Effect of Motion-Gesture Recognizer Error Pattern on User Workload and Behavior

Keiko Katsuragawa, Ankit Kamal, Edward Lank

Effect of Motion-Gesture Recognizer Error Pattern on User Workload and Behavior

Keiko Katsuragawa[1], **Ankit Kamal**[1] **& Edward Lank**[1,2]
[1]Cheriton School of Computer Science, University of Waterloo, Ontario, Canada
[2]University of Lille, Lille, France
kkatsura@uwaterloo.ca, anktkml@gmail.com, lank@uwaterloo.ca

ABSTRACT

Bi-level thresholding is a motion gesture recognition technique that mediates between false positives, and false negatives by using two threshold levels: a tighter threshold that limits false positives and recognition errors, and a looser threshold that prevents repeated errors (false negatives) by analyzing movements in sequence. In this paper, we examine the effects of bi-level thresholding on the workload and acceptance of end-users. Using a wizard-of-Oz recognizer, we hold recognition rates constant and adjust for fixed versus bi-level thresholding. Given identical recognition rates, we show that systems using bi-level thresholding result in significant lower workload scores on the NASA-TLX and accelerometer variance. Overall, these results argue for the viability of bi-level thresholding as an effective technique for balancing between false positives, recognition errors and false negatives.

Author Keywords

Usability Testing and Evaluation; Handheld Devices and Mobile Computing; Interaction Design; Recognition; Thresholding; Gesture

ACM Classification Keywords

H.5.2. Information Interfaces and presentation (e.g., HCI): User Interfaces

INTRODUCTION

In interaction design, a significant body of work has explored free-space hand movements, i.e., motion gestures, as an input modality to computing systems such as large displays [6, 37], desktop computers [4], smart environments [20, 38] and mobile devices [5, 15, 28, 39]. While motion gestures are attractive from an end-user perspective, they present challenges to designers and developers. Specifically in the domain of smartphone-based motion gestures [15, 25, 40], researchers have explored techniques for discriminating deliberate motion gestures input from everyday movement [31], teaching appropriate movements necessary to invoke commands [17], and designing reliable recognizers [25].

In this paper, we are particularly concerned with issues of reliability in recognition for gestural input systems. Challenges include: How can one discriminate everyday movement from intentional movement, thus preventing false positives [31]? How can one control error rate [29, 36, 42]? The typical approach to limiting false positives and recognition errors in many real-world recognition systems is through criterion values [9]. These criterion values are essentially thresholds which ensure that only input that is sufficiently close to an individual category is recognized; other input is considered ambiguous and is typically left unrecognized [24]. This approach has been used in many domains – speech recognition, handwriting recognition, optical character recognition – primarily because of the high cost of being wrong. False positives in gestural input mean that systems respond without the user intending to invoke a command. Recognition errors result in systems doing the wrong thing. In both of these cases, the need to identify the error, unroll system state, and potentially repeat actions is considered significantly more costly than if the system does nothing at all. The problem, of course, is that one ends up with rejected input [24], i.e. input that was intended as an action but, because of imprecision on the part of the user, is rejected. This tension between false positives, errors, and false negatives is well-known in recognizer design [8, 24].

Systems that leverage free-space motion gestures are particularly susceptible to the challenges associated with tuning criterion values, for two reasons. First, whether using a device or using our hands for input, everyday manipulations of devices or everyday actions as we move result in natural gesturing [31, 37]. If a system interprets every hand movement, every device jiggle, as input, then forced choice recognition means systems would be firing events constantly, an effect known as the Midas Touch effect [37]. Second, the actual gestures that users wish to use when performing input compound this problem; elicitation studies have shown that these gestures correspond almost exactly to gestures in everyday movement [32]. We are left with two options. We can tighten criterion values to avoid false positives, placing a constraint of precision on the user; unfortunately, this can make it difficult for the user to perform the gesture precisely enough to fire the desired action. The other option is to loosen the threshold,

IUI 2017, March 13 - 16, 2017, Limassol, Cyprus

ISBN 978-1-4503-4348-0/17/03. . . $15.00
DOI: http://dx.doi.org/10.1145/3025171.3025234

making it easier to perform the gesture, but at the cost of false positives and recognition errors.

Given the high cost of Midas touch effects and the high cost of errors, it seems wise to default to a higher false negative rate to ensure that erroneous actions of the system are rare. There is some evidence that this choice may be acceptable to end users; in everyday interactions with computer systems, one thing that we have frequently observed is that false positives are relatively commonplace. Users frequently double-click on an icon or single-click on a button but the event does not fire; rather than become concerned, users simply perform the action again. Only after repeated failures do users become concerned that something serious may need to be addressed. It appears that the cost of a single input false negative is relatively low, and that the cost of false negatives only increases if they occur repeatedly. The natural question is, then, whether or not we can use this propensity of users, if at first they do not succeed, to try again.

To address this, we leverage a technique called bi-level thresholding [25]. The key idea behind bi-level thresholding is to look at interaction as a sequence of user actions and to leverage one aspect of context, near misses, as an additional data point in analyzing input. Specifically for gestural input systems, if a motion gesture is performed with sufficient precision, the gesture can be recognized via a classically tuned threshold. However, if a gesture is performed slightly incorrectly, a near miss, the system examines subsequent input. If another near miss gesture is observed, the gesture is recognized. We are not the first to consider bi-level thresholding as a technique to enhance interaction in gestural submissions; in a short-paper, work-in-progress submission, Negulescu et al. proposed and evaluated bi-level thresholding [25] as a means to improve recognition rate (2/3 of gesture attempts were recognized with the bi-level threshold model). The problem with their early work on bi-level thresholding is that we have little guidance on why or whether bi-level thresholding benefits users, beyond the fact that it improves recognition. More particularly, is bi-level thresholding a technique to improve overall recognition accuracy? If so, we can improve recognition in other ways, e.g. enhanced recognition, better sensors, etc. More particularly, if recognition rates are similar, is it the repeated errors or the recognition rate that causes problems? Would any equally reliable recognizer work as well? Is there any benefit to bi-level thresholding beyond the observed improvement in recognizer accuracy?

The research in this paper was motivated by a belief that bi-level thresholding may be a broadly beneficial recognition strategy, regardless of recognition rate. If we plan to incorporate computational intelligence into interfaces, we posit that it is necessary to understand the costs associated with recognition errors, reject rates, and repeated errors/rejection. To begin to address the inter-relationship between these, this paper looks specifically at reject rate versus repeated rejection. We conduct a study where we contrast two different reject strategies: One reject strategy attempts to perform the best guess possible given different criterion values; the other tightens the first criterion value, resulting in a higher initial reject rate, but then leverages dual action to lower the reject rate on second attempt. We create a wizard-of-Oz recognition system that allows us to contrast the two approaches, and demonstrate statistically significant benefits to the bi-level thresholding strategy, *even when overall reject rates are identical*.

In the remainder of this paper, after providing an overview of related work, we discuss our experimental design. We then describe the results of our study and discuss their implications.

RELATED WORK

Designing Motion Gesture Input

Much of the past research on gestural interaction has been from the perspective of gestures in support of human discourse [41]. However, free-space hand gesture interaction (e.g., as shown 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 [6] designed a "put-that-there" system in 1980 that combined pointing with voice commands. More recently, freehand interaction has been frequently explored as a modality for interacting with large vertical displays [11, 18, 37]. Multimodal interaction frequently leverages physical gestures alongside other input modalities [23, 38], and toolkits have been developed to simplify the design and deployment of gesture sets [4, 13].

When used as input to smartphones, a motion gesture leverages on board sensors such as the accelerometer and gyroscope to sense changes in orientation. In this vein, early work by Rekimoto [28] demonstrated how mapping motion to tilt can be used for selecting menu items, interacting with scroll bars, panning or zooming around a digital workspace, and performing complex tasks such as 3D object manipulation. Tilt sensors have also been used to navigate through widgets on mobile devices [28, 40]. Modern smartphones use tilt to change screen orientation, an innovation credited to Hinckley *et al.* [15]. Additionally, motion input has also been used for a variety of other input tasks, such as text input [16, 19, 27, 40], controlling a cursor [39], user verification [22], and accessing data on virtual shelves around a user [20].

Motion gestures as an input modality have been studied by the research community, particularly in work by Ruiz *et al.* [32]. Ruiz *et al.* elicited a consensus set of motion gestures for a set of smartphone tasks. In analyzing the consensus set, they noted that their participants tended to specify gestures that had low overall degrees of freedom to the movement, i.e., gestures that represented translation or rotation around a single axis (e.g. double-flip, flick-left, flick-up, etc.). As well, movements tended to exhibit low to moderate intensity in magnitude and change in acceleration, i.e., low kinematic impulse, a result of the propensity of end-users to bias toward movement profiles that minimize abrupt changes in acceleration [10]. Ruiz and Li [31] also examined everyday smartphone movement and proposed using a specialized motion gesture, the double-flip, as a delimiter for other motion gestures. The use of a delimiter partially mitigates the challenges associated with discriminating between everyday smartphone movement and intentional motion gestures, but at the expense of performing two input actions per command.

The use of motion gestures as an input modality for invoking commands on smartphones has seen some commercial success. 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, while some modern smartphones allow the user to place the smartphone face down on a desk to mute the ringtone for an incoming phone call [3] and the Moto X [2] leveraged Ruiz and Li's double-flip gesture [31] to activate the camera.

Recognizing Motion Gestures

Computational recognition of gestural input has a long history. In the domain of surface gestures, Rubine's recognizer [30] is a widely used, single-gesture recognizer that uses a simple set of geometric properties to interpret a gesture. Other variants of spatial recognizers exist, notably variants of elastic matching [35], including the 1$ recognizer [42] and Protractor [21]. At the same time, recognition of gestures need not be limited to elastic matching of spatial templates and machine learning algorithms such as Hidden Markov Models [29, 33] have also been used to interpret gestural input.

When interpreting spatial movement of a smartphone, the displacement of the phone is sensed indirectly through sensors including an accelerometer and gyroscope. As a result, input data streams provide data that is not purely spatial. While simple spatial template algorithms such as elastic matchers may be modified to match smartphone sensor data to templates, elastic matchers also assume that the start and end of a template gesture can be accurately identified. This is easy with gestures performed on a display: The gesture is delimited by an explicit pen/finger/mouse down action, and a pen/finger/mouse up action. However, with smartphones, which are always in motion and sensing, cleanly delineating between the beginning and end can be challenging. When start and end of an input signal cannot be clearly identified, there are algorithms that monitor data streams and recognize templates within those streams. Two common algorithms that have been used to recognize motion gestures on smartphones are dynamic time warping [31, 36] and HMMs [26, 29, 33].

The overall goal of any recognition algorithm is to support high precision and recall [29, 36, 42]. More specifically, we want each gesture to be correctly recognized as that gesture and no other (high precision) and we want all instances of the gesture to be identified (high recall). When characterizing the performance of recognizers, techniques used to represent precision and recall include confusion matrices and receiver operating characteristic (ROC) curves [9]. The goal of these representations is to help researchers identify correct thresholds, i.e., criterion values, to discriminate between what is a specific gesture and what is not. However, precision and recall are frequently at odds. To prevent confusion between gestures, a tighter threshold can increase precision and avoid misrecognition, but, with the tighter threshold, recall can suffer as certain gestures may not be recognized at all. In the presence of noisy input, these issues are often discussed using terms such as false positives (where a gesture is misrecognized as another or where random noise such as everyday movement is recognized as a gesture) and false negatives (where specific movement is not recognized as a gesture and is, instead, incorrectly labeled as noise or everyday movement) [25].

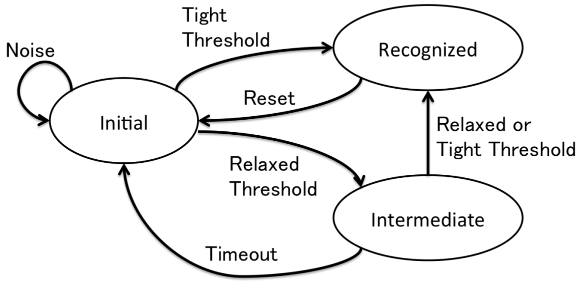

Figure 1. Be-level threshold recognizer

The most basic strategy to optimize precision and recall is to set appropriate criterion values from ROC curves such that the confusion matrix is optimized [9]. Alongside criterion values, recognizers that learn from end-users, either by manually allowing end-users to specify templates [30], by providing some form of feedback [17], or by learning [14] can be used to refine recognition algorithms on the fly. The selection or criterion values and tailored training of recognizers are complimentary and orthogonal approaches to improving recognition. In this paper, we focus specifically on criterion values and, specifically, how one mechanism to mediate between false positives and false negatives – i.e., bi-level thresholding – affects perceived efficacy of motion gesture interaction.

BI-LEVEL THRESHOLD RECOGNIZER

Bi-level thresholding is a recognition strategy that uses two thresholds, a more restrictive threshold (tight threshold) designed to limit false positives and recognition errors, and a more permissive threshold (relaxed threshold) which prevents repeated false negatives. The recognition strategy can be represented via a 3-state state machine (Figure 1). In earlier work on motion gesture input, Negulescu *et al.* [26] explored the utility of motion gestures as an eyes-free input technique. In their study, they noted one problem with motion gesture input: repeated errors seemed to cause particular difficulty for end-users performing motion gestures while walking. In particular, they note that "participants speed was particularly affected by recognition rate" [26]. Motivated by this observation, and by the high false negative rate that they observed in their motion gesture recognizers, they proposed a novel recognition strategy called bi-level thresholding[25].

The recognizer begins in an Initial state. Most sensor data received by a smartphone is simply noise, i.e., everyday device movement, and this everyday movement does not cause a state change. From the Initial state, if the recognizer observes a movement which exceeds the tight threshold for a candidate gesture in the template library, the system moves to the Recognized state and the gesture is recognized. If, in contrast, candidate movement exceeds the relaxed threshold for a template gesture, the system moves to an Intermediate state. In this state, if the system receives either a tight threshold or relaxed threshold input for the same gesture, the system moves to the Recognized state and the gesture is recognized.

If no such gesture occurs the system moves back to the Initial state after a timeout, set to 3 seconds in our implementation.

Bi-level thresholding is designed to protect against these repeated failures while preserving a tight initial threshold that prevents false positives and recognition errors by ensuring that the system has high confidence in any inference. In early, offline experimental results [25], bi-level thresholding seemed to enhance recognition: 95.3% of gestures were recognized with bi-level thresholding. Only 35% of gestures would have been captured within two attempts using a single, tight threshold (25% on first attempt and an additional 10% on second attempt).

THE EXPERIMENT

To assess the usability of the bi-level threshold recognizer, we conducted an experiment that evaluated bi-level thresholding (BL) against fixed-level thresholds (FL). We simulated 3 levels of recognition rates: 50%, 60% and 70%.

The design of our study is inspired by the work of Negulescu *et al.* on eyes-free motion gesture interaction [25], which evaluated whether motion gestures support eyes-free input by designing a study where participants interacted with a smartphone while walking. To replicate low cognitive workload scenarios, e.g. walking a familiar path, we asked the participants to walk while they are performing the motion gesture task.

Recognition strategy, i.e., fixed-level vs. bi-level, was a 'within-subjects' measure, as this was our most salient data with respect to our research question. Recognition rate was a 'between-subject' factor to avoid biasing toward higher recognition rates, i.e., to tease out the effect of recognition strategy. As a result, our experimental design consisted of a 2x3 mixed design with bi-level/fixed-level threshold (BL/FL) as a within subjects factor and recognition rate (50, 60, 70) as between subjects factors.

Participants

We recruited 36 participants (24 male, 12 female, ages 20 -39) 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. 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 $10 after the completion of the experiment.

Apparatus

Our experimental software was developed in Java using the Android SDK [1]. Software ran on Nexus 5 phones with a 2.26 GHz quad-core Krait 400 processor and a three-axis accelerometer and gyroscope. The Android version used was KitKat 4.4.4.

Measures

Experimental measures varied per between-subject factors. For all subjects, we captured cognitive workload measures using the NASA-TLX Weighted Workload (WWL), as per standard practice [7, 12, 43]. Additionally, we administered a short questionnaire at the end of the study asking participants to assess their performance and the system performance with the two recognition strategies. There were eight 10-point Likert questions on the survey (similar in presentation to the NASA-TLX questions participants completed after each session) and questions that asked them to compare the two sessions. The eight Likert questions included how much participants liked each system, how easy the system was to use, whether the system was fun to use, how comfortable the participant was, how relaxed the participant was, whether the system was stable, and, subjectively, how well they performed. The comparison questionnaire included their preference for session one or two, if they perceived any difference between the two sessions, whether they performed differently in the two sessions (and why), and whether the application performed differently in the two sessions (and how).

Alongside these questionnaires, we collected quantitative data on walking speed. This data was collected using a stopwatch to measure time between marked points on the floor as participants walked.

Task

During experimental sessions, we used five gestures in our experiment - right flick (1), left flick (2), flick up towards face (3), flick down away from face (4) and double flip (5), again mimicking Negulescu et al. [25] and Kamal and Lank [17]. Each gesture was performed ten times per session, yielding 50 gestures per session. The order of gestures displayed to the participant was randomized within the session. The images of the five gestures are shown in the Figure 2.

The gestures were drawn directly from the consensus set of motion gestures obtained through an elicitation study by Ruiz *et al.* [32]. The rationale for the selected gestures in previous experimental evaluations is that this subset represents the simplest set of useful motion gestures for smartphone control [32]. Nominally, the gestures correspond to next, previous, zoom-in, zoom-out and mode switch (delimiter) gestures, labeled 1 through 5 respectively in Figure 2. We chose these gestures both because they represent a useful subset of potential commands issuable via motion gestures and to preserve consistency with other studies [32].

Recognizer

To discriminate between a deliberate gesture and noise, we used a simple threshold recognizer. Two expert users iteratively decided on the threshold values for each of the 5 gestures within the recognizer. We found that, for our experiment where participants were constantly performing motion gestures, a permissive threshold ensured that participants would need to perform a reasonable action to activate the threshold, but the thresholds were sufficiently permissive that we observed no false negatives. On the other hand, if used in practice these permissive thresholds would result in a high false positive rate.

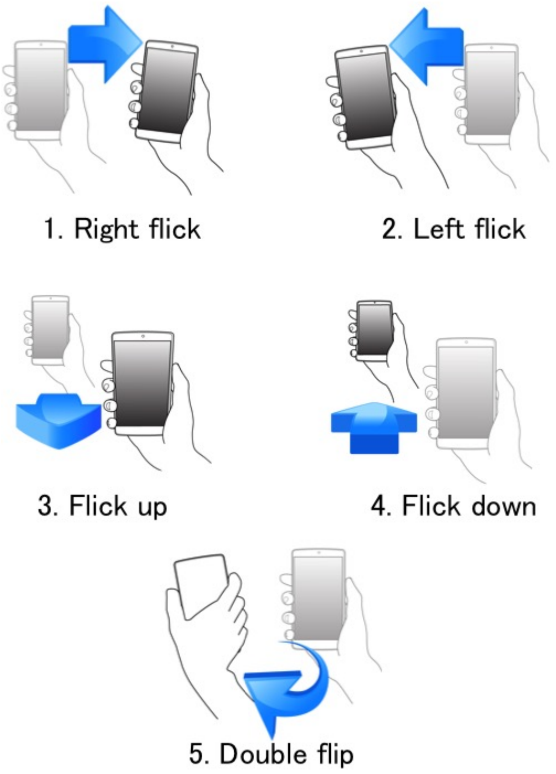

Figure 2. Five gestures for the experiment

The thresholds were only appropriate for a situation where a simulated recognition algorithm was being used.

To control error rate, we used a wizard-of-Oz technique. Gestures that exceeded our base-level threshold for a gesture (tuned above) were either correctly or incorrectly recognized. We controlled error rate by using a probability function in conjunction with a count of gesture attempts to equalize the number of attempts that participants needed to perform 50 successful motion gestures.

To understand how recognition rate was equalized, consider Table 1. Columns 1 to 6 show the number of gestures attempts and their corresponding row values indicate the frequency for each attempt. For example, for a recognition rate of 70% and for the fixed threshold recognizer, recognition was reported as correct on the first attempt 35 times, correct on the second attempt 11 times, correct on the third attempt 3 times and correct on the fourth attempt once. This gives a total of 35x1 + 11x2 + 3x3 + 1x4 = 70 gesture attempts and 20 unsuccessful gesture attempts (errors). Similarly for the 70% case and bi-level threshold recognizer, there were a total of 30x1 + 20x2 = 70 gesture attempts and 20 unsuccessful attempts. As a result, 50 correct gestures were recognized out of 70 gesture attempts, giving an overall recognition rate of 50/70 or 71.4% for both the fixed-level and bi-level recognition strategy.

The error rate was controlled over all gestures, but we took some pains to ensure balanced error rate per individual gesture as well. So, for example, in the 70% case and bi-level threshold recognizer condition, as first attempts, 6 out of 10 gestures are successfully recognized for each gestures.

	Required attempts						total attempt
	1	2	3	4	5	6	
70%-Fixed	35	11	3	1			70
70%-Bi-level	30	20					70
60%-Fixed	30	12	5	2	1		82
60%-Bi-level	18	32					82
50%-Fixed	25	13	6	3	2	1	97
50%-Bi-level	0	50					100

Table 1. The number of required attempts and frequency corresponds its recognition rate.

The 50% case is particularly interesting from the perspective of bi-level thresholding. To preserve parity in recognition rates (so that we could determine whether overall recognition rate or bi-level thresholding was most effective at enhancing usability) participants performed 50 correct gestures out of approximately 100 gesture attempts. For the bi-level case, this means that, for 50% recognition, the gesture was always reported as incorrect on the first attempt and correct on the second attempt.

While we could have chosen different error rates, for the length of our study, we were reluctant to raise the error rate above 70% for two reasons. First, at 80%, it becomes highly unlikely that more than two attempts are needed to recognize a gesture: at 80% recognition, only two gestures would have used a third attempt, making 80% virtually identical for bi-level and fixed-level thresholding. As well, past experience preventing false positives in motion gesture input results in first-instance recognition rates that are closer to 30% [25], not the 50% rate that is the lowest recognition rate we use in this study. Given that gestures are often single-axis movements with low kinematic impulse, we find it unlikely that recognition rates would ever reach as high as 80% for first-instance recognition without resulting in prohibitively high false-positive rates.

Study Procedure

The study lasted approximately 30 minutes for each participant and consisted of three sessions – a training session, a fixed-level session and a bi-level session. The order of the fixed versus bi-level session was counterbalanced between participants. After each gesture attempt where the simulated recognizer reported a correct result, a check mark was displayed on the screen for 1 second. Following the check mark, there was a 3 second pause before the next task (gesture) was presented as an image.

Before the study began, we conducted a briefing where detailed instructions about the study were communicated to the participants. During this briefing, we informed participants that they would perform a training session and two experimental sessions, but did not inform them that the recognizers were different in the two experimental sessions.

This was followed by collection of participants' baseline measurements. The baseline was just a measure of normal walking

	Fixed	Bi-level	F	p
WWL	34.3	29.2	4.478	<.05
Mental Demand	25.1	25.2	0.003	.96
Physical Demand	31.1	29.5	0.267	.609
Temporal Demand	31.7	28.3	1.371	.250
Performance	28.9	19.2	8.352	<.01
Effort	33.0	28.0	2.198	.148
Frustration	32.8	29.1	1.304	.261

Table 2. MANOVA result of NASA-TLX scores.Values of mean are shown for weighted workload (WWL) and for components of the TLX

speed to and from two specified points, 14.5 meters apart. We had spotters to ensure participants' safety while walking.

Participants then performed the experimental sessions. At the end of each experimental session, participants completed the computerized version of the NASA-TLX on a laptop computer (which took approximately two minutes).

At the end of the study, a three minutes questionnaire asked them to compare the sessions. We then conducted a short, semi-structured interview assessing their perspective on the recognizers and motion gestures as a smartphone input modality.

Analysis

We analyze data to examine the relative effects of recognition strategy on measures collected during experimental sessions (walking speed, magnitude of the gesture) and on measures collected post-hoc (NASA-TLX scores, relative preferences.

RESULTS

Measures Collected During Experimental Sessions

There was no statistically significant difference in Walking speed in our study. Interestingly, this result is at odds with observational data reported by Negulescu *et al.* that motivated the design of the bi-level thresholding strategy [25]. Specifically, they note that, if repeated errors occur, participants slow down, stop, and try to diagnose errors. In both our study and the Negulescu *et al.* study, the errors that occurred were false negatives, i.e., the recognizer would fail to correctly recognize a motion gesture. In Negulescu *et al.* thresholds for recognition were tightened such that misrecognitions (i.e., the recognition of one gesture as another) did not occur [25]. Thus, we are unable to validate the qualitative observations of Negulescu *et al.* on the effect of error-prone recognition on walking.

Post-Hoc Measures (NASA-TLX, Questionnaire Data)

NASA-TLX

Table 2 depicts the weighted workload (WWL) scores for the NASA-TLX for recognition strategy. A three-way MANOVA of between-subjects and within-subjects effects for threshold strategy (bi-level versus fixed-level) and recognition rate (50%, 60%, and 70%) was performed, identifying an interaction between recognition rate and WWL. As shown in the first row of Table 2, the WWL shows statistically significant differences, with lower workload for bi-level compared to the fixed-level threshold recognizer (F(1,33)=4.478, p <.05).

	Fixed	Bi-level	Z	p
Likable	5.3	5.7	-1.189	.234
Easy to use	5.6	6.1	-1.704	.088
Fun to use	4.7	4.8	-.118	.906
Comfortable	5.3	5.6	-.945	.344
I felt relaxed	5.4	6.0	-1.632	.103
App was stable	5.6	6.3	-2.055	<.05
I performed well	6.5	7.2	<.01	<.005
App performed well	5.3	5.8	-1.402	.161

Table 3. Wilcoxon rank test result of Likert questionnaire.

	70%	60%	50%	Total
Fixed prefered	1	2	4	7 (19.4%)
Be-level prefered	6	9	7	22 (61.1%)
Indifference	5	1	1	7 (19.4%)

Table 4. The number of participants who preferred each session.

In Table 2, we also report component scores for the various elements of the NASA-TLX - mental demand, physical demand, temporal demand, performance, effort, and frustration - as is common in the HCI literature [7, 43]. MANOVA indicates that significant differences exist for Performance F(1,33) = 8.352, p <.01. There was no statistically significant effect of recognition rate.

Questionnaire Data

To triangulate our NASA-TLX data with participants overall impressions, we also collected Likert data assessing the two systems (as described earlier) at the end of the study. Mean values are shown in Table 3. To analyze the influence of recognition strategy, a Wilcoxon Signed-Rank test was performed. Recognition strategy had a significant effect on the perception of applications stability and performance although overall number of the error was same.

Participants' preference broken out by two recognizers is depicted in Table 4. The majority of participants (n = 22, 61.1%) chose the session with the bi-level threshold (BL) recognizer as their preferred session, followed by fixed threshold (FL) (n = 7, 19.4%) and Indifference (EL) (n = 7, 19.4%). A Chi-square test showed that there was a significant difference in these numbers ($\chi 2$ (2) = 12.5, p <.01, significant difference was shown between BL-FL and BL-EL). This result suggests that, if the overall recognition rate is the same, the bi-level threshold recognizer is more preferred than the fixed threshold recognizer.

While some participants were aware that they were using two different artificial recognizers, we did not provide any additional indication on whether one was better or worse than the other. We probed participants to see if they noted a difference between the two sessions. In the 70% recognition rate case,

	70%	60%	50%	Total
Difference perception	5/12	7/12	11/12	23/36 (63.9%)

Table 5. The proportions of participants those observed any difference between two sessions on each recognition rate

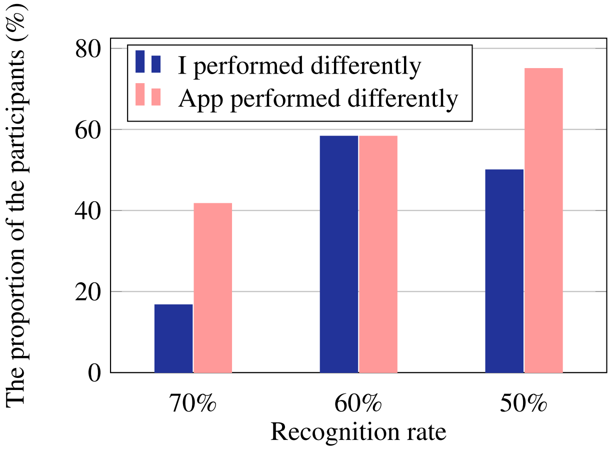

Figure 3. The proportion of the participants who perceived the difference either on the application or their performance.

more than half (58.3%) of the participants declared they did not observe any difference between the two sessions (Table 5). As recognition rate decreased, however, we saw an increase in the number of participants who noted a difference between two sessions. Overall, only 23 out of 36 participants noticed a difference in the two sessions, yet 80.5% (29 out of 36) had a preference for one session of the experiment.

Differences between the two sessions could be attributed either to differences in the application or in the participants themselves (i.e., they performed better or worse). On probing, 15 out of 36 participants thought they themselves made the difference in each session (Figure3), perhaps because their performance improved later in the experiment or their fatigue increased. 21 declared the application performed differently and 8 out of 21 participants thought both of the application and they themselves made the difference.

A two-way ANOVA of between-subjects and within-subjects effects for the perceived difference, recognition rate was performed. ANOVA indicates that significant differences exist for recognition rate (F (2,33)=4.892, p<.05) but there was no significant differences in the perceived cause of the difference.

Observations

Repeated errors were present in our system. In the 50% case, for example, with bi-level thresholding, all gestures were recognized on the second attempt, whereas with fixed thresholding, half were recognized on a first attempt, one quarter on the second, and so on. One thing we did note on repeated attempts was a variation in user behavior. A single failure had limited effect on participants; they would simply try again. However, subsequent failures caused participants to begin to vary in unanticipated ways their input patterns by holding the phone differently, moving at a different angle, or shaking the phone to 'reset' recognition. We call this process of varying input movement *annealing*, a term borrowed from simulated annealing in optimization. Essentially, participants are randomly varying attributes of their movement to better explore the search space. While we did not see an effect of this behavior on walking speed (perhaps because, despite conducting the experiment in a public place, we were careful to help participants avoid collision as per requirements from our research ethics office), one question we had was whether any quantitative measures might triangulate with this qualitative observation of annealing.

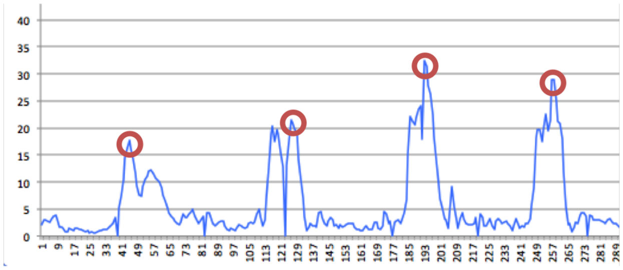

Figure 4. Magnitude of accelerator peak of repeated "Left Flick" gesture as a function of time. Peaks of the magnitude of each gesture are marked with red circle.

To explore this quantification of the annealing process, we examined a series of measures from the accelerometer. Using a multivariate ANOVA for accelerometer measures, we found a statistically significant difference for accelerometer variance. To understand this measure, consider Figure 4, which shows the peak magnitude or acceleration of an input gesture. The accelerometer magnitude is the square root of sum of squares of x,y and z axis of accelerator value, i.e. the Euclidean norm of x, y, and z accelerometer values. Although there was no significant difference in overall average magnitude of the accelerometer, MANOVA indicates that significant differences exist for standard deviation of the magnitude of the peak F(1,70) = 4.546, p <.05. and the difference between the minimum and maximum of the peak F(1, 70) = 4.371, p <.05.

To understand how this result triangulates with the qualitative observation of the annealing process, consider what happens when people try to vary input. They vary their movement patterns, their acceleration, and the force with which they perform the gesture. Changes in the way they accelerate (slower and more carefully, more firmly, etc.) will create higher variance and higher differences between min and max acceleration.

Interestingly, in our system, because of the wizard-of-Oz nature of the recognizer, annealing had no effect on performance. In the real world, it may be possible to leverage this annealing process in training systems. For example, if we know that kinematics of the gesture is off, perhaps because the gesture was performed too softly, telling the user to gesture more forcefully would be a welcome feedback. Similarly, if the motion path was incorrect, providing guidance on how to vary the path of movement would also be welcome, as users could then correct their input both in the present and in future interactions.

DISCUSSION

Revisiting our results, we note that bi-level thresholding places statistically significantly lower subjective workload on our participants and accelerometer variance. The participants prefer the bi-level thresholding strategy more.

However, walking speed did not reveal statistically significant differences. One reason that we may not have seen an effect on walking speed is that, overall, our recognizers did have relatively high performance compard to the 25% recognition rate in Negulescu *et al.* [25] using a tighter criterion value. When recognition is this poor, repeated error streams will be even longer than the 6-attempt limit in our worst-case 50% recognition rate and more than two attempts will be required for 60% of the gestures, compared to only 25% of the gestures at 50% recognition rate in our study when not using bi-level thresholding. We understand the reason the recognizers in the Negulescu *et al.* study performed so poorly was the similarity of walking movement to motion gesture input. However, we have no way to disentangle the effects of more reliable recognition from recognition strategy in Negulescu *et al.*. Our results lend support to the observation that the cost of repeated errors is disproportionately more important than the overall error rate of a system.

In many ways, the success of bi-level thresholding in our work remains surprising. Consider that each of our participants performed exactly the same number of gesture attempts in both sessions of our experiments. There are no physical workload benefits to the bi-level thresholding condition in our experiment because, at any one recognition rate, participants still perform exactly the same number of gesture attempts to complete 50 gestures. It simply is the case that, with bi-level thresholding, they fail more frequently on the first attempt – only 22% first-attempt recognition accuracy at 60% recognition rate with the bi-level strategy and 0% first-attempt recognition accuracy at 50% recognition – but achieve more reliable first or second attempt recognition than does a user providing input to a system with a fixed-level threshold.

Implications for Interaction Design

Usability Improvement with Bi-Level Thresholding

In many situations, designers and developers need to make difficult decisions about allocating resources to improve systems. In the case of motion gestures, one of the trade-offs developers must make is whether to improve overall recognition or simply to improve systems such that repeated errors become less frequent. While doing both would undoubtedly be the ideal, in the real world resources are often tight and designing better recognition algorithms is quite challenging.

Given the need to trade-off limited resources, exploring additional ways to guard against repeated errors may prove an effective long-term solution to enhancing the perceived reliability of recognition algorithms for motion gestures. Our results demonstrate that you can both enhance user satisfaction and improve overall recognition rates by considering any candidate motion in the context of movement immediately preceding or following the candidate motion. Simply leveraging our natural inclination to try again on initial failure (If at first you don't succeed ...) yields improvement in cognitive workload measures and participant preference.

Viable Prevention of False Positives

Errors of commission (false positives) can be very costly in user interfaces. In many ways, a false negative simply requires that a user try again, whereas a system that performs an incorrect action requires that the user determine that an incorrect action was performed, undo that incorrect action, and then try to perform his or her desired action again [34]. One of the tensions in intelligent interaction design is the trade-off that must occur when selecting criterion values: Sufficiently tight that false positives are rare, but sufficiently loose that false negatives are not prohibitively high is the rule of thumb. Given the positive workload scores for bi-level thresholding, our results add support to the argument, first made by Negulescu *et al.* [25], that it may be possible to satisfy both limiting false positives and limiting false negatives through a more restricted criterion function for first attempts followed by a looser function for subsequent attempts. It seems theoretically possible that such a strategy may represent the 'best of both worlds'.

To understand this 'best of both worlds' commentary, consider a scenario where, with basic tuning, the false positive rate is a value x' within the timeout interval of the bi-level recognizer shown in Figure 1, for example, a 2% likelihood of false positives over a 5-second timeout. Using these numbers, then one would expect that over approximately four minutes of use, one false positive would occur. If, instead, bi-level thresholding were used, one could use a much lower criterion value. To preserve an equivalent false negative rate over two attempts, the likelihood of false positives on first attempt would drop to 0.04% (0.02^2). One would, therefore, expect one false positive over every 2500 seconds of use, or one every 40 minutes.

Broader Implications Computational Intelligence In Input

Why is then a bi-level thresholding technique not used more frequently in real world recognition systems? This is a question that we do not know the answer to. Based on our results, it appears that, in any domain where computational intelligence is used to interpret input (sketch recognition, assistive technologies, speech input, freehand gestures), bi-level thresholding could be a viable recognition strategy for improving user satisfaction.

We believe our results argue for a much more agressive exploration of tuned rejection in recognition systems based upon user perception, rather than just analyzing accuracy or workload optimization [24]. Specifically, given that one problem with computational intelligence in interaction is the challenge of managing false positives and recognition errors against the negative cost of reject rates, what we have shown here is that reject rates may have the potential to be much more aggressively leveraged to enhance the *perceived reliability* of intelligent interactive systems. Consider, our 50% case above: Even with 0% first instance accuracy (versus 50% in the competing case), participants preferred bi-level thresholding. This high tolerance for first-instance failure may be very empowering for designers of intelligent user interfaces: error can be aggressively limited through reject rates, while still keeping user satisfaction high through an interpretation of input in the context of the input stream.

Convergence

One thing that was interesting for us was how attuned participants were to the improved performance of bi-level thresholding. In psychology experiments, to test whether there is an awareness of difference, the most common test used is a *Just Noticeable Difference* test. The idea behind this test is to ask participants to choose the best option between competing options. If there is no difference in the conditions, then one would expect the all conditions to perform approximately the same; if there exists a sufficient difference to be noticeable, even subconsciously, then one would see one category being selected much more frequently than another.

In our data, even for our highest recognition rate, we see a strong bias among participants in favor of bi-level thresholding (by a 2:1 margin). This bias is particularly strong given the presence of an "indifference" option. Furthermore, additional evidence exists for ecological validity in our experiment: In Figure 4, we see that, as recognition rate increases, participants do not know exactly why they found one system better than another (i.e., equal numbers of participants blamed their bias on the app's performance and their own performance).

LIMITATIONS

In our study, one challenge with generalizing results is that the task was quite simple and may not be fully ecologically valid. Participants were cued and then performed a specific motion gesture. In real-world use, participants may be more concerned about reliability of interaction than our participants.

Another challenge with any study design that leverages a wizard-of-Oz system is the confound introduced by the lack of a real world recognizer with real world failures. Fortunately for us, our participants seemed unaware of the wizard-of-Oz nature of our experimental study: No participants obviously tried to game the system by seeing how badly they could perform on a second attempt. No participant decided to simply not do anything at all because the recognizer results were canned.

Finally, we should note that, beyond limitations, there are significant strengths to our study design from the perspective of valid hypothesis testing. In particular, note that our study design unfairly penalizes bi-level thresholding. To understand this point, consider first that with bi-level thresholding, recognition rate would increase significantly because fewer second-instance rejects occur. In the small scale study of Negulescu *et al.*, a 23% recognition rate increased to over 70% via bi-level thresholding. In our analysis of raw recognizer behavior, we find that a 70% recognition rate would increase to above 75% and a 50% recognizer's accuracy would increase to approximately 67%. In our study, to keep gesture attempts constant (i.e., to keep the workload constant), we penalized first-instance success when using bi-level thresholding. For 70% recognition, first-instance success with bi-level thresholding is only 60%; for 50% recognition, first-instance success with bi-level thresholding is 0%. In other words, we eliminate the recognizer benefits associated with bi-level thresholding because those are vacuously true. Our quest was to determine whether the benefit was due to improved recognizer accuracy, or if it was due to reduced repeated error. Improved recognizer accuracy is always a benefit. Our results argue that removing repeated error is significantly more beneficial than just improving recognizer accuracy.

CONCLUSION

Overall, the moral to be drawn from this research is simple: If a user's input is a near miss to something that may be a specific command, then that near miss provides valuable information which can be used to enhance the perceived reliability of recognition-based interactions. In this work, we show that, by doing this, we lower the mental workload of end-users and increase their satisfaction, even when the overall number of attempts they make to perform actions remains constant.

The significant effect we see on mental workload remains surprising because, in our experimental design, bi-level thresholding did not save any physical effort. Participants still performed exactly the same number of gesture attempts, but the reduced first-attempt reliability was more than offset by the enhanced second-attempt reliability.

We feel that the overall benefit to these results is specifically in the perceived reliability of interfaces that incorporate recognition algorithms. Overall, the promise seems to be that we can be slightly more aggressive in preventing false positives while leveraging near misses to prevent repeated false negatives.

ACKNOWLEDGMENTS

The authors thank the Natural Sciences and Engineering Research Council of Canada (Discovery Grant Program) and Google's Faculty Fellowship Program for funding this research.

REFERENCES

1. Android studio. `https://developer.android.com/studio/index.html`.

2. Motox. `https://motorola-global-portal.custhelp.com/app/answers/prod_answer_detail/a_id/96251/p/30,6720,8696`.

3. Smart profile. `https://motorola-global-portal.custhelp.com/app/answers/detail/a_id/51540/~/droid-x---smart-profile`.

4. Ashbrook, D., and Starner, T. Magic: A motion gesture design tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, ACM (2010), 2159–2168.

5. Bartlett, J. F. Rock 'n' scroll is here to stay. *IEEE Comput. Graph. Appl. 20*, 3 (May 2000), 40–45.

6. Bolt, R. "Put-that-there". In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques - SIGGRAPH '80* (1980), 262–270.

7. Dai, L., Sears, A., and Goldman, R. Shifting the focus from accuracy to recallability: A study of informal note-taking on mobile information technologies. *ACM Trans. Comput.-Hum. Interact. 16*, 1 (Apr. 2009), 4:1–4:46.

8. Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern classification*. John Wiley & Sons, 2012.

9. Fawcett, T. An introduction to roc analysis. *Pattern Recogn. Lett. 27*, 8 (June 2006), 861–874.

10. Flash, T., and Hogans, N. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of neuroscience 5* (1985), 1688–1703.

11. Haque, F., Nancel, M., and Vogel, D. Myopoint: Pointing and clicking using forearm mounted electromyography and inertial motion sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, ACM (2015), 3653–3656.

12. Hart, G. S., and Staveland, E. L. Development of nasa-tlx (task load index): results of empirical and theoretical research. *Human Mental Workload 52* (1988), 139–183.

13. Hartmann, B., Abdulla, L., Mittal, M., and Klemmer, S. R. Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, ACM (2007), 145–154.

14. Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*, 2 ed. 2009.

15. Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. Sensing techniques for mobile interaction. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, ACM (2000), 91–100.

16. Jones, E., Alexander, J., Andreou, A., Irani, P., and Subramanian, S. Gestext: Accelerometer-based gestural text-entry systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, ACM (2010), 2173–2182.

17. Kamal, A., Li, Y., and Lank, E. Teaching motion gestures via recognizer feedback. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, ACM (2014), 73–82.

18. Katsuragawa, K., Pietroszek, K., Wallace, J. R., and Lank, E. Watchpoint: Freehand pointing with a smartwatch in a ubiquitous display environment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, ACM (2016), 128–135.

19. Katsuragawa, K., Wallace, J. R., and Lank, E. Gestural text input using a smartwatch. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, ACM (2016), 220–223.

20. Li, F. C. Y., Dearman, D., and Truong, K. N. Virtual shelves: Interactions with orientation aware devices. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, ACM (2009), 125–128.

21. Li, Y. Protractor: A fast and accurate gesture recognizer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, ACM (2010), 2169–2172.

22. Liu, J., Zhong, L., Wickramasuriya, J., and Vasudevan, V. User evaluation of lightweight user authentication with a single tri-axis accelerometer. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, ACM (2009), 15:1–15:10.

23. Mignot, C., Valot, C., and Carbonell, N. An experimental study of future &ldquo;natural&rdquo; multimodal human-computer interaction. In *INTERACT '93 and CHI '93 Conference Companion on Human Factors in Computing Systems*, CHI '93, ACM (1993), 67–68.

24. Nagy, G. 29 optical character recognitionâĂŤtheory and practice. *Handbook of statistics 2* (1982), 621–649.

25. Negulescu, M., Ruiz, J., and Lank, E. A recognition safety net: Bi-level threshold recognition for mobile motion gestures. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, ACM (2012), 147–150.

26. Negulescu, M., Ruiz, J., Li, Y., and Lank, E. Tap, swipe, or move: Attentional demands for distracted smartphone input. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, ACM (2012), 173–180.

27. Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., and Want, R. Tilttype: Accelerometer-supported text entry for very small devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*, UIST '02, ACM (2002), 201–204.

28. Rekimoto, J. Tilting operations for small screen interfaces. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, ACM (1996), 167–168.

29. Rigoll, G., Kosmala, A., and Eickeler, S. High performance real-time gesture recognition using hidden markov models. In *In Proc. Gesture Workshop*, Springer (1998), 69–80.

30. Rubine, D. Specifying gestures by example. *SIGGRAPH Comput. Graph. 25*, 4 (July 1991), 329–337.

31. Ruiz, J., and Li, Y. Doubleflip: A motion gesture delimiter for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, ACM (2011), 2717–2720.

32. Ruiz, J., Li, Y., and Lank, E. User-defined motion gestures for mobile interaction. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, ACM (2011), 197–206.

33. Sezgin, T. M., and Davis, R. Hmm-based efficient sketch recognition. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, IUI '05, ACM (2005), 281–283.

34. Small, D., and Ishii, H. Design of spatially aware graspable displays. In *CHI '97 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '97, ACM (1997), 367–368.

35. Uchida, S., and Sakoe, H. A survey of elastic matching techniques for handwritten character recognition. *IEICE - Trans. Inf. Syst. E88-D*, 8 (Aug. 2005), 1781–1790.

36. Vintsyuk, T. Speech discrimination by dynamic programming. *Kybernetika* (1968).

37. Vogel, D., and Balakrishnan, R. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, ACM (2005), 33–42.

38. Voida, S., Podlaseck, M., Kjeldsen, R., and Pinhanez, C. A study on the manipulation of 2d objects in a projector/camera-based augmented reality environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, ACM (2005), 611–620.

39. Weberg, L., Brange, T., and Hansson, A. W. A piece of butter on the pda display. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '01, ACM (2001), 435–436.

40. Wigdor, D., and Balakrishnan, R. Tilttext: Using tilt for text input to mobile phones. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, ACM (2003), 81–90.

41. Wobbrock, J. O., Morris, M. R., and Wilson, A. D. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, ACM (2009), 1083–1092.

42. Wobbrock, J. O., Wilson, A. D., and Li, Y. Gestures Without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, ACM (2007), 159–168.

43. Yesilada, Y., Stevens, R., Harper, S., and Goble, C. Evaluating dante: Semantic transcoding for visually disabled users. *ACM Trans. Comput.-Hum. Interact. 14*, 3 (Sept. 2007).