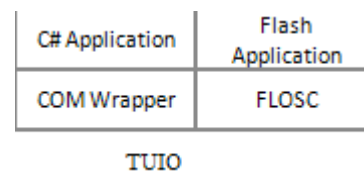


Figure 7. Stack for Flash and C# applications.

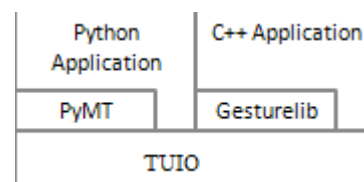


Figure 8. Stack for Python and C++ applications.

3.5 Distances between Touch Points

Algorithms to find the distance between different touch point:

- Step 1: Construct the possible direction in the gesture.
- Step 2: List of direction in the gesture of each step.
- Step 3: Calculating the last point in the direction.
- Step 4: Return the direction list in the gesture.
- Step 5: Calculate the x axis and y axis point direction.
- Step 6: Analysis the angle of the direction point.
- Step 7: If the angle is less than zero then the angel will be 360 degree.
- Step 8: The direction will be integer rounded off 45 degree angle.
- Step 9: If the direction point is differ from direction then,
- Step 10: All direction will be returned.

When you have obtained the direction sequence for the gestures you want to compare, there are different possibilities for how to proceed. The touch distance can be fine-tuned depending on the multi-touch application that the user is working on.

3.6 Featured Applications

All the applications are developed using Action Script, a flash programming language. Since it is hard to implement multi-touch on native operating systems such as Windows, Mac and Linux – we have developed an operating system based on Action Script. From this other applications that can support multi-touch can be launched. In our testing phase, we have found that most of the applications and the operating system itself, to be robust.

List of applications and the operating system developed:

- Spark Touch (the operating system)
- Photo Gallery
- Song River
- Virtual Piano
- Puzzle (game)
- Bloom
- Fire demo (lava lamp simulation)

4. Result and Discussions

4.1 Working Process of the Final System

The multiple windows on Figure 9 show the various versions of the fingers placed on the semi-transparent sheet, which is viewed by the web camera. Numbering the windows on Figure 9 which show the extraction of blobs (white

patches), from 1 through 7. Window 1 shows the real-time input from the web camera. Window 2 shows the black and white version of the camera input. Window 3 shows the infra-red version of the second window. Window 4 shows the background subtracted version. Windows 5 and 6 show the amplified version of Window 4. Amplification helps to brighten the regions that are dim and also helps to extract the desired blobs by controlling the level of amplification needed. The position of the detected BLOBS is shown in Figure 10. A separate window shows the number of blobs detected by the web camera and also their position. The FLOSC stands for flash OSC (Open Source Control), this helps to parse the tracking information to the various Flash based applications that are waiting for tracking-data input. The open-source FLOSC server is started is shown in Figure 11. The circles on the Figure 12 indicate the multiple touch points detected by the system. These circles are nothing but the number of fingers detected by the system and they move as the user moves his/her fingers on the semi-transparent sheet, observed by the web camera. These circles act like multiple mouse pointers. These multiple pointers can be used at the same time and each of the circles respond to the finger movements. The Figure 13 shows the application, Smoke. This application is capable of creating fluid colours for each touch that is made on

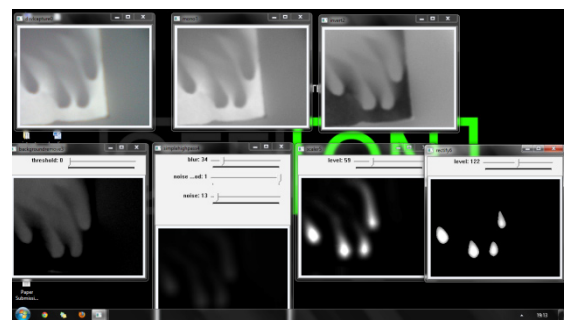


Figure 9. Extracting BLOBS from raw camera input.

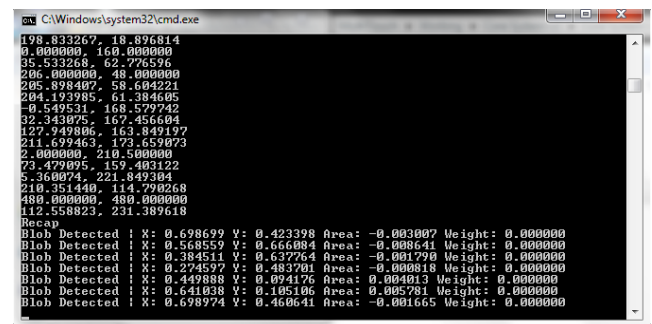


Figure 10. Shows the position of the detected BLOBS.

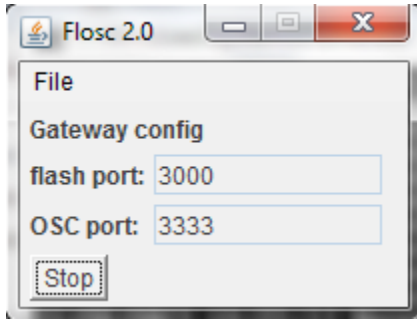


Figure 11. Open-source FLOSC server is started.

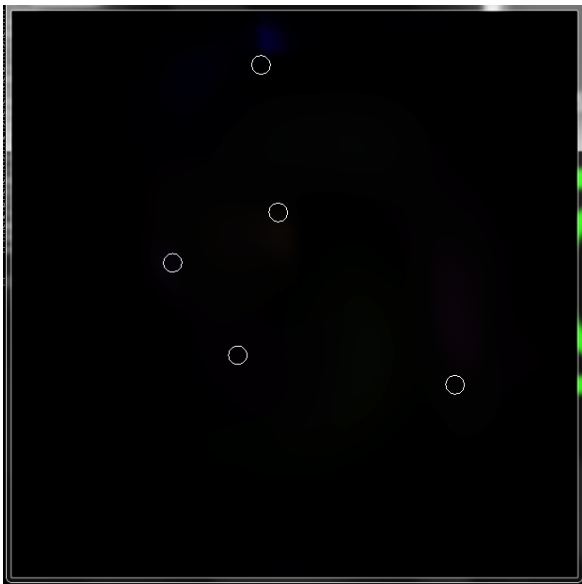


Figure 12. Multiple touch points shown on the screen.

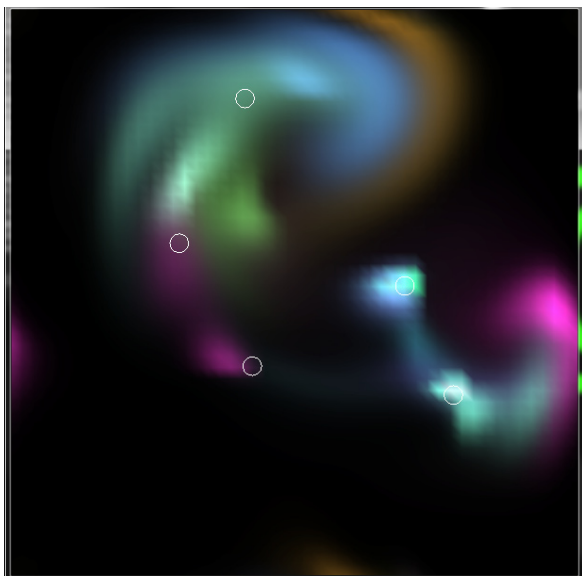


Figure 13. Application: SMOKE.

the semi-transparent sheet and the colours flow according to the movement of the fingers. It can be clearly seen that multiple colours flowing at the same time, indicating a multi-touch environment on a regular Personal Computer. Various flash based applications like Photo gallery, Music pad (plays a musical tone for each touch input), etc., were created and tested upon.

4.2 Low Cost Multi-touch Tables

The user interface is projected onto a plexi-glass panel using a projector from below. The panel prevents the projected user interface from escaping through it. A web camera, which is also present below the panel facing upwards, receives the touch input made by the user. The touches are made on top of the plexi-glass (i.e. the user manipulates the application that is projected onto the panel). The movement of the fingers are tracked using the tracker application and the data is sent to the TUIO protocol. The TUIO protocol then parses it to the application which is capable of decoding the tracking information. In turn, the various objects of the application behave according to the touch input from the user. The application resides on a mini Central Processing Unit (CPU) which provides the power to process all the stages of the system.

5. Conclusion

During the work it became clear that the issue of the touch input technologies is large. The aim of implementation has been fulfilled, and relatively cheap device was built. Future work should be focused on the software optimization and interaction design. Usability of implemented features should be evaluated. An automatic trimming of captured image would be very useful. The tracking algorithm and the methods used to process the information can be fine-tuned to reduce the lag that sometimes can be seen in the current implementation. Integrating all the process involved into a single application can considerably reduce lag.

6. References

1. Amma C, Gehrig D, Schultz T. Air writing recognition using wearable motion sensors. Proceedings of the 1st Augmented Human International Conference, AH '10; 2010. p. 10.
2. Lv Z. Wearable smart phone: Wearable hybrid framework for hand and foot gesture interaction on smart phone. Sydney, NSW: IEEE International Conference on Computer Vision Workshops (ICCVW); 2013. p. 436–43.

3. Chen M, AlRegib G, Juang BH. 6DMG: A new 6D motion gesture database. Proceedings of the third annual ACM conference on Multimedia systems, MMSys '12; 2012. p. 83–8.
4. Hoffman M, Varcholik P, LaViola J. Breaking the status quo: Improving 3d gesture recognition with spatially convenient input devices. IEEE Virtual Reality Conference (VR' 10); 2010. p. 59–66.
5. Lee JC. Hacking the nintendo wii remote. IEEE Transactions on Pervasive Computing. 2008; 7(3):39–45.
6. Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011; 34(7):1409–22.
7. Ruiz J, Li Y, Lank E. User-defined motion gestures for mobile interaction. Proceedings of the 29th International Conference on Human factors in Computing Systems, CHI '11; 2011. p. 197–206.
8. Teather RJ, Pavlovych A, Stuerzlinger W, MacKenzie IS. Effects of tracking technology, latency, and spatial jitter on object movement. Proceedings of IEEE Symposium on 3D User Interfaces, 3DUI'09; 2009. p. 43–50.
9. Mariappan S, Rao GS, Ravindra Babu S. Enhancing GPS receiver tracking loop performance in multipath environment using an adaptive filter algorithm. Indian Journal of Science and Technology. 2014 Nov; 7(Suppl 7).