GESTURE RECOGNITION BASED INTELLIGENT

ALGORITHMS FOR VIRTUAL KEYBOARD DEVELOPMENT

Hafiz Adnan Habib
04-PhD-EE-18

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy.

Thesis Supervisor
Prof. Dr. Muid Mufti

Department of Electrical Engineering
University of Engineering & Technology Taxila

June 2007
GESTURE RECOGNITION BASED INTELLIGENT

ALGORITHMS FOR VIRTUAL KEYBOARD DEVELOPMENT

Approved By:

Prof. Dr. Muid Munir, Research Supervisor

Dr. Naveed A. Malik, External Examiner

Brig. Dr. Younis Javed, External Examiner

Date of Approval: 19 Jun 2007
ACKNOWLEDGEMENT

Thanks to Almighty Allah whose blessings have been showering me all through my life and brought peace and happiness. I am highly grateful to my supervisor Prof. Dr. Muid Mufti for his continuous support and excellent guidance in conducting a result oriented and well directed research. Dr. Muid Mufti made me capable of visualizing and proposing the solution for the complex problem of virtual keyboard design. I am also grateful to Prof. Dr. M. Saleem Mian for his continuous moral and technical support in not only solving this research work but in also many other scientific projects. I would like to thank Prof. Dr. Habibullah, Vice Chancellor UET Taxila, for the development of research supportive environment at UET and let the people conduct research in various disciplines of engineering which otherwise had not been possible in the past. I would like to thank Prof. Khalil Ahmad Khan, Director ASR&TD, for his support and guidance to meet the official requirements for pursuing the PhD degree.
To my family
Virtual keyboard design is an important research & development area in human computer interaction. A keyboard in portable devices is generally inadequate for text entry because of small size and attachment of ordinary keyboard adds to the size. Need for virtual keyboard also exist in human machine interfaces to a wide range of applications in markets as consumer electronics, medical/healthcare, industrial automation, automotive, and public information kiosks.

Current virtual keyboards designs appear in various forms such as finger-joint wearable sensor gloves, thumb code, accelerometer based inputs, laser projected keyboards and gyroscope based sensing. Each virtual keyboard has certain design characteristics. However, performance parameters for the evaluation of keyboards are same such as number of discrete keys, response time and failure rate. Other parameters included can be the ability to remap, key symbol mapping and space requirements. The cost of most of the successful virtual keyboard designs is high due to custom design and expensive technology.

This research provides a new perspective to view the problem of virtual keyboard design by incorporating gesture recognition by using a simple mono vision camera. Human hand and finger movements are considered as gesture movements making keystrokes on any
surface. Human hand and finger movements are recorded in video sequence and gesture recognition algorithms estimate the key pressed. This algorithm uses aggregation of gesture information using fuzzy logic and intelligent learning based approach. Feasibility of mono-vision gesture based virtual keyboard is demonstrated. The proposed system shows lesser requirements in hardware due to software centric design and off the shelf components.
TABLE OF CONTENTS

ABSTRACT ....................................................................................................................... i

Table of Contents .......................................................................................................... vii

Chapter 1 ......................................................................................................................... 1

Introduction .................................................................................................................... 1

1.1 Problem statement ................................................................................................... 1

1.2 Need for Virtual Keyboard ...................................................................................... 1

1.3 Conventional keyboard .............................................................................................. 2

1.3.1 The QWERTY keyboard ....................................................................................... 2

1.4 Alternative Approaches for Text Entry ...................................................................... 3

1.4.1 Chord keyboard .................................................................................................. 3

1.4.2 Phone pad and T9 entry .................................................................................... 4

1.4.3 Handwriting Recognition ...................................................................................... 4

1.4.4 Speech Recognition .............................................................................................. 5

1.4.5 Sign language ....................................................................................................... 5

1.4.6 Virtual Keyboard ................................................................................................ 6

1.5 Characteristics and Metrics ....................................................................................... 7

1.5.1 Methodology for Key-press Detection .................................................................. 7

1.5.2 Number of Discrete Keys .................................................................................... 8

1.5.3 Degree Of Freedom ............................................................................................. 8

1.5.4 Temporal significance interval for key press ....................................................... 8
1.5.5 Failure Rate ................................................................. 8
1.6 Virtual Keyboard Designs .................................................. 9
  1.6.1 Visual panel .............................................................. 9
  1.6.2 Thumbeode ............................................................. 10
  1.6.3 Chording glove ......................................................... 11
  1.6.4 FingeRing .............................................................. 11
  1.6.5 Scurry ................................................................. 12
  1.6.6 SenseBoard ........................................................... 13
1.7 Vision based virtual keyboard ............................................. 13
  1.7.1 VType ................................................................. 13
  1.7.2 VKB Projection ....................................................... 14
1.8 Social and Economic Issues with Virtual Keyboards ................. 15
1.9 Gesture recognition based virtual keyboard ............................ 15
1.10 Summary ........................................................................ 16
1.11 Reference: ...................................................................... 17
Chapter 2 .............................................................................. 19
Gesture Recognition .................................................................. 19
  2.1 Introduction to Gesture Recognition ................................. 19
  2.2 Types of Gestures ......................................................... 21
  2.3 Application Domains of Gestures ..................................... 22
  2.4 Architecture of Gesture Recognition System .................... 23
  2.4.1 Gesture acquisition and Pre-processing ....................... 24
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.2</td>
<td>Gesture Feature Extraction and Representation</td>
<td>24</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Gesture Recognition &amp; Classification</td>
<td>28</td>
</tr>
<tr>
<td>2.5</td>
<td>Hand gesture recognition</td>
<td>39</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Hand detection</td>
<td>39</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Appearance Based Approaches</td>
<td>39</td>
</tr>
<tr>
<td>2.6</td>
<td>Summary</td>
<td>40</td>
</tr>
<tr>
<td>2.7</td>
<td>Reference:</td>
<td>40</td>
</tr>
<tr>
<td>Chapter 3</td>
<td></td>
<td>44</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>44</td>
</tr>
<tr>
<td>3.2</td>
<td>Intersection, Union and Compliment</td>
<td>46</td>
</tr>
<tr>
<td>3.3</td>
<td>T-norm and T-conorm:</td>
<td>46</td>
</tr>
<tr>
<td>3.4</td>
<td>Fuzzy Relation</td>
<td>47</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Fuzzy implication</td>
<td>47</td>
</tr>
<tr>
<td>3.5</td>
<td>Fuzzy Composition</td>
<td>48</td>
</tr>
<tr>
<td>3.6</td>
<td>Fuzzy Inference Engine</td>
<td>48</td>
</tr>
<tr>
<td>3.7</td>
<td>Fuzzy logic as a Universal Approximators</td>
<td>51</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Universal Approximation Theorem</td>
<td>51</td>
</tr>
<tr>
<td>3.8</td>
<td>Fuzzy Measures</td>
<td>51</td>
</tr>
<tr>
<td>3.9</td>
<td>Fuzzy Inferencing</td>
<td>51</td>
</tr>
<tr>
<td>3.10</td>
<td>Performance Metrics</td>
<td>52</td>
</tr>
<tr>
<td>3.11</td>
<td>Reference:</td>
<td>53</td>
</tr>
</tbody>
</table>
Chapter 4

Gesture Recognition Based Virtual Keyboard System

4.1 Architecture

4.2 Algorithmic Design of Mono Vision Gesture Based Virtual Keyboard

4.2.1 Fuzzy Representation of Trajectories

4.2.2 Fuzzy Representation of Dominant Finger

4.2.3 Knowledge Fusion / Fuzzy Rule Base

4.2.4 Degree of Confidence of Key Stroke

4.3 Summary

4.4 Reference:

Chapter 5

Experimental Setup and Results

5.1 Development Environment & Results

5.2 Prototype Setup for Virtual Keyboard

5.3 Summary:

5.4 Reference:

Chapter 6

Conclusion & Future Work

6.1 Conclusion

6.2 Future work
CHAPTER 1

INTRODUCTION

1.1 Problem statement

To design a vision based virtual keyboard which detects key strokes by recognition of hand gestures instead of mechanical transducer operations of key pressing. Mono-vision video of hand gesture for pressing the keys is analyzed by motion matching algorithm to estimate the pressed key.

1.2 Need for Virtual Keyboard

Historically, keyboards are designed for text character printing and text character entry later on in the attached devices. Therefore, keyboard is communication mechanism between human beings and the computing machines and is used to enter textual data and commands. Virtual keyboard are generally assumed to produce the same output as the conventional QWERTY layout produces. The utilization of virtual keyboard appears in space saving situations or requirement in soft programmability of keys or systems avoiding mechanical failure or in movement situations where usability of standard keyboard is limited. Utilization of virtual keyboards in space saving situations is enormous e.g. text entry in PDA's and cellular phones etc. PDA's and cellular phones don't have standard keyboards available with them due to limited keys on their text entry keypads. Soft key programmability of keyboards is also a considerable factor in the
design of virtual keyboards. Similarly virtual keyboards have been designed without mechanical keys. These kinds of keyboards are utilized in hostile environments. Virtual keyboards find their position in transport environments e.g. rail, plane or automotive. Virtual keyboards are also designed for public kiosks and here those designs are suitable which avoid mechanical failure. Industrial environments, sterile and medical environments also require the special requirements from keyboard designs which satisfy the requirements of utilization of a virtual keyboard in such environments.

1.3 Conventional keyboard

Keyboard has long been known since eighteenth century development of fully mechanical board in which actuated levers and arms mechanisms were used to print the corresponding character on some printing material. The new advancements had been possible in 1920's when electromechanical technology was involved in the keyboard character printing functionality. A switch closing operation following the key pressed actuated mechanical mechanism to print the particular character. Later on, computers drove the development of current keyboards in which a switch is closed whenever a key is pressed and a unique electrical signal is generated for the corresponding pressed key.

1.3.1 The QWERTY keyboard

The most common keyboard format is QWERTY layout, named upon the first six letters of the top row of alphabetical keys. The layout for letters and digits are fixed over a QWERTY layout, as shown in Figure 1-1, and non-alphanumeric keys vary on the keyboard. This difference is evident on the British and American keyboards. The most
obvious means of text entry is the plain keyboard, but there are several variations on this
 e.g. chord keyboards that use combination of fingers to enter letters and phone keypads.
Handwriting and speech recognition offer more radical alternatives.

![Figure 1-1: QWERTY keyboard layout](image)

### 1.4 Alternative Approaches for Text Entry

In spite of keyboard, there are many possible alternatives for text entry in the special
circumstances such as cellular phones, PDA's etc. These all situations incorporate
different layouts of keyboard and with variation in key entry mechanisms. Sometimes
there is entirely different methodology which is not based on key to character/alphabet
generation. Following are few examples of such text entry methodologies:

#### 1.4.1 Chord keyboard

These kinds of keyboards are significantly different from standard alphanumeric
keyboards such as QWERTY layout. Chord keyboards usually consist of few keys and
letter emission is controlled by pressing one or more keys at the same time. Such
keyboards offer number of advantages as their size is compact than a traditional keyboard and can be operated with one hand [3].

1.4.2 Phone pad and T9 entry

The phone keypads consists only 0-9 digits and does not have full alphanumeric keyboard. Therefore most phones have at least two modes for the numeric buttons: one where the keys mean digits and the other where the keys mean letters. However typing with such keypads is very laborious and efficient usage requires a lot of practice. A T9 algorithm is incorporated to facilitate the letter entry where a large dictionary is used to disambiguate words by simply writing the relevant letters once [14].

1.4.3 Handwriting Recognition

Hand writing is an attractive method of text entry and can be an intuitive method for interaction with computer. Special scanners specifically designed for hand written text entry with added feature of hand writing recognition have been practically available for number of years. These scanners accompany hand writing recognition software to communicate the host application while importing the text in the appropriate form. However the level of accuracy for hand writing character recognition are still fairly inaccurate when compared with character recognition accuracy for printed characters. The variance in individual human writing is enormous and is the major challenge in the successful development of hand writing recognition [13].
1.4.4 Speech Recognition

Speech recognition is a promising way to communicate with the computer for text entry. However, speech recognition systems have been practically deployed in limited circumstances. Speech recognition has found its positions in many markets such as telephone information system, access for disabled, in hand occupied situation (e.g. military) [3]. Successful recognition rates over 97% have been reported which actually represents one letter error in every thirty which dictates one spelling mistake every six or so words. Such performances are reported for restricted vocabulary. There are many challenges faced by the speech recognition systems such as extremely large vocabulary spoken in different accents with inherent vagueness and imprecision [15], [16].

1.4.5 Sign language

Body language and facial expression are integral parts for the communication in a sign language system and can exhibit more symbols than alphabets in English language. Ultimately higher bandwidth is achieved with fewer atomic elements. Therefore sign language appears to be a natural way of communication with computing devices. However many real time issues have been discovered while practicing the sign language: computer recognition is a challenge, skill penetration in the population is lesser and sign language signals are highly context dependent [17].
1.4.6 Virtual Keyboard

Keyboards are considered to be the de-facto standard for text entry in the computing machines. Keyboards are meant for character emission for the host device. Character emission process is governed by transducer operation which senses the keystroke generated by human fingers. Virtual keyboard is termed from the fact that though output is same but the key generation process may be different or keys arrangement is different or number of keys may be different. It is assumed that virtual keyboard is utilized by the people who can’t use the usual keyboards due to physical limitations. A virtual keyboard may be a feature of computer program or a program in itself that acts as virtual extension of a controller e.g. in case of desktop PCs, the Windows XP on screen program acts as virtual keyboard which can be operated by mouse and stylus is used in personal digital assistants for data entry. However it is also assumed that a virtual keyboard is used by the people who can’t use the usual keyboard due to physical limitations. Analyzing the situation and looking for reasons for the development of virtual keyboard: traditionally following characteristics are assumed for a keyboard.

- Certain layout of keyboard.
- Certain size of keyboard.
- Standard character emitting functionality by pressing the keys on the keyboard.
- Standard defined keys.

The current layout, which is named QWERT layout, of keyboard had long been established since 1874 and it has survived for enough periods though it has come to an
age. There had been many positive aspects of this design. Moreover, it is hard to visualize any other layout for keyboard and appeared to be the de-facto standard for text entry in the computing devices.

Certain short coming of the standard QWERT keyboards is reported in the literature. The layout is assumed to be non-optimal in a number of ways for text entry in portable devices by stylus [18]. Various keyboard layouts optimize the arrangement for frequent alternating use of right and left hands, short travel distances of fingers, decreasing load from index finger to pinky to compensate the decreasing strength of these fingers. Keyboard layout modifications such as split keyboard had been shown to have very consistent impact. Split keyboard was first proposed by Klockenberg [20] in 1926 and it is claimed that split keyboards reduce the physical stress on hands and fingers for all users, novice and experienced [19].

1.5 Characteristics and Metrics

A keyboard is essentially a user interface and it should be analyzed from the user interface design perspectives. A number of characteristics can be defined [5] for comparisons of virtual keyboards.

1.5.1 Methodology for Key-press Detection

Key-press detection can be performed in number of ways e.g. by the touch of a surface or interruption of light beam etc. The detection process actually defines the robustness.
1.5.2 Number of Discrete Keys

This parameter defines how many keys do the device or method have. Also, can the number of keys can be extended i.e. the design of keyboard is flexible so that new keys can be added to the design as in the case of Chinese keyboards where extended keys are required.

1.5.3 Degree Of Freedom

DOF (degree of freedom) is defined by key to symbol mapping. The design has one DOF if each key correspond to exactly one character/symbol. There can be more DOF as one to many characters mapping, disambiguated by either temporal methods i.e. multiple successive keystrokes, or by statistical prediction or by chording methods i.e. multiple keys pressed simultaneously produce one character.

1.5.4 Temporal significance interval for key press

Temporal significance interval is defined by the time required to register a particular key i.e. specific length of duration till the key pressed is actually registered. On a normal keyboard, key press is registered as long as the switch contact is closed. Therefore, temporal interval for registration of key pressed event becomes an important characteristic and measure for the effectiveness of any virtual keyboard design.

1.5.5 Failure Rate

Failure of keyboard occurs from mechanical failure of switches, chemical deteriorations of surfaces and contact and dust deposits. The failures in soft / virtual keyboard occur
from misclassification, deficient learning and software bugs. The aim is to develop a virtual keyboard which has low failure rate along with the above mentioned characteristics.

1.6 Virtual Keyboard Designs

Virtual keyboards appear in variety of forms with many sensing mechanisms. Since this research work relates to the involvement of novel technology for the design of virtual keyboard, therefore only those keyboard designs are presented here which is novel in terms of their design or novel in the way the keys are registered. Therefore designs with various layouts of keyboards but with standard key registering methodology have not been included.

1.6.1 Visual panel

The Visual Panel [4] consists of a camera and a sheet of paper as shown in Figure 1-2. The location of the extended index finger with reference to the paper is located by computer vision methodologies. The primary application is a mouse pointer, clicking is

Figure 1-2: Visual Panel
achieved by resting the fingertip in its current position for three seconds. The authors demonstrated text entry by interpreting pointer locations as the keys of a keyboard, which were printed on the sheet of paper. An audible notification signals the recognition of a character after the 3 second significant interval. The slow response of this system makes it unfit for consumer use.

1.6.2 Thumbcode

The “Thumbcode” method described in [6] defines the touch of the thumb onto the other fingers’ phalanges of the same hand as key strokes, as shown in Figure 1-3. Consequently

![Figure 1-3: ThumbCode](image)

there are 12 discrete keys (three for each index, middle, ring finger and pinky). To produce up to 96 different symbols, the role between keys and operators is broken up: The four fingers can touch each other in eight different ways, each basically representing a mode, or modifier key that affects the mapping for the thumb touch. Tactile user feedback is implicit when touching another finger with the thumb. This system has limited discrete keys and the wearable sensor is subject to mechanical failure.
1.6.3 Chording glove

The Chording Glove [7] employs pressure sensors for each finger of the right hand in a glove to implement a chording input device. Almost all possible finger combinations are mapped to symbols, making it potentially hard to type them. Additional “mode switches”, located along the index finger, are used to produce more than the 25 distinct characters. Yet user experiments suggest otherwise: rates of up to 19 wpm are achieved after ten training sessions “with no signs of leveling off”.

1.6.4 FingeRing

FingeRing [8] uses accelerometers on each finger to detect surface impacts, shown in Figure 1-4. In the wireless version depicted in the figure below these rings communicate with a wrist-mounted data processing unit. The interaction method is designed for one handed use, but could be extended to two hands with obvious implications. In the current version, the finger movements to produce one character are extensive: two chording patterns have to be typed within a time interval, each consisting of a combination of fingers hitting the surface. Due to this piano-style typing method, users with prior piano experience fare much better with this device; in fact, the full 2-stroke chord mapping is rendered too difficult for novice users.
1.6.5 Scurry

Tiny gyroscopes on each finger are the sensing technology in Samsung’s Scurry [9], as shown in Figure 1-5. The prototype suggests that these finger rings communicate with a wrist-mounted unit where the data is processed. Not much is known about this device, yet our assumption is that finger accelerations and relative positions are detected, making it possible to distinguish multiple key targets per finger. We further guess that a surface
impact is required to register a key stroke, also making for the primary sensory feedback to the user. Little LEDs on the rings potentially provide additional feedback [5].

1.6.6 SenseBoard

The Senseboard [10] consists of two rubber pads that slip onto the user’s hands. Muscle movements in the palm are sensed (with unspecified, non-invasive means) and translated into key strokes with pattern recognition methods. The only feedback other than characters appearing on a screen comes from the tactile sensation of hitting the typing surface with the finger [5].

1.7 Vision based virtual keyboard

The previous section encompassed an overview of the novel keyboard designs. These designs were novel in the key registration of key stroke made by human fingers. This section is aimed at the in-depth description of only those novel keyboard designs which incorporate image, video, vision or optical technology in some manner. The selection of these technologies is based on the direct relevance with the proposed research of development of gesture recognition based virtual keyboard. The development of gesture recognition based virtual keyboard also incorporates the vision technology.

1.7.1 VType

Virtual Devices Inc. [11] has developed a computer virtual keyboard. The virtual keyboard operates by the precise positioning of light sources and detection devices, as shown in Figure 1-6. The keyboard is projected over tabletop or some other surface. This
acts as visual aid for the user to localize the individual keys. The device senses the pressed key by simply touching the light area displaying the particular character. Technology appears to be registering the position of the displayed keys and some mechanism is involved e.g. IR range sensing etc. to detect the position where projection light is cut by the finger for key stroke. This system is costly due to the use of special IR sensors and is prone to detection errors.

1.7.2 VKB Projection

The virtual keyboard technology developed by VKB [12] is a tabletop unit that projects a laser image of a keyboard on any flat surface. Infrared cameras detect key strokes of all ten fingers. Word disambiguation techniques are employed despite this 1 DOF mapping. Therefore, our assumption is that engagement of all distinct key locations is detected, yet with a fairly low accuracy. These two characteristics in combination should result in fairly good recognition rates. Surface impacts of the fingers serve as a typing feedback [5]. VKB has fairly low accuracy and can operate in room environments.
1.8 Social and Economic Issues with Virtual Keyboards

QWERTY was started in the eighteenth century and has been acting as de-facto standard in spite of its non-optimality while analyzing the sustainability and adaptability of keyboards other than QWERTY layout, extreme difficulty is evident due to large social and economic factors. Generations of public have experienced the QWERTY keyboard layout and immense number of moderate and experienced users exists. Economically, a bulk of investment had been made which was distributed over multiple sectors from the manufacturer to the end user. All this investment has to be phased out to adapt the alternative. Therefore this alternative deployment requires massive resources and this apparently is quite impossible [3].

1.9 Gesture recognition based virtual keyboard

This thesis presents a novel gesture recognition based virtual keyboard system which replicates the transducer based keyboard system. Such gesture recognition based virtual keyboard will have communication capability with portable text entry systems such as mobile phones, PDA’s etc. Gesture recognition is based upon the tracking of human movement and interpretation of that movement as semantically meaningful commands. Gesture recognition has the potential to be a natural and powerful tool for intuitive interaction between the human and computer [1]. Gesture recognition has been successfully applied in virtual reality, human computer interaction, game control, robot interaction, remote controlling of home and office appliances, sign language, activity
recognition, human behavior analysis and training systems etc. Gesture recognition system is designed in four stages: gesture acquisition, feature extraction, classification, and learning. Gesture acquisition is accomplished by position sensors, motion / rate sensors, and digital imaging. Feature extraction and classification are real time stages to analyze the acquired gesture while learning stage is off-line activity to learn the relationship between gesture and information or command. The gesture recognition based virtual keyboard system follows the standard QWERTY layout while it benefits from its diminished size due to its software centric mechanism rather a hardware centric mechanism. The selection of QWERTY layout is made due to the general and wide acceptance of this standard in the mass. Therefore training requirements for adaptation of such keyboard is minimal.

1.10 Summary

This chapter presented an overview for the text entry methods in standard and portable environments such as PDA’s and cellular phones. Alternative text entry approaches were presented such as chord keyboards, phone pads and T9 entry, speech recognition, handwriting recognition and sign language. Finally concept of virtual keyboard was described. Justification for virtual keyboard was highlighted and its importance and usefulness in text entry for PDA’s, cellular phones, hostile environments, medical and industrial environments and transportation such as rail, plane and automotive has been presented. Then the merits and characteristics for keyboard functionality have been described in the theoretical perspective of human computer interaction paradigm. Different useful
terminologies for keyboard performance have been defined such as DOF, number of discrete keys, sensing methodologies etc. Then some interesting virtual keyboard designs were briefly described. These designs were presented in two categories: vision based virtual keyboards and non-vision virtual keyboards. Finally the social and economical issues for the wide deployment of virtual keyboard had been described.

1.11 Reference:


   http://boole.stanford.edu/thumbcode


[7] Samsung Scurry; PCWorld article at


CHAPTER 2

GESTURE RECOGNITION

2.1 Introduction to Gesture Recognition

Gesture may be defined as the physical movement of hands, arm, face or body with the intent to convey information or command. Gesture recognition consists of tracking human movement and interpretation of that movement as semantically meaningful commands [1]. Gesture recognition has the potential to be a natural and powerful tool for intuitive interaction between human and computer [14]. Gesture recognition has emerged as one of the most important research areas in the field of human-computer interaction. Gestures have long been considered a promising approach to enabling a natural and intuitive method for human-computer interactions for myriad computing domains, tasks, and applications [4]. The first gestures that were applied to computer interactions date back to the PhD work of Ivan Sutherland [3], who demonstrated Sketchpad, an early form of stroke-based gestures using a light pen to manipulate graphical objects on a tablet display. This form of gesturing has since received widespread acceptance in the human-computer interaction (HCI) community, inspiring the stroke-based gesture interactions commonly used for text input on personal digital assistants (PDAs), mobile computing, and pen-based devices [5], [6]. Since then, the notion of using gestures to facilitate a more expressive and intuitive style of computer interactions has gained popularity among researchers seeking to implement novel interactions with computers. Gloves augmented
with electronic motion and position sensors were developed to enhance interactions with virtual reality applications, enabling users to manipulate digital objects using natural hand motions [7], [8], [9] and polhemus sensors that provide six DOF (3-position and 3 orientation) tracked arm movements for controlling large screen displays from a distance, presented by Bolt [10] in the “Put That There” system. By the mid 1980s, computer vision technology was gaining popularity within the computing sciences, however it was not until the early 1990s that Freeman & Weissman [11] first demonstrated a vision-based system that enabled gestures to control the volume and channel functions of a television. While this work represented a new direction of perceptual, device-free gestures, computer-vision interactions to-date, remain a technique restricted to laboratory studies [4].

Current problems in the field include interface design with computers using gestures of the human body, typically hand movements. In gesture recognition technology, a camera reads the movements of the human body and communicates the data to a computer that uses the gestures as input to control devices or applications. For example, a person clapping his hands together in front of a camera can produce the sound of cymbals being clashed together when the gesture is fed through a computer. One way gesture recognition is being used is to help the physically impaired to interact with computers, such as interpreting sign language. The technology also has the potential to change the way users interact with computers by eliminating input devices such as joysticks, mice and keyboards and allowing the unencumbered body to give signals to the computer.
through gestures such as finger pointing. Unlike haptic interfaces, gesture recognition
does not require the user to wear any special equipment or attach any devices to the body.
The gestures of the body are read by a camera instead of sensors attached to a body such
as data glove. In addition to hand and body movement, gesture recognition technology
also can be used to read facial and speech expressions (i.e., lip reading), and eye
movements.

A primary goal of gesture recognition research is to create a system which can identify
specific human gestures and use them to convey information or to send commands for
device control.

2.2 Types of Gestures

There are three types of gestures [12] reported in the literature: mimetic, deictic, and
arbitrary. In mimetic gestures, motions form an object's main shape or representative
feature. These gestures are intended to be transparent. Mimetic gestures are useful in
gesture language representations. Deictic gestures are used to point at important objects,
and each gesture is transparent within its given context. These gestures can be specific,
general, or functional. Specific gestures refer to one object. General gestures refer to a
class of objects. Functional gestures represent intentions, such as pointing to a chair to
ask for permission to sit. Deictic gestures are also useful in gesture language
representations. Arbitrary gestures are those whose interpretation must be learned due to
their opacity. Although they are not common in a cultural setting, once learned they can
be used and understood without any complimentary verbal information. An example is the set of gestures used for crane operation [13]. Arbitrary gestures are useful because they can be specifically created for use in device control. These gesture types are already arbitrarily defined and understood without any additional verbal information [12].

### 2.3 Application Domains of Gestures

Gestures are considered as an input technique for large number of computing domains. Therefore gesture recognition has been applied to numerous applications such as:

- Virtual Reality.
- Augmented Reality.
- Robotics and tele-presence.
- Graphics and drawing applications for desktop and tablet PC applications.
- Computer supported collaborative work (CSCW).
- Ubiquitous computing and smart environments.
- Tangible computing.
- Pervasive and mobile computing.
- Telematics.
- Adaptive technology.
- Communication.
- Games.

Gesture recognition technique had been considered as standard input in successful development of applications like virtual and augmented reality, pervasive and ubiquitous
computing etc. Similarly pervasive, mobile and wearable computing has been employing gestures to enable eyes-free interactions that enable multitasking while reducing distraction to primary activities such as walking, driving etc.

2.4 Architecture of Gesture Recognition System

Like any pattern recognition system, gesture recognition system consists of mainly three components: acquisition, feature extraction / parameterization and recognition. Acquisition is the first stage in the gesture recognition system to acquire representative raw data of gesture. Variety of options exists for acquisition of gesture. Vision sensors are considered among the most common source for gesture acquisition. Usually pre-processing stage also accompanies acquisition stage; though pre-processing stage may be optional for certain scenarios. Feature extraction / parameterization stage converts the raw data or pre-processed data into more meaningful and suitable form for further processing. Finally recognition stage identifies the currently executing gesture by comparing the real time features to the pre-stored features of gesture classes. Briefly stating, architecture of gesture recognition system consists of following components

- Gesture Acquisition and Pre-processing.
- Gesture Feature Extraction and Representation.
- Gesture Recognition or Classification.
2.4.1 Gesture acquisition and Pre-processing

Variety of transducers or are available for gesture acquisitioning ranging from discrete components such as magnetic sensor, position sensors, accelerometer sensor to fully developed functional units such as gesture gloves and more sophisticated systems such as mono-vision, stereo-vision and range sensors etc. Technologically these sensors employ different sensing methodologies ranging from basic physics principals to sophisticated range imaging and lasers. This research work is more related to optical sensing of gestures. Therefore, methodologies for hand gesture acquisition by vision sensors are discussed in length.

Vision sensors are installed in mainly two configurations: mono-vision and stereo-vision. Mono-vision sensors incorporate one sensing camera naming CCD or CMOS with multiple possible interfacing such as IEEE 1394, USB 2.0, Camera link, Ethernet, PAL, NTSC or CCIR etc. for their video signal transmission. Similar kind of acquisition sensors are utilized for stereo-vision. However, the primary difference exists in the further interpretation of stereo-imaging.

2.4.2 Gesture Feature Extraction and Representation

Feature extraction for the purpose of gesture recognition consists of segmentation of image components that contribute to the formation of gesture inputs. Both raster (skin tone blobs, colored gloves, shape signatures etc.) and vector informations (joint
geometry, facial animation parameters etc.) form the basis of feature extraction. Some commonly features for gesture recognition are given below:

- Image moments.
- Skin tone Blobs.
- Colored Markers.
- Geometric Features.
- Multi scale shape characterization.
- Motion History Images (MHI) and Motion Energy Images (MEI).
- Shape Signatures.
- Polygonal approximation-based Shape Descriptor.
- Shape descriptors based upon regions and graphs.

2.4.2.1 Image moments

These features are defined in analogy to the moments of descriptive statistics [15]. These are used to describe objects with variance to translation, rotation and scale. However such algorithms are computationally very expensive. For random variable suppose $X$, moments and central moments are characterized by the density function $p(x)$ which is defined in terms of expectance of $X$.

2.4.2.2 Skin tone Blobs

Skin color is the obvious feature for the body parts such as hand and faces. Therefore skin color detection algorithms detect skin areas in the captured images by classifying sub images into skin and non-skin classes [16]. Now the detected skin areas are further
split and merged to filter out false skin areas. These finalized skin detected areas are then applied to different processes to compute the pose or geometric features. The derived pose or geometric features actually represent the features for gesture recognition or classification.

2.4.2.3 Geometric Features

There are varieties of geometric features proposed in the literature focusing at solving many real time problems. Few geometric features are specifically designed for certain applications while many geometric features are general purpose in nature and suitable for number of applications. Geometric features are well suited for trajectory representation which is developed by the tracking of particular gesticulating body part. Geometric features are suitable whenever shape of the object is important and trajectory may be assumed as shape of arbitrary nature [17].

2.4.2.4 Motion Energy Images (MEI) and Motion History Images (MHI)

Motion energy images and motion history images are the view based template approach for the representation of action. Motion energy image (MEI) coarsely describes the spatial distribution of energy for a given view of a given action. Motion energy image provide coarse level index to the pre-stored motion images [18]. Motion history image is a static image whose pixel intensity is a function of the recent motion in a sequence. Statistical feature are further computed from motion history images. These methods are suitable for fixed view systems only and interestingly vision based gesture recognition systems are mostly fixed view based systems.
2.4.2.5 Shape Signatures

Shape signatures is the representation of shapes in which one or more 1D signals are produced that somehow describe the 2D shape [19]. It is also fascinating to point that 1D signatures can be obtained from both contour based and region based representation. Representation of shape in the form of 1D representation results in the application of 1D signal processing techniques such as scale-space and wavelet methods. For dynamic gesture feature extraction and representation, boundary based signatures takes more attention as dynamic gestures are mostly represented by the trajectories when motion analysis part is under consideration. There are many possibilities in which 1D signature can be generated e.g. distance to centroid based signature, slope-density function based boundary signature, arc-height function and orientation along the contour etc.

2.4.2.6 Polygonal approximation-based Shape Descriptor

Shape contours can also be represented by polygonal approximations and can be utilized for trajectory feature representation e.g. following features can be derived from polygonal approximation of the contours [17].

- Number of corners and vertices.
- Angle and sides statistics such as mean, median, variance and moments.
- Major and minor sides lengths.
- Major and minor sides ratio.
- Major and minor angle ratio.
- Ratio b/w the major angle and the sum of all angles.
• Ratio b/w the standard deviations of the sides and angles.
• Mean absolute difference of adjacent angles.
• Symmetry measures

2.4.2.7 Shape descriptors based upon regions and graphs

Shapes can also be represented by region descriptors. For dynamic gestures of hands and heads, shape descriptors based upon regions and graphs can be derived from motion energy images and motion history images [17]. Following shape descriptors based upon regions and graphs can be derived

• Number of constituent parts.
• Number of junctions.
• Number of extremities.
• Number of branches.
• Branches size.
• Convex hull and convex deficiency measure such as area, perimeter etc.
• Distance transforms statistics.
• Number of self intersections.
• Geodesics statistics.

2.4.3 Gesture Recognition and Classification

Finally, gesture recognition and classification stage classifies the reported features belonging to certain pre-stored category. Following are the list of gesture recognition or classification methods proposed in the literature so far:
• Hidden Markov Model (HMM).
• Time Delay Neural Network (TDNN).
• Elman Network.
• Dynamic Time Warping (DTW).
• Dynamic Programming.
• Bayesian Classifier.
• Multi-layer Perceptrons.
• Genetic Algorithms.
• Fuzzy Inference Engine.
• Template Matching.
• Condensation Algorithm.
• Radial Basis Functions.
• Self Organizing Map.
• Binary Associative Machines.
• Syntactic Pattern Recognition.
• Decision Tree.

2.4.3.1 Hidden Markov Model (HMM)

Dynamic gestures prolong over certain duration of time, due to which gestures usually appear in the form of sequences or spatiotemporal information. Ultimately some kind of sequence or spatiotemporal matching process is required for successful gesture recognition. Theoretically, a sequence can be characterized as being generated by some
parametric random process. Modeling of such parametric random process can be successfully accomplished by hidden Markov model (HMM) whose parameters can be learned from training sample of example sequences by a Baum Belch algorithm [13].

2.4.3.2 Time delay neural network

Time delay neural networks are special artificial neural networks which receive input over several time steps [20]. Theoretically, time delay neural networks are also considered as an extension of multi-layer perceptron. TDNN is based on time delays which gives individual neurons the ability to store the history of their input signals. Therefore the network can adapt to sequence of patterns. Due to the concept of time delay, each neuron has access not only to present input at time $t$ but also to the inputs at time $t-1, t-2, \ldots, t-n$. Therefore each neuron can detect relationship b/w the current and former input values which might be a typical pattern in the input signal. Also, the network is able to approximate functions that are derived from time sampled history of input signal. Learning of typical TDNN can be accomplished by standard back propagation as well as its variants.

2.4.3.3 Dynamic time warping

Dynamic time warping is an algorithm for measuring similarity between two sequences which may vary in time or speed. For instance, similarities in walking patterns would be detected, even if in one video the person was walking slowly and if in another he or she were walking more quickly, or even if there were accelerations and decelerations during the course of one observation. DTW has been applied to video, audio, and graphics --
indeed, any data which can be turned into a linear representation can be analyzed with DTW. A well known application is automatic speech recognition, to cope with different speaking speeds.

In general, DTW is a method that allows a computer to find an optimal match between two given sequences (e.g. time series) with certain restrictions [21]. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. This sequence alignment method is often used in the context of hidden Markov models.

One example of the restrictions imposed on the matching of the sequences is the monotonic nature of the mapping in the time dimension. Continuity is less important in DTW than in other pattern matching algorithms; DTW is an algorithm particularly suited to matching sequences with missing information, provided there are long enough segments for matching to occur. The optimization process is performed using dynamic programming, hence the name.

2.4.3.4 Dynamic Programming

Dynamic programming is a method of solving problems exhibiting the properties of overlapping sub-problems and optimal substructure (described below) that takes much less time than naïve methods. The term was originally used in the 1940s by Richard Bellman to describe the process of solving problems where one needs to find the best decisions one after another. By 1953, he had refined this to the modern meaning. The
word "programming" in "dynamic programming" has no particular connection to computer programming at all. A program is, instead, the plan for action that is produced. For instance, a finalized schedule of events at an exhibition is sometimes called a program. Programming, in this sense, is finding an acceptable plan of action [22].

2.4.3.5 Bayesian Classifier

Bayesian classifier is one of the basic classification methods from supervised classification category [23]. Baye’s rule is stated as follows:

\[ P(C \mid x) = \frac{P(C)p(x \mid C)}{p(x)} \]  \hspace{1cm} (2.1)

Where \( P(C) \) is the prior probability, which in particular can be said the prior probability of gesture. \( p(x \mid C) \) is the class likelihood and is the conditional probability that an event belonging to \( C \) has observation \( x \). Likelihood for gesture recognition can be specified as the conditional probability that gesture belonging to class \( C \) has feature vector \( x \). \( p(x) \) is the evidence in the sense that a particular feature vector for some gesture appears with this probability. Finally posterior probability \( P(C \mid x) \) is calculated by combining the prior, likelihood and evidence. For multiple classes, the posterior probability can be calculated as:

\[ P(C_i \mid x) = \frac{P(C_i)p(x \mid C_i)}{p(x)} \]  \hspace{1cm} (2.2)

Finally for the minimum error, the Bayesian classifier selects the class with the highest posterior probability i.e.

Select \( C_i \) if \( P(C_i \mid x) = \max_i P(C_i \mid x) \)
2.4.3.6 Multi-layer Perceptron

Conceptually, a perceptron is a kind of binary classifier that maps its input vectors to an output value [24]. Inputs are directly fed to the output unit via the weighted connections, therefore the perceptrons can be considered as simplest kind of feed forward network. Mathematically, a perceptron can be defined as follows:

\[ y = \phi \left( \sum_{i=1}^{n} w_i x_i + b \right) = \phi(w^T x + b) \]  \hfill (2.3)

where \( w \) denotes the vectors of weights, \( x \) the vector of inputs, \( b \) is the bias and \( \phi \) is the

![Diagram of a multi-layer perceptron](image)

**Figure 2-1: Single Layer Perceptron**
activation function. Following figure shows the signal flow graph of perceptron. Single layer perceptrons are not useful for classifying non-linear classification problems. However these perceptrons can be utilized as basic building block for multilayer perceptrons which have more functional capabilities. Multilayer perceptron typically consist of a set of source nodes forming the input layer, one or more hidden layers as computational nodes and a linear output layer. Multilayer perceptron are mostly utilized to solve non-linear classification problems. Signal flow diagram for multilayer perceptron is shown in Figure 2-2.

![Multilayer Perceptron Diagram](image-url)

*Figure 2-2 Multilayer Perceptron*
2.4.3.7 Genetic algorithm

Genetic algorithms are considered to be global search heuristics. Particularly, genetic algorithms are class of evolutionary algorithms which are inspired by evolutionary biology such as inheritance, mutation, selection and crossover [25].

2.4.3.8 Fuzzy Inference Engine

Fuzzy inference engine are mainly involved in decision making of imprecise data or overlapping data. Fuzzy inference engines have been incorporated for feature analysis, clustering and classifier design [26]. Fuzzy inference engines have been explored for solving the problem of gesture recognition to some extent. This research work had real benefit from fuzzy logic paradigm as gesture movements were not very much distinct, therefore fuzzy logic inference has been incorporated.

2.4.3.9 Template Matching

Template matching is most general class for recognition of a pattern in the pre-stored patterns. Pattern matching operates at both levels: raw shapes are recognized or features are initially extracted and later on matched with the pre-stored classes. Pattern matching operates both in spatial as well as in the frequency domain. Basically, template matching is match based segmentation which works by moving an image of the object to be detected across the current image and computing similarity measure at each position. If one position yields a significant similarity value, the algorithm concludes that an instance of the respective object is present at that position. This procedure is easily generalized to an arbitrary number of object patterns to be detected at an arbitrary number of positions.
Matching can be based on different similarity measures, however correlation coefficient being the most common [27]. A template of width \( w \) and height \( h \), which has to be classified, is translated over all positions in the image starting from the top left position. Correlation coefficient of this template and corresponding image section at any particular location is calculated by the Equation 2-4.

\[
\begin{align*}
    r(x, y) &= \frac{\sum_{s=0}^{w-1} \sum_{t=0}^{h-1} g_{x+s\cdot r, y+t\cdot r} \cdot I_{x+t, y+s} - N \cdot g \cdot t}{(N - 1) \sigma_x \sigma_t} \\
    &\quad \text{(2.4)}
\end{align*}
\]

Where \((x, y)\) the current position in the image plane are, \(g_{x+s\cdot r, y+t\cdot r}\) is the gray level in the image which corresponds to the template gray level \(I_{x+t, y+s}\). Also

\[
\begin{align*}
    g &= \frac{1}{N} \sum_{s=0}^{w-1} \sum_{t=0}^{h-1} g_{x+s\cdot r, y+t\cdot r} \\
    &\quad \text{(2.5)}
\end{align*}
\]

\[
\begin{align*}
    t &= \frac{1}{N} \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} I_{x, y} \\
    &\quad \text{(2.6)}
\end{align*}
\]

\[
\begin{align*}
    \sigma_x &= \sqrt{\frac{1}{N - 1} \left( \sum_{s=0}^{w-1} \sum_{t=0}^{h-1} g_{x+s\cdot r, y+t\cdot r} - N \cdot g^2 \right)} \\
    &\quad \text{(2.7)}
\end{align*}
\]

\[
\begin{align*}
    \sigma_t &= \sqrt{\frac{1}{N - 1} \left( \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} I_{x, y} - N \cdot t^2 \right)} \\
    &\quad \text{(2.8)}
\end{align*}
\]

The basic template matching method described above performs full correlation with all points of the template for all image pixels which requires intensive computation. Variety of optimizations are proposed for the achieving computational efficiency such as coarse to fine sampling and sub-sampling approaches. Similarly, rather calculating correlation
coefficient at equidistant positions, it is possible to adapt for optimization methods such as self-organizing neural networks.

2.4.3.10 Radial Basis Function

The methodology of radial basis function originates from the techniques in performing the exact interpolation of set of data point in some multidimensional space. In radial basis function, hidden units are activated by computing the distance b/w input vector and prototype vector rather than computing a non-linear function of the scalar product b/w input vector and weight vector. Construction of radial basis function involves three layers: input layer, hidden layer and output layer. Radial basis functions were developed with the aim of speedier training stage while maintaining similar efficiency levels obtained through multi-layer perceptrons [24].

2.4.3.11 Hopfield Neural Network

Hopfield network is considered suitable for classification of binary patterns. The properties of binary classification are helpful from gesture recognition point of view because dynamic gestures are represented by motion trajectories which are essentially a point in the 2D image space. Therefore trajectory representation is possible in the form of
2D binary images. The Hopfield network has finite set of neurons which serve as processing units [24]. Each neuron has value (or state) at time $t$ denoted by $a_i(t)$. A neuron in the Hopfield net can be in either of two states, -1 or +1; therefore $a_i(t) \in \{-1, +1\}$.

2.4.3.12 Binary Associative Memory

A binary associative memory is a vector space transform which may or may not be linear. A bidirectional associative memory is a generalization of the Hopfield net. The domain and range of the BAM need not to be of the same dimension. Similar to Hopfield network, binary associative machines can also be utilized for recognition of gesture trajectories.

2.4.3.13 Decision Tree

A decision tree is a hierarchical data structure implementing the divide and conquers strategy [28]. Basically, decision tree is an efficient non parametric method. In non parametric estimation, the input space is divided into local regions, defined by some distance measure such as Euclidean norm. a decision tree is a hierarchical model for supervised learning and is composed of internal decision nodes. Each decision node say $m$ implements a test function $f_m(x)$ with the branches discretely labeled. Each $f_m(x)$ is a discriminant in the $d$-dimensional input space which is divided into smaller regions. Gesture recognition is also helped by the decision trees.
2.5 Hand gesture recognition

Hand is one of the modalities for gesture recognition. There are other modalities such as head, face or body.

2.5.1 Hand detection

Hand detection is the first step in automated hand gesture recognition. The reliability of accurate hand detection and localization has major role in the overall performance of hand gesture recognition system. Given a single image or video, ideal hand detector should be able to identify and locate single or both hands irrespective of their position, scale and orientation. Hand detection can be performed based upon several cues [29]: skin color (for hands in color images and videos), motion (for hands in videos), hand pose, hand appearance, or combination of these parameters. Hand detection is posed as classifying the pattern in the sub-window as hand or non-hand. The hand/non-hand classifier is learnt from hand and non-hand training examples.

2.5.2 Appearance Based Approaches

With the appearance based approach [30], hand detection is treated as a problem of classification of each scanned window as one of two classes (i.e. hand and non-hand). Appearance based methods avoid difficulties in modeling 3D structures of hands by considering possible hand appearances under varying illumination conditions. A hand/non-hand classifier may be learned from training set composed of hand examples taken under possible conditions. Development of such classifier is possible as pixels in
the hand are highly correlated whereas non-hand sub-window shows much less regularity.

2.6 Summary

This chapter served as an overview of gesture recognition. Vision based gesture acquisition has been considered. Gesture recognition system performs in two stages: feature extraction and classification. Therefore algorithms utilized for feature extraction had been considered in length. Geometric features and statistical features suitable for gesture recognition are described at length. Similarly recognition and classification methods specifically tailored for gesture recognition are described such as Hidden Markov model, time delay neural networks and dynamic time warping etc. Finally hand gesture recognition has been considered at length. Hand detection is the first step in hand gesture recognition which is accomplished by skin color detection and appearance features.

2.7 Reference:


CHAPTER 3

FUZZY LOGIC

3.1 Introduction

Fuzzy logic is one of several means by which an intelligent system accommodates uncertainty. Both the inputs and outputs of a system may be imprecise and uncertain. The process of knowledge acquisition may also be quite imprecise [8]. It is likely that the knowledge acquired does not exactly capture the knowledge of the expert, especially since the expert is often not aware of all the tools used in the reasoning process. The knowledge that an expert reasons with may itself contain uncertainty. This must be effectively captured if the expert’s reasoning process is to be emulated. If knowledge is not expressed in some formal language, then its meaning cannot be interpreted exactly. Fuzzy reasoning technique can provide the basis for representing the imprecision inherent in the expert’s knowledge. There are at least four circumstances in which the concept and techniques of fuzzy-set theory are uniquely helpful in intelligent decision making [6]. Firstly, where a fuzzy set serves as an interface between a linguistically formatted feature (non-numeric or symbolic feature) and quantitative measurements. The second circumstances is at the class-membership level, rather than at the feature level. This is useful in classification of uncertain object into different classes. The third circumstance is
where fuzzy membership values are used to provide an estimate of missing or incomplete information. The fourth circumstance is in the context of passing structure and classes.

Fuzzy logic allows for an effective means of formally representing uncertain data. For example, one way of saying Joe's age is about 30 mathematically is as follows:

\[ \frac{2}{27} + \frac{5}{28} + \frac{8}{29} + \frac{1}{30} + \frac{8}{31} + \frac{5}{32} + \frac{2}{33} \]

This implies that possibility of Joe's age being 27 is .2, 28 is .5, 29 is .8, 30 is 1 and so on. If age of about 30 is classified as medium then another way of stating this fact could be Joe's age is medium. The variable age now takes linguistic values (young, medium, old, etc.) instead of numerical values. Hence it is called a linguistic variable. The interval of ages 27 to 33 is called the support of the set Joe's age is medium and the associated value in the interval \([0,1]\) are the degrees of membership. Finally, a fuzzy set is defined as follows:

**Definition:** If \( X \) is a collection of objects \( x \), then a fuzzy set \( A \) in \( X \) is the set of ordered pairs \([1]\)

\[ A = \{(x, \mu_A(x))| x \in X \} \]  \hspace{1cm} (3-1)

\[ \mu_A(x): X \rightarrow [0,1] \]  \hspace{1cm} (3-2)

The entity \( \mu_A(x) \) is the membership function, the value of which is the grade of membership of \( x \) in \( A \). It is also the degree to which the deterministic measurement \( x \) is compatible with (the vague concept of) \( A \).

\( X \) is called universe of discourse and all the elements of \( X \) for which \( A \) is nonzero comprise the support of \( A \):
\[ \text{sup}(A) = \{ x \in X | \mu_A(x) > 0 \} \] (3-3)

If the support of fuzzy set is a single point, then it is called a fuzzy singleton. The center of the fuzzy set is the point(s) \( x \in X \) at which \( \mu_A(x) \) achieves its maximum value.

### 3.2 Intersection, Union and Compliment

Let \( A \) and \( B \) be two fuzzy sets in \( X \). Then the intersection \( A \cap B \) is a fuzzy set in \( X \) with membership defined as follows:

\[ \mu_{A \cap B}(x) = \min \{ \mu_A(x), \mu_B(x) \} \] (3-4)

The union \( A \cup B \) is a fuzzy set in \( X \) with membership defined as follows:

\[ \mu_{A \cup B}(x) = \max \{ \mu_A(x), \mu_B(x) \} \] (3-5)

Usually the intersection and the union operator are denoted by \( \wedge \) and \( \vee \), respectively.

### 3.3 T-norm and T-conorm:

A T-norm denoted by \( \odot \) is a function \( [0,1] \times [0,1] \rightarrow [0,1] \). Numerous definitions of T-norms are available [5], [6], [7]. Some popular definitions are given below.

\[ \mu_A(x) \odot \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \]
\[ \mu_A(x) \odot \mu_B(x) = \mu_A(x) \mu_B(x) \]
\[ \mu_A(x) \odot \mu_B(x) = \max\{0, \mu_A(x) + \mu_B(x) - 1\} \]
\[ \mu_A(x) \odot \mu_B(x) = 1 - \min \left[ 1, \left( 1 - \mu_A(x) \right)^\nu + \left( 1 - \mu_B(x) \right)^\nu \right] \text{ for } p \geq 1 \] (3-6)

A T-conorm, also called an S-norm, denoted by \( \oplus \) is a function \( [0,1] \times [0,1] \rightarrow [0,1] \). Some popular definitions of S-norm are given below.
\[ \mu_A(x) \cdot \mu_B(x) = \max(\mu_A(x), \mu_B(x)) \]

\[ \mu_A(x) + \mu_B(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x) \]

(3.7)

\[ \mu_A(x) = \min\{1, \mu_A(x) + \mu_B(x)\} \]

\[ \mu_A(x) + \mu_B(x) = \min\left[1, (\mu_A(x))^p + (\mu_B(x))^p\right], p \geq 1 \]

T-norms and T-conorms are used in the fuzzy inference process by which the gesture recognition algorithm estimates the pressed key.

### 3.4 Fuzzy Relation

Let \( U \) and \( V \) be two universes of discourse. A fuzzy relation \( R \) is a fuzzy set in the product space \( U \times V \) having the membership function \( \mu_R(x, y) \), where \( x \in U \) and \( y \in V \).

Fuzzy relations form the basis of gesture recognition process by establishing an association of the input gesture with the pre-stored gesture. Fuzzy relation in a gesture recognition environment is a mapping from feature space to the decision space.

#### 3.4.1 Fuzzy implication

Fuzzy implication is a special subclass of a fuzzy relation. There exist several ways of introducing fuzzy implications [2]. Let \( A \) and \( B \) be fuzzy sets in \( U \) and \( V \), respectively. A fuzzy implication is denoted by \( A \rightarrow B \) in \( U \times V \). The most classical forms involving premise \( X \) is \( A \) and conclusion \( Y \) is \( B \), \( \forall x \in U, \forall y \in V \) are as follows:
\[\begin{align*}
\mu_{A \circ B}(x, y) &= \mu_A(x) \circ \mu_B(y) \\
\mu_{A \odot B}(x, y) &= \mu_A(x) \odot \mu_B(y) \\
\mu_{A \land B}(x, y) &= \min(\mu_A(x), \mu_B(y)) \\
\mu_{A \lor B}(x, y) &= \min(1 - \mu_A(x) + \mu_B(y), 1) \\
\mu_{A \ltimes B}(x, y) &= \max(1 - \mu_A(x), \mu_B(y)) \\
\mu_{A \triangleright B}(x, y) &= \text{iff} \mu_A(x) \leq \mu_B(y)
\end{align*}\]  

\[\text{(3-8)}\]

### 3.5 Fuzzy Composition

Let \( R \) and \( S \) be two fuzzy relation \( U \times V \) and \( V \times W \), respectively. For \( x \in U, y \in V, z \in W \), few popular fuzzy compositions are given below.

\[\begin{align*}
\mu_{R \circ S}(x, z) &= \max_{y \in V} \left[ \min(\mu_R(x, y), \mu_S(y, z)) \right] \\
\mu_{R \odot S}(x, z) &= \sup_{y \in V} \left[ \mu_R(x, y) \odot \mu_S(y, z) \right] \\
\mu_{R \ast S}(x, z) &= \inf_{y \in V} \left[ \max(\mu_R(x, y), \mu_S(y, z)) \right]
\end{align*}\]  

\[\text{(3-9)}\]

### 3.6 Fuzzy Inference Engine

As mentioned above, a fuzzy relation is mapping from feature space to the decision space. A fuzzy inference engine employs the mapping defined by the fuzzy relation to generate the output. It uses fuzzy logic principles to combine the fuzzy IF-THEN rules in the fuzzy rule base into a mapping form the fuzzy sets in \( U = U_1 \times \ldots \times U_n \) to fuzzy set in \( V \). Where \( U \) is the input (fault features) domain and \( V \) is the output (decision) domain.

Suppose that there is L number of rules of the following form:

\[R^i : IF \ x_1 \ is \ F_{i1}^i \ and \ \ldots \ and \ x_n \ is \ F_{in}^i, \ THEN \ y \ is \ G_i \]  

\[\text{(3-10)}\]

Where \( F_i^i \) and \( G_i^i \) are fuzzy sets in \( U_i \) and \( V \), respectively, and

\[x = (x_1, \ldots, x_n) \in U, x_i \in U_i \text{ (inputs) and } y \in V \text{ (outputs)} \]

are the linguistic variables, and
where \( L \) is the total number of rules. Each fuzzy IF-THEN rule in Equation 3-10 is defined as the fuzzy implication \( F^*_1 \times \ldots \times F^*_n \rightarrow G^*_1 \) in the product space \( U \times V \). Let this implication be represented by a fuzzy relation \( R \) as:

\[
\mu_R(x, y) = \mu_{F^*_1 \times \ldots \times F^*_n \rightarrow G^*_1}(x, y)
\]  

(3-11)

Let \( A \) in \( U \) be the input to the fuzzy inference engine, each fuzzy IF-THEN rule of Equation 3.10 determines a set \( B' \) in \( V \) using sub-star composition \( (B' = R' \circ A) \):

\[
\mu_B(y) = \sup_{x \in U} \left[ \mu_{F^*_1 \times \ldots \times F^*_n \rightarrow G^*_1}(x, y) \ast \mu_A(x) \right]
\]

(3-12)

The output fuzzy set \( B \) is the defuzzified from a fuzzy set in \( V \) to a crisp point \( y \in V \).

Among the most popular defuzzifiers are maximum defuzzifier and center of gravity defuzzifier. Maximum defuzzifier can be defined as

\[
y = \arg \sup_{y \in V} (\mu_B(y))
\]

(3-13)

Where \( (\mu_B(y)) \) is given be Equation 3.12.

Similarly, center of gravity defuzzifier is defined as

\[
y = \frac{\sum_{i=1}^{L} y^i \left( \mu_B(y^i) \right)}{\sum_{i=1}^{L} \left( \mu_B(y^i) \right)}
\]

(3-14)

If \( x \) is the input space, then a fuzzy logic systems \( f \) with center of gravity defuzzifier and product implication from Equation 3.8 and singleton fuzzifier are of following form:

\[
y = f(x) = \frac{\sum_{i=1}^{L} \left( y^i \prod_{j=1}^{n} \mu_{F^*_j}(x_j) \right)}{\sum_{i=1}^{L} \left( \prod_{j=1}^{n} \mu_{F^*_j}(x_j) \right)}
\]

(3-15)

**Proof:** using product-implication rule in Equation 3.9:
\[ \mu_{\mu} (y') = \sup_{x, t} \left[ \prod_{i=1}^{n} \mu_{F_i} (x_i) \mu_{G_i} (y') \mu_{A} (x) \right] \]  

(3-16)

For a singleton fuzzifier, \( \mu_{A} (x) = 1 \) when \( x \) is at the input crisp point and 0 otherwise.

Also assume that \( \mu_{G_i} (y') = 1 \) (peak value of output is usually 1). Hence:

\[ \mu_{\mu} (y') = \prod_{i=1}^{n} \mu_{F_i}. \]  

(3-17)

Substituting this value in Equation 3.14 to obtain Equation 3.15. Assuming that the fuzzy sets \( F_i \) are Gaussian in form, that is, \( \mu_{F_i} (x_i) = \exp \left( -\frac{x_i - x_i^f}{\sigma_i^2} \right) \) where \( x_i \) and \( \sigma_i^2 \) are adjustable parameters, then equation 3.15 becomes:

\[ f(\bar{x}) = \frac{\sum_{t=1}^{L} \left( x_i^f \prod_{i=1}^{n} \exp \left( -\frac{x_i - x_i^f}{\sigma_i^2} \right) \right)}{\sum_{t=1}^{L} \prod_{i=1}^{n} \exp \left( -\frac{x_i - x_i^f}{\sigma_i^2} \right)} \]  

(3-18)

The fuzzy logic inference mechanism in designed for discrete points only. These points are included in the rule-set and a fuzzy relation is designed. However, the inputs to the inference engine may not exactly match the rule-base. This is where the fuzzy tools provide support for approximate reasoning.

Consider a fuzzy logic system \( f \) trying to approximate a continuous function \( g(x) \), where \( x = (x1, x2) \) is the input to the system. Since the training set of the rule-base are fuzzy in nature, they spread out in a Gaussian form and overlap regions of universe of discourse.
Thus, even if the singleton input is not fully covered by one set, it can be partially covered by two overlapping sets and still be adequately analyzed by the inference mechanism.

### 3.7 Fuzzy logic as a Universal Approximator

Fuzzy logic inferencing described above is capable of approximating any set of nonlinear functions in $U$, if $U$ is compact.

#### 3.7.1 Universal Approximation Theorem

[3] For any given real continuous function $g$ on a compact set $U \subset \mathbb{R}^n$ and arbitrary $\varepsilon > 0$, there exists a fuzzy logic system $f$ in the form of Equation 3.18 such that

$$\sup_{x \in U} |f(x) - g(x)| < \varepsilon$$

(3.19)

### 3.8 Fuzzy Measures

A wide variety of fuzzy measures have been defined in the literature but similarity and distance measures are of special interest in the context of gesture recognition. They provide the basic foundation for fuzzy classification and clustering. Several measures of similarity distance are available for fuzzy sets [4].

### 3.9 Fuzzy Inferencing

Inferencing or decision making is done by a set of IF-THEN rules. Let the input universe of discourse be $T$ and $G$ and the output universe of discourse is $K$. Let $J_{i,i}(m)$ be the
location of $i^{th}$ joint in the $m^{th}$ frame in the input universe of discourse, and $J_{j,i} (m)$ be the location of same joint in the pre-stored gesture $G_j$.

The input (trajectory features) variables and the pre-stored gestures are combined by the following fuzzy rule base for the decision of key stroke in the output space $K$.

If $\mu_{f,i} \text{ is } G_A \land \mu_{i,1} \text{ is low} \land \mu_{i,2} \text{ is low} \land \mu_{i,3} \text{ is low} \land \mu_{i,4} \text{ is high}$ THEN $K$ is $k_\gamma$.

If $\mu_{f,i} \text{ is } G_Y \land \mu_{i,1} \text{ is high} \land \mu_{i,2} \text{ is low} \land \mu_{i,3} \text{ is low} \land \mu_{i,4} \text{ is low}$ THEN $K$ is $k_y$.

If $\mu_{f,i} \text{ is } G_E \land \mu_{i,1} \text{ is low} \land \mu_{i,2} \text{ is high} \land \mu_{i,3} \text{ is low} \land \mu_{i,4} \text{ is low}$ THEN $K$ is $k_e$.

If $\mu_{f,i} \text{ is } G_C \land \mu_{i,1} \text{ is low} \land \mu_{i,2} \text{ is high} \land \mu_{i,3} \text{ is low} \land \mu_{i,4} \text{ is low}$ THEN $K$ is $k_c$.

... Typically one rule is sufficient for one key stroke estimation. Hence, the number of rules is equal to the number of keys.

### 3.10 Performance Metrics

Once a decision about the key stroke is made, then degree of certainty (DOC) is assigned as a measure of confidence to the decisions. It is used to take into account the uncertainty inherent in the system. DOC gives an indication for the closeness of the actual decision to the original training output. The value of DOC indicates the robustness of the decision making logic of the inferencing mechanism.
3.11 Reference:


4.1 Architecture

The proposed virtual keyboard system performs gesture acquisition by a mono vision sensor which can typically be a CCD or CMOS imaging sensor. The functional architecture of the proposed system is shown in the Figure 4-1.

Figure 4-1 Architecture of Gesture Recognition System
Sensor senses the demonstrated gesture and presents the information in the signal form. In case of vision based gesture recognition system, 2D signal is captured this is then combined to form a temporal signal. Gesture acquisition stage captures this signal and records in an appropriate way. Usually there is pre-processing stage which is an optional stage and filters the noise from the acquired signal. Then gesture feature extraction stage extracts the features from the acquired and pre-processed information. These derived features are further utilized in the gesture recognition stage.

![Diagram](image)

**Figure 4-2 Keyboard Comparison**

Suppose the output of the keyboard system is defined as

\[ C = \{c_1, c_2, \ldots, c_L\} \]  \hspace{1cm} (4.1)
Where $c_j$ is either alphanumeric character such as $c_1 = 'A'$, $c_2 = 'B'$ etc. or some control key, $j = 1, 2, ..., L$ where $L = 63$ is the total number of keys on the keyboard (as $L = 63$ in [9]). Both the traditional and gesture based virtual keyboards emit $c_j$ as output as shown in Figure 4-2.

In a traditional keyboard, transducer action is performed in electro-mechanical switch function fashion. While mono vision gesture based virtual keyboard analyzes the hand and finger gestures in the video sequence. Concept of making key stroke by both traditional keyboard and gesture based virtual keyboard is shown in Figure 4-3.

![Figure 4-3 Keyboard Functional Comparison](image)

Hand video is captured continuously. Concept of dominant finger is introduced which defines the dominant finger as responsible finger making key stroke. Whenever dominant finger is triggered, a gesture estimation procedure is initiated, as shown in Figure 4-4. This gesture estimation procedure reveals the key stroke $c_i$ where $i = 1, 2, ..., j, ..., L$. 

56
4.2 Algorithmic Design of Mono Vision Gesture Based Virtual Keyboard

Algorithmic detail of gesture based virtual keyboard is shown in Figure 4-5. The proposed virtual keyboard captures the digital video sequence $V_j$ by mono vision sensor like CCD or CMOS sensors [7]. Frames $F(x, y)$ are extracted where $(x, y)$ are the spatial coordinates in the image plane.
Pre-processing stage performs segmentation where the hand and finger joints are extracted. This has been thoroughly investigated [3], [4] and is not discussed here.

This research work adapted parametric approach, shown in Figure 4-6, where each hand is modeled as set of joints, $J_i$, where $i = 1, 2, ..., N$, $N = 19$ for either right or left hand.
Each $J_i$ is defined as $J_i = [J_{i,x}, J_{i,y}]$ where $J_{i,x}$ and $J_{i,y}$ defines the position of $i^{th}$ joint in $(x,y)$ coordinates of frame $F(x,y)$. Recoding the positions of particular joint for $m$ number of frames represent the trajectory of that joint and is defined as:

$$T_i = [J_{i,1}, J_{i,2}, \ldots, J_{i,m}, J_{i,M}]$$ (4.2)

Cumulatively, the trajectories of all joints are defined as:

$$T = [T_1, T_2, \ldots, T_m]$$ (4.3)

Where a particular $T_i$ represents the trajectory of $i^{th}$ joint over $m$ number of frames. $[\ \ ]'$ represents the transpose. The overall information contained by matrix $T$ of dimension $M \times N$ can be defined as:

59
\[
\begin{bmatrix}
J_i(1)J_i(2)\cdots J_i(M) \\
J_j(1)J_j(2)\cdots J_j(M) \\
\vdots \\
J_k(1)J_k(2)\cdots J_k(M)
\end{bmatrix}
\]

\[ T = \text{ } \]

\[ (4.4) \]

### 4.2.1 Fuzzy Representation of Trajectories

Let
\[
G = \{ G_i, G_{i2}, \ldots, G_{ij}, \ldots, G_{ik} \}
\]

represent the pre-stored templates of reference gestures of key strokes. Each \( G_j \) corresponds to particular key stroke \( c_j \). \( G_j \)'s are learned and stored in off-line training phase. Let \( J_{i,j}(m) \) be the location of \( j^{th} \) joint in the \( m^{th} \) frame in the instantaneous trajectory \( T_i \), and \( J_{i,j}(m) \) be the location of the same joint in the pre-stored gesture \( G_j \).

Fuzzy membership function [8] \( \mu_{i,G_j} \) is computed that defines the degree of membership of the instantaneous trajectory \( T \) with the pre-stored trajectories of gestures \( G_j \) over a universe of discourse (set of all possible gestures) as

\[
\mu_{i,G_j} = 1 - d_j
\]

\[ (4.6) \]

Where
\[
d_j = \sum_{m} \sum_{i} D_{ij} / W
\]

\[ (4.7) \]

While
\[ D_v = \| J_{r,T}(m), J_{s,G}(m) \| \]  

(4-8)

And

\[ W = \max \left( \sum_{i} \sum_{m} D_v \right) \]  

(4-9)

Output of \( \mu_{r,G} \) membership function appears in the range [0 1]. '0' corresponds to minimum membership of instantaneous trajectory in some pre-stored gesture while '1' corresponds to the maximum.

### 4.2.2 Fuzzy Representation of Dominant Finger

The finger responsible for making key stroke is extracted and is called dominant finger. The same joint \( J_s \) is now classified as \( J_{pq}(x, y) \) where \( J_{pq}(x, y) \) is the position of joint \( p \) in finger \( q \) as shown in Figure 4-7, where \( p = 1,...,4 \) representing total number of joints in the finger, and \( q = 1,...,5 \) representing total number of fingers in either hand.

Let \( R_q(m) \) be the motion of finger \( q \) between two successive frames is given below

\[ R_q(m) = \sum_{p=1}^{4} \left( \frac{dJ_{pq,1}(m)}{dm} \right)^2 + \left( \frac{dJ_{pq,2}(m)}{dm} \right)^2 \]  

(4-10)

Aggregate motion of \( q^{th} \) finger is defined as

\[ S_q = \sum_{m=1}^{m_{max}} R_q(m) \]  

(4-11)
Then membership of dominant finger in making key stroke is computed by the membership function $\mu_i$, which is mathematically represented as

$$
\mu_{I,\text{dom}}(S_q) = \begin{cases} 
1 & 0 \leq S_q \leq 0.2S_{q,\text{max}} \\
-5S_q + 2 & 0.2 < S_q \leq 0.4S_{q,\text{max}} \\
0 & S_q > 0.4S_{q,\text{max}}
\end{cases}
$$
\[
\mu_{f_{\text{medium}}}(S_q) = \begin{cases} 
0 & 0 \leq S_q \leq 0.2S_{q,\text{max}} \\
4S_q - \frac{4}{5} & 0.2S_{q,\text{max}} < S_q \leq 0.45S_{q,\text{max}} \\
-4S_q + \frac{4}{5} & 0.45S_{q,\text{max}} < S_q \leq 0.7S_{q,\text{max}} \\
0 & S_q > 0.7 
\end{cases}
\]

\[
\mu_{f_{\text{high}}}(S_q) = \begin{cases} 
0 & 0 \leq S_q \leq 0.5S_{q,\text{max}} \\
5S_q - \frac{5}{2} & 0.5S_{q,\text{max}} < S_q < 0.7S_{q,\text{max}} \\
1 & S_q \geq 0.7S_{q,\text{max}} 
\end{cases}
\]

(4-12)

### 4.2.3 Knowledge Fusion / Fuzzy Rule Base

The recorded trajectory information is compared to the pre-stored gestures by fuzzy rule base approach. Thirty two fuzzy rules are defined which are simultaneously applied to both hands. Each time, key stroke defined by one hand is reported. For example, the general architecture of fuzzy rule base may be defined as follows:

If \( \mu_{f_{i,k}} \) is \( G_j \) \( \wedge \) \( \mu_{f_1} \) is high \( \wedge \) \( \mu_{f_2} \) is low \( \wedge \) \( \mu_{f_3} \) is low \( \wedge \) \( \mu_{f_4} \) is low THEN \( K \) is \( c_j \).

Implementation of above rule based for particular characters is given below:
If $\mu_{g_i} \wedge \mu_{t_1}$ is low $\wedge \mu_{t_2}$ is low $\wedge \mu_{t_3}$ is low $\wedge \mu_{t_4}$ is high \text{ THEN } K \text{ is } k_{t_3}$.

If $\mu_{g_i} \wedge \mu_{t_1}$ is high $\wedge \mu_{t_2}$ is low $\wedge \mu_{t_3}$ is low $\wedge \mu_{t_4}$ is low \text{ THEN } K \text{ is } k_{t_2}$.

If $\mu_{g_i} \wedge \mu_{t_1}$ is low $\wedge \mu_{t_2}$ is high $\wedge \mu_{t_3}$ is low $\wedge \mu_{t_4}$ is low \text{ THEN } K \text{ is } k_{t_1}$.

If $\mu_{g_i} \wedge \mu_{t_1}$ is low $\wedge \mu_{t_2}$ is high $\wedge \mu_{t_3}$ is low $\wedge \mu_{t_4}$ is low \text{ THEN } K \text{ is } k_{t_1}$.

... 

The above rule based is extended to all keys on the keyboard. Finally, defuzzifier[6] is incorporated based upon centroid to declare the key pressed.

### 4.2.4 Degree of Confidence of Key Stroke

DoC (Degree of confidence) is an essential parameter for evaluation of the accuracy of key stroke revealed by gesture recognition algorithm. DoC gives the level of confidence in the declared decision of key stroke. If DoC is below a threshold, then additional supportive tools e.g. dictionary look up tables etc. can be employed. DoC is defined as follow

$$DoC = K_a (K_a - K_b) \times 100$$

(4-13)

Where

$$K_a = \max(K_i)$$

(4-14)

for $j = 1, 2, ..., J, ..., L$ and

$$K_b = \max(K_i), \forall j, b \neq a$$

(4-15)
4.3 Summary

This chapter was devoted to the description of the developed virtual keyboard algorithm. The proposed algorithm is based upon fuzzy logic based gesture recognition implementation. A parametric approach for hand representation had been followed where joints in the finger become reference point. This parametric information had been dealt in two different streams whose outcome is combined by fuzzy rule base approach. DoC (Degree of Confidence) [5] is consulted to check the health of the declared key stroke. The threshold for DoC had been selected empirically.

4.4 Reference:


[9] www.vkb.co.il
CHAPTER 5

EXPERIMENTAL SETUP AND RESULTS

5.1 Development Environment & Results

The analysis and development of the algorithm for virtual keyboard is accomplished in MATLAB®. Experimental video sequences were captured for the key strokes of twenty six characters of English (A-Z) in off-line fashion. Gesture library was developed to store the gestures. The developed gesture library is utilized by gesture based virtual keyboard algorithm as shown in Figure 5-1.
The training set consisted of ten instances of typing of each character by the same subject. The motion trajectories from each of the sequence had been extracted and averaged for each character. These served as templates for fuzzy knowledge base. The experimental video sequences were fed into the fuzzy inference engine and the results were tabulated. Figure 5-2 shows the typical motion of various joints.

![Diagram of motion trajectories](image)

**Figure 5-2: Motion of joints in the finger b/w two successive frames**

It is evident from Figure 5-2 that the instantaneous motion of fourth finger $f_4$ has the highest motion which is represented by peak in $R_4(m)$ at m=6, 7. Rest of the fingers shows much less motion. Hence $f_4$ is the dominant finger for character ‘A’.
Figure 5-3: Aggregate motion by fingers for the specified letters

Figure 5-3 represents the aggregate motion $S_q$ as an example, the aggregate motion of character ‘G’, ‘C’, ‘W’, ‘A’ is shown. It is clear from the figure that dominant fingers for the letters ‘G’, ‘C’, ‘W’, ‘A’ are $f_1$, $f_2$, $f_3$, and $f_4$ respectively. Finally, the degree of confidence (DoC) for each declared key stroke is computed. This reduces the possibility of false characters to be reported to the host application so that accuracy of the system is increased. Degree of confidence and percentage success for each key stroke is shown in Table 5-1.
The design of gesture recognition based virtual keyboard assumed professional typing. Professional typing involves both hands in entering the text. Proposed algorithm partitions the character set between two hands. This reduces the possible number of character choices to the half for each hand. Proposed algorithm suggests smaller rule base for each hand and considers the symmetric nature of the problem.

5.2 Prototype Setup for Virtual Keyboard

Laboratory prototype setup comprised a standard analog interface video camera, camera mounting platform and the processing PC. Different tilt angles are experimented to evaluate the amount of hand exposed in the respective field of view. Suitable tilt angle is
selected subjectively by hit and trail. The videos are captured for typing by different subjects. Six separate videos for each character typing are recorded; therefore 156 videos are captured in total. These captured videos are divided in two sets: one set is used in learning and the other used for the testing.

Practically, gestured virtual keyboard algorithm operates on any PDA in standard environment. It analyzes the video captured by digital camera connected to the PDA through SDIO interface. Gesture recognition algorithm reveals the key stroke and communicates the revealed key to the host application as shown in Figure 5-4.

![Figure 5-4: Application Interface](image)

In the above setup CCD camera captures the video of hang gestures. The gesture recognition algorithms are being processed over the PDA and the revealed key is automatically sent to the host application. The practical setup is shown in Figure 5-5.
5.3 Summary

This chapter showed the experimental and practical setup. CCD camera with SDIO interface has been attached to the PDA where the gesture recognition algorithm processes the captured video. Letters are declared from the processed video. These letters are then communicated to the host application.

This chapter also reported the results experienced in the development of the gesture recognition based virtual keyboard algorithm. Three different level experimental results have been reported. Instantaneous motion for each finger b/w two successive frames has been reported. Similarly motion made by each finger over specified interval to emit different characters has also been presented. Finally the degree of confidence for all characters to be typed by the left hand is presented.
5.4 Reference:


CHAPTER 6

CONCLUSION & FUTURE WORK

6.1 Conclusion

Results showed a very reliable system which can be easily deployed on a consumer device. The proposed system has low cost due to its software centric algorithm and hardware dependency is low [1]. Similar approach can be tested for the other data entry interfaces like kiosks etc. Performance of system had been tested over PDA and it will be tested over mobile phones and other portable devices.

The proposed system works well over modern devices since their processing power is increasing day by day. The key issue of sensing the keystrokes while maintaining the standard layout has been successfully addressed. Hand gesture recognition technology has been applied in number of applications such as virtual reality, human computer interaction, game control, robot interaction, remote controlling of home and office appliances, sign language, activity recognition, human behavior, and training systems etc. The proposed system is advancement to the hand based gesture recognition technology applications. Hidden markov models, time delay neural networks, multi-layer perceptron and template matching have been widely used for gesture recognition technology. The proposed system incorporated fuzzy rule base approach in its development. Fuzzy logic has not been widely used in the development of gesture recognition systems. Fuzzy logic
has been opted for the development due to its ability for interaction with uncertainty or ambiguous data. Therefore proposed system introduced a novel application for fuzzy rule base. Also, the proposed system suggested a new development methodology for keyboard. Though there is no advancement in the existing layout or functionality of keyboard, still it is a novel methodology for the functionality of keyboard.

The performance of the keyboard can be compared with the standard keyboard from a number of important factors such as speed and accuracy. Speed of standard keyboard is mostly limited by the typing speed of the typist. Speed for gesture recognition based virtual keyboard system is defined by image acquisition system. A number of parameters are usually defined for image acquisition system such as field of view, depth of field, frames per second, spatial resolution etc. Frames per second are the parameter which actually defines the typing speed of the gesture recognition based virtual keyboard. It was found by experimentation that ten frames exhibit sufficient information for the pressed key estimation. The camera utilized in the development of system was capable of capturing thirty frames per second. Therefore the speed for the proposed gesture recognition based virtual keyboard is assumed to be three characters per second. This speed is less than standard keyboard and the need for more number of characters per second is evident. However this speed is reasonable keeping in view the typing speed of an average professional typist. Accuracy of the standard keyboard is superior than gesture recognition based virtual keyboard in normal practice. But gesture recognition based virtual keyboard is free from mechanical failures such as its performance can’t be
disturbed by sticky keys. Therefore accuracy for gesture recognition based virtual keyboard is better over elongated periods of time.

Comparative analysis between existing systems and the proposed system shows the contribution of the work. Unfortunately, no research work has been reported yet in literature for virtual keyboard development based upon gesture recognition technology. Benchmark videos for further comparison are developed and made available at the URL www.uettaxila.edu.pk/computer/labs/vip/projects/vk.htm for further research advancements.

6.2 Future work

Current experimental analysis assumes that the key pressing gestures are not overlapping in time which means that a key pressing activity is completed before the next key press activity is initiated. The future work can embark on the analysis of overlapping gestures i.e. simultaneous key pressing motions.

The speed of the gesture recognition based virtual keyboard has been limited due to the ordinary number of frames per second of the image acquisition system. The performance of the system can be enhanced by incorporating the image acquisition system with more number of frames per second. Image acquisition systems with three hundred frames per second are available in standard form factor, while their availability in portable consumer devices such as PDA's, cellular phones etc. is still awaited. However, algorithm
development and testing can be performed over lab prototypes with higher number of frames per second for remittance of sufficient number of characters per second.

Gesture recognition based virtual keyboard required maximum amount of hand available in the video sequence captured by image acquisition system. The system also required the typing in the professional style where both hands are properly utilized for key entry. Therefore this keyboard has poor performance for novice users which are not familiar with the professional typing. Therefore research work can be focused on the development of gesture recognition based virtual keyboard for novice users.

The use of gesture analysis can be applied to areas other than virtual keyboard implementation. Prominent candidates for gesture based applications are medical surgery, rescue operations, training, forensic applications, development of aid systems for handicapped, automotive and industrial control etc. A key upcoming application is gesture aware environment.