International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue (4) July, Technovision-2014, ISSN 2249–071X

Technovision-2014: 1st International Conference at SITS, Narhe, Pune on April 5-6, 2014

Gesture Recognition Engine using Golden Section Search Algorithm for Touch Tables

Mrs. Rohita P. Patil
Department of Electronics and Telecommunication
Smt. Kashibai Navale College of Engineering, Pune, India
Email: rohita_jagdale@yahoo.com

Shreyansh Daga
Department of Electronics and Telecommunication
Smt. Kashibai Navale College of Engineering, Pune, India

Manoranjan Singh
Department of Electronics and Telecommunication
Smt. Kashibai Navale College of Engineering, Pune, India
Email: manoranjankr.singh07@gmail.com

Nitin Lunawat
Department of Electronics and Telecommunication
Smt. Kashibai Navale College of Engineering, Pune, India

Abstract **– Today world is dominated by all kinds of electronic devices, ranging from giant mainframes to hand held Smart phones. Over the years one of the goals in this scenario was always to have a better user interface with the machine. As time went by effective means were devised to achieve this goal. Many such devices like the Keypad, Keyboard, Mouse, Joystick, Console etc. were deployed. As the technology moved ahead, one of the prime milestones was the use of a Touch Screen. The touch screen enables the interaction of user through touch. This touch could be a finger touch or a stylus touch. Touch technology has penetrated almost everywhere, as devices like Tablet computers, smart Phones etc have employed touch screen a part of User Interface. Means to interact using touch screens are being invented. Single touch gestures are being used currently in many applications. As the need grows, the limitations of single touch gestures inhibit the versatility of the Touch User Interface. Hence there is a strong need to explore the Multi Touch Gesture Recognition for a Touch User Interface. This paper constructs algorithm and a user interface with which single and multi touch gestures can be recognized efficiently.**

Keywords **– Multi-Touch, Uni Stroke, Multi Stroke, User Interface, Gesture, Golden Section Search, Euclidean Distance, Gesture Template.**

I. INTRODUCTION

Gesture recognition is a practice that has been around since 1964, when W. Tiedemann developed the first trainable gesture recognizer. Since then, gesture recognition has been used in a wide range of applications, such as commercial CAD applications since the early 1970's, and more recently, sign language interpretation, hand and facial gesture recognition, and multi-touch gesture recognition. With the recent rise and commercialization of gesture-recognition enabled mobile devices, the need for algorithms that minimize the computational load on the limited resources of mobile devices, whereas still remaining strong in recognition speed and accuracy has become of great importance [4].

As the research and development for touch systems grows, there will be a strong need to have an effective, flexible and a robust medium. This system presents a way, where user can interact with the device using Multi Touch Gestures. Multi Touch Gesture Recognition Engine primarily deals with simply recognition of multi touch gestures. As today's systems thrive on user experience, it is important to understand and provide the user with best possible way of interacting with the system. With touch devices, the user gets more degree of freedom when he uses gestures to interact with the device. So, this system is made adaptable to the user where the user experience is excelled. Multi-Touch surface computer technology turns ordinary surfaces into multi-touch interactive surfaces such as multi-touch tables, bars and counters [5]. Multi-touch table is in demand by clients for their interactive displays, exhibits and touch screen computing systems. Though in research stage many touch gesture recognition systems are coming which are targeted at various devices, majority in them, the Tabletop Devices and Touch Devices.

Figure 1 show the multi touch table used to design different gesture recognition technique.

A. Hardware Specification of Magic Table is as following

1. CPU: Core i5 2.67 GHz
2. HDD: 250 GB
3. Touch Screen: Overlay up till 40'' IR Sensor based, with 32 Point Touch Sense and USB Connectivity.

B. Software Specification of Magic Table is as following

1. Base Operating System: MAC OS X
2. Boot Camp Operating System: Windows 7 Ultimate
3. TUIO protocol embedded Multi Touch Screen Device Driver.
4. Magic Table Rich Application Suite

The contributions of this paper are to present an easy-to implement gesture recognition algorithm for use by UI prototypes that may have little or no knowledge of pattern recognition. This includes an efficient scheme for rotation invariance.

Fig.1. Touch Table

II. GESTURE RECOGNITION

Computationally, a gesture as a human action that begins with placing one or more fingers on the interactive surface, and ends when no fingers remain on the surface. Gestures are mostly used as a form of input, which is like a command or an event for a particular application. The main job of a gesture is to trigger the event every time it is performed or drawn. A gesture is a form of non-verbal communication in which bodily visible actions communicate particular messages, either in place of speech or together and in parallel with words [5]. Gestures include movement of the hands, face, or other parts of the body. A more formal definition of touch gestures, the drawing of symbols with a pointing device or finger(s) to convey a particular message of interaction to the machine or the device in use.

A. Classification of Gesture

Classification of gesture is based on two categories like Online and Offline Gestures, based on event. Which are further divided into single touch gestures and multi touch gestures based on number of strokes. In case of multi touch gestures, there is more than one number of strokes in the gesture which is drawn. Thus from a mathematical point of view there is more than one functions of strokes which define the particular gesture. The main goal here is to identify these individual functions or strokes and use them for the recognition, when an input is presented by the user.

1. Online Gestures are those gestures which give instantaneous response to the user as soon as to draw the gesture. They do not wait for the user to complete the gesture. As the name suggests online gestures means their action or event or response is instantaneous irrespective of completion of the gesture [3].
2. Offline gestures are those gestures which have to drawn completely by the user in order to invoke the associated event.

Offline gestures are processed once the gesture is drawn and the finger is taken from the touch screen. Offline gestures require completion to trigger or invoke the action or event or response of the gesture.

B. Gesture Recognition Techniques

At present there are many algorithms presented for recognition of gestures [1][2], however these are mainly targeted for single touch or uni-stroke gestures. However, methods to obtain a credible result for recognition of multi touch gestures change from those used for recognition of single touch gestures [8]. Fundamentally, the prime feature used for recognition of gestures in general is the Euclidean distance. It can be defined as,

$$Xe = \frac{1}{n}\sum_{1}^{n}\left|Ai - Bi\right| \qquad (1)$$

Where, Xe is the distance and Ai and Bi are two consecutive points in plane being considered. Generalize this function for individual strokes as well where in A = {A1, A2, A3, ….. , An} and B = {B1, B2, B3, …. , Bn} represent two different stroke combinations. Thus A can represent the gesture template and can represent the input gesture, and Xe can then form a basis of the hierarchical calculation for estimating the matching of the two strokes.

C. Multi Touch Recognition Engine

The system works closely with online recognition model implemented using [1][2], golden section search algorithm. The steps used to design this algorithm as follows.

1. Points received from the user are stored and gesture map is created.
2. Gesture map is normalized as mentioned above.
3. Feature set is extracted from the gesture map, with number of cursors, start vector angle, end vector angle and angle between the cancroids and first point.
4. Gesture templates are loaded and pre-processed, which includes the same normalization and feature set extraction methods.
5. For every gesture template which is loaded all possible combinations of gesture map are also computed and stored in a temporary list [8].
6. First criterion filter: NUMBER OF CURSORS.

a. The input gesture is compared with every combination of the gesture template for same number of cursors.
b. If the criterion is not met, the next gesture combination is selected from the temporary list and the step is repeated.
c. If criterion is met, the algorithm advances to the next criterion.

7. Second criterion filter: START AND END ANGLE VECTORS.

a. The input gesture is compared with every gesture template to see whether the differences of their start and end vector angle are within the specified threshold which is decided before.
b. If the criterion is not met, the next gesture combination is selected from the temporary list and the step is repeated.
c. If criterion is met, the algorithm advances to the next stage.

8. Golden Section Search

a. The Golden Section Search (GSS) takes input gesture points, gesture template point, minimum and maximum

bounding angle (a and b) and difference threshold as arguments[6][7].
b. Within the bounding interval [a, b] two points are computed x_1 and x_2 using the golden ratio such that

$$\frac{(x_2 - a)}{(x_1 - a)} = 1.618:1 \quad (2)$$

Where

$$x_1 = b - 0.618 * (b - a)$$

$$x_2 = a + 0.618 * (b - a)$$

c. Now the input gesture is rotated around its centred by these two values x_1, x_2 and the mean path distance is calculated against the gesture template. This gives us the two values of path distances fx_1 and fx_2 [8].
d. If fx1 < fx2 the interval is re-divided as,
b = x2, x2 = x1 and x1 = b - 0.618 * (b – a) (3)
e. If fx1>fx2 the interval is re-divided as,
a = x1, x1 = x2 and x2 = a + 0.618 * (b – a) (4)
f. And step 3 is repeated until the absolute difference between fx_1 and fx_2 is less than the given threshold.
g. Once step 6 is achieved, then the minimum mean path distance between input gesture and template gesture obtained at respective angle x_1 or x_2 along with the number of iterations wherein the interval was re-divided in order to achieve exact angle x_1 or x_2.
h. In order to be a valid match between input gesture and template gesture this mean path distance must be as low as possible.
9. Calibrating the score in terms of Probability
a. After computing the minimum mean path distance between the input gesture and template gesture, the score is calibrated in terms of percentage as follows:
Score = [1.00 – (m) / (HL)] * 100 (5)
Where, m = Minimum path distance computed using GSS and HL = Half diagonal length of standard bounding box
b. We can infer from these equations that a 100% match will have minimum (0.00) path distance and a 0% match will have maximum path distance (half diagonal length).
c. If the score in terms of probability is greater than 60% then the gesture is recognized correctly, otherwise it is not a valid gesture.
10. Event Triggering: After correct recognition of gesture, its respective event is triggered with help of respective event handlers.

D. Other methods and approaches

Designing a gesture recognition system consist of pattern representation, feature extraction and selection, classifier design and in some cases learning, training and testing, and performance evaluation [6]. The best known approaches to pattern recognition are template matching, statistical classification, syntactic or structural matching.

1. Template Matching

Template matching involves finding small subsections of an image, being the image of the input gesture, which matches a template image.

2. Statistical Classification

Statistical classification involves each pattern being represented as features in an n-dimensional space, based on quantitative information on one or more characteristics, and assumes that patterns are generated by a probabilistic system.

3. Structural Matching

In the synaptic or structural matching approach, patterns are seen as being made up of simpler sub-patterns, which are themselves built up of even simpler sub-patterns. These approaches are implemented via a number of methods [3], like Bayesian Networks, Hidden Markov Models, Fuzzy Logic, Artificial Neural Networks, and Dynamic Time Warping.

III. GESTURE RECOGNITION SYSTEM

The block diagram of the multi touch gesture recognition system is shown in Fig. 2 (a). It consists of the blocks as, Gesture Recognition Engine, Gesture Database, Gesture Event Handlers and Event Handlers. The interaction of user with the Magic table may result in a touch gesture.

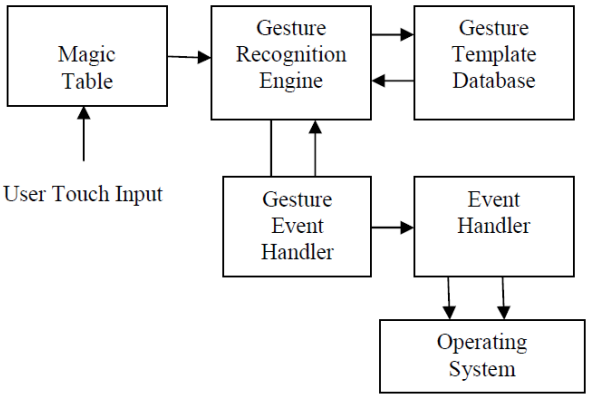


Fig.2. (a) Gesture Recognition System

Whenever the user provides a touch input at the touch screen, the device driver senses this touch input and sends the touch points at which the touch was sensed to the Gesture Recognition Engine, which is the first block.

A. Gesture Recognition Engine

This block performs the recognition of the gesture, if it is drawn by the user on the Magic Table. This block is responsible for correctly receiving the touch points, decoding them, identifying and representing them in form of a raw gesture and conditioning this raw gesture information in order for its correct recognition. This block contains the algorithm for the recognition of the gesture. This block uses Gesture Templates [1] from the Gesture Template Database for recognition of the gesture. It finally gives the result in form of a score on the scale of 0 to 1, which will decide whether the input gesture has been recognized correctly. The design of the Multi Touch Gesture Recognition Engine involves following modules

like TUIO tracker module, TUIO client module, Gesture normalization module, Gesture recognition module and Event triggering module as shown in Fig.2 (b).

B. Gesture Database

This block is a database which has the collection of all the gesture templates, which are required for recognition. The gesture templates are recorded and stored in form of XML files.

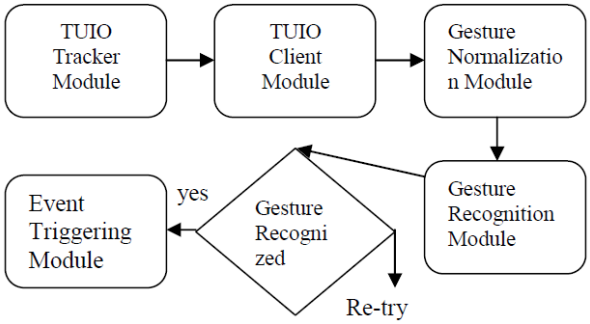


Fig.2. (b) Recognition Engine Design Flow

These template files contain the name of the gesture, the number of fingers used to draw the gesture and the touch points traced by each finger along with the timestamp of each touch point and the total gesture duration in milliseconds.

C. Gesture Event Handlers

Gesture Event Handlers are a group of generic call back methods which are invoked or called whenever an event pertaining to any gesture activity is fired. The event may be raised whenever a gesture is started, a gesture is changed, a gesture is recognized or a gesture is complete. This simplifies the approach with which the gesture events and the appropriate responses are needed to be handled [2].

D. Event Handlers

This block contains the collection of event handlers which are called once the gesture is correctly recognized and which are dedicated to respective events the user wants to perform with gestures. These handlers will then perform required system calls, as the operating system is responsible for the event action required whenever the respective gesture is recognized correctly. These event handlers may require event specific arguments, as they are meant to handle exact events which the user desires through the gesture.

IV. Result and Discussion

Table 1 shows the list of Name and type of gestures implemented on magic table. Browser, Notepad, E-mail, Run, Task Manager and close gestures are performed for single touch and Desktop, Minimize, Winlock, Switch, Swipe left and Swipe right are performed for multi touch. Initially GUI is also developed to visualize loading, recording, representation and recognition of different characters and figures.

A. Single Touch Recognition

In single touch recognition six gestures are implemented. Browse gesture is explained below in Fig. 3 and remaining five are shown in Fig.7.

Table 1: Tasks Performed with Single and Multi Touch Gesture

Gesture name	Number of touch points	Type of Gestures
Browser	1	Single touch
Notepad	1	
E-mail	1	
Run	1	
Task Manager	1	
Close	1	
Desktop	5	Multi touch
Minimize	4	
Winlock	3	
Switch	2	
Swipe left	2	
Swipe right	2	

1. Browser – To open Google

Character ‘G’ is used to open Google. It is as shown in Fig. 3 (a), the visual representation of the Browser gesture template as recorded by the user. This representation is realized from the touch points which are stored in the template file. In this representation the gesture template is loaded first and positioned with centroid at the center.

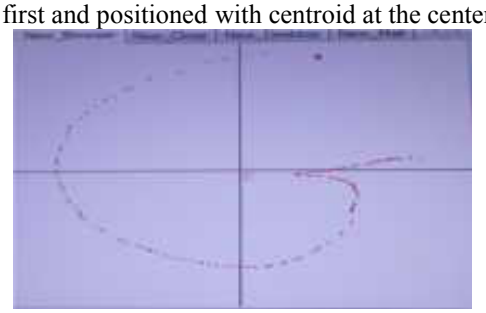


Fig.3. (a) Recorded gesture template for browser

For each cursor there is an independent list, as a result we have separate independent lists with corresponding gesture points. Once when sufficient numbers of cursor points are obtained/the gesture tracing is complete, these points are then integrated to form one gesture object. Now this gesture object is normalized by this module. The normalization is done to fit the gesture object in a standard size and thus these normalized gestures are independent from the position, orientation and size. Thus no matter how small, how large or wherever the gesture is drawn, normalization will take care of it.

Figure 3 (b) shows the gesture drawn by the user for Browser on the Magic Table. It contains the collection of event handlers which are called once the gesture is correctly recognized and which are dedicated to respective events the user wants to perform with gestures.

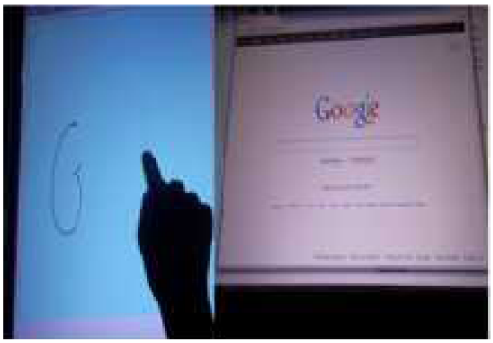


Fig.3. (b) User Gesture input G and Browser event

These handlers will then perform required system calls, as the operating system is responsible for the event action required whenever the respective gesture is recognized correctly. These event handlers may require event specific arguments, as they are meant to handle exact events which the user desires through the gesture. When this gesture matches with the stored template for browser then its respective event is triggered and it open Google Browser as shown in Fig 3 (b). Comparison of input gesture and the gesture template shown in Fig. 4.

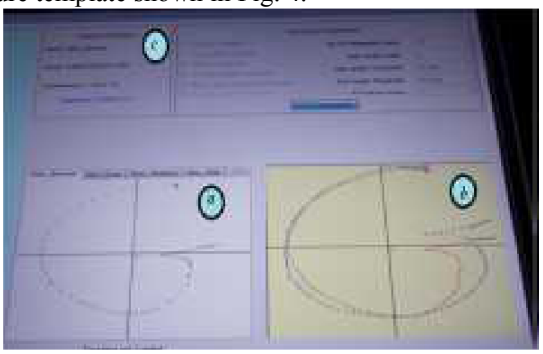


Fig.4. Browser Recognition: (a) Loaded Gesture template.
(b) Comparison of loaded template and input gesture.
(c) Result: Name, Score, No of comparison.

B. Multi Touch Recognition

Multi Touch Recognition is implemented for five gestures as listed in Table 1. Task to open Desktop is shown in Fig. 5 and remaing four are shown in Fig. 8.

1. Task to open Desktop

Figure 5 (a) shows the task performed to open Desktop with five fingers. Visual representation of the Desktop gesture template is recorded by the user. It consists of five touch points with fingers.

This representation is realized from the touch points which are stored in the template file. In this representation the gesture template is loaded first and positioned with centric at the centre.

Figure 5 (b) shows the gesture drawn by the user for Desktop using five fingers on the Magic Table. When this gesture matcheds with the stored template for desktop then its respective event is triggered and desktop is open.

Fig.5. (a) Recorded gesture template for desktop.

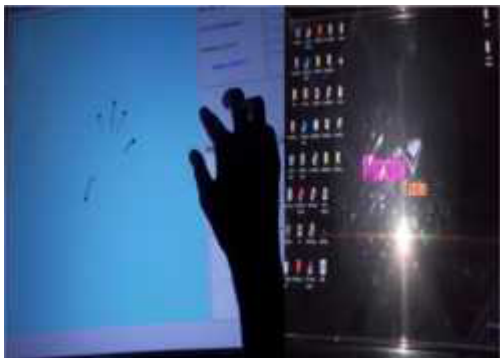


Fig.5. (b) User Gesture Input & Desktop open triggered event

Figure 6 shows the comparison of input gesture from user and the gesture template stored in database.

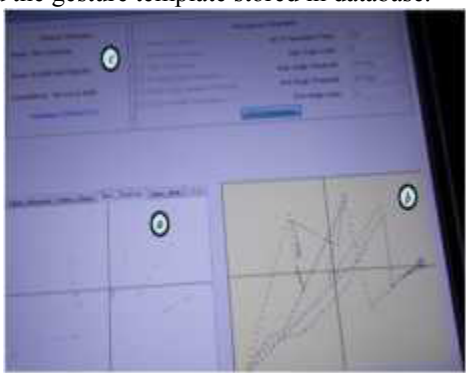
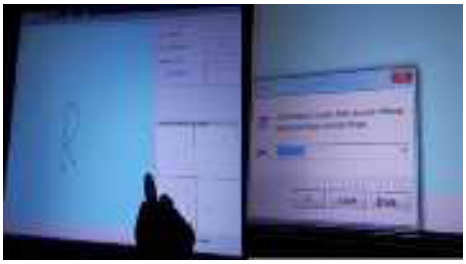


Fig.6. To open Desktop (a) Loaded Gesture template.
(b) Comparison of loaded template and input gesture.
(c) Result: Name, Score, No. of comparison.

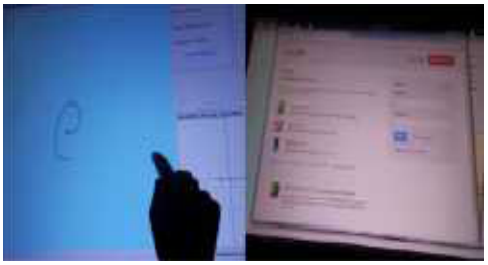
(a) Character “O” for Task Manager



(b) Character "R" to open Run

(c) Character "N" to open Notepad

(d) Symbol cross to close all windows



(e) Character "e" to open mail

Fig. 7 Single touch recognition

(a) Lock Windows

(b) Minimise operation

(c) Switch to other application

Fig. 8 Multi touch recognition

V. CONCLUSION

Touch Gesture Recognition offers many advantages to any touch based system to make the interface to user very easy and flexible, applications can be launched with help of single and multi gesture, easy to use and learn as it involves interacting with finger touch from the use, make optimum use of the hardware capability of the Multi Point Touch screen. The system is to be designed in the fore mentioned way and would also incorporate any future changes in its design aspect.

Golden Section Search algorithms are used for the Gesture Recognition module on their space complexity and timing complexity. The entire system is tested first on the Magic Table with demonstration of Touch Gestures with respective applications.

REFERENCES

[1] P.O.Kristensson & L.C. Denby, "Continuous Recognition and Visualization of Pen Strokes and Touch Screen Gestures" EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modelling 2011.

[2] Anderson, D., Bailey, C. and Skubic, M. (2004) Hidden Markov Model symbol recognition for sketch-based interfaces. *AAAI Fall Symposium.* Menlo Park, CA: AAAI Press, 15-21.

[3] Cao, X. and Balakrishnan, R. (2005) Evaluation of an on-line adaptive gesture interface with command prediction. *Proc. Graphics Interface '05.* Waterloo, Ontario: CHCCS, 187-194.3

[4] Jacob.O.Wobbrock, *Andrew D. Wilso, and Yang Li* "A $1 Recognizer for User Interface Prototypes".

[5] Daniel Wood, "Methods for Multi Touch Gesture Recognition", *UIST'07*, October 7-10, 2007, Newport, Rhode Island, USA.

[6] Floyd Hanson. MCS 471 Class Optimization Notes: Method of Golden Section Search. Available at http://www.math.uic.edu/» hanson/mcs471/classnotes.html.

[7] Ward Cheney and David Kincaid. Numerical Mathematics and Computing. Third Edition, Brooks/Cole Publishing Company, 1994.

[8] Curtis F. Gerald and Patrick O. Wheatley. Applied Numerical Analysis. Seventh Edition, Addison- Wesley, 2004.