Teaching Motion Gestures via Recognizer Feedback

Ankit Kamal
University of Waterloo
Waterloo, ON, Canada
a6kamal@uwaterloo.ca

Yang Li
Google Research
Mountain View, CA, USA
yangli@acm.org

Edward Lank
University of Waterloo
Waterloo, ON, Canada
lank@cs.uwaterloo.ca

ABSTRACT
When using motion gestures, 3D movements of a mobile phone, as an input modality, one significant challenge is how to teach end users the movement parameters necessary to successfully issue a command. Is a simple video or image depicting movement of a smartphone sufficient? Or do we need three-dimensional depictions of movement on external screens to train users? In this paper, we explore mechanisms to teach end users motion gestures, examining two factors. The first factor is how to represent motion gestures: as icons that describe movement, video that depicts movement using the smartphone screen, or a Kinect-based teaching mechanism that captures and depicts the gesture on an external display in three-dimensional space. The second factor we explore is recognizer feedback, i.e. a simple representation of the proximity of a motion gesture to the desired motion gesture based on a distance metric extracted from the recognizer. We show that, by combining video with recognizer feedback, participants master motion gestures equally quickly as end users that learn using a Kinect. These results demonstrate the viability of training end users to perform motion gestures using only the smartphone display.

Author Keywords
Motion Gestures; sensors; smartphone; Android; Recognizer feedback.

ACM Classification Keywords
H.5.2. User Interfaces – Interaction Styles.

General Terms
Human Factors; Design; Experimentation.

INTRODUCTION
Hand motion—pointing, gesturing, grasping, shaking, tapping—is a rich channel of communication. We point and gesture while we talk; we grasp tools to extend our capabilities; we grasp, rotate, and shake items to explore them. Inspired by these everyday movements to extend conversation, researchers [13, 14, 16, 17, 18] have begun to explore motion gestures, i.e. deliberate movements to issue commands to a device, as an input modality. Motion gestures have been applied to, for example, large-screen displays, desktop computers, and smartphones.

In this paper, we are particularly interested in motion gestures as an input modality for modern smartphones. The reasons for this are twofold. First, modern smartphones contain an evolving set of sensors for recognizing movement of the phone, including accelerometers, gyroscopes and cameras, so the technology already exists to support motion gesture input. Second, using a motion gesture provides many attendant benefits, including an expanded input space and the ability to issue commands eyes-free without using the touch screen by leveraging proprioception [14].

One of the most significant barriers to widespread adoption of motion gesture input involves teaching end-users to perform motion gestures. Motion gestures are not self-revealing; end-users need to be taught the set of motion gestures supported by a smartphone device. As well, for each of these gestures, end-users need to understand exactly how to perform the gestures to ensure maximum recognition accuracy. Constraints on movement include the shape of the movement of the motion gesture (its three-dimensional path in space) and the kinematics of the motion gesture (the tolerances for fast or slow motion gestures).

The process of instructing and correcting the actions of a learner is typically called scaffolding. Scaffolding involves both a depiction of the desired activity and assessment to correct inaccuracies. Many questions arise when considering how to depict motion gestures. Can we simply show icons of motion gestures that depict movement (see Ruiz et al. [17]). Do we instead need to show a brief video of movement on the smartphone display? Or do we require an external display to see movement in larger scale? Alongside techniques for depicting the form of a motion gesture, recognizing a motion gesture involves contrasting the gesture performed with some desired template, i.e. providing some form of feedback that guides a user more quickly to the correct action. We wish to also understand how feedback can be used to help a user converge to the ideal motion gesture more quickly.

In this paper, we contrast three techniques for teaching motion gestures: icons, smartphone videos, or Kinect plus

IUI'14, February 24–27, 2014, Haifa, Israel.

http://dx.doi.org/10.1145/2557500.2557521

videos on external displays. For each of these techniques, we also study the effect that simple recognizer feedback – more specifically a visualization of the distance between a desired template and the actual input movement of the user – on the ability of end-users to accurately perform motion gestures.

We show that Kinect-based instruction, where the movement is displayed as a 3D wireframe and participant movement is captured and replicated for direct contrast teaches motion gestures very quickly for a group of participants. As well, for Kinect-based feedback, the presence of a visualization of recognizer distance had limited effect on accuracy, primarily because the Kinect's contrasting of input motion from template was sufficient. However, we also show that, while video on a smartphone screen is worse than Kinect, video on a smartphone screen plus a simple visualization of recognizer feedback causes the smartphone video condition to converge to the performance of Kinect-based instruction.

The significance of these results lies in the training of end-users on motion gestures as input to smartphones. Before embarking on this research, we were unsure whether it was possible to train end users effectively on motion gestures without someone present to demonstrate the motion gestures, or without external hardware (e.g. a Kinect) to allow users to master the kinematics of the motion gesture commands. Given our results, it now seems plausible to construct a teaching aid for motion gestures that uses only the smartphone display.

The remainder of this paper is organized as follows. We first highlight related work in motion gestures and on techniques for teaching gestures. Next we detail a series of pilot studies and a final study evaluating techniques for teaching motion gestures. Finally, we discuss our results and their implications more fully.

RELATED WORK

Free-space hand gesture interaction (as in the movie *Minority Report*) has been perceived of as a novel, futuristic input technique, despite known problems with fatigue, i.e. gorilla arm. Bolt designed a "put-that-there" system in 1980 that combined pointing with voice commands [3]. Vogel and Balakrishnan [18] explored the design space for freehand gestural interaction for large vertical displays.

Motion gestures are a known, albeit underutilized, technique for controlling smartphones. Hinckley et al. [7] proposed using tilt on mobile devices to allow a user to change screen orientation—a feature now commonly found on many smartphones. In addition to navigation, tilt sensors have also been used for text input [8] and accessing data on virtual shelves around a user [11]. Commercially, the use of a shake motion gesture to shuffle music is one common example of controlling a smartphone or personal music player (e.g. iPod) via a motion gesture. As well, some modern smartphones allow the user to place the smartphone face-down on a desk to mute the ringtone for an incoming phone call. Finally, the Google App for iPhone turns on voice search if the iPhone is brought to your ear.

Previous research on motion gestures

Ruiz et al. [17] created a taxonomy describing the attributes of smartphone motion gestures and their natural mappings onto smartphone commands. They showed that a consensus exists among users on parameters of movement and on mappings of motion gestures onto commands. They also enumerated a user-defined motion-gesture set for smartphone input.

Alongside work on motion gesture input, Ruiz and Li [16] explored how best to discriminate between deliberate motion gestures and everyday movement of a smartphone. They proposed "DoubleFlip", a motion gesture designed as an input delimiter for smartphone motion gestures. The DoubleFlip delimiter is performed by quickly rotating the wrist such that the phone's display is facing away from the user and back to the original position with the display of the phone facing the user. They showed that DoubleFlip is easy to invoke and unlikely to be accidentally invoked by users.

Negulescu et al. [14] analyzed the relative cognitive cost of taps, surface gestures, and motion gestures for distracted input on smartphone devices. They show that there is no significant difference in reaction time for motion gestures, taps, or surface gestures on smartphones, and that the use of motion gestures results in participants in a study spending significantly less time looking at the smartphone during walking than taps, even with eyes-free optimized input interfaces.

Negulescu et al. [13] also explored techniques for limiting false positives and false negatives for motion gesture input. They devised a "bi-level threshold" recognizer which helped lower the rate of recognition failures by accepting either a tightly thresholded gesture or two consecutive gestures recognized by a looser-threshold model.

Previous research on teaching surface gestures

Our research in this paper focuses specifically on teaching motion gestures to smartphone users. Significant past work exists in teaching users gestural input languages. Kurtenbach's [10] Marking menus, an extension of pie menus [5], combine feed-forward and feedback to provide a fluent transition between novice and expert use. Marking menus take advantage of novice user's hesitation when they are unsure of a gesture or command. Users flick the pen or mouse in a particular direction in order to indicate a command. After a "press and wait" gesture, a circular feed-forward display appears around the mouse cursor, showing each available command. Highlighting the current selected item during input provides feedback on how a user's input is being interpreted. This approach offers a good compromise between learning and efficient use. Novices often pause to take advantage of the feed-forward display. As they become experts, they move more quickly and no longer needing the feed-forward menu, significantly increase overall performance.

In the same vein as marking menus, Bau et al. [2] designed a dynamic guide called "Octopocus" that combines on-screen feed-forward and feedback to help users learn, execute and remember surface gesture sets. Octopocus continuously updates the state of the recognition algorithm by gradually modifying the thickness of possible gesture paths, based on its 'consumable error rate'. They show that users can better learn, execute and remember gesture sets if one reveals, during input, what is normally an opaque process, the current state of recognition, and represents gestures in a graphical form that shows the optimal path for the remaining alternatives.

One challenge with gesture-based systems is that end-users need to be made aware of the gestures that can be performed to invoke commands. Alongside this awareness, as users are learning the mechanics of gestures, they must also have the opportunity to practice and receive feedback on the gestures they attempt. To satisfy these goals, Bragdon et al. [4] designed a unique training system, GestureBar, which can be incorporated into gesture-based systems for pen-tablet computers. GestureBar is, conceptually, a simple scratch pad which allows the user to select a gesture and then attempt the gesture within a region of the display. Feedback depicting the deviation between desired input and the user's input is displayed so the user can modify and correct any errors in the pen strokes that they draw on the screen. In their research, Bragdon et al. describe the design iterations, the final GestureBar system, and its effectiveness as a training tool based on subjective user feedback.

One of our goals is to adapt aspects of the training systems described above to motion gestures on Smartphones. However, how we communicate motion gestures to end users is somewhat ambiguous. With Marking menus, Octopus, and GestureBar, because users were drawing on a two-dimensional surface, the system could render the two-dimensional shape. Users could start out with an animation of the movement, then over time simply see the final, complete gesture. However, a smartphone cannot move itself through space. Communicating the relative displacement obviously requires some form of a movie that displays motion relative to the end-user. To the best of our knowledge, no previous research has been done to train people to perform motion gestures.

GESTURE RECOGNIZER DESIGN

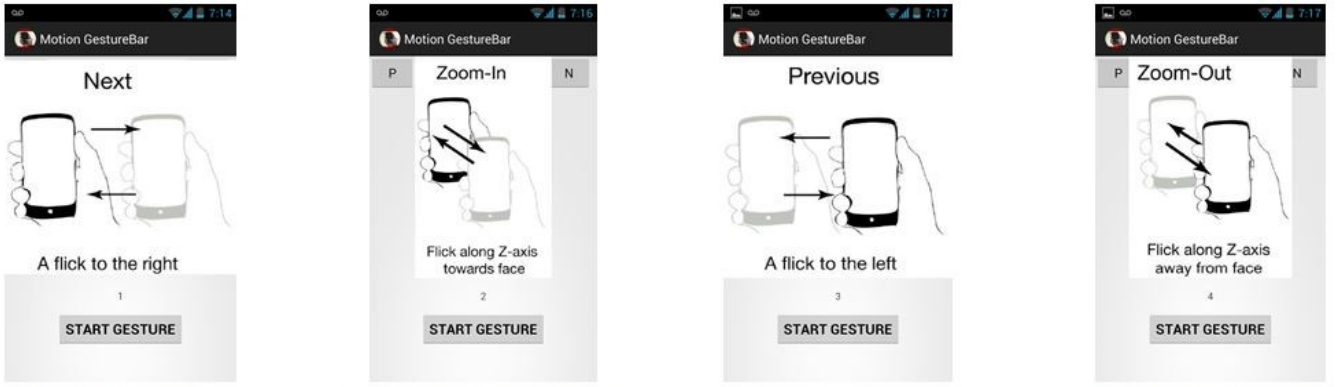
Ruiz et al. [17] note that, when end-users design motion gestures, the gestures they select tend to be simple (non-compound), single-axis movements with low kinematic impulse. As a result, we base our study around four single-axis gestures – *right flick, left flick, flick up towards face* and *flick down away from face*.

Our four gestures were chosen from the user defined set in Ruiz et al. [17], and we would argue that they represent the simplest set of useful motion gestures for smartphone control. Nominally, the gestures correspond to next, previous, zoom-in and zoom-out gestures respectively. Essentially, we chose the gestures we did because these are the types of gestures – single axis, low kinematic impulse – users specify when we elicit gestures from them [17].

Our recognizer was developed in Java using the Android SDK [1] for use on Nexus S phones with an ARM Cortex A8 1GHz processor and a three-axis accelerometer. Sensor input, i.e. filtered acceleration data, is matched to gesture templates using Dynamic Time Warping (DTW) [12]. DTW is a dynamic programming algorithm that measures the similarity of two time series with temporal dynamics [12] when given a function for calculating the distance between the two time samples. The result is a warp distance that can be used to determine how similar a set is to the reference set. A warp distance of 0 (zero) indicates absolute identical sets. The bigger the distance, the more different the sets are. Our implementation of our gesture recognizer uses a weighted Euclidean distance function for calculating the distance between the quantized time series of acceleration data to the a template. As a full discussion of DTW is beyond the scope of this paper, we refer the reader to Wobbrock et al. [19] for more information. The sampling rate of acceleration data was 32 Hz.

One challenge with the gestures we select is that, because they are single-axis and because they have low kinematic impulse, the gestures are virtually indistinguishable from everyday movement of a smartphone. The typical way designers of recognizers address a collision between noise and signal is via a tight criterion function to discriminate true positives from false positives [13]. The challenge with a tight criterion function is the propensity to cause false negatives. In other words, seeking to avoid accidental activation of a motion gesture, we require greater precision in the performance of a motion gesture. This, in turn, makes it more essential to teach end users the careful kinematics needed to successfully invoke a motion gesture; otherwise, they repeatedly fail to invoke their desired motion gesture.

To simulate this tight criterion function, the DTW templates for each gesture type were created by an expert user, specifically one of the authors of this paper. The expert performed the correct gesture 20 times. Each gesture was compared to the 19 other gestures using DTW. Then, the average warp distance for the respective gesture was calculated, and the gesture with lowest average warp distance from all other gestures was selected as the gesture template for that particular gesture. This is a common approach found in related work (Kar et al. [9]). In a second step, the selected gesture template was compared to the remaining 19 gestures. The 19 warp distances were then used to calculate the mean, median, minimum, maximum and standard deviation of distances. These values were used to calculate the threshold of the DTW Distance metric within which an input gesture is considered as valid. The result of the use of a single expert user is that, to successfully invoke a motion gesture on a smartphone, the



(a) Right Flick (b) Flick away from face (c) Left Flick (d) Flick towards face

Figure 1: Iconic representation of the motion gestures

end-user must perform the gesture in nearly the same manner as the expert from whom the template was elicited.

In the following sections, we describe our experiments where we explore various types of mechanisms to teach end users motion gestures, examining two factors. The first factor we explore is how to represent motion gestures: as icons that describe movement, video that depicts movement using the smartphone screen, or as video on an external screen. The second factor we explore is feedback, i.e. a simple representation of the proximity of a motion gesture to the desired motion gesture based either on a distance metric extracted from the recognizer or based upon movement tracked by Kinect.

PRELIMINARY STUDY: DEPICTING MOTION GESTURES

We performed a preliminary study to compare the performance of two basic representations of motion gestures as teaching methods: icons describing the movement (see Ruiz et al. [17]) and short videos depicting the movement on the smartphone screen. Feedback mechanisms were not explored in this study. The reason for conducting this study was to check if very basic representations of the motion gesture, e.g. icons or simple videos, are sufficient to teach motion gestures.

Participants

We recruited 12 participants (8 male, 4 female, ages 20 -35) from the general student body of our institution. We advertised the study widely to get a sample of participants with diverse backgrounds and levels of experience using computers. All participants owned a smartphone. and knew what motion gestures were, but not with respect to movement of the smartphone device. All were familiar with motion gestures pertaining to the Nintendo Wii or Kinect based games, but none were familiar with smartphone-based motion gestures (beyond shake-to-shuffle).

Experimental Design

We used a between-subjects design with the two conditions - teaching via icons and via videos. The reason for choosing a between-subjects design is that if a user masters a gesture using one technique, the evaluation of the other technique becomes invalid. Six participants were asked to perform motion gestures based on the iconic representation of the motion gesture shown and the other six were asked to perform gestures based on the video shown on the phone. The iconic representations that were displayed on the Android device are shown in Figure 1. These iconic representations of the motion gestures were taken from the user-defined set created in the work done by Ruiz et al. [17]. The videos of gestures were captured from gestures performed by an expert user, and they depict the gesture used for the correct template in our recognizer from an eyes-view. i.e. as if one was looking at the smartphone while performing the gesture, Figure 2.

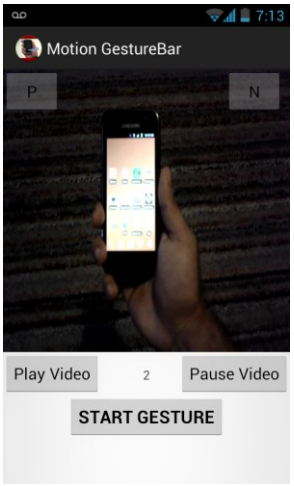


Figure 2: A screen-grab of the video on the phone describing the correct motion gesture

Procedure

Each participant was asked to perform four blocks (corresponding to the four gesture types) of thirty gestures each, i.e., 4 x 30 = 120 gestures. The order of the blocks of gestures to be performed was presented randomly. We did not give any hint to our participants regarding the correct gesture. When our DTW recognizer recognized a correct gesture, a beep sound was generated, indicating the completion of the correct gesture. Participants could refer to the icons or watch the videos as many times as they wanted. A total of 4 x 30 x 12 = 1440 gestures were performed.

Metrics

We extracted two metrics from our participants:

No. of correct gestures: The number of correct gestures out of the total of 120 performed by each user. This is a measure of performance of the user.

Average converging gesture count: This value is the average number of gestures it took for the participant to converge (or learn) to the correct gesture. Convergence is essentially, the point after which the gesture is performed consistently correctly. In our data, it works out to reaching 80 - 100% success rate, and represents the speed of learning a specific gesture.

Results

Figure 3 shows the number of correct gestures (out of 120) performed by all 12 participants for the two conditions – videos and icons. A Student's t–test showed significant differences for the number of correct gestures performed. Participants performed significantly better ($p < .001$) in terms of number of correct gestures performed with videos ($M = 98.5$, $S.D = 1.87$) versus icons ($M = 83$, $S.D = 3.74$). Figure 4 shows the average number of gestures (out of 30) over the four kinds of gestures at which the participants converged to the correct gesture. A Student's t–test showed significant differences for the average converging gesture count. Participants performed significantly better ($p < .001$) in terms of average number of gestures to converge to the correct gesture with videos ($M = 5.5$, $S.D = 0.55$) than with icons ($M = 9.5$, $S.D = 1.04$). The primary reasons for not performing a correct gesture were differences in speed and direction (acceleration of the device along a particular axis to be precise). If the DTW distance (which was based on acceleration along a particular axis) was within a specified threshold, then the gesture was considered correct.

Our initial results indicate that, by only showing icons, participants take a significant amount of time to converge to the correct gesture (10 gestures on an average) and perform poorly. Videos perform significantly better (6 gestures on an average) than icons as a teaching method, but scope for improvement exists. In the next section, we describe a set of designs that support feedback on the accuracy of performing motion gestures. We also evaluate our mechanisms for teaching gestures and assessing gesture accuracy.

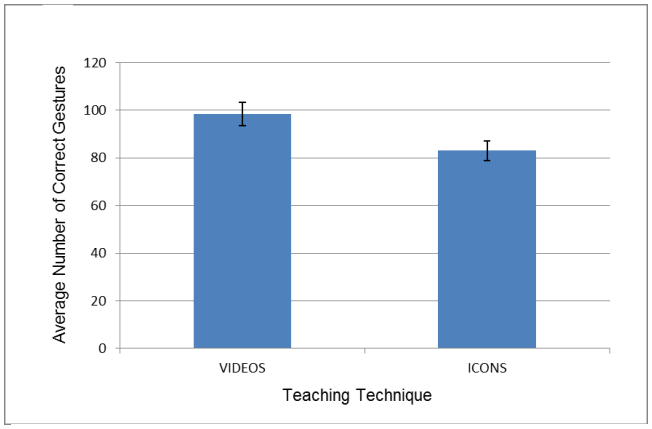


Figure 3: Average number of correct gestures (out of 120) performed by participants. 95% CI shown

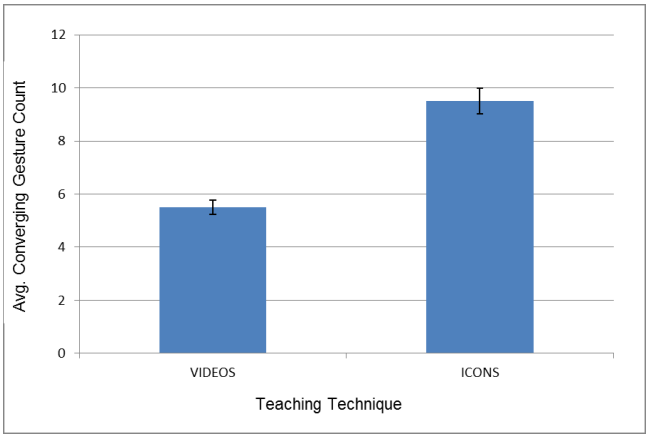


Figure 4: Average gesture count (out of 30) over the four types of gestures at which participants converged to the correct gesture. 95% CI shown

EVALUATION OF TECHNIQUES TO TEACH MOTION GESTURES

Our second user study explores additional representations of motion gestures, specifically examining the presentation of motion gestures on an external screen. We also explore additional feedback mechanisms for depicting motion gestures. These include using the Kinect to provide feedback, and also providing feedback from our recognizer using a distance metric extracted from the DTW algorithm.

Recognizer feedback design

In their work on bi-level thresholding, Negulescu et al. [13] note that, when users repeated fail to perform a motion gesture, they begin to vary the parameters of movement, attempting, essentially, to re-acquire the correct movement parameters needed to perform the motion gesture. We use the term *annealing* to describe this process of exploration.

Feedback that allows end users to assess the accuracy of a gesture exists on a continuum, from simple to more complex forms. The simplest form of feedback is some indication of correct versus incorrect from a recognizer.

However, given the annealing process of users who fail to perform motion gestures, the goal of our feedback designs was to guide users to the correct gesture, i.e. to guide this annealing process. We design two feedback mechanisms, one on smartphone and one on an external display.

For feedback on the smartphone, our approach was one of minimal feedback, as in we tried to adopt the simplest feedback we could while still guiding the annealing process. We performed a series of pilot studies to design our recognizer feedback. We began with simple, three-level textual feedback (correct, near, far), a feedback mechanism that corresponds to the children's game Cold, Warm, Hot, where someone hides an object and then guides a child's search. In early pilots, we found that the textual information was difficult to acquire and so was ignored. We also explored simple colors to provide feedback, but this, too, seemed insufficient. As a result, we moved to a numerical scale, based directly upon DTW distance.

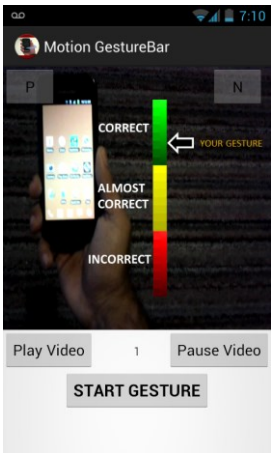


Figure 5: A screen shot of the recognizer feedback

Figure 5 shows a screen shot of the final design of our recognizer feedback mechanism. An arrow maps to the DTW distance metric of the recognizer. A feedback bar provides some basic context for the position of the arrow. The feedback bar is divided into 3 equal mini bars: Green is sufficiently close to be correct, yellow is within a loosened threshold, and red quite far from the correct template.

When feedback is enabled, after every gesture performed by the user, the recognizer displays the performance feedback bar with the arrow indicating the proximity to the gesture template. A distance of 0, i.e. a perfect gesture, would result in the arrow being positioned at the top of the green region. We continue to provide auditory feedback of recognition, specifically using three distinct sounds depending on which of the three regions the arrow points to. For gestures that pass the threshold for correctness, the bar comes up immediately after the gesture is performed. For gestures that land in the almost correct or incorrect area, the feedback bar is displayed after a pause of 2-3 seconds.

We also designed a more complex feedback mechanism that used an external screen and Kinect to provide users with a three dimensional depiction of movement and an ability to directly contrast their movement with the desired template, shown in Figure 6. Users were shown a Kinect skeleton performing a motion gesture and a video of the expert performing the gesture. As a user performed the motion gesture, the movement was displayed in an adjacent skeleton. After completing the gesture, they could replay their movement and the correct template simultaneously to identify deviations between their motion and that of the perfect template.

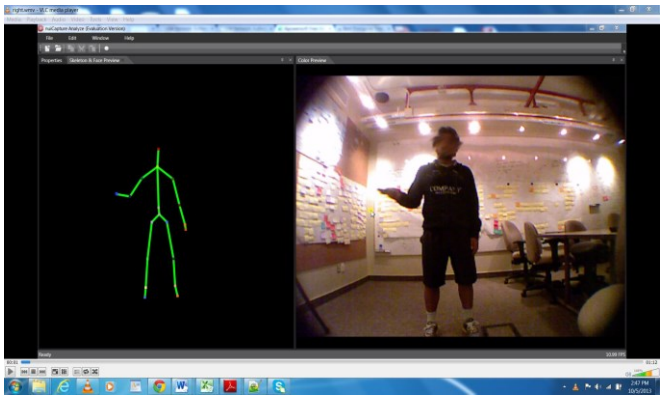


Figure 6: A screen shot of the Kinect-based teaching mechanism that captures and depicts the gestures on an external display in three-dimensional space

We used the nuiCapture Analyse (Trial Version) software [15] to capture and display the motion data from the Kinect sensor. The Kinect depth sensor provides various motion tracking views. We captured videos of the expert performing the correct gestures along with the corresponding skeletal movement as shown in Figure 6. On one PC screen, the user could see the captured video of the human and skeletal movement of the correct motion as shown in Figure 6. On another adjoining screen they could see their own motion- both skeletal and human. We could also record the motion of the participant and play it against the correct motion to compare movements.

The end result is that, for the purposes of evaluating feedback, we provide three different levels of feedback to end users: correct/incorrect (no feedback); numerical scale depicting DTW distance (DTW feedback); and full motion feedback with Kinect (Kinect). Our initial hypothesis was that Kinect feedback would best teach users to perform motion gestures. However, using the Kinect requires external hardware, whereas our other forms of training and feedback can be provided using only the smartphone device. The specific question we ask is how much worse than Kinect other forms of presentation and feedback for teaching motion gestures are.

Participants

We recruited 50 participants (23 male, 27 female, ages 20 - 35) from the general student body of institution. As in our

earlier study, we advertised the study widely to get a sample of participants with diverse backgrounds and levels of experience using computers. All participants owned a smartphone and knew what motion gestures were, but not with respect to movement of the smartphone device. Some of the participants were familiar with some hand gestures above the screen that can be performed on the Samsung Galaxy S4 Android device. All participants were remunerated with a $10 Tim Horton's gift card after the completion of the experiment.

Experimental Design

We again used a between-subjects design for this study. The rationale for choosing a between-subjects design is that, if a user masters a gesture using one technique, the evaluation of the other technique becomes invalid. In this experiment, we evaluate the following 5 motion gesture teaching techniques – icons with DTW feedback, videos, videos with DTW feedback, Kinect, and Kinect with DTW feedback. We did not evaluate icons as a teaching mechanism in this study due to their poor performance in our preliminary study.

PROCEDURE

As in the preliminary study, the gestures that the participants were asked to do were *right flick, left flick, flick towards the face* and *flick away from face*. Participants were required to perform the gesture presented to them 30 times. Thus, each participant was asked to do four blocks (corresponding to the four gesture types) of thirty gestures, i.e., 4 x 30 = 120 gestures. Each gesture block was presented to them randomly. We described the presentation and feedback mechanisms for the desired gesture, but did not provide any guidance on when or how to use feedback during the experiment. The goal of the participants was to perform as many correct gestures as possible. Given 50 participants in our study, for each of the 5 teaching/feedback mechanisms, we had 10 participants. Thus a total of 4 x 30 x 50 = 6000 gestures were performed, 1200 per feedback mechanism. In the case of the Kinect with DTW feedback, the DTW feedback, was displayed on the smartphone after each gesture. After performing all the gestures, each participant was asked to complete an exit questionnaire, followed by a semi-structured interview. The questionnaire examined the subjective preferences of our participants, and the interview was intended to obtain their opinion on motion gestures in general as an input modality for smartphones.

Metrics

As in our previous study, we capture the following measures:

No. of correct gestures: The number of correct gestures out of the total of 120 performed by each user. This is a measure of performance of the user.

Average converging gesture count: This value is the average number of gestures it took for the participant to converge (or learn) to the correct gesture. Convergence is essentially, the point after which the gesture is performed consistently correctly. In our data, it works out to reaching 80 - 100% success rate, and measures how quickly users can learn motion gestures.

Results

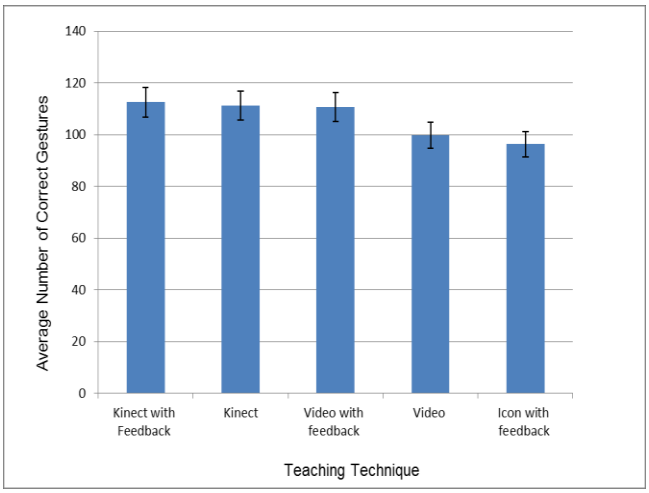


Figure 7: Average number of correct gestures (out of 120) performed by participants. 95% CI shown

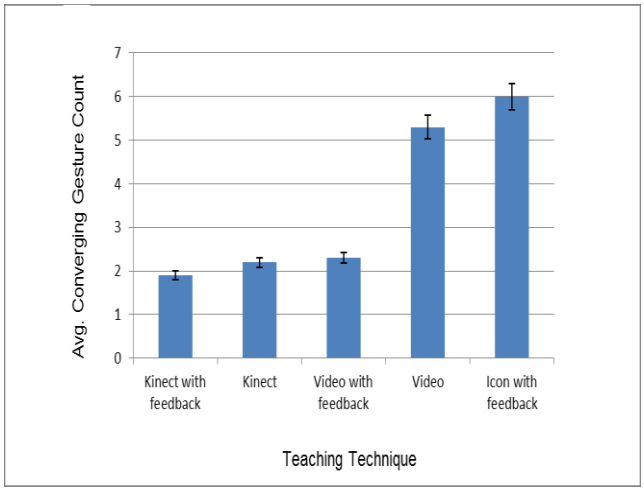


Figure 8: Average gesture count (out of 30) over the four types of gestures at which participants converged to the correct gesture. 95% CI shown

Figure 7 shows the number of correct gestures (out of 120) performed by all 50 participants for the 5 teaching mechanisms - icons with the DTW feedback, videos, videos with DTW feedback, Kinect, and Kinect with DTW feedback. A one-way analysis of variance shows that teaching technique had a significant effect on number of correct gestures performed ($F_{4,45} = 43.9$, $p < 0.001$). Post-hoc analysis using Bonferroni correction showed

significant differences between Kinect with feedback and videos ($p < 0.001$), Kinect with feedback and icons with feedback ($p < 0.001$), Kinect and videos ($p < 0.001$), Kinect and icons with feedback($p < 0.001$), videos with feedback and icons with feedback($p < 0.001$) and finally between videos with feedback and videos($p < 0.001$). Participants performed significantly better in terms of number of correct gestures with video with feedback (M = 110.7, S.D. = 3.5), Kinect (M = 111.3, S.D. = 3.4) and Kinect with feedback (M=112.5, S.D. = 2.9) than with videos (M = 99.8, S.D. = 3.5) or icons with feedback (M = 96.3, S.D. = 4.2). No significant differences were found in terms of number of correct gestures among Kinect with DTW feedback, Kinect, and videos with DTW feedback teaching mechanisms ($p = 1.00$ for all). This shows that, by combining video with DTW feedback, participants perform almost equally well as end users that learn using a Kinect or Kinect with DTW feedback.

Figure 8 shows the average number of gestures (out of 30) over the four kinds of gestures at which the participants converged to the correct gesture. A one-way analysis of variance shows that the teaching technique had a significant effect on the average converging gesture count ($F_{4,45} = 37.9$, $p < 0.001$). Post-hoc analysis using Bonferroni correction showed significant differences between Kinect with feedback and videos ($p < 0.001$), Kinect with feedback and icons with feedback ($p < 0.001$), Kinect and videos ($p < 0.001$), Kinect and icons with feedback ($p < 0.001$), videos with feedback and icons with feedback ($p < 0.001$) and between videos with feedback videos ($p < 0.001$). Participants performed significantly better in terms of average number of gestures to converge to the correct gesture with video with feedback (M = 2.3, S.D. = 1.15), Kinect (M = 2.2, S.D. = 1.03) and Kinect with feedback (M=1.9, S.D. = 0.73) than with videos (M = 5.3, S.D. = 1.2) or icons with feedback (M = 6, S.D. = 1.05). No significant differences were found in terms of the average number of gestures to converge to the correct gesture among Kinect with feedback, Kinect and videos with feedback teaching mechanisms ($p = 1.00$ for all). This shows that, by combining video with DTW feedback, participants perform equally quickly (i.e., just after 2 incorrect gestures) as end users that learn using a Kinect or Kinect with DTW feedback.

These results demonstrate the viability of training end users to perform motion gestures using only the smartphone display. In particular, given some graphical representation of distance from correct gesture and a video depicting kinematics of movement, participants performed as well and learned as quickly as participants trained using full graphical feedback via the Kinect.

Subjective preferences of exit questionnaire

We further examined the subjective preferences of our participants via an exit questionnaire. Participants were to circle choices on a Likert Scale from -3 to 3. The following questions were asked in the questionnaire:

1. How did you like the motion gesture teaching technique in this experiment? Here, a rating of -3 corresponded to very poor and 3 to very good.

2. Would you like to have motion gestures for device commands along with surface gestures? Here, a rating of -3 corresponded to least preferred and 3 to most preferred.

Figure 9 shows the results of the first question. i.e., How much did the participant like the teaching technique. A one-way analysis of variance shows that teaching technique had a significant effect on the average rating of how much the users liked it ($F_{4,45} = 6.142$, $p < 0.001$).

Figure 9: Median Likert rating from -3 to 3 for how much participants liked the teaching technique. The bars show 95% CI for median

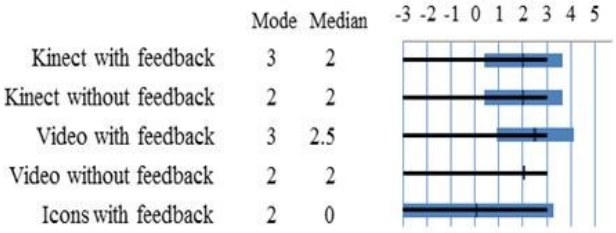


Figure 10: Median Likert rating from -3 to 3 for participant's opinion of motion gestures as an input modality along with surface gestures. The bars show 95% CI for median

Post-hoc analysis using Bonferroni correction showed significant differences between Kinect with feedback and icons with feedback ($p < 0.05$), Kinect and icons with feedback ($p < 0.05$), videos with feedback and icons with feedback ($p < 0.05$) and finally, videos and icons with feedback ($p < 0.05$). Participants gave significantly better ratings for video (M = 2.1, S.D. = 0.87), video with feedback (M = 2.4, S.D. = 1.15), Kinect (M = 1.7, S.D. = 0.96) and Kinect with feedback (M=1.9, S.D. = 1.28) than icons with feedback (M = 0.1, S.D. = 1.37). Video with feedback got the highest average ratings. One of the reasons for this could be that videos along with the recognizer feedback are much more suitable to display on the smartphone and no external display is needed.

Figure 10 shows the results of the second question. i.e., if participants would like to have motion gestures as an input modality along with touch (surface) gestures. A one-way analysis of variance again shows that teaching technique had a significant effect on the average rating of whether

participants would like to have motion gestures as an input modality along with surface gestures ($F_{4,45} = 5.045$, $p < 0.05$). Post-hoc analysis using Bonferroni correction showed significant differences between Kinect with feedback and icons with feedback ($p < 0.05$), videos with feedback and icons with feedback ($p < 0.05$) and finally, between videos and icons with feedback ($p < 0.05$). Participants gave significantly better ratings in the case of video (M = 1.5, S.D. = 1.5), video with feedback (M = 2.4, S.D. = 0.69) and Kinect with feedback (M= 2, S.D. = 1.05) than icons with feedback (M = 0.3, S.D. = 1.7)

Again, video with feedback got the highest average ratings, higher than both the Kinect-based teaching mechanisms. One of the reasons for this could be that videos along with the recognizer feedback are much more believable as a prospective teaching method on the smartphone than those that require any external display like the Kinect based-mechanisms.

User's opinions on motion gestures

We also conducted an exit semi-structured interview after each participant completed the experiment. Transcripts of the recorded interviews were used to identify common themes that emerged from our study.

Subtle gestures

16 out of 50 participants commented about the kind of motion involved in the motion gesture. A common theme that emerged was that the gesture should involve as little movement as possible.

Well, I mean motion gestures are ok and all, but I would rather use my wrist than using my arm.[P28].

The four kinds of gestures in this study did involve some lateral and vertical arm movement. Participants felt that too much arm movement in any gesture would be strenuous and might also invade an adjoining person's private space.

If I'm in a packed place or say on the bus, my arm might accidentally bump into the person next to me while doing the gesture. [P12].

Social Acceptability

36 out of 50 participants indicated their fondness for motion gestures and mentioned that, just like any new technology, motion gestures would eventually be accepted and used in public.

I don't mind these in public. I think they're pretty cool. [P21]

I think motion gestures could go mainstream really soon. It's kind of a cool new technology after all. Eventually everyone would be using them. [P40]

However a few of the participants indicated that motion gestures may become "awkward" in public places.

I would feel weird doing them in public. If all of us start doing motion gestures, it'll feel like a crazy world.[P3]

Fatigue

12 out of 50 participants mentioned that with prolonged use, motion gestures may cause some damage to the arm, especially for older people.

With prolonged use, my arms could pain and the older folks, say my grandfather, wouldn't want to do these at all. [P11]

Individual privacy

9 out of 50 participants indicated that, if motion gestures are standardized, then observers may be more aware of their actions, i.e. that the observability of motion gestures may result in a loss of privacy.

If all motion gestures are the same, your motion might indicate what you're doing. Other people might see me doing actions on the phone which I, you know, don't want to show them. [P12]

False positives/negatives

The last theme that came up from the quotes of many participants (33 out of 50) was the problem of false positives and negatives. Participants mentioned the problem of distinguishing everyday motion from motion gestures and minimizing false positives. They also said that the recognizer should be very responsive and should have a minimal false negative rate.

What if I'm like, running with the phone in my pocket or maybe stretching? Then if I accidentally start calling someone, that would be a big problem. [P35]

As we note earlier, significant past work addresses the question of balancing false positives and false negatives [13, 16].

Gamification of Recognizer Feedback

31 out of 50 participants indicated that one of the reasons they liked the DTW feedback in our study was that it challenged them to get the arrow point to the correct (green) area of the bar and as high as possible on each attempt.

I felt like, you know, I can totally do this. I just didn't want to let that arrow to drop down. It was fun. [P17]

DISCUSSION AND LIMITATIONS

Our experiments demonstrate that, as a teaching mechanism, showing a video of a desired gesture on the phone along with some feedback of how close a gesture is to optimal can effectively aid learning of motion gestures. This clearly demonstrates the viability of training end users to perform motion gestures using only the smartphone display.

We acknowledge that the four gestures in our evaluation were simple gestures, requiring only lateral or vertical motion of the phone. For complicated gestures, e.g. gestures using twists or curves, only providing feedback about how close a person is to the desired gesture may not be sufficient.

CONCLUSION

This paper addresses the challenge of teaching people to do motion gestures. Specifically, we examine two factors. The first factor is how to represent motion gestures: as icons that describe movement, video that depicts movement using the smartphone screen, or a Kinect-based teaching mechanism that captures and depicts the gesture on an external display in three-dimensional space. The second factor we examine is recognizer feedback, i.e. a simple representation of the proximity of a motion gesture to the desired motion gesture based on a distance metric extracted from the recognizer. We show that, by combining video with recognizer feedback, participants master motion gestures almost equally quickly as end users that learn using a Kinect and perform equally well.

ACKNOWLEDGEMENTS

We thank the participants in our studies. Funding for this research was provided by the Natural Science and Engineering Research Council of Canada (NSERC), the Networks of Centres of Excellence Program (NCE-GRAND), the Ontario Ministry of Innovation, and the Google Faculty Fellowship Program.

REFERENCES

1. *Android Open Source Project*. Google Inc.
2. Bau, O., and Mackay,W., "OctoPocus: A Dynamic Guide for Learning Gesture-Based Command Sets.*", Proc. of UIST'08*, 37-46.
3. Bolt, R. "Put-that-there: Voice and gesture at the graphics interface." *Proc. Computer Graphics*, 14:3, 1980, 262- 270
4. Bragdon, A., Zeleznik, R., Williamson, B., Miller, T., and Laviola, J. J., "Gesturebar: improving the approachability of gesture-based interfaces." *Proc. CHI 2009*, 2269–2278.
5. Callahan, J., Hopkins, D., Weiser, M. & Shneiderman, B.,"An empirical comparison of pie vs. linear menus", *Proc. CHI'88*, 95-100.
6. Bucolo, S., Billinghurst, M. and Sickinger, D., "User experiences with mobile phone camera game interfaces," *Proc, MUM 2005*, 87-94.
7. Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. "Sensing techniques for mobile interaction." *Proc. UIST 2000*, 91–100.
8. Jones, E., Alexander, J., Andreou, A., Irani, P., and Subramanian, S. "GesText: Accelerometer-based Gestural Text-Entry Systems." *Proc. CHI 2010*, 2173-2182.
9. Kar, B., Dutta, P. K., Basu, T. K., Vielhauer, C., Dittmann, J., "DTW based verification scheme of biometric signatures", *Proc. ICIT 2006*, 381-386.
10. Kurtenbach, G. (1993) "The Design and Evaluation of Marking Menus", *Ph. D. Thesis*, Dept. of Computer Science, University of Toronto.
11. Li, F., Dearman, D., Truong, K.N. "Virtual Shelves: Interactions with Orientation Aware Devices." *Proc. UIST 2009*, 125-128.
12. Myers, C. and Rabiner, L., "A comparative study of several dynamic time-warping algorithms for connected word recognition", *The Bell System Tech Journal 60*, 7 (1981), 1389- 1409.
13. Negulescu, M., Ruiz, J., and Lank, E., "A Recognition Safety Net: Bi-Level Thresholding for Mobile Motion Gestures", *Proc. MobileHCI 2012, 147-150.*
14. Negulescu, M., Ruiz, J., Li., Y. and Lank, E., "Tap, Swipe, Move: Attentional Demands for Distracted Smartphone Input", *Proc. AVI 2012, 173-180.*
15. *nuiCapture Analyze*, Cadavid Concepts Inc.
16. Ruiz, J. and Li, Y., "DoubleFlip: a motion gesture delimiter for mobile interaction", *Proc. CHI 2011,* 2717–2720.
17. Ruiz, J., Li, Y. and Lank, E., "User-Defined Motion Gestures for Mobile Interaction", *Proc. CHI 2011*,197 - 206.
18. Vogel, D., and Balakrishnan, R. "Distant freehand pointing and clicking on very large high resolution displays." *Proc., UIST 2005,* 33-42.
19. Wobbrock, J.O., Wilson, A.D., and Li, Y., "Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes", *Proc. of UIST '07*, ACM (2007), 159–168.