DEVELOPING A HAND GESTURE RECOGNITION SYSTEM FOR MAPPING SYMBOLIC HAND GESTURES TO ANALOGOUS EMOJI IN COMPUTER-MEDIATED COMMUNICATION

A Thesis

by

JUNG IN KOH

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Tracy Hammond
Committee Members,	Yoonsuck Choe
	Cara Wallis
Head of Department,	Dilma Da Silva

August 2017

Major Subject: Computer Science

ABSTRACT

Recent trends in computer-mediated communications (CMC) have not only led to expanded instant messaging through the use of images and videos, but have also expanded traditional text messaging with richer content, so-called visual communication markers (VCM) such as emoticons, emojis, and stickers. VCMs could prevent a potential loss of subtle emotional conversation in CMC, which is delivered by nonverbal cues that convey affective and emotional information. However, as the number of VCMs grows in the selection set, the problem of VCM entry needs to be addressed. Additionally, conventional ways for accessing VCMs continues to rely on input entry methods that are not directly and intimately tied to expressive nonverbal cues. One such form of expressive nonverbal that does exist and is well-studied comes in the form of hand gestures. In this work, I propose a user-defined hand gesture set that is highly representative to VCMs and a two-stage hand gesture recognition system (trajectory-based, shape-based) that distinguishes the user-defined hand gestures. While the trajectory-based recognizer distinguishes gestures based on the movements of hands, the shape-based recognizer classifies gestures based on the shapes of hands. The goal of this research is to allow users to be more immersed, natural, and quick in generating VCMs through gestures. The idea is for users to maintain the lower-bandwidth online communication of text messaging to largely retain its convenient and discreet properties, while also incorporating the advantages of higher-bandwidth online communication of video messaging by having users naturally gesture their emotions that are then closely mapped to VCMs. Results show that the accuracy of user-dependent is approximately 86% and the accuracy of user-independent is about 82%.

DEDICATION

To my mother, my father, my aunts, my uncles, my grandmothers and all my family, who always support me in everything.

ACKNOWLEDGMENTS

I would like to sincerely express my gratitude to all the people who helped me in the process of completing this thesis. I am especially thankful my advisor, Dr. Tracy Hammond for her constant support and guidance, and inspiring me with her passion for research and excellence.

I would also like to thank other committee members, Dr. Yoonsuck Choe and Dr. Cara Wallis, for their insights and encouragement for this endeavor.

I would like to thank all the members of the Sketch Recognition Lab at Texas A&M University whose invaluable experience and feedback during my time here helped navigate through difficult times. I am especially indebted to Paul Taele, Josh Cherian, Aqib Niaz Bhat, Vijay Rajann, and Stephanie Valentine.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Dr. Tracy Hammond (advisor) and Dr. Yoonsuck Choe of the Department of Computer Science & Engineering and Dr. Cara Wallis of the Department of Communication.

All work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported by a Research Assistantship from Dr. Tracy Hammond from the Department of Computer Science & Engineering at Texas A&M University.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

1.1 Computer-Mediated Communication

Current trends in computer-mediated communication (CMC) has enabled people to take advantage of a large variety of options for communicating with other people online when co-located face-to-face communication is not available. These online communication options can be represented on a spectrum.

- **Higher-End Spectrum:** People can access tools for communicating video or audio or both in real-time. This is achievable through software such as Skype, or alternatively, embed them as media clips into existing online messaging systems such as iMessage or WeChat.
- **Lower-End Spectrum:** People can also access tools for communicating primarily with text in numerous real-time online messaging systems such as Facebook Messenger, or even primarily with images such as through Snapchat.

With the large availability of options at people's disposal for communicating online, each set of tools offer strengths that people may find appealing. Higher-bandwidth communication from primarily video- and audio-driven messaging software allow people to closely experience the intimacy and expressiveness that is found in co-located face-to-face communication, as individuals can see and hear each other's gesticulations and intonations, respectively, at distinct locations from each other. In contrast, lower-bandwidth communication from primarily text- and image-driven offers people the opportunity to communicate conveniently and discreetly, since such tools require less setup to communicate text and images, and since people may be more receptive to use lower-bandwidth communication in a larger range of settings (e.g., crowded places, privately at home) compared to higher-bandwidth communications.

1.2 Traditional Visual Communication Markers

However, the advantages offered by one type of bandwidth communication proves to be lacking in the context of the other. For example, the expressiveness and intimacy that people enjoy in higher-bandwidth video- and audio-driven communication does not necessarily invoke as strongly to people in lower-bandwidth text- and image-driven communication, and vice versa in terms of convenience and discreetness. As a result, one of the most common ways that people have more closely bridged the advantages between both bandwidth types of communication is through the use of visual communication markers (VCMs) that are embedded in existing text messages [5]. As more people are shifting their attention to lower-bandwidth online communications, these markers provide relatively more recent forms of visual or nonverbal communication cues in digital interaction [6] for uniquely expressing emotions via graphic icons [7, 8, 9, 10]. In addition, there are a number of ongoing studies of VCMs [9, 11, 10]. These markers also range in different types in terms of input and visualization [12].

- **Emotions:** Typographic displays of representations used to convey emotion in a text-only medium.

- **Emoji:** Picture-equivalent of emoticons and supported character-set extensions in most operating systems.

- **Kaomoji:** Emoticons that expand to the fuller set of typed Japanese; and stickers, which are larger custom pictures – some of which are animated – and are used in a number of instant messaging clients.

VCMs can be easily accessible with their own keypads separate from text keypads on mobile devices, and are also highly popular globally such as in East Asia [13].

1.2.1 Advantages of VCMs

Several factors that have led to the popularity and appeal of VMCs to people in lower-bandwidth online communication range from their ability to convey users' emotions and enrich their communication [14], substituting for nonverbal cues that is missing in lower-bandwidth CMCs [5], and even a growing cultural acceptance of replacing text itself (e.g., [15]). Various research work have further supported the advantages of VCMs in lower-bandwidth online communications for reasons including users' enjoyment and satisfaction of them and their valuable addition to how they communicate compared the absence of them [8, 7], their ability to convey emotions to users without the cognition of faces [14], as well as users' positive effect on attributes such as personal interaction, perceived information richness, and perceived usefulness [7].

1.2.2 Disadvantages of VCMs

On the other hand, VCMs present their own limitations either prevent or do not as fully capture the advantages found in higher-bandwidth online communications. These limitations include their difficulty in achieving emotional valence from strictly text-only VCMs for groups such as non-native speakers [16], and also lack analogous mapping to expressive gesturing frequently found with higher-bandwidth online communications [17, 18].

One other major limitation of VCMs — especially emoji — is that a large set of them makes naively inputting them difficult for users to quickly select from [13]. That is, as the number of emoji grows in the selection set, researchers have tried to alleviate the problem of emoji text entry [19, 20].

1.3 Online Video Chat Communication

Unlike VCMs that present lower bandwidth information, online video chat provides a richer form of long-distance communication that more closely emulates intimate face-to-face communication [21]. With improvements in internet speeds and greater ubiquity in both mobile computing devices, online video chat has become a reliable form of communication. However, there are factors that users may find that online video chat may not be an appropriate form of communication. One such factor is privacy, where "privacy during video chat is a key concern of end user," [22] and typically expressed concerns of online video chat in public areas [23]. Users not only emphasized privacy from others outside of the conversation, but within the conversation also. For example, users who communicated with strangers online preferred to express themselves through text versus video communication [24]. Even with people who were familiar with each other in online conversation, people also felt strongly in maintaining privacy due to situations that were not appropriate for conversations in face-to-face situations (e.g., coming out of the shower) or to retain their self-image (e.g., not being publicly presentable) [25]. Another factor stems from accessibility, where despite major strides in internet speeds, there may still be challenges in internet bandwidth availability to sufficiently serve online video communication [26], which may make online video chat communication and its expressiveness less accessible compared to its lower-bandwidth alternative of text communication embedded with VCMs.

1.4 Hand Gesture Communication

While VCMs play an important role in supplementing richer nonverbal communication cues that are already inherent in online video chat communication, conventional ways for accessing VCMs continues to rely on input entry methods that are not directly and intimately tied to expressive nonverbal cues. One such form of expressive nonverbal that does exist and is well-studied come in the form of hand gestures [27, 18]. As a nonverbal communicative cue, "hand gestures constitute an important source" as "either complementary to the speech, or the primary one for people with disabilities," [28] and are especially significant since it supplements verbal communication at a very high proportion [17, 18]. However, while there is interesting potential for utilizing hand gestures as a form of expressive nonverbal input to intuitively select corresponding nonverbal VCMs in online text communication, research work that investigates this relationship correspondence for emotion-enhanced CMC remains largely lacking [29].

1.5 Proposed Work

In this work, I focus on eliciting user-defined gestures that are highly representative to VCMs, and implementing a two-stage hand gesture recognition system (trajectory-based, shape-based) that distinguishes the gestures. Gestures have seen strong successes in capturing the expressiveness of face-to-face communication such as through facial (e.g., smiling [30]) and limb (e.g., hugging [31]) interactions [29]. My idea involves users to be more immersed, natural, and quick in generating VCMs through emoting actions that employ in-air gesturing. The idea is for users to maintain the lower-bandwidth online communication of text messaging to largely retain its convenient and discreet properties, while also incorporating the advantages of higher-bandwidth online communication of video messaging by having users naturally gesture their emotions that are then closely mapped to VCMs.

To this end, the goal of this research is to address the following four questions:

1. **Symbolic Gestures:** What kinds of hand gestures (emblem) do users express that they feel properly symbolize popular emojis?

2. **Quality Features:** Which features are valuable for recognizing the user-defined gestures?

3. **Robust Recognition:** Can we achieve reasonable hand gesture recognition for these emojis using our method?

The rest of the paper address related work on emotion-enhanced interaction in CMC and hand gesture recognition, provide our approach for eliciting user-defined hand gesture from participants and results, our two-stage hand gesture recognition experiments using five classifier algorithms including random forest, sketch our future research steps and draw a conclusion.

2. RELATED WORK

This chapter outlines similar aims in recognition or interaction provides supporting insight to the direction of the two-stage system. The related work is divided into three different perspectives: emotion-enhanced interaction in CMC and trajectory-based gesture recognition, and shape-based gesture recognition.

2.1 Emotion-enhanced Interaction in CMC

The development of multimodal interactions including visual, haptic, or auditory components has increased the interest in the recognition and conveying of emotions in computer mediated communications (CMC). Researchers are currently studying the effect of embedding emotions in CMC such that users can better express their emotion, which in turn would reflect their emotional availability using facial expression, multi-touch gesture, body posture or even electrical resistance of the skin.

Many researchers associate emotion-enhanced CMCs with facial expression. Kaliouby et al. [32] introduce an application called Facial Affect IM (FAIM) based on MSN Messenger 6.0. FAIM analyzes facial expressions of users in real time and maps them analogously maps them to the facial expressions of an emotive avatar. KinChat provided by Wang et al. [25] does so by expanding text communication with visualizations of users' body movements such as facial expressions and head movements via an illustration, to similarly replicate the experiences of face-to-face chat. For example, if the person in one end of the communication smiles, the illustration on the other end would also smile. Even though FAIM and KinChat provide a real-time commentator for facial expression as a complementary to the text-based communication and protect visual anonymity, users might not benefit from the avatar or the illustration since attention is split between the animated graphics and the actual message in text. Filho et al. [33] augments mobile online text chat by detecting user's emotional reaction using facial expression. For emotion detection, the facial expression is analyzed by computer vision technologies from a commercial solution. Depending on the level of emotion detection, either three types of emotion, "happy," "surprised," and "calm," is notified to conversation thread as text. Ali et al. [20] propose Face2Emoji that filters out the emojis that are relevant to a user's facial expression. This study is similar to my study with regard to conveying emotion via emojis.

There is also much effort on studying tactile interfaces to support emotional expression in CMC. The multi-touch gesture interface proposed by Pirzadeh [34] identifies text-based emotional cues and created a set of analogous multi-touch gestures that more expressively map to those emotional cues. The multi-touch gestures are an author defined. For example, drawing a heart shape represents love, drawing a circle present agreement and a long tabbing and draw a horizontal line represents 'hahaha.' Although these type of gesture sets can be clearly performed, it results in somewhat arbitrary gestures sets whose members may be chosen out of concern for reliable recognition. Rovers et al. [35] suggest augmenting text messaging communication using haptic effects and hapticons. The research described HIM (haptic instant messaging), a framework for instant messaging, which supports to exlore the design and to use hapticons and haptic IO devices. Moreover, Shin et al. [36] introduce a tactile emotional interface. In the proposed methods, users send and receive 6 emotional expressions, e.g., grin, cry, anger, surprise, kiss, sleepy through vibration, which cannot achieve direct evocation of emotion.

Additionally, Tan et al. [37] defines gestures as the user's body postures that occur in ubiquitous computing environments in order to infer the user's affective state with a system called mASqUE [37]. Wang uses animated texts and galvanic skin response (GSR) as methods conveying emotional information in order to encourage online users to interact with each other efficiently [38].

Similar to my proposed system in sending VCMs to others online, there have been prior online messaging systems that have expanded beyond conventional means of embedding visual media in existing mainstream online messaging systems. Two such systems include mobile apps React Messenger [12] and SmileChat [39], both of which perform similarly by allowing users to use their mobile device's video camera to record themselves and then send these recordings as looping animated images. While such systems allow for more intimate communications to others, doing so also similarly sacrifices privacy of the sender while also prove more difficult for users to use with strangers. These systems support my proposed system's motivation of richer anonymous online communications.

Although gestures are one of the most ubiquitous and expressive forms of human communication [29], there are relatively small number of research works involving gestures and gesture-based interfaces for emotion-enhanced CMC compared to facial expression or touch-based interactions. Therefore, the research with no doubt plays important role in the field of emotion-enhanced interaction.

2.2 Hand Shape-Based Gesture Recognition

Since this thesis work also relies heavily on automatically identify the hand and fingers for the gestures in my approach, it is similarly important to provide an overview of the research area that specialize in hand shape-based gesture recognition. While the works in this research area investigate different approaches, they also generally overlap in the way they pursue these techniques.

- **Hand Detection:** Many of the prior approaches focus on trying to automatically recognizing the location of the hand in complex backgrounds, either through computer vision algorithms or directly from the sensor.

- **Finger Detection:** Once the hand is detected, many of the prior approaches then use the hand's location as context to subsequently detect the fingers.

- **Sensor Technologies:** Many of the prior approaches rely on two types of sensors: RGB video cameras, which use the red-green-blue color spectrum for recording video; and depth sense cameras, which also includes depth of the human user as additional context in user interface interactions.

- **Recognition Techniques:** Many of the prior approaches rely on a combination of machine learning and computer vision algorithms to handle the bulk of the automated hand shape recognition.

- **Real-Time Tracking:** Many of the prior approaches focus on detecting the hand fro real-time tracking data.

2.2.1 RGB Video Camera-Based Approaches

The first group of hand shape-based gesture recognition approaches stem from algorithms that were designed for RGB video cameras. These algorithms rely on purely the color information from video cameras—generally low-cost cameras such as commercial webcams—for generally filtering the hand and fingers from complex backgrounds. For example, work from [40] focuses on a survey of implementations that use machine learning techniques—specifically, artificial neural networks, fuzzy logic, and genetic algorithms—in recognizing hand shapes in gesturing. The authors discovered from the survey that regardless of approach, the majority of researchers relied on fingertip detection at some capacity to improve their approaches in identifying the user's hand in real-time video. Such works include a system that relies on finger tracking and hand pose detecting approaches for recognizing 24 different types of hand poses [41], a system that also includes skin segmentation to supplement hand pose detection and curve-fitting for hand motion detection [25], and a system that relies on a variety of computer vision algorithms and rule-based approaches for recognizing 13 different types of hand poses [42].

Other hand shape-based gesture recognition approaches for RGB video cameras emphasize their efforts on the contour of the hand and its convex hull. One work by Hongyong et al. [43] relies on computer vision and machine learning techniques to segment the skin color, detect the hand's edges, and perform blob tracking of the overall hand, which was then evaluated by performing hand gestures for manipulating a virtual keyboard application. Another work by Siddharth et al. [44] similarly relies on computer vision techniques for detecting the hand's contour and its convex hull for recognizing 7 different hand poses.

Additional approaches for RGB video cameras expand on solely detecting the hand shape and its different poses. One work focuses on a hand shape detection algorithm that can handle recognition quickly using multiple algorithms such as template matching, and can accommodate recognition for 15 different hand poses [28]. Other works pursue the use of geometric shape descriptors in conjunction with conventional machine learning and computer vision techniques to try to provide more sophisticated recognition [45, 46]. One other work also takes into account the user's head in detection for greater context of the hand's location, and was evaluated in a simple application selector program for simple hand poses [47].

One last work that strays from the conventions of RGB video cameras comes from [48], which does not rely on video but instead on scans from a flatbed scanner to detect hands. This work emulates a video snapshot from the hand scan, so that the authors could focus on developing an approach for recognizing different hand poses. While the system relies on a variety of algorithms such as k-means clustering, independent component analysis, and Hausdorff distance for detecting the hand contour in recognition, the approach remains untested for actual real-time video interactions.

2.2.2 Depth Sense Camera-Based Approaches

With the growing accessibility of commercial depth sense cameras, which can also sense depth of the user's body and limbs for richer digital interactions, researchers are taking advantage of these sensors to provide more sophisticated tracking of the hands and fingers beyond the default hand location detection that conventional depth sense cameras provide. Some approaches for detecting such hand shape gestures rely on identifying the contour of the hand and its convex hull. One work relies on hand contour algorithms to recognize 18 different hand poses [49]; another work uses a hand convex hull algorithm to recognize a small set of gestures that was evaluated in a mobile gaming app [50]; the other work uses a hand contour algorithm and dynamic time warping to recognize 23 different hand poses [51].

More detailed hand shape gesture recognition approaches also incorporate detection techniques for other parts of the body to potentially handle more sophisticated hand gesturing interactions. For example, Some works also utilize finger detection for gestures that have richer finger interactions [52, 53, 54, 55], while other works utilize head detection to handle hand gestures which require the location of the user's head for correct context in the domain of Japanese sign language [56].

2.2.3 Hand Pose Usability Design

While most of the prior works in this section for hand shape poses focused on detection, there has also been other work that focuses on the usability design of hand poses in existing signing domains to bring greater insights to designers of automated recognition systems specific to detecting the hand and fingers. One such study investigates 33 different hand poses from the sign language domain for hand pose tasks, and was motivated on how these hand shapes can affect detection of real-world usage of hand gestures [57]. One of the primary takeaways from this study relies on the discomfort level of different hand poses to users, and whose behaviors can better inform designers to better accommodate users for these types of poses during automated recognition.

2.3 Trajectory-Based Hand Gesture Recognition

Expanding even further from hand shape-based gesture recognition approaches that specialize more on recognizing static hand poses are trajectory-based hand gesture recognition approaches that explore recognizing dynamic hand gestures and hand motion trajector. One of the early works for this research area relies on the user wearing a wired dataglove and tracks the user's static hand poses and dynamic hand gestures in a gestural command set called Charade to manipulate interactions for presentation slides [58].

As computer vision and machine learning algorithms became more sophisticated for non-invasive detection of a user's hand, researchers began investigating how RGB cameras can be leveraged to recognize their dynamic hand gestures and hand motion trajectory as well. For example, Tran et al. implemented their own skeletal joint tracking on the user's upper body motion tracking such as their hands to recognize a set of six dynamic hand gestures [59], Bhuyan et al used a variety of features to recognize different hand poses that changes the interaction type for when the hand is motioning in mid-air [60, 61], Mckenna used hidden Markov models, moment features, and template matching on users' skin color for modeling hand trajectoriesby [62], and Singha provided a heuristic-based approach for dynamic hand gestures on 40 gesture classes solely without depth information [63] .

With the introduction and growing ubiquity of reliable depth sense cameras, the inclusion of depth information enables researchers to rely on these sensors for global skeletal joint motion trajectories such as for the hand. By having these sensors automatically handle the calculations for retrieving these global hand motion trajectories, approaches that utilize such sensors focus on more sophisticated dynamic hand gesturing with motion trajectories. For example, work focuses on techniques for recognizing more dynamic signs in Indian sign language [64], work focuses on techniques that recognize hand gestures with richer motion trajectories for a video gameby [65], and work focuses on more local hand motion trajectories to supplement the depth sensor's existing automated detection for global hand motion trajectories [66].

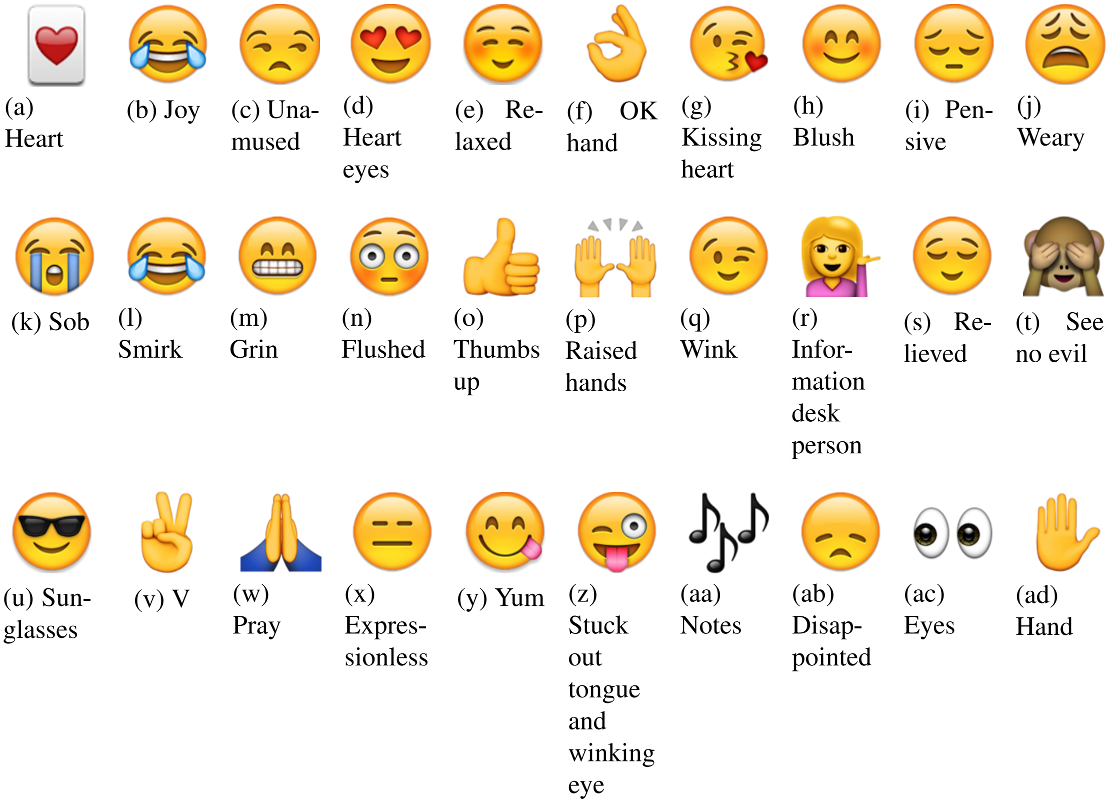
3. IDENTIFYING SYMBOLIC HAND GESTURES TO ANALOGOUS EMOJI

The user-defined design is a cornerstone of human-computer interaction. Conceivably, one could design a system in which all commands were executed with gestures, but not only would this be difficult to learn [67], it would also be inaffective since users are not designers. Care must be taken to elicit user behavior profitable for design. This section describes the approach to developing a user-defined gesture set through three experiments. Prior to these experiments, I select 30 emojis that are popular, emotion-related or show hand gesture from the literatures [68, 69, 70]. With the 30 emojis, I have conducted the first and the follow-up experiments to elicit gestures to analogous emoji. The last experiment was conducted to evaluate and compare these gestures based on their performance.

3.1 Gesture Collection: Maximizing high intuition with participants

It is possible to design a highly intuitive gesture set by acquiring gestures from participants. In this study, participants are first recruited to create symbolic hand gestures for specific emojis. The more participants, the more likely the resulting symbolic hand gesture set will be intuitive to external users. In other words, if more people propose the same gesture for an emoji, the gesture is more intuitive than other proposed gestures. The goal is to obtain commonly used and intuitive hand gestures from which to create the resultant symbolic hand gesture set.

With 10 participants and 30 emojis, 300 gestures are made with one hand or two hands. Participants create their hand gesture for each emoji in turn. They are given the instruction of the initial and the final states, and the image of an emoji. They have unlimited time to create a gesture for the emoji. Once they have decided on a gesture, they reproduce it once to be video-recorded by the experimenter. To avoid bias [71], no specific application domain was assumed.



(a) Heart (b) Joy (c) Unamused (d) Heart eyes (e) Relaxed (f) OK hand (g) Kissing heart (h) Blush (i) Pensive (j) Weary

(k) Sob (l) Smirk (m) Grin (n) Flushed (o) Thumbs up (p) Raised hands (q) Wink (r) Information desk person (s) Relieved (t) See no evil

(u) Sunglasses (v) V (w) Pray (x) Expressionless (y) Yum (z) Stuck out tongue and winking eye (aa) Notes (ab) Disappointed (ac) Eyes (ad) Hand

Figure 3.1: 30 Popular Emojis

For each emoji, identical gestures are grouped together. An emoji can have more than one group. The group with the largest size is then chosen to be the symbolic gesture for the emoji for our user-defined gesture set. A scoring function determines how gestures are commonly consistent. The score should be 1 when proposed hand gestures are identical, and ≈0 when they are unique. Equation 3.1 expresses this as a function.

$$score_{group_i} = \frac{Size\, of\, group_i}{total\, number\, of\, groups_{emoji}} \tag{3.1}$$

For example, in 10 hand gestures for an emoji *e*, if 7/10 are of one form, 2/10 are of another, and 1/10 of are of a third, the gesture with the score 7/10 would be most probably assigned to *e*.

Figure 3.2 shows the highest score on each emoji. As shown in the graph, the scores of 'OK hand,' 'Thumbs up,' 'See no evil,' 'Peace sign hand,' and 'Pray' emojis are 1. This means that the proposed gestures were identical for those emojis respectively. Since most of the emojis describe hand gestures, it is easy for the participants to get an idea of creating a hand gesture directly from the emoji.

3.2 Gesture Selection: Achieving high agreement with participants

This experiment was designed to develop the final user-defined gesture set in light of the agreement participants exhibited in choosing gestures for each emoji. 17 Participants consist of the 10 people who participated in the first procedure and 7 people who did not. They are asked to pick the best match for the emoji's intended meaning while watching videos of the proposed hand gestures for each emoji. The videos do not include any audio, and people's faces in the videos are blotted out by a blur technique to make participants focus solely on the gesture itself without facial expressions. The videos on the survey last from one to three seconds and are played on a loop while participants answer the question. Figure 3.3 shows a screenshot of a part of the question.

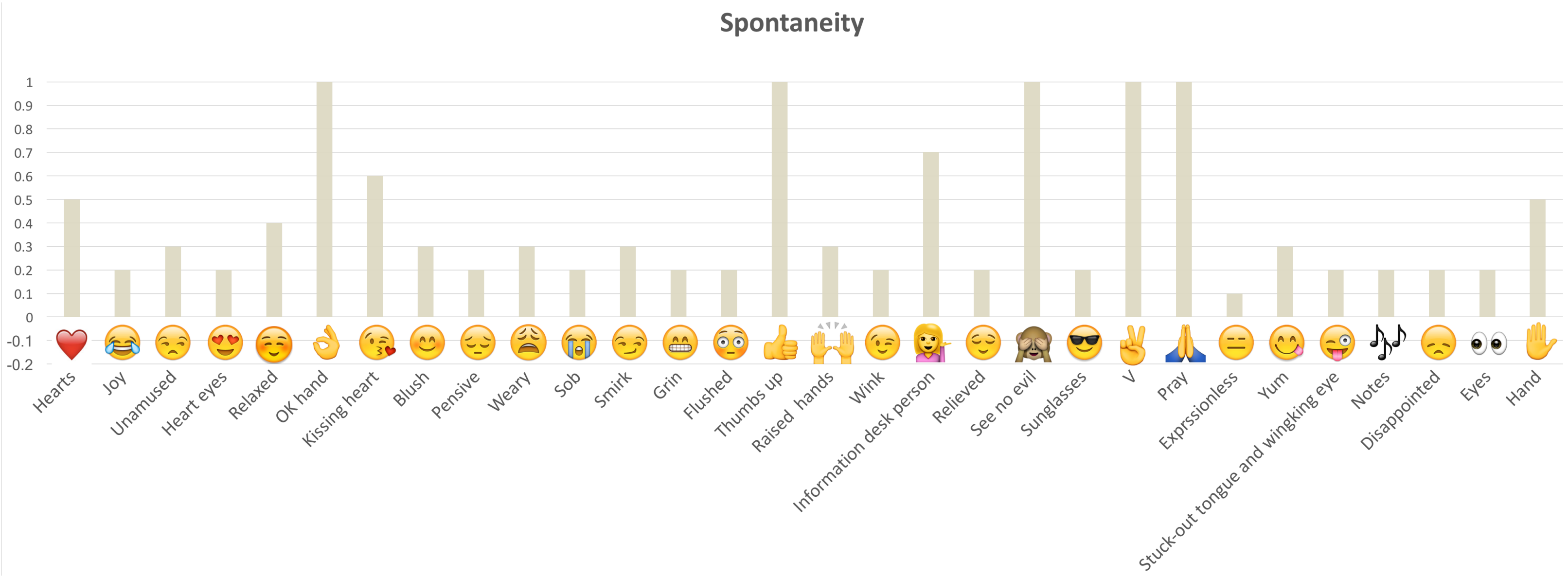
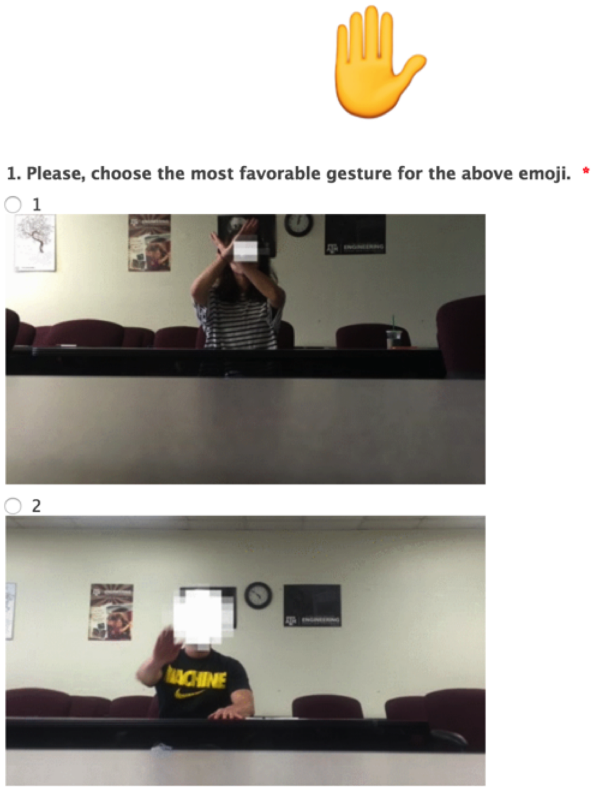


Figure 3.2: The degree of intuition



1. Please, choose the most favorable gesture for the above emoji. *

○ 1

○ 2

Figure 3.3: A part of a gesture selection question

To evaluate the degree of consensus among the participants, I adopted the process of calculating an agreement score for each emoji [1] in equation 3.2.

$$A = \frac{\sum\limits_{e \in E} \sum\limits_{P_i \subseteq P_e} (\frac{|P_i|}{|P_e|})^2}{|E|} \tag{3.2}$$

In equation 3.2, e is an emoji in the set of all emojis E, P_r is a set of proposals for emoji e, and P_i is a subset of identical symbols from P_r. The range for A is $[|P_r|^{-1}, 1]$. For instance, in 10 proposals for emoji e_1 and e_2 respectively, if e_1 has three different forms—5/10 are of one form, 3/10 are of another, and 2/10 are of a third, and e_2 has two different forms—6/10 are one form, and 4/10 are of another, then the agreement of e_2 $([(6/10)^2 + (4/10)^2]/1 =$

0.52) is higher than that of e_1 ($[(5/10)^2 + (3/10)^2 + (3/10)^2]/1 = 0.38$).

I applied the process to three data: 1) data from gesture collection, 2) data from gesture selection only with the 10 people who participated the both experiment, and 3) data from gesture selection with all the 17 participants including 7 people who have not experienced the first experiment. Figure 3.4 shows the results of Wobbrock's agreement [1] on the three data. As the gestures scored 1 from the gesture collection experiment are excluded in the second experiment, no values are shown in the second and the third data. In general, the two data from gesture selection acquire higher agreement than the data from gesture collection. Particularly, compared gesture collection data and gesture selection data from the 10 participants involved in the both experiments, the agreement of the latter is mostly higher than of the former. This is because the participants find out that another gesture proposed by others is more associated with an emoji than the gesture created by their own for the emoji in the first experiment. Moreover, the third data from the 17 participants also has a higher agreement than the first data. Through this result, we conclude that expressibility is not necessarily intuitive and is more influential for mapping gestures to emojis. Therefore, this demonstrates that the result of the second experiment is more reliable for building the final user-defined gesture set.

Deciding a gesture for an emoji in this experiment is similar to the method of the previous experiment. The user-defined gesture set is developed by taking gestures with the most votes for each emoji and assigning those gestures to the emoji. The final gesture sets of the first experiment and the second experiment are partially different. However, the second experiment achieves higher agreement.

Furthermore, interestingly, a high degree of consistency exists in our user-defined set. The same gestures are reselected for similar emojis. In the gesture set, there are four cases in which the same gesture has been used to indicate different emojis. Figure 3.6 describes the four cases. For example, the gestures for 'Relieved,' 'Blush,' and 'Relaxed' emojis

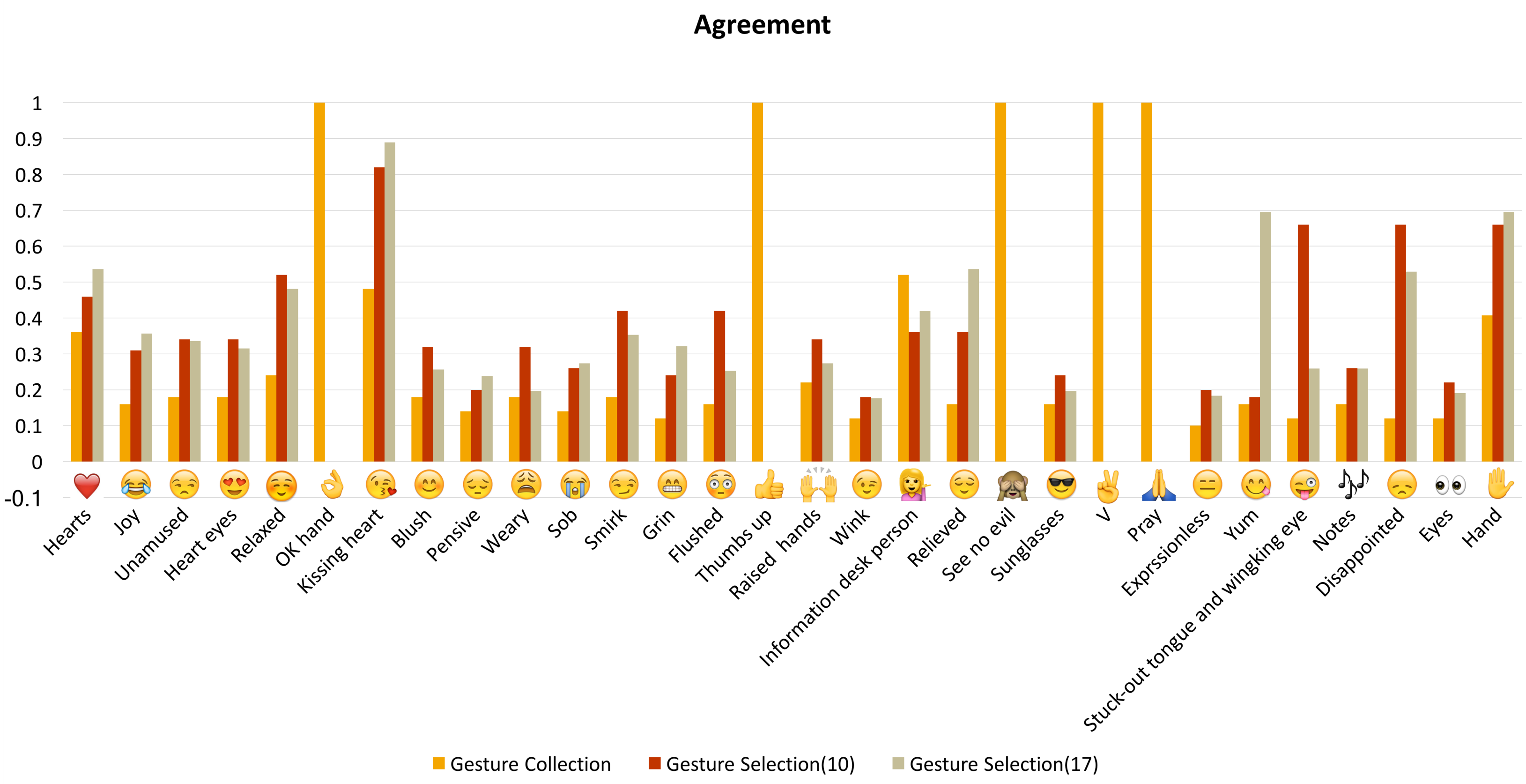
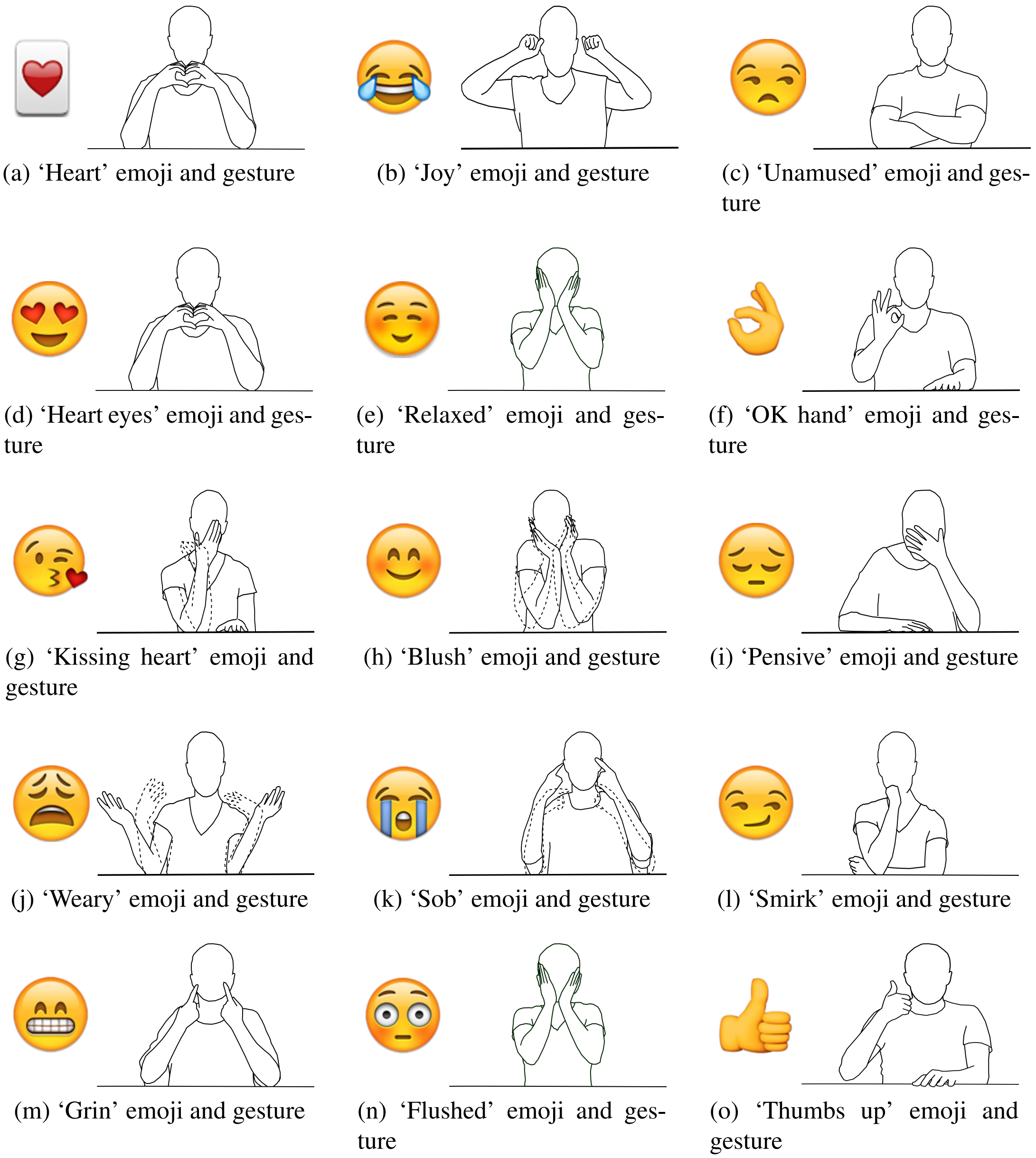


Figure 3.4: Wobbrock's agreement [1] on the three data: data from gesture collection, data from gesture selection only with the 10 people who participated the both experiment, and data from gesture selection with all the 17 participants



(a) 'Heart' emoji and gesture

(b) 'Joy' emoji and gesture

(c) 'Unamused' emoji and gesture

(d) 'Heart eyes' emoji and gesture

(e) 'Relaxed' emoji and gesture

(f) 'OK hand' emoji and gesture

(g) 'Kissing heart' emoji and gesture

(h) 'Blush' emoji and gesture

(i) 'Pensive' emoji and gesture

(j) 'Weary' emoji and gesture

(k) 'Sob' emoji and gesture

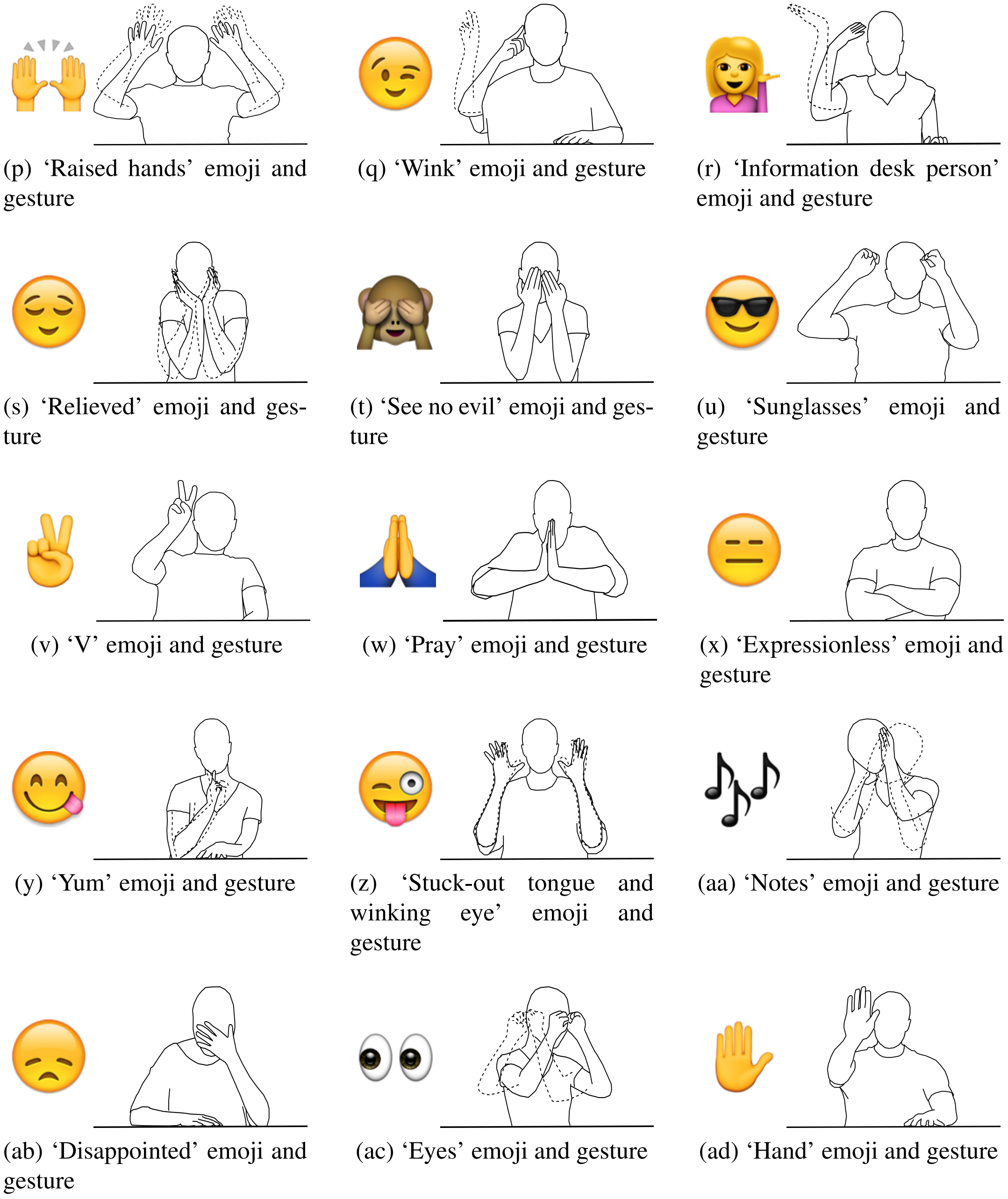
(l) 'Smirk' emoji and gesture

(m) 'Grin' emoji and gesture

(n) 'Flushed' emoji and gesture

(o) 'Thumbs up' emoji and gesture

Figure 3.5: A user-defined gesture set



(p) ‘Raised hands’ emoji and gesture

(q) ‘Wink’ emoji and gesture

(r) ‘Information desk person’ emoji and gesture

(s) ‘Relieved’ emoji and gesture

(t) ‘See no evil’ emoji and gesture

(u) ‘Sunglasses’ emoji and gesture

(v) ‘V’ emoji and gesture

(w) ‘Pray’ emoji and gesture

(x) ‘Expressionless’ emoji and gesture

(y) ‘Yum’ emoji and gesture

(z) ‘Stuck-out tongue and winking eye’ emoji and gesture

(aa) ‘Notes’ emoji and gesture

(ab) ‘Disappointed’ emoji and gesture

(ac) ‘Eyes’ emoji and gesture

(ad) ‘Hand’ emoji and gesture

Figure 3.5: Continued

are practically the same. This is because the three emojis relatively resemble each other. Another case is the gestures for ‘Unamused’ and ‘Expressionless.’ Although the two emojis do not look alike, the meaning of the emojis is similar. When facing these questions, one might try to answer the questions consistently. However, participants did not intend to make a consistent answer for the questions nor reverse their decision, but they followed their instinct. Hence, these cases can be evidence that the final user-defined gestures and emojis are highly correlated. The gestures resulting from the second experiment in Figure 3.5 will be used in further studies.

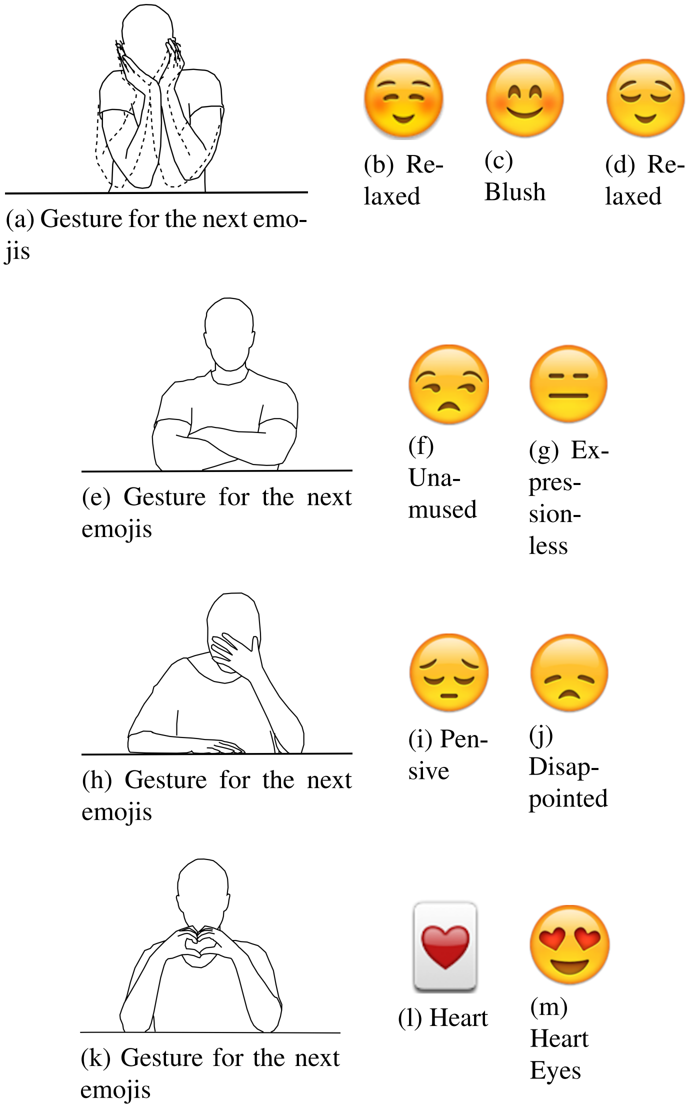
3.3 Gesture Evaluation

This experiment gives some of the performance measures and preference ratings for gesture articulation. Participants who have joined neither of the previous two experiments were recruited and informed of the specific application domain. The subjects watch the video of a user-defined gesture for an emoji then are asked to follow the gesture two times on cue. Immediately after performing the gesture, they answer a multiple choice question that asks them to pick an emoji that is the best match for the gesture as shown in Figure 3.7. Out of the five choices, the emoji mapped onto the gesture always exists, and other four emojis are randomly selected. Then, the subjects are asked to rate on both modes of chat, gesture-driven and search-driven for sending the emoji, on six seven-point Likert scales on the following criteria: intuition, intimacy, expressiveness, enjoyability, efficiency and ease. The possible responses ranged from 1 (low) to 7 (high) (9 subjects).

Wobbrock at el. define the guessability in symbolic input (for our study, hand gestures) as:

> The quality of symbols which allows a user to access intended referents via those symbols despite a lack of knowledge of those symbols.

A user’s initial attempts at performing gestures must be met with success despite the



(a) Gesture for the next emojis

(b) Relaxed

(c) Blush

(d) Relaxed

(e) Gesture for the next emojis

(f) Unamused

(g) Expressionless

(h) Gesture for the next emojis

(i) Pensive

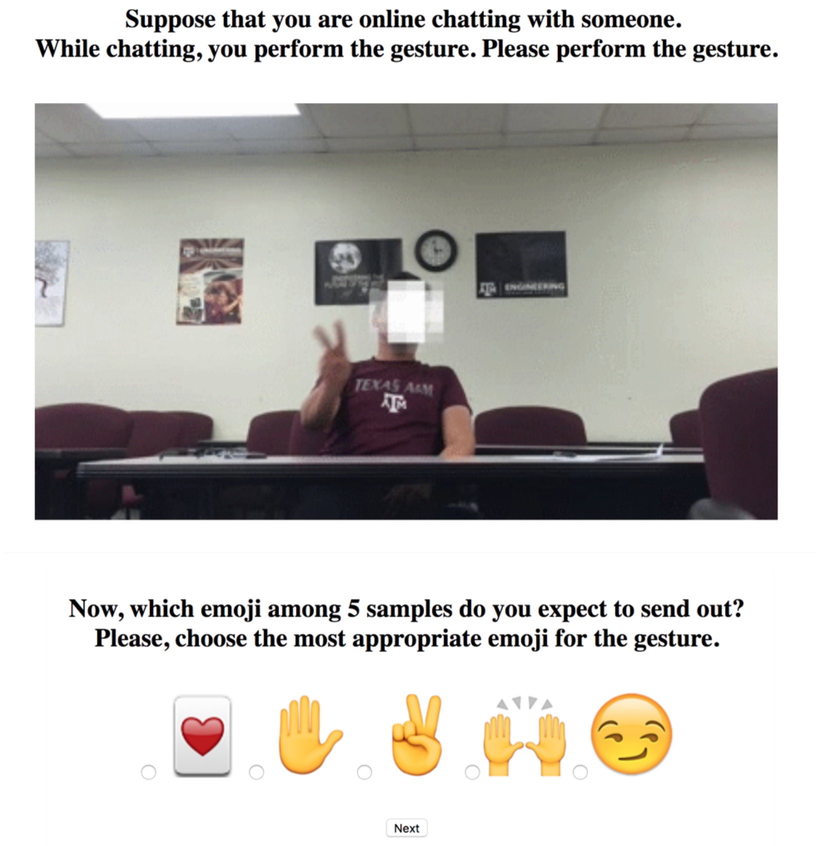
(j) Disappointed

(k) Gesture for the next emojis

(l) Heart

(m) Heart Eyes

Figure 3.6: The association between emojis and gestures



Suppose that you are online chatting with someone.
While chatting, you perform the gesture. Please perform the gesture.

Now, which emoji among 5 samples do you expect to send out?
Please, choose the most appropriate emoji for the gesture.

○ 🂱 ○ ✋ ○ ✌️ ○ 🙌 ○ 😏

Figure 3.7: A part of a gesture evaluation question

user's lack of knowledge of the usage. This requires high guessability. Even though they have their own formula for the gusessability, I think one measurement for the degree of the guessability can be also f-measure (detailed discussion in Section 4.2.3). Thus, with the results from 9 participants, I use f-measure to show shown inFigure 3.8. The total f-measure is 0.79.

The graph of the preference rating data shown in Figure 3.9 helps us to understand the impact of the user-defined gesture set. Overall, the average scores of gesture-driven mode are higher than those of search-driven mode on all criteria especially enjoyability. In terms of standard deviation, gesture-driven mode is superior to search-driven mode on three criteria, ease, enjoyability, and intuition.

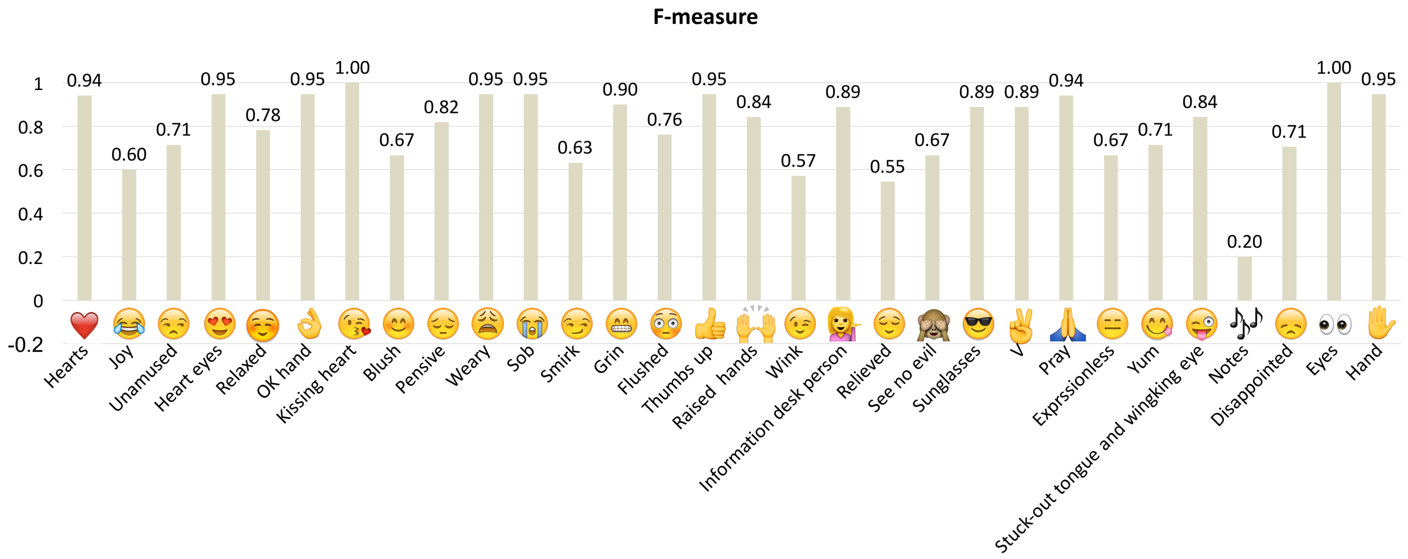


Figure 3.8: F-measure on the final gesture set with 9 participants

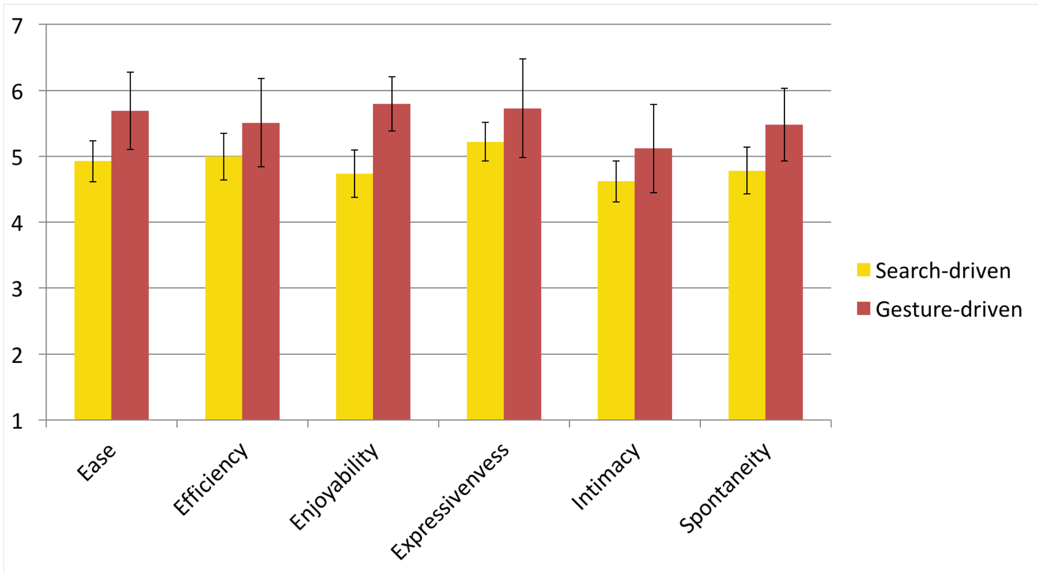


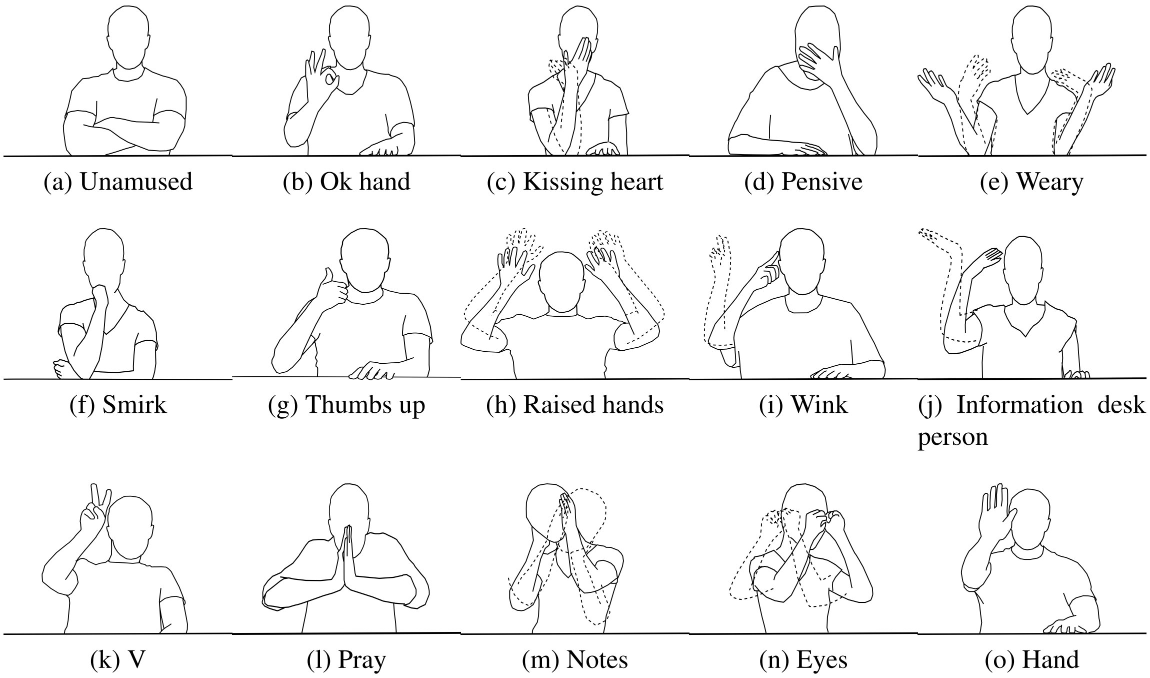
Figure 3.9: Comparison between search- and gesture-driven emojis

4. HAND GESTURE RECOGNITION SYSTEM

Hand gestures can be classified into two types: static and dynamic gestures. Static hand gestures are defined as orientation and position of the hand in the space during an amount of time without any movement, and this is often called pose. On the contrary, dynamic gestures are defined as orientation and position of hand in the space if a movement exists during the time above. For instance, the waving of a hand is considered as a dynamic gesture, while a 'thumbs up' pose is a static gesture.

I use 15 gestures, a subset of our user-defined gesture set, which consist of nine single-handed gestures—'OK hand,' 'Thumbs up,' 'V,' 'Hand,' 'Information desk person,' 'Smirk,' 'Wink,' 'Pensive,' and 'Kissing'—or sex two-handed gestures—'Unamused,' 'Weary,' 'Raised hands,' 'Pray,' 'Eyes,' and 'Notes'—for the proposed hand gesture recognition system. The gestures and the corresponding emojis are shown in Figure 4.1. Among them, the 5 gestures, 'OK hand,' 'Thumbs up,' 'V' (peace sign), and 'Hand' (hi gesture), and 'Information Desk Person' are static.

As the goal of our recognition system is to distinguish the 15 gestures that are either static or dynamic, the hand gesture recognition system should enable to detect them. Thus, I implement a two-stage recognition system: trajectory- and shape-based recognition. To detect dynamic gestures, the feature-based recognition is used, and for static gestures, the shape-based recognition is used. First, the trajectory-based recognition categorizes the 15 different gestures in 11 groups because the five static gestures are grouped as called 'Hand Shape' due to their similar movement. Second, these gestures in the 'Hand Shape' group are run through the shape-based recognition in order to distinguish the five gestures with their hand shape.



(a) Unamused (b) Ok hand (c) Kissing heart (d) Pensive (e) Weary

(f) Smirk (g) Thumbs up (h) Raised hands (i) Wink (j) Information desk person

(k) V (l) Pray (m) Notes (n) Eyes (o) Hand

Figure 4.1: 15 user-defined hand gestures

4.1 Data Collection

I used Microsoft Kinect, a depth camera, to capture hand gestures as shown in Figure 4.2. Instead of a data glove, the Kinect has been widely used in computer vision research due to its natural, intuitive operation and low cost. The sensor can detect and segment hands robustly. However, due to the low-resolution of the Kinect depth camera of 640 * 480, it does not work well when tracking a small object, e.g., a hand with complex articulations. To address the problem, I present a shape-based recognition (detailed discussion in Section 5).

Based on the assumption that a user sits in front of a computer and performs one of our user-defined gestures to send an emoji while chatting online, only the user's upper body movement data is necessary. With using the Kinect SDK 2.0[1] that provides body-tracking

[1] https://msdn.microsoft.com/en-us/library/dn799271.aspx

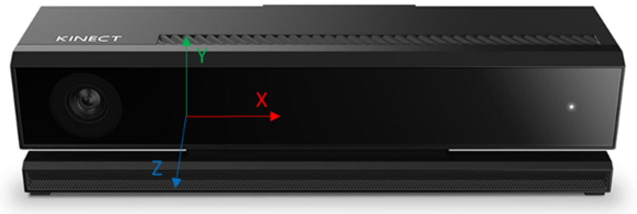


Figure 4.2: The image of the Microsoft Kinect. Reprinted from [2]

methods for the Kinect, I collect the upper body joints, e.g., head, neck, shoulder-spine, mid-spine, right and left shoulders, right and left elbows, right and left wrists, right and left hands, right and left hand tips, and face points, e.g., forehead, eyes, nose, cheeks, mouse and chin. The Kinect is mounted on the place where could cover overall motions of a user. Figure 4.3 is the graphic user interface (GUI) that was developed in C# for interaction with the Kinect. The data collection is captured by the Kinect mounted in upper front of the user. The big dots indicate the upper body joints, and the small red dots are face points. Once the Kinect starts tracking the joints and face points, the dots are displayed on the screen. Also, when the GUI detect the data points, the start button in the GUI is activated. If the start button is clicked, the button is changed to stop button and the user's gesture is recorded. If the stop button is clicked, the recording is stopped and the gesture data has been saved.

The Kinect produces a sequence of depth images at 30 frames per second, and the sequence of depth image frames indicates a hand gesture. The gesture has approximately 60–100 frames as our user-defined gestures take two or three seconds for each. The each frame contains the values of x, y and z coordinates of the upper body joints and the face points i.e. the position of the hand in 3D space. As shown in figure 4.2, the origin ($x = 0$, $y = 0$, $z = 0$) is the center of the IR sensor on the Kinect; x grows to the sensor's left, y grows up, and z grows out in the direction of the sensor is facing. The data of the gesture are derived in the Cartesian coordinate space directly from the gesture frames, and saved

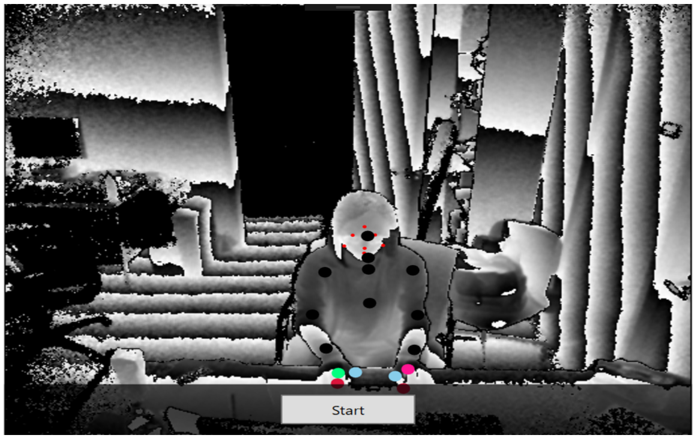


Figure 4.3: The graphic user interface of data collection

as an XML file.

21 subjects participated in this study. In the case of two-handed gestures, they were asked to do a gesture twice. In the case of single-handed gestures, they were asked to do a gesture four times, twice with the right hand and twice with the left hand. The total number of collected gestures is 1008, but the number of processed data is 988 due to detection failure.

4.2 Trajectory-based Recognition

In this section, I describe features, the implementation of recognition system, and the results of distinguishing between the 11 hand gestures in our user-defined gesture set.

4.2.1 Proposed features

Good feature selection plays an important role in hand gesture recognition performance. For matching the trajectory, 44 traditional features are adapted from different kinds of literatures [72, 73, 74, 63]. These features are derived for training. In Table 4.1, the extracted features are shown. These features are inputted into several classification algorithms with 10-fold cross validation.

Adapted Traditional Features
Average (X, Y, Z) (Left/Right), 6
Standard Deviation (X, Y, Z) (Head/Left/Right), 9
Min/Max (X, Y, Z) (Left/Right), 12
The length of the bounding box 3D diagonal (Head/Left/Right), 3
The angle of the bounding box (Head/Left/Right), 3
The ratio of the bounding box volume of hands, 1
The length of 3D gesture (Head/Left/Right), 3
Average velocity (Left/Right), 2
Maximum speed (Head/Left/Right), 3
The total traversed angle (Left/Right), 2

Table 4.1: 44 Adapted Traditional Features

4.2.2 Recognition algorithms

The 44 extracted traditional features are tested on five supervised classifications provided by the WEKA[2], the most popular machine learning algorithm library: decision tree, k-nearest neighbor, naive Bayes, multilayer perceptron, and random forest. To avoid overfitting to the training data, I used 10-fold cross-validation that evaluates predictive model by partitioning data into 10 equal sized subsamples and testing the model 10 times with using a single subsample as test data and 9 remaining subsamples as training data.

4.2.2.1 Decision tree

A decision tree is a simple machine learning algorithm that creates a tree-like structure as a predictive model with training data. The completed tree classifies test data. Due to the simplicity, the decision tree can elucidate the underlying relationships in the data.

[2] http://www.cs.waikato.ac.nz/ml/weka/

However, as the decision tree is susceptible to overfitting when training data has noise, pruning the data must be required. I tested J48, the implementation of algorithm ID3 (iterative Dichotomies 3).

4.2.2.2 K-nearest neighbors (kNN)

kNN is used for classifying data based on closest training examples in the feature space. It is a type of lazy learning where the process is only estimated locally, and all computation s are deferred until classification. An instance is classified based on the majority vote of its k-nearest neighbors. It is helpful to assign k to an odd number because this prevents tied votes [75]. However, it is challenging to find the optimal value of k. I tested this classifier with setting up the value of k to be the odd numbers from 1 to 9. For my case, the highest accuracy amongst the trials was achieved when k is 1. It is a simple classification that test instances are assigned to the group of the closest neighbor.

4.2.2.3 Naive Bayes

Naive Bayes is a probabilistic classifier based on applying Bayes' theorem in equation 4.1 with the assumption that features are independent. Even if features can be correlated, naive Bayes lets all of the features independently contribute to the probability of classification, and this is why it is known as 'Naive.' In spite of the oversimplification, the classifier works well on large data sets and is easy to build. However, one drawback of naive Bayes is "Zero Frequency." For example, if test data set has a new class that is not observed in training data, the model are unable to make a prediction since the model will assign a zero probability.

Equation 4.1 seeks the posterior probability of $Class_i$ given predictor x from 1) the prior probabilities of the class ($P(Class_i)$) and the predictor ($P(x)$), and the probability of the predictor given the class ($P(x|Class_i)$).

$$P(Class_i|x) = \frac{P(Class_i)P(x|Class_i)}{P(x)} \tag{4.1}$$

4.2.2.4 Multilayer perceptron

The multilayer perceptron consists of an input layer, an output layer, and one or more hidden layer. The multilayer perceptron is a feed forward artificial neural network by training a model with backpropagation. This was introduced to find non-linear pattern in data since initially standard perceptron can only work well on finding linear patterns.

4.2.2.5 Random forest

Random forest is an ensemble learning method that builds a large number of decision trees with random sample sets and exploits bootstrap aggregating, so called bagging [76], with the decision trees for training. As random forest resolves the decision trees' habit of overfitting as mentioned in the section 4.2.2.1, it has been widely used in various problems, consistently achieving high degrees of accuracy [77]. I tested random forest with 100 decision trees.

4.2.3 Result

To evaluate the performance of these algorithms in distinguishing between user-defined gestures, I used accuracy and F-measure.

In terms of binary classification that has two possible labels, "positive" and "negative", there are the four following possible cases: 1) true positives (TP), the number of positives labeled as positive, 2) true negatives (TN), the number of negatives labeled as positive, 3) false positives (FP), the number of positives labeled as negatives, and 4) false negatives (FN) labeled as positive. Accuracy is formulated with the possible cases in equation 4.2. If TP < FP, then accuracy will may increase when a classification rule is changed to always output negative category. Conversely, If TN < FN, the same may happen when a classification rule is changed always to output positive. The idea that some algorithms with less predictive power may have higher accuracy is called accuracy paradox.

To avoid this problem, we instead use recall as defined in Equation 4.3, precision as defined in Equation 4.4 [78], and f-measure as defined in Equation 4.5. Recall is the number of true positives divided by the number of all positives. Precision is the number of true positives divided by the number of all true positives and false negatives. F-measure is the harmonic mean of recall and precision, and this is widely used as a measure of accuracy in statistical analysis of binary classification.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.2}$$

$$Recall = \frac{TP}{TP + FP} \tag{4.3}$$

$$Precision = \frac{TP}{TP + FN} \tag{4.4}$$

$$F\text{-}measure = 2 \cdot \frac{1}{\frac{1}{Recall} \cdot \frac{1}{Precision}} = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} = \frac{2TP}{2TP + FP + FN} \tag{4.5}$$

To evaluate the effectiveness of our features, I tested their ability to distinguish between 11 gestures. Our features were run through each of the classifiers with 10-fold cross-validation. Cross-validation is one of model validation techniques to asses results of statistical analysis. This is used to determine how accurately a predictive model (e.g., features, shape context) performs in several experiments. I used 10-fold cross-validation

Classifier	Accuracy	F-measure
J48	70.11%	0.698
kNN (k=1)	68.08%	0.674
Naive Bayes	66.26%	0.665
Multilayer perceptron	85.61%	0.856
Random Forest	80.75%	0.792

Table 4.2: The performance of classifier for identifying 11 gestures using the 44 adapted traditional features

that evaluates the predictive model by partitioning data into 10 equal sized subsamples and testing the model 10 times with using a single subsample as test data and 9 remaining subsamples as training data.

The results are shown in Table 4.2. The 44 adapted traditional features are able to distinguish between 11 gestures fairly accurately.

4.2.4 Novel features

I introduce additional features that are specified to our user-defined gesture set in an effort to improve the recognition of 15 user-defined gestures. As can be seen in Table 4.3, 33 features are added by observing and quantifying differences in the data between the user-defined gesture set.

These features are divided into two groups: conditional or non-conditional. Conditional features are extracted given a certain condition. I made 3 assumptions: 1) when the hand is in the highest position, 2) when the hand is closest to the body (or the face), and 3) when the distance between two hands is the shortest. Mainly, the extracted features under the conditions are x, y, and z differences between two hands or between the hand and the head whereas non-conditional features are the distances between the hand and the head,

Novel features
The longest distance between the head and the hand (Left/Right)
The shortest distance between the head and the hand (Left/Right)
The average distance between the head and the hand (Left/Right)
The longest distance between hands
The shortest distance between hands
The average distance between hands
The sub-length of 3D gesture
The sub-length of 2D gesture (XY)
The sub-length of 2D gesture (XZ)
Condition 1. When y is the highest,
The y difference between the hand and the head (Left/Right)
The x difference between the hand and the head (Left/Right)
The y difference between hands
The x difference between hands
Condition 2. When z is the furthest from the sensor,
The y difference between the hand and the head (Left/Right)
The x difference between the hand and the head (Left/Right)
The y difference between hands
The x difference between hands
Condition 3. When the shortest distance between two hands,
The y difference between the hand and the head (Left/Right)
The x difference between the hand and the head (Left/Right)
The z difference between the hand and the head (Left/Right)

Table 4.3: 33 novel features

the distances between two hands, the sub-lengths of gesture.

Additionally, the sub-length of gesture is the length of gesture that includes the trajectory of the hand above the neck. The sub-length are extracted in three dimensional and two dimensional spaces.

4.2.5 Result

The classification performance of algorithms trained on these features with 10-fold cross-validation is shown in Table 4.4. Table 4.5 is the confusion matrix. Table 4.6 shows the f-measure of the 11 gestures. These results shows that when applying additional features to the classification, a marked improvement in recognition performance occurred. For example, with random forest, the accuracy improved from 80.75% with the original features to 92.40% with the new features; the f-measure increased from 0.792 to 0.922. Although naive Bayes shows the less improvement, I presume the low performance of this classifier is not because our features performed poorly but because these features, especially the new features, are highly correlated to each other. As mentioned in the previous Section 4.2.2.3, the naive Bayes requires the assumption of the independence of features. Evaluating the effect of the new features on recognizing 11 gestures is as clear cut. The result demonstrates that the addition of new features is more sufficient to reliably discriminate the 11 gestures.

Classifier	Accuracy	F-measure
J48	81.05%	0.811
k-NN (k=1)	89.66%	0.896
Naive Bayes	69.30%	0.711
Multilayer perceptron	92.50%	0.925
Random Forest	92.40%	0.922

Table 4.4: The performance of classifier for identifying 11 gestures using 77 features (44 traditional + 33 novel)

Activity	Classified As										
	Eyes	Hand Shape	Kissing Heart	Notes	Pensive	Pray	Raised Hands	Smirk	Unamused	Weary	Wink
Eyes	100	0	0	0	0	0	0	0	0	0	0
Hand Shape	0	99.28	0	0	0.24	0	0.24	0	0	0.24	0
Kissing Heart	0	9.52	78.57	0	1.19	0	0	10.71	0	0	0
Notes	2.38	0	0	97.62	0	0	0	0	0	0	0
Pensive	0	6.41	0	0	85.90	0	0	2.56	0	0	5.13
Pray	0	0	0	0	0	92.86	0	0	7.14	0	0
Raised Hands	0	0	0	0	0	0	100	0	0	0	0
Smirk	8.54	9.76	0	7.32	0	0	0	74.39	0	0	0
Unamused	0	0	0	0	0	4.85	0	0	95.12	0	0
Weary	0	2.38	0	0	0	0	2.38	0	0	95.24	0
Wink	0	18.18	0	0	0	0	0	0	0	0	81.82

Table 4.5: Confusion matrix for distinguishing between the 11 gestures using the random forest and combining of the traditional features and the novel features

Gesture	**F-measure**
Eyes	0.988
Hand shape	0.956
Kissing heart	0.835
Notes	0.988
Pensive	0.876
Pray	0.940
Raised hands	0.976
Smirk	0.792
Unamused	0.940
Weary	0.964
Wink	0.875

Table 4.6: F-measures of the 11 gestures

5. SHAPE-BASED RECOGNITION

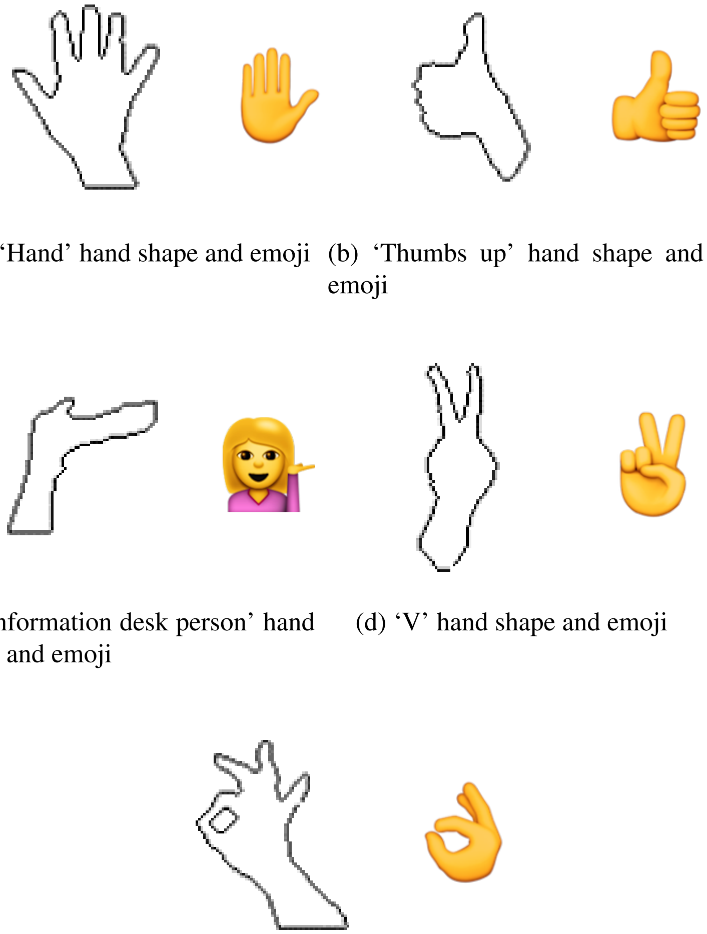
In this chapter, I describe our proposed shape-based recognition that employs four basic steps: pose segmentation, hand segmentation, hand shape tracing, and hand shape classification. As a reminder, in the trajectory-based hand gesture recognition, five shapes have been grouped into a single activity 'Hand Shape.' The shape-based recognition classifies those five gestures that were grouped into 'Hand Shape.' Images of the five different hand shapes are shown in Figure 5.1.

5.1 Pose segmentation

A gesture is a sequence of poses and is saved as a sequence of frames as mentioned in the previous Section 4.1. In order to capture the certain pose considered as one of the static gestures within the frames, two conditions should be satisfied. First, the algorithm finds the highest y coordinate of the hand joint. Second, the algorithm checks to see if the number of the pixels of the contour in the frame is equal or greater than a threshold. (extracting contour will be explained in Section 5.3.) In our case, the threshold is 100. Unless the second condition is satisfied, the algorithm searches either the next frame or the previous frame based on the index of the currently selected frame in the sequence. To be specific, if the current index is in the middle or the second half of the frame sequence, the algorithm searches the previous frame. On the other hand, if the current index is in the first half, the algorithm find the next frame. The algorithm keeps searching until the threshold condition is met.

5.2 Hand segmentation

The hand segmentation aims to extract the hand region from a depth image. Since the Kinect SDK 2.0 provides methods to detect the hand joint, the hand tip joint, and the



(a) ‘Hand’ hand shape and emoji

(b) ‘Thumbs up’ hand shape and emoji

(c) ‘Information desk person’ hand shape and emoji

(d) ‘V’ hand shape and emoji

(e) ‘OK hand’ hand shape and emoji

Figure 5.1: The hand shapes

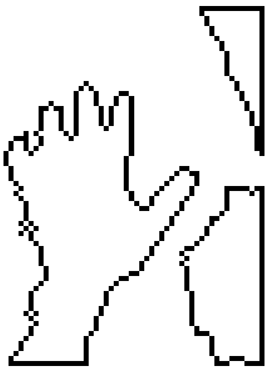


Figure 5.2: The error after the hand segment

thumb joint, the search area of the hand can be narrowed down. First, the length between the hand joint and the hand tip joint, and the length between the hand joint and the thumb joint are calculated. The longer of the two is chosen to be the length of the bounding box of the search area. While scanning the depth values in the area of depth image, any values will be excluded if the values do not fall between the desired range. Thresholding converts the search area to a binary black-and-white image; the pixels that belong to the hands assigned 1 and the remaining pixels are assigned 1. As a result, the shape of the hand can be extracted.

5.3 Hand shape tracing

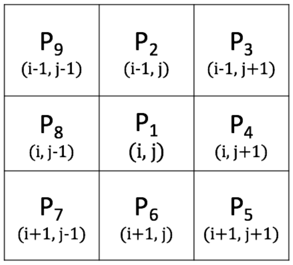
I adapt Rajan's stroke extracting algorithm [3, 79, 80] to trace a hand shape. The hand shape tracing approach is chosen to resolve an unexpected noisy issue caused by the hand segment process. Figure 5.2 is an example of the issue. Likewise, the results of the hand segment process, occasionally, contain more shapes other than a hand shape. One way to discriminate a hand shape from others is to compare the length of the shapes. It highly possible that the shape with the longest length will be the hand shape. Thus, shape segmentation is required, and the hand shape tracing approach enables to segment shapes.

5.3.1 Pre-processing

The preprocessing is to extract the contour information, or the coordinates of the contour, from the hand shape, the outcome of the hand segment. The first step in the preprocessing is to denoising the hand shape images. In reality, the images obtained from the Kinect are often corrupted with noise. As a result, the hand shape usually has noise around the shape outline. Thus, a denoising process is applied. The denoising process removes isolated pixels and isolated small regions or segments. The second step in the preprocessing is to obtaining the shape boundary coordinates. Creating a rough contour of the hand shape, first. Assume that black pixels are denoted as 1's and white pixels are denoted as 0's, if a black pixel that belongs to the shape has adjacent to at least one white pixel in the 8-pixel neighborhood around the black pixel, the algorithm considers the black as a part of the contour of the shape. After finding all the black pixels that meet the criterion, the algorithm thins the contour to a thickness of a single pixel. Otherwise, it is difficult to determine the intended direction of contour. This process is a prerequisite of Rajan's stroke extracting algorithm.

In order to thin the rough contour, I use Zhang-Suen's fast parallel thinning algorithm [81]. Zhang-Suen's thinning algorithm is an iterative algorithm, which eliminates the outer layer of pixel until no more layers can be removed [82]. The algorithm consists of two sub-iteration. The first sub-iteration removes the bottom-right boundary pixels and the top-left corner pixel while the second sub-iteration removes the top-left boundary pixels and the bottom-right corner pixel. In the first sub-iteration, a pixel, P_1 in figure 5.3, is eliminated on the following conditions;

- The number of nonzero neighbors is greater than or equal to 2, or less than and equal to 6
- The number of 01 pattern should be one in the ordered neighbor set, $P_2, P_3, ..., P_9$



P_9 (i-1, j-1)	P_2 (i-1, j)	P_3 (i-1, j+1)
P_8 (i, j-1)	P_1 (i, j)	P_4 (i, j+1)
P_7 (i+1, j-1)	P_6 (i+1, j)	P_5 (i+1, j+1)

Figure 5.3: Designations of the nine pixels in a 3×3 windows

in figure 5.3

- $P_2 \cdot P_4 \cdot P_6 = 0$
- $P_4 \cdot P_6 \cdot P_8 = 0$

In the second sub-iteration, the first two conditions remain same but the last two condition are modified as follow:

- $P_2 \cdot P_4 \cdot P_8 = 0$
- $P_2 \cdot P_6 \cdot P_8 = 0$

5.3.2 Contour tracing

After the contour of a hand shape is thinned into a single pixel thickness, Rajan's stroke extracting algorithm [3] is applied in order to trace the contour. A resultant stroke can be regarded as either a whole contour or a part of the contour in this section. The algorithm traces strokes using both the local and global characteristics of the previously identified stroke points to determine the future points of the stroke.

Initially, a starting point is identified from which to begin stroke extraction. As the starting point, the black pixel closest to the bottom left corner of the hand shape image is selected. The first point belongs in the first stroke and is stored in an array, *stroke history*.

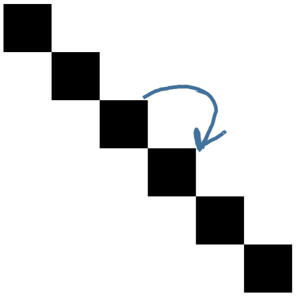


Figure 5.4: The transition when just one pixel is available [3]

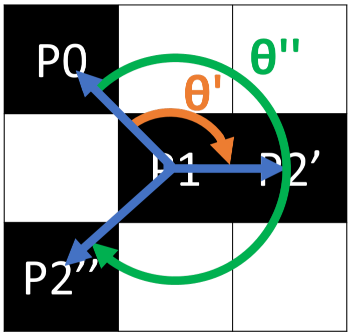


Figure 5.5: Finding minimum angular deviation with the local solver [3]

The *stroke history* maintains the record of the pixel designated as part of the stroke under consideration. The pixels in the array are flagged as 0, which means that these are invisible to the algorithm.

When the algorithm, continuously, explores pixels, it can face either following two cases, the number of nonzero neighbors around the current pixel is one or more than one. The former case is simple to process; the one neighboring pixel is added to the *stroke history* as shown in Figure 5.4. In the latter case, the current pixel is called as a point of ambiguity. Depending on the number of pixels stored in the *stroke history*, either the *local solver* or the *global solver* is employed to select the next pixel. The threshold in this study is 12 (pixels).

The *local solver* and the *global solver* determine the next pixel with minimum angular deviation from the direction given by the last 2 pixels or the last 12 pixels of the *stroke history* respectively. The process of the *local solver* is described in detail in Figure 5.5. It the last pixel and the current pixel are $P0$, $P1$ respectively, the *local solver* finds the angles of $P2'$ and $P2''$, θ' and θ''. On the contrary, Principle Component Analysis (PCA), a well-known statistic pattern recognition, serves as the basis of the global pixel selector. In this study, I used the Accord.NET Framework [83] to implement the PCAs, which are eigenvectors of the covariance matrix of the data of 12 pixels in *stroke history*, which is denoted by equation 5.1. The eigenvectors correspond to the direction with the maximum eigenvalues.

$$Cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{n-1}, \text{ where } n = 12 \tag{5.1}$$

Assume that the PCA of the *stroke history* is denoted as $\overrightarrow{P_h}$, and the PCA of the data of 12 pixel placed in one possible direction, so-called *stroke future*, is denoted as $\overrightarrow{{}_iP_f}$, where i is the number of *stroke futures*. Then, as can be seen in equation 5.2, the dot product of $\overrightarrow{P_h}$ and $\overrightarrow{{}_iP_f}$ is computed. The selected *stroke future* is the maximum value of DP_i.

$$DP_i = \overrightarrow{P_h} \cdot \overrightarrow{{}_iP_f} \tag{5.2}$$

Figure 5.6 is an example of the result of choosing the next point with the *global stroke*. Initial point is the $P0$; current point is $P2$, the chosen next point is $P3$, and the blue arrow indicates the direction of the tracing previous pixels. This is the decision of *global solver* because the local solver would choose the bottom left pixel of $P2$, instead of $P3$.

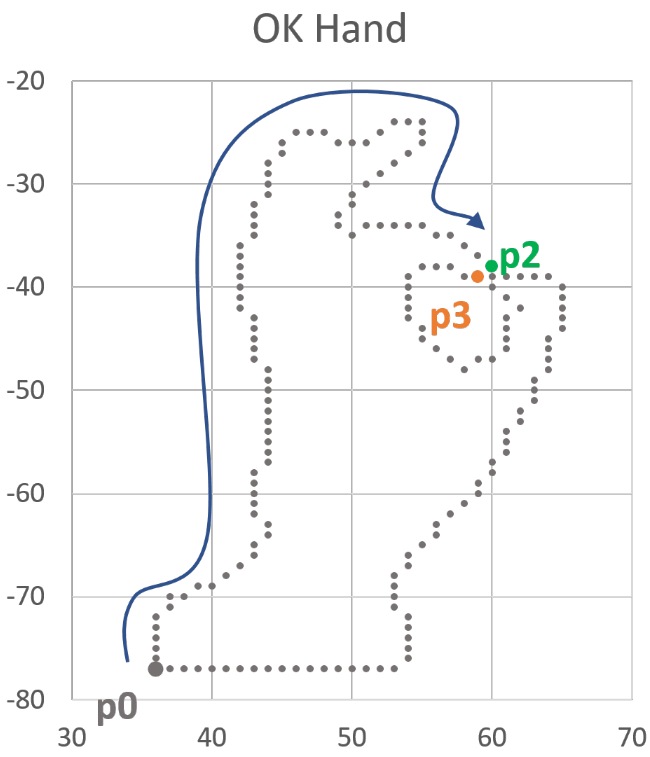


Figure 5.6: A global point of ambiguity in the shape of 'OK hand,' one of the user-defined gestures

5.3.3 Contour selection

Through stroke extraction with the *local solver* and the *global solver*, occasionally, multiple strokes are generated, which compose a contour of a hand shape. Then, these strokes are merged based on the distance between the end points of distinct strokes by a threshold. If the final result has more than one contour, the algorithm finds the longest contour as well as the second longest contour. As illustrated in Figure 5.1e, the second longest contour can be a feature to help identify 'OK hand,' as 'OK hand' has a contour inside the first. With using Ray's Casting algorithm, if our system determines that the second algorithm is located inside of the longest contour, the second longest contour also remains. The left sides of each of the figures in Figure 5.1 show the outcome of the hand shape tracing approach.

5.4 Hand shape classification

To classify the five hand shapes in Figure 5.1, I have explored two approaches: finger detection and shape matching.

5.5 Finger detection

My previous assumption was since the five hand shapes have different number of unfolded fingers, detecting these fingers can lead to classify the gestures. As many researchers have used convex hull [49, 50, 52] or curve detection [84, 85, 86] for fingertip detection, both methods are used in this work.

The convex hull of a set of points is the smallest set of points that contains all the points. Our system calculates convex hull by Graham scan [87] to find the convex hull. Figure 5.7 is an example of the result after finding the convex hull of the hand shape. The red dots including the fingertips are the convex hull. Once the convex hull is acquired, each convex point is examined by curve detection method to determine fingertips. The

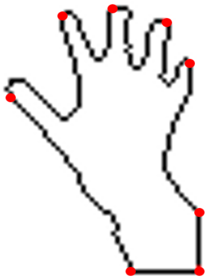


Figure 5.7: The convex hull of the hand shape

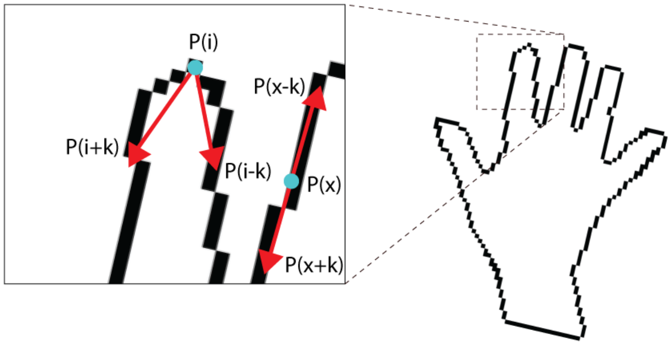


Figure 5.8: The k-curvature [4]

k-curvature is used to implement the curve detection. The k-curvature detects the angle between two vectors at each convex hull point. The vectors start at the convex hull point and end k points away in either direction as shown in Figure 5.8.

As a result, the finger detection approach performed poorly. Due to the low resolution of the Kinect, our captured hand shape images are not as clear as the actual hand shape. The finger detection approach is not robust to unclear hand shapes. For example, Figure 5.9 is one of our captured hand shapes, and the red dots are convex hull. The actual fingertips do not have the characteristic This kind of our hand shapes makes defining k of k-curvature difficult.

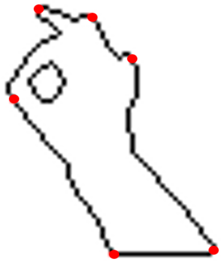


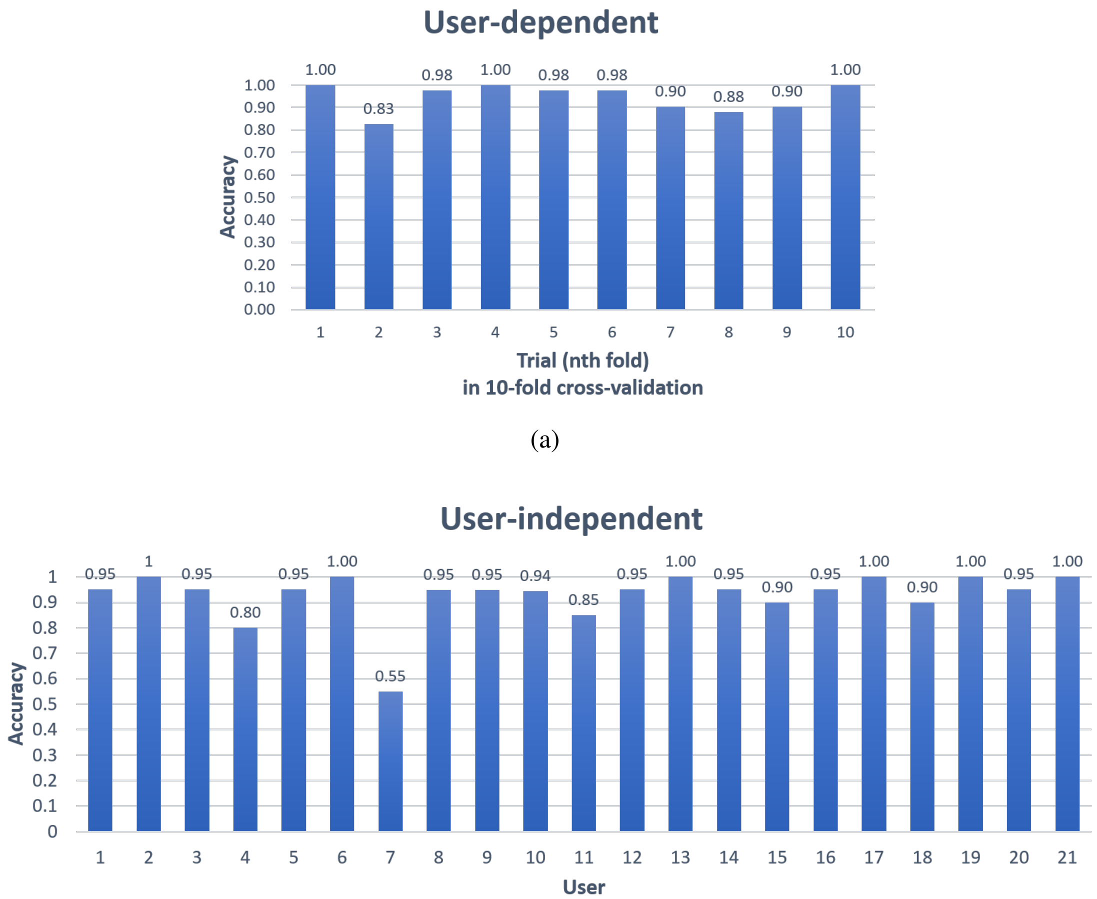
Figure 5.9: One of our unclear hand shapes

5.6 Shape matching

For shape matching, there are various measurement of similarity between shapes, e.g., Hausdorff distance [88], inner distance [89], shape context [90, 91]. I implement the shape context with Emgu CV [92], which allows to use functions of Open CV library [93] in C# framework. The shape context is binning of spatial relationships between points to find the correspondence between two shapes. The shape context has three main methods: log-polar bins (k-bins histogram), shape context cost, point correspondence. Log-polar bins I used quantizes angle into 12 bins, log-distance into 4 bins. The histogram with 48 elements, which is the shape context for a point. For each point on the shape, a coarse histogram of the relative coordinates of the remaining points is computed by taking the vector distance of the point with respect to all other points. Then, Chi-square test statistic is used to get a measure of the cost of matching two points. After getting the set of costs between all pairs of points on the two shapes, bipartite graph matching is done to minimize the total cost of matching.

5.7 Result

The total number of hand shape data is 420 (21 people × 5 different hand shape × 4 times (twice with left and right hand for each)). Figure 5.10 shows the result of solely the shape-based recognition. Whereas Figure 5.10a is the result of 10-fold cross validation with whole data, i.e, a user's different data can be either test data or training data, the result shown in Figure 5.10b is that the whole data of a user is used as test data and the others' data have been used as training data. For this reason, the average accuracy of user-independent data is slightly lower than that of user-dependent data. Table 5.2 shows the performance of individual recognition, and the performance of the two-stage recognition. Overall, as there is not much difference between the accuracy of user-independent data and the accuracy of user-dependent data, it shows that our recognition system is able to generalize the gestures of users.



(a)

(b)

Figure 5.10: The performance of the shape-based recognition

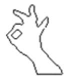
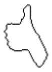
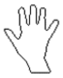
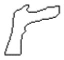
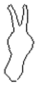
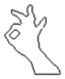
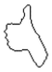
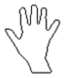
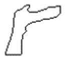
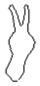
Hand shape	Classified As				
					
	95	4	0	1	0
	0	98	0	1	1
	6	0	93	1	0
	2	7	0	91	0
	1	1	0	1	97

Table 5.1: Confusion matrix for distinguishing between the 5 hand shapes using shape context

	User-dependent	User-independent
Trajectory-based recognition	92.40%	
Shape-based recognition	94.43%	92.31%
Two-stage recognition	86.04%	82.07%

Table 5.2: The performance of proposed two-stage recognition

6. FUTURE WORK

There are many avenues for future development. Foremost, our results shows the possibility that our system can be extended to the rest of user-defined gestures or even new user-defined gestures and to a real-time application.

In this work, we implemented a two-stage recognition system for our user-defined gesture set but has yet to test on the remaining 10 hand gestures. Originally, the number of the remaining gestures was 15 but as considered that there are identical gestures for different emojis, the number of the hand gestures to be classified is 10.

This research has focused on eliciting user-defined gestures and recognizing these gestures. I have yet to recognize the gesture in real-time. I have built the framework of the two-stage recognition system based on the ultimate goal of real-time recognition system. I envision that a user sends an emoji by an gesticulation while chatting on online.

I am looking forward to finding ways to improve the our system by adding more features or bringing more classification methods.

7. CONCLUSION

In this work, I have presented a user-defined gesture set that is highly representative to popular emojis and a two-stage hand gesture recognition that distinguish between the 15 user-defined hand gestures using data collected from a Kinect, a depth camera, with an overall accuracy of 86%.

I present three main contribution in light of the explanation of my work.

1. **Symbolic Gestures:** What kinds of hand gestures (emblem) do users express that they feel properly symbolize popular emojis?

   Through three experiments, I developed 30 user-defined gestures analoguous to emojis. The first experiment was to design a highly spontaneous gesture set by acquiring gestures for emojis from 10 participants. The second experiment is to achieve a high agreement of the gestures resulted in the first experiment with 17 participants by using Wobbrock's agreement [1]. I found that The last experiment was to evaluate the final gesture set by using Wobbrock's guessability [94] and preference ratings as performance measurement.

2. **Quality Features:** Which features are valuable for recognizing the user-defined gestures?

   I developed 32 features specific to 15 user-defined gestures. These 32 additional features improved the system's performance to discriminate between the 15 gestures by over 10%. Therefore, through this work, I show that it is feasible to identify specific user-defined gestures by using features specifically designed to highlight the unique characteristics of those gestures.

3. **Robust Recognition:** Can we achieve reasonable hand gesture recognition for these

emojis using our method?

I developed a two-stage recognition system, where a trajectory-based recognition classifies dynamic hand gestures with above-mentioned features, and a shape-based recognition classifies static hand gesture, particularly hand shapes with shape context. To test the performance of our system, I collected 1008 gesture data from 21 participants. The accuracy of the two-stage recognition system was 86%.

REFERENCES

[1] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, "User-defined gestures for surface computing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1083–1092, ACM, 2009.

[2] "Microsoft Kinect 2 for Windows." Web, 2014.

[3] P. Rajan and T. Hammond, "From paper to machine: Extracting strokes from images for use in sketch recognition," in *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling*, SBM'08, (Aire-la-Ville, Switzerland, Switzerland), pp. 41–48, Eurographics Association, 2008.

[4] M. Truyenque, *Uma Aplicação de Visão Computacional que Utiliza Gestos da Mão para interagir com o Computador. Rio de Janeiro, 2005. 100p.* PhD thesis, Dissertação de Mestrado-Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, 2005.

[5] J. B. Walther and K. P. DâĂŹAddario, "The impacts of emoticons on message interpretation in computer-mediated communication," *Social science computer review*, vol. 19, no. 3, pp. 324–347, 2001.

[6] K.-A. Hsiao and P.-L. Hsieh, "Age difference in recognition of emoticons," in *International Conference on Human Interface and the Management of Information*, pp. 394–403, Springer, 2014.

[7] A. H. Huang, D. C. Yen, and X. Zhang, "Exploring the potential effects of emoticons," *Information & Management*, vol. 45, no. 7, pp. 466–473, 2008.

[8] K. Rivera, N. J. Cooke, and J. A. Bauhs, "The effects of emotional icons on remote communication," in *Conference companion on human factors in computing systems*,

pp. 99–100, ACM, 1996.

[9] H. Cramer, P. de Juan, and J. Tetreault, "Sender-intended functions of emojis in us messaging," in *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 504–509, ACM, 2016.

[10] J. Y. Lee, N. Hong, S. Kim, J. Oh, and J. Lee, "Smiley face: why we use emoticon stickers in mobile messaging," in *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pp. 760–766, ACM, 2016.

[11] G. W. Tigwell and D. R. Flatla, "Oh that's what you meant!: reducing emoji misunderstanding," in *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pp. 859–866, ACM, 2016.

[12] "React Messenger." Web, 2014.

[13] Y. Urabe, R. Rzepka, and K. Araki, "Emoticon recommendation system to richen your online communication," *International Journal of Multimedia Data Engineering and Management*, vol. 5, no. 1, pp. 14–33, 2014.

[14] M. Yuasa, K. Saito, and N. Mukawa, "Emoticons convey emotions without cognition of faces: an fmri study," in *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, pp. 1565–1570, ACM, 2006.

[15] "No Time to Text? Say It With Stickers." Web, 2014.

[16] A. M. Hautasaari, N. Yamashita, and G. Gao, "Maybe it was a joke: emotion detection in text-only communication by non-native english speakers," in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pp. 3715–3724, ACM, 2014.

[17] S. Goldin-Meadow, *Hearing gesture: How our hands help us think.* Harvard University Press, 2005.

[18] D. McNeill, *Gesture and thought.* University of Chicago press, 2008.

[19] H. Pohl, D. Stanke, and M. Rohs, "Emojizoom: emoji entry via large overview maps," in *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 510–517, ACM, 2016.

[20] A. El Ali, T. Wallbaum, M. Wasmann, W. Heuten, and S. C. Boll, "Face2emoji: Using facial emotional expressions to filter emojis," in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 1577–1584, ACM, 2017.

[21] C. Neustaedter and S. Greenberg, "Intimacy in long-distance relationships over video chat," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 753–762, ACM, 2012.

[22] K. O'Hara, A. Black, and M. Lipson, "Everyday practices with mobile video telephony," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 871–880, ACM, 2006.

[23] J. Procyk, C. Neustaedter, C. Pang, A. Tang, and T. K. Judge, "Exploring video streaming in public settings: shared geocaching over distance using mobile video chat," in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pp. 2163–2172, ACM, 2014.

[24] L. E. Sherman, M. Michikyan, and P. M. Greenfield, "The effects of text, audio, video, and in-person communication on bonding between friends," *Cyberpsychology: Journal of psychosocial research on cyberspace*, vol. 7, no. 2, 2013.

[25] H.-C. Wang and C.-T. Lai, “Kinect-taped communication: using motion sensing to study gesture use and similarity in face-to-face and computer-mediated brainstorming,” in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pp. 3205–3214, ACM, 2014.

[26] S. Jana, A. Pande, A. Chan, and P. Mohapatra, “Mobile video chat: issues and challenges,” *IEEE Communications Magazine*, vol. 51, no. 6, pp. 144–151, 2013.

[27] R. M. Krauss, Y. Chen, and P. Chawla, “Nonverbal behavior and nonverbal communication: What do conversational hand gestures tell us?,” *Advances in experimental social psychology*, vol. 28, pp. 389–450, 1996.

[28] J. Nalepa and M. Kawulok, “Fast and accurate hand shape classification,” in *International Conference: Beyond Databases, Architectures and Structures*, pp. 364–373, Springer, 2014.

[29] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, “Vision-based hand-gesture applications,” *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[30] H. Tsujita and J. Rekimoto, “Smiling makes us happier: enhancing positive mood and communication with smile-encouraging digital appliances,” in *Proceedings of the 13th international conference on Ubiquitous computing*, pp. 1–10, ACM, 2011.

[31] F. Mueller, F. Vetere, M. R. Gibbs, J. Kjeldskov, S. Pedell, and S. Howard, “Hug over a distance,” in *CHI’05 extended abstracts on Human factors in computing systems*, pp. 1673–1676, ACM, 2005.

[32] R. El Kaliouby and P. Robinson, “Faim: integrating automated facial affect analysis in instant messaging,” in *Proceedings of the 9th international conference on Intelligent user interfaces*, pp. 244–246, ACM, 2004.

[33] J. Feijó Filho, T. Valle, and W. Prata, “Non-verbal communications in mobile text chat: emotion-enhanced mobile chat,” in *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, pp. 443–446, ACM, 2014.

[34] A. Pirzadeh, H.-W. Wu, R. Bharali, B. M. Kim, T. Wada, and M. S. Pfaff, “Designing multi-touch gestures to support emotional expression in im,” in *CHI’14 Extended Abstracts on Human Factors in Computing Systems*, pp. 2515–2520, ACM, 2014.

[35] A. Rovers and H. A. van ESSEN, “Him: a framework for haptic instant messaging,” in *CHI’04 Extended Abstracts on Human Factors in Computing Systems*, pp. 1313–1316, ACM, 2004.

[36] H. Shin, J. Lee, J. Park, Y. Kim, H. Oh, and T. Lee, “A tactile emotional interface for instant messenger chat,” *Human Interface and the Management of Information. Interacting in Information Environments*, pp. 166–175, 2007.

[37] C. S. S. Tan, J. Schöning, K. Luyten, and K. Coninx, “Informing intelligent user interfaces by inferring affective states from body postures in ubiquitous computing environments,” in *Proceedings of the 2013 international conference on Intelligent user interfaces*, pp. 235–246, ACM, 2013.

[38] H. Wang, H. Prendinger, and T. Igarashi, “Communicating emotions in online chat using physiological sensors and animated text,” in *CHI’04 extended abstracts on Human factors in computing systems*, pp. 1171–1174, ACM, 2004.

[39] “Smile Chat.” Web, 2015.

[40] A. Chaudhary, J. L. Raheja, K. Das, and S. Raheja, “Intelligent approaches to interact with machines using hand gesture recognition in natural way: A survey,” *Inter-*

national Journal of Computer Science and Engineering Survey, vol. 2, pp. 122–133, mar 2011.

[41] A. Kulshreshth, C. Zorn, and J. J. LaViola, “Poster: Real-time markerless kinect based finger tracking and hand gesture recognition for hci,” in *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 187–188, March 2013.

[42] Z. hua Chen, J.-T. Kim, J. Liang, J. Zhang, and Y.-B. Yuan, “Real-time hand gesture recognition using finger segmentation,” *The Scientific World Journal*, vol. 2014, 2014.

[43] T. Hongyong and Y. Youling, “Finger tracking and gesture recognition with kinect,” in *2012 IEEE 12th International Conference on Computer and Information Technology*, pp. 214–218, Oct 2012.

[44] S. S. Rautaray and A. Agrawal, “Real time gesture recognition system for interaction in dynamic environment,” *Procedia Technology*, vol. 4, pp. 595 – 599, 2012. 2nd International Conference on Computer, Communication, Control and Information Technology(C3IT-2012) on February 25 - 26, 2012.

[45] T. R. Trigo and S. R. M. Pellegrino, “An analysis of features for hand-gesture classification,” in *17th International Conference on Systems, Signals and Image Processing (IWSSIP 2010)*, pp. 412–415, 2010.

[46] E. Sangineto and M. Cupelli, “Real-time viewpoint-invariant hand localization with cluttered backgrounds,” *Image and Vision Computing*, vol. 30, no. 1, pp. 26 – 37, 2012.

[47] A. Agrawal, R. Raj, and S. Porwal, “Vision-based multimodal human-computer interaction using hand and head gestures,” in *2013 IEEE Conference on Information Communication Technologies*, pp. 1288–1292, April 2013.

[48] E. Yoruk, E. Konukoglu, B. Sankur, and J. Darbon, "Shape-based hand recognition," *IEEE Transactions on Image Processing*, vol. 15, pp. 1803–1815, July 2006.

[49] Y. Li, "Hand gesture recognition using kinect," in *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*, pp. 196–199, IEEE, 2012.

[50] G.-F. He, S.-K. Kang, W.-C. Song, and S.-T. Jung, "Real-time gesture recognition using 3d depth camera," in *Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on*, pp. 187–190, IEEE, 2011.

[51] M. Hatano, S. Sako, and T. Kitamura, "Contour-based hand pose recognition for sign language recognition," in *6th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*, p. 17, 2015.

[52] Y. Wen, C. Hu, G. Yu, and C. Wang, "A robust method of detecting hand gestures using depth sensors," in *Haptic Audio Visual Environments and Games (HAVE), 2012 IEEE International Workshop on*, pp. 72–77, IEEE, 2012.

[53] M. Maisto, M. Panella, L. Liparulo, and A. Proietti, "An accurate algorithm for the identification of fingertips using an rgb-d camera," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, pp. 272–283, June 2013.

[54] J. L. Raheja, A. Chaudhary, and K. Singal, "Tracking of fingertips and centers of palm using kinect," in *2011 Third International Conference on Computational Intelligence, Modelling Simulation*, pp. 248–252, Sept 2011.

[55] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Transactions on Multimedia*, vol. 15, pp. 1110–1120, Aug 2013.

[56] K. Fujimura and X. Liu, “Sign recognition using depth image streams,” in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pp. 381–386, April 2006.

[57] D. Rempel, M. J. Camilleri, and D. L. Lee, “The design of hand gestures for humanâĂŞcomputer interaction: Lessons from sign language interpreters,” *International Journal of Human-Computer Studies*, vol. 72, no. 10âĂŞ11, pp. 728 – 735, 2014.

[58] T. Baudel and M. Beaudouin-Lafon, “Charade: remote control of objects using free-hand gestures,” *Communications of the ACM*, vol. 36, no. 7, pp. 28–35, 1993.

[59] C. Tran and M. M. Trivedi, “3-d posture and gesture recognition for interactivity in smart spaces,” *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 178–187, 2012.

[60] M. Bhuyan, D. Ghosh, and P. Bora, “Feature extraction from 2d gesture trajectory in dynamic hand gesture recognition,” in *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, pp. 1–6, IEEE, 2006.

[61] M. K. Bhuyan, D. A. Kumar, K. F. MacDorman, and Y. Iwahori, “A novel set of features for continuous hand gesture recognition,” *Journal on Multimodal User Interfaces*, vol. 8, no. 4, pp. 333–343, 2014.

[62] S. J. Mckenna and K. Morrison, “A comparison of skin history and trajectory-based representation schemes for the recognition of user-specified gestures,” *Pattern Recognition*, vol. 37, no. 5, pp. 999–1009, 2004.

[63] J. Singha, A. Roy, and R. H. Laskar, “Dynamic hand gesture recognition using vision-based approach for human–computer interaction,” *Neural Computing and Applications*, pp. 1–13.

[64] M. Geetha, C. Manjusha, P. Unnikrishnan, and R. Harikrishnan, "A vision based dynamic gesture recognition of indian sign language on kinect based depth images," in *Emerging Trends in Communication, Control, Signal Processing & Computing Applications (C2SPCA), 2013 International Conference on*, pp. 1–7, IEEE, 2013.

[65] Y. Zhu and B. Yuan, "Real-time hand gesture recognition with kinect for playing racing video games," in *Neural Networks (IJCNN), 2014 International Joint Conference on*, pp. 3240–3246, IEEE, 2014.

[66] D. Ramirez-Giraldo, S. Molina-Giraldo, A. M. Álvarez-Meza, G. Daza-Santacoloma, and G. Castellanos-Domínguez, "Kernel based hand gesture recognition using kinect sensor," in *Image, Signal Processing, and Artificial Vision (STSIVA), 2012 XVII Symposium of*, pp. 158–161, IEEE, 2012.

[67] M. Wu, C. Shen, K. Ryall, C. Forlines, and R. Balakrishnan, "Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces," in *Horizontal Interactive Human-Computer Systems, 2006. TableTop 2006. First IEEE International Workshop on*, pp. 8–pp, IEEE, 2006.

[68] "Most popular emoji." Web.

[69] "This is the most-used emoji on Instagram." Web, June 2015.

[70] "The 100 Most-Used Emojis." Web, June 2014.

[71] J. Epps, S. Lichman, and M. Wu, "A study of hand shape use in tabletop gesture interaction," in *CHI'06 extended abstracts on human factors in computing systems*, pp. 748–753, ACM, 2006.

[72] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick, and A. Pentland, "Invariant features for 3-d gesture recognition," in *Automatic Face and Gesture Recog-*

nition, 1996., Proceedings of the Second International Conference on, pp. 157–162, IEEE, 1996.

[73] D. Rubine, *Specifying gestures by example*, vol. 25. ACM, 1991.

[74] D. Xu, X. Wu, Y.-L. Chen, and Y. Xu, “Online dynamic gesture recognition for human robot interaction,” *Journal of Intelligent & Robotic Systems*, vol. 77, no. 3-4, pp. 583–596, 2015.

[75] X. Zabulis, H. Baltzakis, and A. Argyros, “Vision-based hand gesture recognition for human-computer interaction,” in *The universal access handbook*, pp. 1–30, CRC Press, 2009.

[76] L. Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[77] G. Biau, “Analysis of a random forests model,” *Journal of Machine Learning Research*, vol. 13, no. Apr, pp. 1063–1095, 2012.

[78] T. Bruckhaus, “The business impact of predictive analytics,” *Knowledge discovery and data mining: Challenges and realities*, pp. 114–138, 2007.

[79] P. Rajan and T. Hammond, “Applying online sketch recognition algorithms to a scanned-in sketch,” in *Proceedings of the Workshop on Sketch Recognition at the 14th International Conference of Intelligent User Interfaces Posters (IUI)*, (Sanibel, FL), ACM, 2 2009.

[80] P. Rajan, P. Taele, and T. Hammond, “Evaluation of paper-pen based sketching interface.,” in *Proceedings of the 16th International Conference on Distributed Multimedia Systems (DMS)*, pp. 321–326, 2010.

[81] T. Zhang and C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[82] A. Martin and S. Tosunoglu, "Image processing techniques for machine vision," *Miami, Florida*, 2000.

[83] "Accord.NET Machine Learning Framework." Web, 2009.

[84] D. J. Ryan, "Finger and gesture recognition with microsoft kinect," Master's thesis, University of Stavanger, Norway, 2012.

[85] J. Segen and S. Kumar, "Human-computer interaction using gesture recognition and 3d hand tracking," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pp. 188–192, IEEE, 1998.

[86] T. Lee and T. Hollerer, "Handy ar: Markerless inspection of augmented reality objects using fingertip tracking," in *Wearable Computers, 2007 11th IEEE International Symposium on*, pp. 83–90, IEEE, 2007.

[87] R. L. Graham, "An efficient algorith for determining the convex hull of a finite planar set," *Information processing letters*, vol. 1, no. 4, pp. 132–133, 1972.

[88] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[89] H. Ling and D. W. Jacobs, "Using the inner-distance for classification of articulated shapes," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 719–726, IEEE, 2005.

[90] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[91] M. Oltmans, *Envisioning sketch recognition: a local feature based approach to recognizing informal sketches*. PhD thesis, Massachusetts Institute of Technology, 2007.

[92] "Emgu CV." Web, 2008.

[93] "Open cv." Web.

[94] J. O. Wobbrock, H. H. Aung, B. Rothrock, and B. A. Myers, "Maximizing the guessability of symbolic input," in *CHI'05 extended abstracts on Human Factors in Computing Systems*, pp. 1869–1872, ACM, 2005.