

HUMAN-COMPUTER INTERACTION

A Journal of Human-Computer Interaction and Information Systems

Volume 25, Number 4
October-December 2013



Nomadic input on mobile devices: the influence of touch input technique and walking speed on performance and offset modeling

Journal:	<i>Human-Computer Interaction</i>
Manuscript ID:	HCI-2014-1674.R4
Manuscript Type:	Article
HCI Keyword Taxonomy:	Graphical UI < USER INTERFACE, UI Input < USER INTERFACE, Intelligent UI < USER INTERFACE, Mobile HCI <AD-HOC KEYWORDS>, Nomadic input <AD-HOC KEYWORDS>, Offset models <AD-HOC KEYWORDS>, Gait phase <AD-HOC KEYWORDS>, Performance < HCI THEORY

SCHOLARONE™
Manuscripts

Nomadic input on mobile devices: the influence of touch input technique and walking speed on performance and offset modeling

Running Head: Nomadic input: performance parameters and offset modeling

ABSTRACT

In everyday life people use their mobile phones on-the-go with different walking speeds and with different touch input techniques. Unfortunately, much of the published research in mobile interaction does not quantify the influence of these variables. In this paper, we analyze the influence of walking speed, gait pattern and input techniques on commonly used performance parameters like error rate, accuracy and tapping speed, and we compare the results to the static condition. We examine the influence of these factors on the machine learned offset model used to correct user input and we make design recommendations. The results show that all performance parameters degraded when the subject started to move, for all input techniques. Index finger pointing techniques demonstrated overall better performance compared to thumb-pointing techniques. The influence of gait phase on tap event likelihood and accuracy was demonstrated for all input techniques and all walking speeds. Finally, it was shown that the offset model built on static data did not perform as well as models inferred from dynamic data, which indicates the speed-specific nature of the models. Also, models identified using specific input techniques did not perform well when tested in other conditions, demonstrating the limited validity of offset models to a particular input technique. The model was therefore calibrated using data recorded with the appropriate input technique, at 75% of preferred walking speed, which is the speed to which users spontaneously slow down when they use a mobile device and which presents a tradeoff between accuracy and usability. This led to an increase in accuracy compared to models built on static data. The error rate was reduced between 0.05% and 5.3% for landscape-based methods and between 5.3% and 11.9% for portrait-based methods.

CONTENTS

1. INTRODUCTION
2. RELATED WORK
 - 2.1. Understanding touch
 - 2.2 Offset models
 - 2.3. User input on-the-go (nomadic input)
 - 2.4. Alternative approaches to improving user input
 - 2.5. Relation of this paper to previous work
3. MATERIALS AND METHODS
 - 3.1. Participants
 - 3.2. Mobile phone input techniques
 - 3.3. Experimental setup and procedure
 - Walking circuit and selection task
 - Experimental design
 - Mobile phone
 - Pace setting algorithm based on the Hilbert transform
 - 3.4. Definition of error rate, accuracy and tapping time
 - 3.5. Throughput as a performance parameter
 - 3.6. Offset models
4. RESULTS AND DISCUSSION
 - 4.1. Error rate analysis
 - 4.2. Accuracy analysis
 - 4.3. Tapping speed analysis
 - 4.4. Throughput analysis
 - 4.5. Gait phase analysis
 - 4.6. Offset model analysis
5. CONCLUSIONS
- REFERENCES

1. INTRODUCTION

Given the increasing use of flexible, multi-purpose hand-held devices, there is an increasing demand for robust, accurate human-machine interaction through touch interfaces on-the-go, subject to movement of the human body, and being used in a range of poses and tapping techniques. However, if we wish to develop improved inference of human intent in such complex environments we need to better understand the impact of mobile use on touch behavior. While future sensing systems might provide additional information to the device, to improve accuracy with the currently available hardware we have to find innovative ways to interpret current touch actions, as well as designing more sophisticated and robust inference algorithms.

Due to the size and softness of the finger the finger is perceived as a much larger contact region than intended, during the touch event itself, so the user might visually target with a different part of their finger from what actually touches the screen (Holz & Baudisch, 2011). It was discovered that there is a constant trend in the error between the actual target position and the touch location, which can be partially compensated for by the development of user-specific offset models. The offset model approach tries to reduce variation in user input by learning (modeling) the differences between the intended input and the actual touch location. Lately, considerable effort has been put into finding the best possible models that account for most of the variation. Unfortunately, many of the approaches used generic models which did not take account of either the needs of specific users or the particular mobile phone being used. Recent effort has been directed toward this user-specific and device-specific context with the incorporation of information from additional sensors such as accelerometers. However, the literature has few examples of systematic analysis of the influence of input technique, gait speed and gait dynamics on offset models.

By taking device motion into account we are able to identify changes that occur in accuracy, tapping speed and throughput, as well as the underlying offset model once the subject starts to move. The goal of the experimental study in this paper was to gather data from a number of on-board sensors in different input and walking speed scenarios and analyze them to find possible relationships between the variables, as well as their influence on offset models. As we can infer gait phase from accelerometer recordings, we explore the relationship between gait phase, input technique and accuracy. This supports detailed quantitative assessment and provides insight into important questions e.g. which phase of the gait cycle (if any) users prefer for tapping, and how that relates to tapping accuracy and the input technique being used? These questions have so far not been answered for the full range of measurement conditions.

In this paper we derive offset models for every input technique and explore and analyze the differences. These observations can be used in the design of touch screen virtual keys in mobile scenarios as well as aiding the calibration of algorithms to infer which input technique is currently being used, increasing the accuracy and efficiency and improving the user experience.

2. RELATED WORK

2.1. Understanding touch

To understand touch interaction and improve the accuracy requires analysis of the underlying “touch mechanism”. Holz & Baudisch, (2010) challenged the commonly accepted fat finger problem and introduced a perceived input model. In (Holz & Baudisch, 2011) they thoroughly investigated the “touch mechanism” (or users’ mental model of touch) when a user acquires a target for varying target sizes and varying angles of approach. Previously there had been an implicit assumption that users acquire targets with the center of the contact area between finger and device, but this failed to explain systematic error offsets recorded in user input. The research found that users generally look for visual cues to align the top part of the finger that is not intended for touch (e.g. the top of the user’s fingernail) with target vertical/horizontal lines. Based on that finding, a projected center model could be constructed to increase the accuracy of input, but this is technically challenging with currently available mass-production mobile hardware. The ‘Shift’ method proposed in (Vogel & Baudisch, 2007) tries to avoid finger occlusion and ambiguity about which part of the finger defines selection, by creating a callout showing a copy of the occluded screen area and placing it in a non-occluded location. However, a higher level of effort per touch is needed, since the whole touching procedure is different from standard interaction strategies.

2.2. Offset models

Offset models are the class of models that calculate the deviation between the intended and actual touch location and correct for it regardless of the input scenario (e.g. keyboard layout or language models are not necessary). This effect and its potential for improvement in user interaction were recognized early in the research. Sears (1991) experimented with large touch screens fixed to a table with a fixed tilt angle. The results obtained showed that when the device was tilted away from the user the actual touch location was below the target while if it was tilted toward the user the touch location was below and to the left of the target. Implementing the offset model corrections enabled a reduction in the size of the virtual keys while maintaining the same error rate. This phenomenon was also recorded in (Himberg, Häkkinen, Kangas & Mäntyjärvi, 2003), where an on-line learning algorithm was used to move soft keys according to the spatial distribution of touch events made on a customized screen laptop PC. The result obtained in the experiments demonstrated the tendency of right-handed subjects to hit to the left of the intended target, while left-handed subjects tended to hit to the right of the actual target. Offsets were noticed in (Parhi, Karlson & Bederson, 2006) where the authors set to investigate the optimal target size in discrete (single) and serial (successive) target selections. Users were static (standing) and used their thumbs for interaction, holding the device in a single hand. In particular, they noticed a right-leaning trend for targets in the right part of the screen. It was additionally noticed that after certain target dimensions a further increase had no statistically significant effect on error rates, but that the target size did have a significant impact on tap time. This was attributed to the fact that users took extra care with smaller targets. Park, Han, Park and Cho (2008) explored the practical

1
2
3 impact of offsets and measured error rate, success rate and pressing convenience of 25
4 predefined regions on 240 x 320 pixel PDA screen when using one-handed thumb input
5 as the only input technique. The results indicated two regions (an appropriate and an
6 inappropriate) for virtual key placement, and the benefit of the correction algorithm based
7 on these offsets was demonstrated to be statistically significant.
8
9

10 The largest study to date on offset models, in terms of collected touch events, was
11 carried out in (Henze, Rukzio & Boll, 2011), where crowd-sourcing via Android Market
12 was used. A custom-designed game collected data on user touch events and intended
13 targets, which enabled analysis of offsets. The paper demonstrated that on such a large
14 sample set there was a systematic offset in touch positions and that it depended on the
15 target size and on target screen location. It should, however, be kept in mind that
16 although large amounts of data were collected, there was little control over the
17 participants and how they held their phone and whether or not they were stationary or on-
18 the-go while playing. The reported results showed several interesting findings. The error
19 value increases significantly when targets smaller than 15 mm were used, reaching 40%
20 for targets smaller than 8 mm. This high error rate might be attributed to instructions
21 given to players to “touch and move as fast as you can”, since players might put greater
22 emphasis on speed and not accuracy. Contrary to some earlier research, it was noted that
23 error rates were lower for targets in the center of the screen compared to targets on the
24 border. When the offset model was derived and applied to the data, the error rates
25 dropped by about 8% and improved player precision. There was no attempt made to
26 customize the offset models to be user-specific, device-specific or input technique-
27 specific.
28
29
30
31

32 *User-specific offset models:* These were analyzed in (Weir, Rogers, Murray-Smith &
33 Löchtefeld, 2012), where a machine learning approach based on Gaussian process
34 regression was used to capture the systematic error between the intended and actual touch
35 positions. The inferred functions were highly nonlinear and varied over screen location
36 and users. One novel aspect of this research was that it was based on data from the raw
37 capacitive sensor array as well as the typical data provided by the mobile phone’s touch
38 API. The results demonstrated a similar performance in terms of accuracy improvement
39 ranging from about 5%-24% depending on target size. To further drive home the point
40 about user-specific nature of offset models, the accuracy of each individual model (8
41 participants) was compared to the accuracy of model generated by data collected from all
42 subjects. Statistically significant improvements in individual models were observed. The
43 cross-device context of offset models was discussed in (Buschek, Rogers & Murray-
44 Smith, 2013) where 13 different mobile phones were used in a game-like setup. In order
45 to improve the accuracy (5-6%) of the interaction, cross-device and cross-user transfer
46 functions were proposed. These transfer functions adapted the user offset model recorded
47 on different mobile phones (or different user models on same phone) to obtain
48 improvement. Although the authors note some limitations of their study (e.g. the lack of
49 feedback and use of linear regression with quadratic terms for modeling) several
50 additional interesting observations were made. First of all, about 60 randomly chosen
51 training examples (targets) in portrait one-thumb mode were sufficient for obtaining
52 meaningful offset models which is a significant reduction compared to similar research
53
54
55
56
57
58
59
60

(200 in (Weir, Rogers, Murray-Smith & Löchtefeld, 2012)). Secondly, a negative effect of screen size on effectiveness of offset models was observed.

Role of finger pose: In (Holz & Baudisch, 2010) a slightly different approach to offset model analysis was taken, employing specialized hardware in the form of a fingerprint reader. The authors demonstrated that the offset model is user-specific and depends on the posture of the finger used in the interaction. The results demonstrated that roll, tilt and yaw all have a statistically significant impact on touch location and thus the offset model. Based on these results, one could argue that a particular subject, using a specific input technique (e.g. landscape orientation with one thumb) could be uniquely identified, since the angle of a finger approaching the target (taking into account the screen position of the target) should have distinct values. We explore this further in our analysis of offset models. (Rogers, Williamson, Stewart & Murray-Smith, 2011) demonstrated that such use of finger pose could have a practical application. They used low-resolution, longer-range capacitive sensors in a custom-built case intended to mimic a mobile device. They developed an explicit model of a finger in contact with the sensor array and a theoretical sensor model in order to infer the finger pose from raw measurements. The accuracy was significantly better than the case of a simple interpolation algorithm, converging to a stable error rate for button sizes of about 3 mm (in contrast, the simple interpolation algorithm converged at about 5 mm).

Posture: The influence of posture was recognized as an important factor in designing more accurate text entry interfaces through posture adaptation, as was the case in (Azenkot & Zhai, 2012). Here, the authors demonstrated that different postures result in different typing patterns (e.g. different offsets) on a soft keyboard during text entry for a particular key. Three postures were analyzed: two-thumbs, one-thumb and one-finger and corresponding error rates and typing speeds calculated. Goel, Wobbrock and Patel (2012) expanded on these findings. In their work they could differentiate between the use of right or left thumb and could infer which hand is holding the device (or if the device is lying on a flat surface). This was achieved by the inclusion of additional interaction elements such as swiping motions and sensors (gyroscope) which enabled reasonable estimation after 5 interaction steps. Authors report 84.3% accuracy for hand posture detection and 99.7% accuracy for table vs. hand detection. The same research group designed a text entry system named ContextType which improved text entry accuracy using posture information in (Goel, Jansen, Mandel, Patel, Wobbrock, 2013). Based on inferred posture, ContextType switched between different keyboard models, which reduced total text entry error rate by 20.6% compared to the control condition. With the inclusion of several additional features (like time elapsed between taps) posture detection accuracy was increased to 89.7% and the decision was made, on average, within 4.9 taps. The authors also noted that the typing speed (expressed in words per minute) was not significantly affected with inclusion of a posture-adaptive keyboard. This approach was further refined in (Yin, Ouyang, Partridge & Zhai, 2013) to include a user-specific posture adaptive keyboard. In the paper, an elaborate hierarchical model was used which reduced the language-model independent error rate by 13.2% over the baseline model. This approach had an on-line posture classification accuracy of 86.4%. However, in this and other works presented in the paragraph, no effort has been made to study how the typing pattern behavior changes while users are on-the-go, or how the proposed on-line

adaptation performed under nomadic conditions (i.e. all data collection was made while subjects were seating/stationary). Further, all studies used soft keyboards using only the lower part of the screen, which might affect the generalization of the results.

2.3. User input on-the-go (nomadic input)

In most of the research presented, the experimental setup was limited to a static setting (e.g. sitting or standing still). This is not sufficient for an average user of a mobile device, since motion (e.g. walking) can potentially have a significant influence on conclusions. This was recognized quite early in (Johnson, 1998), where a number of scenarios are explored and the associated problems discussed. In a mobile setting, new variables emerge and the experimental setup needs to be adjusted accordingly, making controlled studies difficult in real-world scenarios. Thus in (Kjeldsov & Stage, 2004) an experiment was carried out to compare evaluation techniques for mobile devices in a laboratory setting (but not a static scenario) and in a real world situation. It was observed that the laboratory setting provides a good approximation in terms of user performance (e.g. accuracy), but a more realistic setting is needed for an evaluation of required mental workload. Similar conclusions were made in (Barnard, Yi, Jacko & Sears, 2005), with the additional inclusion of a treadmill as possible experimental tool. In (Kane, Wobbrock & Smith, 2008) five additional environmental effects were identified: weather, pedestrians, public events, safety and clothing. For example, cloudy weather reduced the task time by 18.5% compared to partly cloudy and sunny conditions, due to the varying visibility of the device’s screen. This confirmed the influence of lighting conditions on a user’s behavior (Barnard, Yi, Jacko & Sears, 2005). The main analysis of the study included the performance of a walking user interface. This interface adapted the size of the GUI control elements when user was on-the-go. To control the walking pace in public, a human pacesetter was used. Despite this, the actual button size change was not automatic but rather controlled by the experimenter. A number of observations were made: the number of task errors varied by target size and that motion unevenly affects how size affects errors. Handedness and gender had no effect on task time. Most importantly, significant interaction between size and movement (in instances without adaptation) was detected, which is in contrast to (Mizobuchi, Chignelli & Newton, 2005) where no such effect was observed. This might be due to the different input techniques used. No effect of walking speed or input speed on accuracy was observed, while the decrease in walking speed while using a mobile phone compared to the usual walking speed was noted. However, the authors do caution that the experiment was carried out in a long hallway without obstacles (either static or dynamic).

An interesting study of the influence of walking speed on touch accuracy was made in (Bergström-Lehtovirta, Oulasvirta & Brewster, 2011). Here, a number of test subjects were placed on a treadmill, with the speed varied from 0-160% of preferred walking speed (PWS). PWS was suggested as a means to assess usability in (Pirhonen, Brewster & Houlguin, 2002). It was noted that no matter how slowly the user walks, it comes at a cost (in terms of accuracy) compared to the static case. The relationship between speed and accuracy was found to be highly non-linear, with a local optimum between 40-80% of PWS. Two additional observations were made: firstly, the mean decrease in speed when using a mobile phone was about 24% from PWS compared when walking without

the device. Secondly, it was noted that both hand and body oscillations had a significant negative effect on accuracy, with hand oscillations being the stronger predictor. A similar decrease in walking speed (25%) was observed in (Schildbach & Rukzio, 2010). Here, a resistive touch-screen mobile device was used to examine the negative effects of walking on target selection and text reading, and their relation to size of targets and text. A pedestrian-free obstacle course was used for testing since mental workload is higher than on a treadmill (Lin, Goldman, Price, Sears & Jacko, 2007). In the experiment there was a 31% increase in task time, 23% decrease in accuracy and 24% increase in workload compared to the static condition. In (Yatani & Truong, 2007), similar effects of walking on accuracy, typing speed and workload were observed when interacting with a PDA using a stylus-based input technique, showing progressive deterioration in accuracy as the mobile task got more demanding (from static to walking, to climbing stairs).

One factor in the design of user interfaces that was not considered in previously presented research, but was studied in (Crossan, Murray-Smith, Brewster, Kelly & Musizza, 2005) is the relationship between gait phase and interaction accuracy. This was studied using a PDA with a stylus input. When walking, the arm holding the device oscillates in relation to the gait cycle. This was also noted in later work (Goel, Findlater & Wobbrock, 2012). Analysis of results collected from over 20 subjects showed that there was part of the gait cycle which was about 3 times more likely to be used for tapping, and for which tapping accuracy was higher, with a lower error mean and lower variability. The preferred phase corresponded to moments when the device was moving downwards along with the supporting arm. This could be used in design of user interfaces presenting feedback to encourage interaction at certain gait phases.

Yamabe and Takahashi (2007) identify three significant factors that need to be taken into account when designing a user interface for a mobile context: physical condition, degree of attention and social ethics. This is in contrast to (Kane, Wobbrock & Smith, 2008). Yamabe and Takahashi (2007) nicely summarize the problem: *'The main cause of the usability deterioration is a gap between a pre-assumed computer-human-interaction model and a model in the actual working environment. Implicitly or explicitly, most of services are designed to be operated in users' stationary situations, such as "sitting on a chair" and "standing on the street".'* To account for the user movement, they propose automatic adaptation of the user interface, based on the 3D accelerometer reading. Their system performed comparably to the static case in most of the test scenarios, but again here a simple straight line path was taken by test subjects in a controlled environment. In (Goel, Findlater & Wobbrock, 2012) an accelerometer-based adaptive virtual keyboard named "WalkType" was also used but a more sophisticated algorithm was employed. It was inspired by image stabilization algorithms for cameras. Several interesting observations in line with (Crossan, Murray-Smith, Brewster, Kelly & Musizza, 2005) were made, although here a portrait, two-thumb input technique was used. Target selection tended to shift towards the center of the keyboard if a tap occurred when users' feet hit the ground, while a shift to the left was dominant when the left foot hit the ground. A right shift was dominant when the right foot hit the ground. The final model reduced uncorrected errors in the walking condition by 45.2% and increased the typing speed by 12.9%.

2.4. Alternative approaches to improving user input

Beyond learning offset models, there are other examples of adaptive methods to improve touch accuracy. Examples of multimodal feedback (beyond visual) include (Hoggan, Brewster & Johnston, 2008) for tactile and (Brewster, 2002) for audio feedback.

Existing solutions for motor-impaired users are being leveraged into the small-device domain like mobile phones because able-bodied subjects using their mobile phones while on-the-go are situationally impaired by both the device and the context of use (Yesilada, Harper, Chen & Trewin, 2010). This hypothesis was proved in static experiments where most of the typing and pointing errors recorded were common in both people with and without a motor impairment. Testing was only carried out in static conditions. The “TrueKeys” system for motor-impaired users in (Kane, Wobbrock, Harniss & Johnson, 2008) combines a word frequency model, keyboard layout and error patterns to automatically identify typing mistakes and offer potential corrections via visual feedback. Although the algorithm performed well, correcting more mistakes than other popular spell checkers, the typing speed was lower. This was attributed to the challenge of forcing users to learn new typing habits and unlearn already established ones.

Baldwin and Chai (2012) implemented a dynamic, user-specific adaptation of virtual keyboard key sizes, based on a language model and observed typing behavior. Special care was given to online data collection methods which dynamically updated the resizing model. Three approaches were used, with each one having a certain level of tradeoff between accuracy and speed. The authors reported a reduction in relative error up to 10.4% over non-user specific models. A related topic is ‘key anchoring’, presented in (Gunawardana, Peak & Meek, 2010). This approach is similar to offset model building (e.g. the same input data is used) but is only applicable for keyboard input, since they are based on natural language technologies. A similar approach to model building was taken in (Findlater & Wobbrock, 2012) where a machine learning approach inferred which key was targeted. This approach, however, required keyboard layout and key-press classification models which could dynamically adapt to individual users.

2.5. Relation of this paper to previous work

A significant amount of research effort has gone into developing innovative ways for improving user interaction, especially in terms of accuracy and typing speed. This is typically achieved by adapting keyboards or offset models. It has been recognized that static and dynamic use conditions have a significant effect on the outcome (Johnson, 1998). More recently, it was shown that the inclusion of movement data (usually via accelerometer) has the potential to significantly improve the interaction experience (Goel. Findlater & Wobbrock, 2012). However, usually this research was limited to one input technique and/or one walking speed. Our study is the first attempt (to our knowledge) to systematically analyze the interactions among a wide range of input techniques for different walking speeds and to investigate how that interaction translates to tapping time, accuracy, offset model construction and gait phase, with associated design implications.

3. MATERIALS AND METHODS

3.1. Participants

In total, 20 subjects (17 males, 3 females) participated in the study (all were volunteers from academic staff and students). The participants' mean age was 28.25 years with a standard deviation of 2.79 years. All participants were self-rated as experienced mobile phone users, owning a smart phone for an average of 3.07 years and with an average mobile phone interaction time (excluding phone calls) of 60 minutes per day. Three of the participants were left-handed, although two of them often used their right hand while using a mobile phone with particular input techniques. It is also interesting to note (for possible cross-device considerations and implications) that the most widely owned phones among users at the time of testing were: Sony Xperia series (9), HTC series (3), Samsung Galaxy series (3) and iPhone series (2). A within-subject 5x5 factorial experimental design was used, requiring all participants to use all input techniques for all walking speeds.

3.2. Mobile phone input techniques

In the study, five input techniques were used, as depicted in Figure 1:

- **portrait 1 thumb pointing (P1TP)** – the mobile phone is in portrait orientation, held by one hand and the user interacts with the phone with the thumb of the hand holding the phone. The user chose which hand was most comfortable to use for interaction.
- **portrait index finger pointing (PIFP)** – the mobile phone is in portrait orientation, held by one hand and the participant interacts with the phone with the index finger of the opposite hand. Which hand to use for interaction and which for holding the device was left to the user. Also the manner in which the mobile phone is held (i.e. how it is held in place by the holding hand as depicted in Figure 1) was left to the user.
- **portrait 2 thumbs pointing (P2TP)** – the phone is in portrait orientation, held by both hands and the participant interacts using both thumbs. The user chose which thumb to use to interact with particular target (with regard to its on-screen position).
- **landscape index finger pointing (LIFP)** – the mobile phone is in landscape orientation, held by one hand and the participant interacts with the phone with the index finger of the opposite hand. The user chose which hand to use for interaction, which for holding the device and the manner in which the mobile phone is held
- **landscape 2 thumbs pointing (L2TP)** – the phone is in landscape orientation, held by both hands and the participant interacts using both thumbs. Which thumb to use to interact with particular target (with respect to its on-screen position) is left to user.

These input techniques were chosen in order to take into account all possible input scenarios and were explained to test subjects. Only landscape with one thumb was not chosen, since in pre-testing interviews none of the subjects seemed to use it. It should be

noted that during the measurements all test subjects chose to use PIFP and LIFP variants presented in the lower row of Figure 1. Also, in pre-testing interviews, participants were asked to state their two most used input techniques in order of frequency of use. 60% of all participants stated that P1TP is their most used input technique, followed by 15% for PIFP and L2TP and 10% for P2TP, while none of the subjects chose LIFP. For the occasionally-used input techniques, 30% of all participants chose P1TP and L2TP, with PIFP in second place with 20% followed by P2TP with 15% and LIFP with 5%. These percentages might be useful in the design process.

(Figure 1 about here)

3.3. Experimental setup and procedure

The goal of the study is to systematically analyze the interactions among a wide range of input techniques (with their particularities, including target size) for different walking speeds and to investigate how that interaction translates to performance measures, along with associated design implications.

Walking circuit and selection task

In the experiment, participants were asked to walk on an indoor circuit (as depicted in Figure 2) which included three right- or left-hand turns (depending on walking direction), as well as going in clockwise and counterclockwise directions. A full lap was ca. 39 m long. This design of walking path was used because it placed realistic mobility demands on the subject, simulating a real world environment where users have to keep track of where they are and interact with the mobile phone at the same time.

During that time, users had to maintain the desired walking pace and tap 150 targets for each measurement condition as quickly and as accurately as possible while trying not to overemphasize either condition. This number was chosen as a compromise between the time it takes to finish a trial (60 minutes of walking + time for static measurement and pre- and post-measurement interview = 75 minutes per subject) and the recommended training set size which in (Buschek, Rogers & Murray-Smith, 2013) was shown to be about 60 targets. These results were also confirmed in (Weir, Buschek, & Rogers, 2013).

Target positions presented to users were random and could appear anywhere in the whole screen area, except for a 20 pixel wide column on the right of the screen used for providing feedback to users about current step frequency. This scenario is akin to the range of someone using a web browser or text editor on a mobile phone, and not only the virtual keyboard, which usually occupies only the lower half of the screen. The user was not provided with feedback that a target selection was correct, but was notified implicitly that a selection was made by drawing a new target on screen. This is in line with related studies like (Gunawardana, Peak & Meek, 2010).

The target size varied depending on whether the screen was in portrait or in landscape orientation and was constant throughout the particular condition. This was motivated by the fact that virtual keys (including a virtual keyboard) change their appearance and size

to accommodate for screen size (height versus width) of a particular phone orientation. When choosing target sizes we were guided by the Apple (Apple, 2010; Clark, 2010) and Nokia (Ribeiro & Carvalhais, 2012; Nokia, 2011) recommendations. For example, iPhone guidelines recommend that “*..in the keyboard, for example, keys are 44 pixels tall but only 30 pixels wide – similarly, in landscape mode view, the buttons are 44 pixels wide but 38 pixels high... That means that practical minimum size of any tap target is 44 x 30.*” Taking into account the physical screen size and screen resolution of 251 ppi that means that 44 pixels roughly corresponds to 7 mm, and 30 pixels to 4.5 mm. Nokia resources, on the other hand, state that target size is determined based on which finger is used: for index finger it should be 7 x 7 mm and for thumb it should be 8 x 8 mm. Taking into account 163 ppi resolution from the datasheet and screen resolution of 854x480 pixels along with physical dimensions of the screen, one can calculate recommended target size in pixels. For our particular experiment we opted for the Apple recommendation and calculated target dimensions in pixels as depicted in Figure 3a.

It should be noted that use of different target sizes for different mobile phone orientations might influence direct comparison of the results obtained. However, the intention was to be as realistic as possible to provide practical recommendations to designers. Thus we are particularly interested in pose performance (with all of its particularities, including virtual key size) compared to other ones as they are available in real world (Nicolau & Jorge, 2012). More direct comparison of performance of target selection is also provided through the use of the throughput parameter since it takes into account (effective) target size.

(Figure 2 about here)

Experimental design

There were two factors in the design, all manipulated within subjects: input technique (5 levels) and walking speed (5 levels – including static condition). Walking speed levels were defined relative to individual subjects' preferred walking speed (PWS), including the static condition. These levels were: 0PWS (static), 0.5PWS, 0.75PWS, PWS and 1.25PWS. These levels were chosen because they pertain to a subject's normal walking speed (PWS), the speed he spontaneously slows down to when using a mobile phone (0.75PWS) and two extremes (0.5PWS and 1.25PWS) which the subject could comfortably perform and which form a uniformly spaced grid. Thus 20 test subjects performed 25 trials in total, resulting in total of $25 \times 150 \times 20 = 750000$ tap events. Static and dynamic measurement conditions were counterbalanced separately (since they were recorded separately at two different time instances) using a balanced Latin square based technique, in order to minimize learning effects. Gait speed was not directly controlled for, due to limitations of the instrumentation, as discussed later in the section. However, since for a particular subject his/her stride length is constant during normal, undisturbed gait, the gait speed and gait frequency ratio are also constant (Bertram & Ruina, 2001). Thus, by controlling gait frequency we controlled gait speed.

Mobile phone

The experiment was carried out on the Nokia N9 mobile phone, using Python and PyGame running natively. This phone was chosen both because it met the technical requirements (the ability to access raw accelerometer signals at a high sampling frequency and enough computational power to run the filter in real time) and because none of participants had it as a personal phone. Thus, no participant would have extra familiarity with the specific device.

(Figure 3 about here)

The mobile phone's role was threefold. First, it served as a sensor, with an accelerometer and a touch screen capacitive array. Secondly it was used as preprocessing unit for the real-time filtering and data processing needed for the calculation of the current user pace. Thirdly, it was used as data logger to save raw data (accelerometer signal along all three axes, target location and actual touch point as interpolated by the mobile phone, and the time stamp of each touch event) and some of the calculated values (gait phase inferred from accelerometer data, current user walking pace etc.). Subsequently, all data analysis was done in two software environments: MATLAB and R. An example of recorded hit distributions for different input techniques and different subject handedness is presented in Figure 4, in accordance with the prior literature (Himberg Häkkinen, Kangas & Mäntylä, 2003).

In the experiment, the mobile phone was used as a pacesetter. Although there are several other approaches commonly used in the literature we dismissed them either as impractical (e.g. a human pacesetter, due to the number of experiments and the fact that actual pace variations between pacesetter and subjects were not recorded) or intrusive (e.g. setting the tempo with metronome or some other audio feedback might influence the typing style and speed, as was shown for gait synchronization (Murray-Smith, Ramsay, Garrod, Jackson & Musizza, 2007)). Frequency synchronization between gait and audio feedback was also undesirable, since we wanted to investigate the possible patterns between gait style/speed and touch events.

(Figure 4 about here)

Pace-setting algorithm based on the Hilbert transform

For the subject pacesetting, an algorithm was developed and implemented to run in real time on the mobile phone. This was enabled by our choice of gait frequency as a measure of a person's speed. Different people have different stride lengths (generally dependent on a person's height) and to achieve the same speed a subject with a larger stride would have to have a lower frequency and consequently his gait would be steadier (Renaudin, Susi & Lachapelle, 2012). Additionally, a pacesetter based on accelerometer measurements is more robust and easier to implement if it is based on frequency and not the speed. This is because the speed would require the integration of the accelerometer signal and thus accumulate error over time. Also, taking into account that different participants felt more comfortable walking at different speeds, we chose to define these

frequencies with respect to individuals' PWS. Before each measurement the user was instructed to keep the phone in their hands, look straight ahead and make three laps around the test circuit with the walking speed/frequency which felt most comfortable. This was recorded by our algorithm as the PWS for that subject.

To calculate the frequency, the raw accelerometer signals were firstly low-pass filtered to preserve only the gravity component. From it, and with the assumption that e.g. for portrait, the z -axis was vertical and the y direction was in the direction the user was walking, one could compensate all raw accelerometer signals for small variations in pitch and tilt angles that participants introduce into the measurement, due to their unique style of holding the phone. This is depicted in Figure 5. All subsequent data analysis for the pacesetting algorithm is based on the raw, compensated accelerometer signal in the direction of the z -axis, since it best captures the cyclic nature of human gait.

(Figure 5 about here)

This signal is then band-pass filtered with a band-pass frequency determined by subjects' walking frequency and configured before the measurement. The band-pass gap should be kept as narrow as possible, because of the Hilbert transform's sensitivity to low frequency trends. Filtered signals are then fed to the Hilbert transform filter (a real-time approximation of the analytical Hilbert transform). This filter had a 5-point size which enabled the minimum latency between the actual movement and user feedback. Based on it, the phase angle can be calculated, which can take on values from the $[-\pi, +\pi]$ interval. In our research we were not concerned about finding the correspondence between the phase angle and gait phase but rather about the frequency of particular gait phase angles. For simplicity we chose this angle to be 0 and implemented a zero detection algorithm but only when there is sudden change from $-\pi$ to $+\pi$ range. This is because a zero crossing occurs twice during one gait cycle. In this manner, the frequency of gait is obtained. It should be noted that the gait phase data can be then further used in analysis to detect whether certain gait instances are more likely to contain tap events than others. Once the frequency is known, it is presented to the user, making him aware of his relationship with the desired frequency, while not adding significantly to the workload (according to users' subjective opinion in post measurement interviews). Achieving the exact frequency would be nearly impossible, due to the variability of human gait and the data processing involved. We chose a frequency band that was acceptable ($\pm 10\%$ of desired frequency). This information was presented to the user in the form of a dynamic bar with changing color, as depicted in Figure 3b.

During the measurement trials, users reported that they found this type of pacesetting easy and fun to use and the obtained results were sufficiently close to the desired walking frequency. The only measurement condition that provided significant problems to our algorithm and to users was 0.5PWS. It is believed that the issue with this condition was caused by the magnitude of the signals fed to the Hilbert transform filter. Specifically, lower accelerations associated with slower walking, made it harder for the pitch and tilt compensation algorithm to distinguish acceleration from the effect of the force of gravity, ultimately leading to degraded performance of the Hilbert transform filter due to its sensitivity to low frequency trends. This effect was reduced as the walking speed started

to increase. We determined the extent of the influence of slower walking speeds on the performance of our pacesetting algorithm by analysis of the normalized estimation error. The median recorded gait frequency and required gait frequency for each walking speed were compared (normalized relative to the required gait frequency), so that direct comparison with the allowed frequency band could be made. Taking into account all test subjects, the following mean errors (and their standard deviations) were obtained, as percentage of the target frequency: 67.91%±41.26% for 0.5PWS, 19.46%±26.6% for 0.75PWS, -9.52%±18.73% for PWS and -24.97%±16.13% for 1.25PWS. Lower absolute mean values are associated with more stable frequency tracking (keeping in mind that ±10% deviance from desired frequency was allowed).

3.4. Definition of error rate, accuracy and tapping time

A rectangular-shaped virtual key was used instead of a crosshair. This was motivated by the intention to better simulate a real world situation (e.g. different target sizes in portrait and landscape mode). The *error rate* was defined as the percentage of wrong selections (outside the virtual key) from the total number of selections made in one trial for one measurement condition while *accuracy* was defined as the reciprocal value of the Euclidian distance between the center of the intended and actual target locations.

The center of the virtual key was moved to the screen coordinate frame origin and the distance to the actual touch point (which was also moved for same distance both in value and direction) was calculated. If the distance between the target center and touch location was too large (outside the $[q_3 - 1.5(q_3 - q_1), q_1 + 1.5(q_3 - q_1)]$ interval where q_1 and q_3 are first and third quartile of all measured distances, respectively), the touch event was considered an outlier and was removed from further analysis (22 touch events in total). These situations occurred on average once per measurement condition, due to uncontrolled user finger movement (due to walking) or unintentional selections (double taps). Since those movements were uncontrolled they were not representative of the user's actual intent and were considered noise. Also, error rates indicate the frequency of the selection point being outside the virtual key.

An additional parameter calculated was tapping time. This time was calculated as the difference between two time stamps associated with two consecutive touch events. It was used as indicator of user tapping speed for a particular input technique, since users were instructed to keep selecting targets as they are presented, without any pause.

3.5. Throughput as a performance parameter

Throughput was introduced as a part of the ISO 9241-9 standard ("*Ergonomic design for office work with visual display terminals (VDTs) - Requirements for non-keyboard input devices*") as an objective parameter for evaluation of non-keyboard input devices. It is based on Fitts' law, which tries to quantify the capacity of the human motor system to transmit information content. Several formulations of Fitts' law have been proposed over the years, the main difference being in the definition of the index of difficulty (MacKenzie, 1992). Also, 2D pointing and selection is different from the original 1D Fitts experiment, so certain adjustments need to be considered (MacKenzie & Buxton,

1992; Accot & Zhai, 2003; Wobbrock, Cutrell, Harada & MacKenzie, 2008). This is further complicated by the fact that the targets used in our experiment were rectangular, which complicates the notion of target width.

Several decisions had to be made during the analysis. First, since our error rate was well above the required 4% (from the original Fitts' law formulation) as presented in section 4.1, we needed to take into account the effective target width (MacKenzie, 1992). This was done by defining the target width using the standard deviation (σ) of selection points with respect to the target center and with equation,

$$W_e = 4.133\sigma,$$

(as proposed in MacKenzie & Buxton, 1992), which leads to the equation for calculating the effective index of difficulty (ID)

$$ID_e = \log_2 \left(\frac{D}{W_e} + 1 \right),$$

where D is the distance from the starting point to the target. Calculation of D is somewhat problematic in our experiment, as further discussed in the Results and Discussion section. The +1 term ensures that the ID will always be positive. The throughput is then defined as,

$$T = \frac{ID_e}{MT}, \quad (1)$$

where MT is the measurement time needed to go from the starting point to the target.

3.6. Offset models

An offset model is a function f that maps the API-reported location to the intended touch locations. The function is derived using some form of machine learning or regression from training examples. A training example is the quadruplet (x_a, y_a, x_i, y_i) , where (x_a, y_a) defines the actual touch location and (x_i, y_i) defines the intended (target) touch location. These models have been shown to be both user- and device-specific. In the experiment, dependencies on measurement condition were examined as well as possible changes between models derived from static data and dynamic data. Additionally, we examined whether inclusion of a more sophisticated model resulted in better estimates, capturing more variance. Thus, two well known model groups were used: a quadratic model and cubic model linear regression (as suggested in (Buschek, Rogers & Murray-Smith, 2013)). Models were built independently for the x and y directions. The purpose of this paper is not to find the best model but rather to analyze trends which can then be applied to a particular problem. Offset models were thus constructed in two scenarios: for a particular measurement condition and for a particular input technique (regardless of walking speed). Examples of offset models are depicted in Figure 6.

(Figure 6 about here)

As with most machine learning based problems, training and prediction phases are required. For the training phase, 80% of the touch events for particular measurement conditions were randomly selected, while the remaining 20% were used for prediction and evaluation. During the training phase N tuples containing touch and target locations (2D coordinates) are available. Depending on the model used, M basis functions $\Phi_1, \Phi_2, \dots, \Phi_M$ are used to model non-linear surfaces. For a quadratic model, $M=5$, with $\Phi_1(x, y) = c_1$, $\Phi_2(x, y) = x$, $\Phi_3(x, y) = y$, $\Phi_4(x, y) = x^2$ and $\Phi_5(x, y) = y^2$. For a cubic model, $M=7$, with $\Phi_1(x, y) = c_1$, $\Phi_2(x, y) = x$, $\Phi_3(x, y) = y$, $\Phi_4(x, y) = x^2$, $\Phi_5(x, y) = y^2$, $\Phi_6(x, y) = x^3$ and $\Phi_7(x, y) = y^3$. When using quadratic and cubic models an $N \times M$ matrix X is built with, the m th basis function evaluated for n th training example. Each row of matrix X contains one training example and each column contains one input dimension. If all x coordinates of all N training examples are stacked in vector t_x and all y coordinates in vector t_y , the maximum likelihood solution of equations $w_x = (X^T X)^{-1} X^T t_x$ and $w_y = (X^T X)^{-1} X^T t_y$ can be derived. The resulting weight vectors w define the offset model using $f_x(a) = w_x^T$ and $f_y(a) = w_y^T$. We define two independent models: one for the x -axis and one for the y -axis.

The aim of the offset model is to predict (x_i, y_i) from the recorded input (x_a, y_a) . In general, during real world application, exact knowledge of the intended touch location is unknown, so the offset model should provide better accuracy than the mobile phone interpolation alone. If a quadratic or cubic model is used for a new touch event t vector $t' = \Phi(t)$ is constructed, i.e. $t' = (c_1, x, y, x^2, y^2)^T$ or $t' = (c_2, x, y, x^2, y^2, x^3, y^3)^T$ depending on the model used. Now, predictions on $t'(x', y')^T$ can be made using $x' = f_x(t)$ and $y' = f_y(t)$. Also, the variance of the predictions can be calculated using $\sigma^2 = \hat{\sigma}^2 a'^T (X^T X)^{-1} a'$, where $\hat{\sigma}^2$ is the maximum likelihood estimate for the dataset noise variance, defined as $\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N (t_i - w^T t_n)$. The prediction variance can then be used as an indicator for uncertain user intention for autonomy handover (H-Metaphor in (Flemisch et. al. 2003)), as demonstrated in (Rogers, Williamson, Stewart & Murray-Smith, 2010; Weir, Pohl, Rogers, Vertanen, Kristensson, 2014).

4. RESULTS AND DISCUSSION

4.1. Error rate analysis

We analyze the error rate for the static case for all input techniques. The error rate is the percentage of selections made outside the virtual key from the total number of selections made in one trial for one measurement condition. If the means and

corresponding standard deviations are calculated, Figure 7 is obtained while Figure 8a presents a box-and-whiskers plot for the error rate in the static condition for each input technique.

The results for error rate are in line with similar research in the literature (e.g. Henze, Rukzio, & Boll, 2011). From the mean data alone, one could conclude that the landscape orientation provides better accuracy, possibly due to a slightly larger virtual key, while for a given orientation, the index finger pointing input technique gives better accuracy than single-handed techniques. This might be due to the fact that in the two-handed approach the user dynamics are more flexible and can better adjust to outside interference. One-way repeated measure ANOVA testing showed that input technique is a statistically important variable ($p < 0.001$, $F(4,76) = 4.18$, $\eta^2 = 0.27$), while a post-hoc Tukey's Honest Significant Difference (THSD) test showed that the statistical difference is between pairs P1TP-L2TP, P2TP-L2TP, P1TP-LIFP, P2TP-LIFP and PIFP-LIFP (with $p < 0.05$). Please note that the THSD test corrects for the family-wise error rate which arises when multiple pair-wise comparisons are performed (this correction is performed in all post hoc analysis presented in the remainder of the manuscript). It is also interesting to note that for portrait-based techniques, the mean error rates in the y direction (i.e. when touch location was outside the virtual key along the y -axis) were twice that of the x direction, possibly due to the smaller target width. This was not present in the landscape-based input techniques, where target width and height are almost identical.

(Figure 7 about here)

If the error rate is examined for the dynamic condition for each individual speed and input technique, the box-and-whiskers plot in Figure 8b is obtained. From the plot it can be concluded that the error rate increases as a person starts to move, regardless of the current speed, along with a slight increase in standard deviation. It is also interesting to note from the figure that, for a given input technique, the error rate median has a minimum at PWS. Here, several observations can be made. First, it seems that a higher penalty is paid (in terms of error rate) when walking faster than when walking slower (although almost every subject noted higher concentration was needed for slower speeds). Secondly, our conclusion is somewhat different from that of (Bergström-Lehtovirta, Oulasvirta & Brewster, 2011). This could be attributed to different experimental setups used (treadmill vs. free walk, type of target and target size used, mental workload needed in obstacle course, finer speed increments) or to inaccuracies in our pacesetting algorithms (using a treadmill, a more accurate speed selection can be made).

(Figure 8 about here)

Nevertheless, a speed with a minimum error rate is observed, as was the case in similar studies. Effects of larger mean errors in the y -direction in portrait modes were also present here, as in the static case, but now the difference was about 1.5 times smaller. When one-way repeated measure ANOVA analysis was performed for a particular input technique, with speed as an independent variable (with inclusion of static condition where speed is 0), statistically significant differences were observed: P1TP ($p < 0.001$, $F(4,76) = 24.38$, $\eta^2 = 0.26$), PIFP ($p < 0.001$, $F(4,76) = 9.18$, $\eta^2 = 0.20$), P2TP

($p<0.001$, $F(4,76)=11.60$, $\eta^2=0.18$), LIFP ($p<0.001$, $F(4,76)=33.44$, $\eta^2=0.32$) and L2TP ($p<0.02$, $F(4,76)=3.07$, $\eta^2=0.10$). A post-hoc THSD test was performed (for each ANOVA) which revealed the following: for all input techniques there was a statistically significant difference between the static condition and the dynamic conditions (with the exception of 0.75PWS in P2TP scenario). For dynamic scenarios, the significant difference was detected for: the P1TP input technique (PWS-1.25PWS), the PIFP input technique (0.75PWS-1.25PWS and PWS-1.25PWS), the P2TP input technique (0.5PWS-0.75PWS, 0.75PWS-PWS and 0.75PWS-1.25PWS), the LIFP input technique (0.75PWS-1.25PWS and PWS-1.25PWS), and for the L2TP input technique (0.5PWS-1.25PWS and PWS-1.25PWS). All statistical differences were detected at $p<0.05$ level. Two-way ANOVA was also performed in order to determine if there was a significant effect between the input technique and walking speed for error rate. A significant interaction effect was observed ($p<0.001$, $df=16$, $F=3.03$).

It is interesting to see how the error rate changes with respect to the target position on the screen, both within one particular condition and among input conditions. Within conditions variability implies that there are parts of the screen where the error will be larger/smaller, as expected and is the motivation for non constant offset models. Between conditions variability implies that the input technique has an influence on the error rate as well as walking speed. These dependencies will be analyzed in the offset models.

In Figure 9 we present average error rates for different parts of the screen in changing measurement conditions for the P1TP input technique over all subjects. For this purpose, the screen was divided into 3 sections along the y-axis and 4 sections along the x-axis. From the figure it is evident that the static input has the lowest error rates over the whole screen area, when compared to other input scenarios. Looking at the general color patterns over the whole screen area, it can be concluded that the heat map for the PWS condition has lower error rates for a particular screen slice (compared to other dynamic conditions and the same slice). It is interesting to note the line-like pattern in all dynamic scenarios (less in 1.25PWS) in the second row from the top, which roughly corresponds to the height at which the thumb of the holding hand is positioned (remember that this is P1TP pose). This is potentially interesting finding for interface designers and we expect this phenomenon to be reflected in corresponding offset model. Similar changes were detected for all other input techniques.

(Figure 9 about here)

No improvement in performance was observed, in terms of error rate between users who do not prefer a particular technique and the ones who do, both in static and dynamic scenarios. Our analysis was limited to P1TP techniques, since for all other input techniques too few test subjects chose them as their preferred one. Also, we analyzed whether target-to-target distance (without consideration for angle of approach) had an influence on the error rate e.g. do some distances have a higher probability of error in selection than others. It was demonstrated that, both for successful and unsuccessful target selection, a similar distribution of target-to-target distances is obtained. This was observed for all input techniques in both static and dynamic settings. However, since the

target-to-target parameter was not controlled for, further research is needed to make conclusive observations.

4.2. Accuracy analysis

As stated earlier, we make a distinction between error rate and accuracy. Here, we define accuracy as a function of distance from actual touch location to target center (lower total error implies higher accuracy and vice versa). Although the target center, in general, might not be something the user is intentionally targeting it is a widely used approach in the literature (Azenkot & Zhai, 2012), especially when considering offset models. Additionally, the target size used in the study was small compared to the finger size (Lee & Zhai, 2009) so users did not have a large area to target (as would be the case with the space bar, for example). Formal analysis was also done to confirm the correlation between accuracy with respect to the virtual key center and virtual key border (which is an alternative choice). It showed moderately strong correlation with a factor of $-0.52 (\pm 0.34)$ for the static case and $-0.22 (\pm 0.35)$ for the dynamic case. This value varied across users and input postures and should be kept in mind during the accuracy analysis presented in the following sections. Figure 10a presents the box-and-whiskers plot for accuracy in the static condition for each of the input techniques, across all test subjects.

From Figure 10a it can be seen that different techniques have different medians, with landscape-based techniques having better accuracy than portrait-based techniques, but with a smaller margin than in the case with error rates. Corresponding mean values are presented in Figure 7. For a particular mobile orientation, the index finger pointing input technique has better accuracy than other input techniques. These findings have statistical significance ($p < 0.02$, $F(4,76) = 3.25$, $\eta^2 = 0.10$), according to one-way repeated measure ANOVA analysis. A post-hoc THSD test revealed that difference between LIFP and L2TP techniques was significant at a $p = 0.05$ significance level. Similar conclusions can be drawn when looking at the accuracy along the x and y axis individually, although there is an increase in the mean in the x direction compared to the y direction.

(Figure 10 about here)

If the error value is examined for the dynamic condition for each individual speed and input technique, the box-and-whiskers plot in Figure 10b is obtained. The figure shows that, for a given input technique, the median accuracy has a maximum at PWS. This observation is in line with results for error rate and data from the literature. When one-way repeated measures ANOVA analysis was performed for a particular input technique, with speed as the independent variable (with the inclusion of the static condition), statistical significance was observed in all cases. A post-hoc THSD test was performed (at $p = 0.05$ level - for each ANOVA), which revealed the following: for the P1TP input technique ($p < 0.001$, $F(4,76) = 12.90$, $\eta^2 = 0.16$), the only statistically significant change in accuracy was between the static condition and all dynamic ones, while for all dynamic cases there was significant difference only between PWS-1.25PWS pair. For the P1FP input technique ($p < 0.001$, $F(4,76) = 19.34$, $\eta^2 = 0.33$) the static condition was statistically different from all dynamic cases, while for dynamic cases a statistically significant changes were detected between 0.5PWS-PWS, 0.75PWS-1.25PWS and PWS-1.25PWS.

For the P2TP input technique ($p<0.001$, $F(4,76)=8.64$, $\eta^2=0.16$), the statistical difference was between the static condition and 0.5PWS as well as 1.25PWS and between pairs 0.5PWS-1.25PWS and 0.75PWS-1.25PWS for dynamic condition. For the LIFP input technique ($p<0.001$, $F(4,64)=9.46$, $\eta^2=0.21$) a statistical difference was observed for the static condition and 1.25PWS as well as between 0.75PWS-1.25PWS and PWS-1.25PWS pairs. Finally, for the L2TP input technique ($p<0.001$, $F(4,76)=17.5$, $\eta^2=0.15$) the only statistically significant difference, except between the static condition and dynamic conditions, was between PWS and 1.25PWS. Additionally two-way ANOVA was performed in order to test for interaction between input technique and speed for accuracy, and no significant effect was detected ($p=0.082$, $df=16$, $F=1.56$). No statistically significant changes were detected in individual directions (e.g. along the x and y axis).

Smoothed scatter plots of normalized (with respect to the coordinate frame origin) touch and target positions are depicted in Figure 11. These plots show the 2D distribution of touches with an emphasis on maximum and deviation. Two examples are presented, one for portrait-based mode and other for landscape-based mode. Both techniques are based on index finger pointing. In both cases, it is evident that for the static condition the 2D distribution is tight and centered, with the larger variation in portrait mode possibly due to the smaller target size. This distribution changes in dynamic conditions in both maximum position and width of distribution, which suggests that offset models developed for the static condition might not work as well in dynamic conditions, and that the offset model should also be walking speed dependent. From both figures it can also be observed that for PWS there seems to be a decrease in variation compared to other walking speeds, which is in accordance with the findings for accuracy and error rate.

(Figure 11 about here)

4.3. Tapping speed analysis

Tapping speed was defined as the time between two consecutive taps, regardless of whether the resulting selection was correct or incorrect. Four observations were removed from this part of the analysis because the test subjects were in some way distracted and the time between two taps was extremely large (>5 s). The distraction caused them to raise their heads and to stop typing (unnecessarily increasing typing time). Afterwards, the typing continued as before. The issue did not affect typing accuracy. Calculated tapping time means (and standard deviations) for all input methods in the static condition are presented in Figure 7. A one-way repeated measure ANOVA was performed and a significant effect ($p<0.001$, $F(4,76)=21.90$, $\eta^2=0.22$) was observed. A post-hoc THSD test (performed for each ANOVA) revealed that the P2TP and L2TP input methods had a statistically significant difference from all other methods at $p=0.05$ level (e.g. P2TP was the slowest while L2TP was the fastest method). Since the differences are so small, we have visualized these differences via a cumulative distribution function (CDF) of tapping-time log, shown in Fig. 12a. From the figure it is clear that the L2TP diagram reaches peak values before other methods, while P1TP does that slightly later. The remaining three methods are very close, such that they can hardly be distinguished in the plot.

(Figure 12 about here)

Analysis of the correlation between tapping speed and total error (accuracy) was performed, with the results presented in Figure 13. From the results, unsurprisingly, an increase in tapping time has a slight but significant negative correlation with the total error, i.e. a decrease in tapping speed is associated with increased accuracy.

A similar analysis for dynamic conditions was performed. Figure 12b compares tapping time means for different walking speeds and input techniques. It can be seen that test subjects generally performed faster typing when compared to the static condition for 0.5PWS and 1.25 PWS, while they were generally slowest for PWS. This might explain why for that walking speed the accuracy was highest, as demonstrated in the previous section. It can also be seen that, in general, L2TP is again the fastest input technique, while P1TP is again the slowest. However, if Figures 11 and 12 are compared, it can be seen that this increase in tapping time is not reflected in increased accuracy (compared to other input techniques for a particular walking speed). Statistical analysis (one way repeated measures ANOVA + post-hoc THSD for each ANOVA) revealed the following: for the P1TP technique there was a significant difference ($p < 0.001$, $F(4,76) = 56.30$, $\eta^2 = 0.52$) between the following cases: static-PWS, static-0.75PWS, 0.5PWS-PWS, 0.5PWS-0.75PWS, 0.75PWS-1.25PWS and PWS-1.25PWS. For PIFP a statistical significance ($p < 0.005$, $F(4,76) = 44.76$, $\eta^2 = 0.45$) was observed between the static condition and all other conditions except 0.5PWS as well as between 0.5PWS and all other dynamic conditions and PWS and all other dynamic conditions. A similar scenario was observed for the P2TP method ($p < 0.005$, $F(4,76) = 40.12$, $\eta^2 = 0.22$) where statistical significance was detected between 0.75PWS and all other conditions (including static) and PWS and all other conditions (including static), while for the LIFP method ($p < 0.001$, $F(4,76) = 50.20$, $\eta^2 = 0.56$) post-hoc analysis revealed a significant difference between all conditions except 0.5PWS-1.25PWS pair. For the L2TP method ($p < 0.01$, $F(4,76) = 44.53$, $\eta^2 = 0.38$) statistical significance was detected between 1.25PWS and all other conditions as well as pairs static-PWS, 0.5PWS-0.75PWS and 0.75PWS-PWS. Two-way ANOVA for an interaction between input technique and walking speed for tapping time provided a statistically significant result ($p < 0.001$, $df = 16$, $F = 26.65$).

(Figure 13 about here)

An analysis of tapping time and total error correlation was performed for the dynamic conditions and similar results as for the static case were obtained (small but statistically significant negative correlation within a range of -0.40 and -0.04 and with $p < 0.001$ in all cases), except for the P1TP-PWS case, where there was no statistical significance for correlation ($CR = -0.004$, $95\%CI = [-0.04, 0.03]$; $p = 0.84$). The correlation coefficients varied depending on speed, but no general pattern was observed.

4.4. Throughput analysis

Throughput was calculated as described in Equation (1) and presented in Figure 14. The means and standard deviations using values marked as outliers in the figure, are presented in Figure 7. One-way repeated measures ANOVA analysis revealed a

significant effect ($p<0.005$, $F(4,76)=7.27$, $\eta^2=0.11$) between input technique and throughput. A post-hoc THSD test showed a significant difference between P1TP and all other techniques except P2TP and between LIFP and L2TP techniques (at $p=0.05$ level). The data suggest that index finger pointing techniques have a higher throughput than thumb-based methods, while landscape-based methods in general have higher values than portrait-based methods. Interestingly, the P1TP technique was subjectively ranked as the preferred input technique by 60% of test subjects in pre-measurement interviews, but actually has the lowest observed throughput value.

As mentioned in the Materials and methods section, the distance-to-target parameter needs to be further discussed. In controlled experiments users are instructed to place a pointer (or finger) into a known starting position after every selection, so the exact value of D is known. In our experiment this was not the case. Users selected targets consecutively without the need to go to a ‘home’ position after each selection. The distance between two consecutive targets was known and that is the distance used in the calculation of the index of difficulty.

Some special scenarios might emerge which have an influence on the final throughput value. In two-handed techniques the next target might be presented on the opposite side of the screen (D close to maximum) but near the other inputting finger, thus reducing the D and thus the effective ID and hence the MT and making throughput higher. We have used the one-handed ID as the baseline task description for comparison among conditions. The reduction in distance to target is inherent to two-handed approaches and the reason why their throughput is higher, but without improved sensing of which hand was used for each trial, it is not trivial to adapt the distance to target for a more realistic ID . Also, in all approaches, the next target might be presented in a location obscured by the user’s finger or hand (which might be close to the previous target, thus having low D) making the user search for it for longer decreasing throughput. Additionally, our targets were random, which might bias the results (depending on user reaction). However, averaged over all test subjects these effects should not present a problem. These issues prevent direct comparison between these results and means from the literature for other devices, such as a standard computer mouse (4-5 bits/s). However, they are indicative of the amount of data which can be accurately input by a particular method on a mobile device touch screen and can, in our view, be mutually compared and conclusions drawn based on that comparison (i.e. their relative values are important). This discussion also pertains to the dynamic scenario. It is also interesting to note from Figure 14a that a number of outliers are detected (both higher and lower value points which could correspond to the cases outlined above).

(Figure 14 about here)

Results are presented in Figure 14b for the dynamic case. Similar trends in throughput can be seen. Although people are more accurate for PWS, it seems that due to the higher selection (tapping) time, throughput is lowest for that speed across all methods. For 0.75PWS throughput is somewhat higher, while it is highest for 0.5PWS and 1.25PWS, possibly due to faster typing. It should, however, be kept in mind that the throughput calculation does not take into account error rates. 0.75PWS appears to be the optimal

speed in the experiment due to a good compromise between throughput and error rate/accuracy, which is in line with subjects' preferred walking speed when interacting with a mobile phone. A number of outliers can be seen in Figure 14b, as in the static case.

The means (and corresponding standard deviations) in all dynamic conditions (taking into account outliers) are presented in Figure 15. When compared to the static case it can be concluded that index finger pointing methods have a higher loss (in terms of magnitude) of throughput when compared to other methods (between 0.476-0.883 bits/s for PIFP and 0.702-1.422 bits/s for LIFP), while the smallest loss is for L2TP (between 0.361-0.557 bits/s). However, there was no significant effect at $p=0.05$ level - a one-way ANOVA detected an almost significant effect ($p=0.059$, $F(4,15)=5.57$, $\eta^2=0.03$) of input technique on reduction in throughput.

(Figure 15 about here)

One-way repeated measures ANOVA analysis of the influence of walking speed on throughput for each individual method revealed significant effects with varying values of p . A post-hoc THSD test was performed (for each ANOVA) and it revealed the following. For the P1TP input method ($p<0.001$, $F(4,76)=35.84$, $\eta^2=0.57$) a significant effect was observed between the static condition and all dynamic ones and between pairs 0.5PWS-PWS and 0.5PWS-1.25PWS (at $p=0.05$ significance level). For PIFP ($p<0.001$, $F(4,76)=29.75$, $\eta^2=0.54$), the static condition also exhibited a statistically significant difference compared to all other dynamic scenarios, while in the case of walking no difference was observed. For P2TP ($p<0.001$, $F(4,76)=23.02$, $\eta^2=0.48$) and LIFP ($p<0.001$, $F(4,76)=35.89$, $\eta^2=0.59$), identical results were obtained in terms of post-hoc analysis with addition that for LIFP technique difference was also observed for pairs 0.5PWS-PWS and 0.5PWS-1.25PWS. For L2TP the significance effect ($p<0.001$, $F(4,76)=6.20$, $\eta^2=0.19$) was observed between static and all dynamic cases except 0.5PWS (at $p=0.05$ significance level). Two-way ANOVA was performed in order to determine if there is interaction between input technique and walking speed for throughput data. A significant effect was recorded with ($p<0.001$, $df=16$, $F=5.91$).

4.5. Gait phase analysis

The acceleration data can be used to explore possible relationships between tapping accuracy and gait phase as well as relationships between input technique and gait phase. We now answer two questions: 1.) "Do subjects have regions of the gait phase cycle where they prefer to tap and is this reflected in performance parameters like accuracy?", and 2.) "Is the distribution of gait phase values at tap instances unique to a particular input technique and does it change with walking speed?". This analysis is limited to the dynamic/walking scenarios. We explore gait phase distributions at time instances when touch events occurred in various measurement scenarios. Figure 16 illustrates the relationship between gait phase zero-crossings and actual movement for one of the test subjects, using the P1TP input technique. The figure also contains histograms of right knee angle values (kinematic parameter recorded using an in-house built optoelectronic motion capture system) sampled at time instances when zero-crossings occurred (over 90 s time period). Note the existence of a single peak in all subplots, demonstrating the

consistency of our gait phase estimation algorithm. Further analysis is needed to precisely define the relationship between the gait phase recorded by the mobile phone and kinematic parameters of the gait.

(Figure 16 about here)

The distributions of gait phase angle along the z -axis (as the dominant axis during gait) are presented in Figure 17. The angle is defined over the $[-\pi, +\pi]$ interval. Distributions were calculated taking into account the periodic nature of the signal under consideration (i.e. phases at $-\pi$ and $+\pi$ corresponds to two consecutive time instances) which is most heavily reflected in the boundary values. Note that for a particular condition, PDF estimates at $-\pi$ and $+\pi$ have similar values. From Figure 17a it can be seen that for a particular speed, all input techniques have similar general trends, although with some minor deviations depending on the input technique. For example, PIFP has two distinct peaks for all speeds, including 1.25PWS while other techniques do not have such pronounced maxima. This two peak trend (connected with left and right leg movement during one full step) is present for all speeds, while this phenomenon is weaker for faster walking (1.25PWS) where one dominant peak is present.

(Figure 17 about here)

This clearly demonstrates that there are certain phases during the gait cycle that are more likely to have more touch events associated with them. If the associated vertical accelerations are examined it can be observed that for both peaks (around -2 and 2 radians) acceleration values are similar: slightly negative and close to zero. Taking into account the following facts: (1) acceleration along the z -axis is equal to zero about the time the foot makes contact with the surface (heel strike), (2) our phase gait cannot discriminate between the right and left foot, and (3) positive orientation of the accelerometer coordinate frame (Figure 5), we can conclude that these instances are related to the time just before the heel strike, when the foot descends. This is in accordance with observations made in (Crossan, Murray-Smith, Brewster, Kelly & Musizza, 2005), but makes a certain level of generalization since here we tested it on all possible input styles. It should be noted that some discrepancies do exist, which could be attributed to measurement conditions i.e. larger targets, different type of device (PDA) and use of a rigid physical stylus in (Crossan, Murray-Smith, Brewster, Kelly & Musizza, 2005). However, because no simultaneous video recording was done it is not trivial to connect gait phase angles to actual gait phases or events for all measurements.

Next, it is interesting to explore whether there is any statistical significance between distributions over the same input methods and different speeds and different speeds over all input techniques. Since the distributions in general are not Gaussian but rather multimodal, we employ the Kolmogorov-Smirnov two sample test (Conover, 1999, Böhm & Hornik, 2012). Looking at Figure 17a, we see that there are 5 signals in each subfigure, so 10 individual tests (per subfigure) are necessary to cover all possibilities (in total, 40 tests were performed). A comparison with adjusted p values (Holm-Bonferroni method) over all subjects and methods for different speeds and over all subjects and speeds for different input techniques was performed (due to the large number of tests and

taking into account space constraints we do not provide more details on the test results). It is interesting to observe that for all measurement conditions the resulting adjusted p -values suggest there is a different underlying process responsible for the generation of the observed densities. These results seem to suggest that for a particular walking speed, the gait phase data could be used as an additional feature (especially in on-the-go scenario) to infer which input technique is being used and possibly improve on detection rates reported in the literature (Goel, Wobbrock & Patel, 2012, Ouyang, Partridge & Zhai, 2013). The same analysis could be achieved for Figure 17b, i.e. analyzing different speed distributions over a particular input method (30 tests in total were performed, 6 per subfigure). From these results it can be concluded that gait phase analysis could be used to detect significant deviations in the gait speed due to the fact that people seem to change or adapt their gait phase/touch event dynamic with changing speed.

Over all methods it seems that 0.75PWS and PWS have the most similar distributions, while this structure disappears at lower and higher speeds. Additionally, it should be kept in mind that the presented results are means across all subjects for each measurement condition. Our analysis of individual subjects suggest that their gait phase distribution in general does follow the trends present in the means in Figure 17, but with two side notes: 1.) the gait phase angle at which the maximum/minimum appears differs slightly from subject to subject and 2.) for certain measurement scenarios some subjects tend to have significantly different distributions from the mean. This implies that an individualized approach might provide better insight into the possible application of gait phase data in identification of current input method. For completeness, we analyzed plots for gait phase distributions along the other two axes and it was found that for the same walking speed there seems to be more variation in distribution trends than compared to the z -axis, with two peak structures in general not being visible. We concluded that z -axis motion is dominant and restricted our analysis to it.

One additional analysis which can be carried out on the available data to provide more insight into the interaction dynamics is a comparison of accuracy/total error with gait phase angle. We divided the gait phase (along the z -axis) into 10 bins of equal size (approx. 0.314 radians) for each input technique-walking speed combination and for each bin we calculated the mean and standard deviation. An example of such a result is presented in Figure 18, where bin1 corresponds to the lowest phase angles. From the examples, it can be seen that indeed there seems to be a local minimum slightly offset from the positions of the two peaks. For P1TP-0.75PWS, the two peaks for tapping distributions are at around -3 rad and +2.51 rad, while accuracy is best for bins 4 (from -1.5679 rad to -0.9410 rad) and 7 (from 0.3128 rad to 0.9397 rad). For L2TP-0.75PWS, the two peaks for tapping distribution are at around -2.51 rad and +2.51 rad, while accuracy is best for bins 1 (from -3.1416 rad to -2.8217 rad) and 7 (from 0.3128 rad to 0.9397 rad), although note that the bin 8 (from 0.9397 rad to 1.5666 rad) median is very close to the bin 7 median. Similar results were observed for all other scenarios. The result suggests that although accuracy is greatest (i.e. total error is lowest) in the general vicinity of the gait phase where most of the taps occur, it leads it slightly in phase.

In order to gain further insight, a one way repeated measures ANOVA was carried out, along with post-hoc THSD testing (for all ANOVAs) on accuracy bin means for

particular pose-speed combination and across all subjects. What this testing showed was highly statistical significance for all pose-speed combinations (for brevity only minimum values are listed here, $p < 0.001$, $F(9, 171) > 194$, $\eta^2 > 0.9$). Post-hoc analysis revealed that there was either a single bin (70% of cases) or two neighboring ones (25% of cases) with significantly lower pixel error (i.e. higher accuracy). Only in one case (1.25PWS-L2TP) were the two bins with lowest pixel error not neighbors (bins 2 and 4). Also, in all but two cases (0.75PWS-L2TP and PWS-PIFP) there were at least two bins that did not have a statistically significant difference (the value ranged between one and five bin pairs). The maximum absolute difference between bin means ranged from 3.8 to 7.6 pixels, depending on the pose-speed combination.

(Figure 18 about here)

It is interesting to examine if there is a particular bin that has in general the lowest pixel error (highest accuracy) across all conditions and subjects. This is depicted in Figure 19, which presents histograms of bins with lowest pixel accuracy obtained from post-hoc analysis: in the case of a tie (i.e. two bins), both were considered. It can be seen that bin 7 is the bin with highest probability of having lowest pixel error. In Figure 18 it can be seen it is just before the gait phase where most of the taps occur, which is in line with our previous conclusions. This is also in line with some previous research (Crossan, Murray-Smith, Brewster, Kelly & Musizza, 2005) that suggested the existence of such gait phase intervals. This leads us to hypothesize that people might be less likely to select targets at a particular time instance during the gait cycle (i.e. where most taps occur) not because their accuracy is best there but rather because it relates in some way to human physiology (e.g. right leg left arm synchronization during the gait cycle).

(Figure 19 about here)

4.6. Offset model analysis

The advantages of touch offset model compensation on target selection accuracy have been demonstrated in the literature, as discussed in section 2.2. However, no compensation analysis was performed with subjects on-the-go. In this section we answer the following questions: “Are offset models walking speed and input technique specific?” and “What is the performance of offset models built on static condition compared to offset models built on data recorded while user was on-the-go if they are both applied to nomadic scenario data?”. Based on the results, we make recommendations on which model to use for on-the-go scenario.

Before model construction, we analyzed various parameters such as screen position, input technique, mobile phone tilt and pitch angles, in order to verify and rank their impact on offset models. These were tested as predictors in a linear regression via random forests variable importance analysis (Breiman, 2001, Genuer, Poggi & Tuleau-Malot, 2010). In general, the on screen position and input method were always ranked among the first three, with varying order. It was interesting to note that if the offset model along the x -axis was built, the x location on screen had more impact than the y position and vice versa. This analysis only confirmed that the input method (and, indirectly, the

approach angles, as outlined in the literature) had a significant effect on the offset model and that further investigation is justified.

In the analysis, only the on-screen location feature (for a particular input method) was used in model building. We used 10-fold verification for all cases, where 80% of randomly selected cases were used in each iteration for training (ideally $150 \times 0.8 = 120$) while the remaining 20% ($150 \times 0.2 = 30$) were used for testing performance parameters like error rate and accuracy. When between-conditions model building and testing was done, the whole dataset from the testing condition scenario was used. First, the error rate was examined for the static case and the dynamic case. In this scenario, data from a particular measurement condition was used for model building for each subject individually, due to the well-known subject dependency of the offset models. Various other cases are examined later.

Figure 20 depicts the error rates for static and dynamic conditions for various measurement conditions, when that condition data is used for building a model. From the figure it can be seen that the application of an offset model significantly improved the error rate (a statistically significant improvement was observed in all cases). The improvement varied depending on condition from 6.4% to 16.7% (mean 12.43%) in the static condition, and from 14.5% to 30.1% (mean 22.9%) for the dynamic condition.

(Figure 20 about here)

It is also interesting to note the relationship between different linear regression models. Based on the obtained results it seems that in the static case, a quadratic model is adequate (an accuracy-computation tradeoff) which was also confirmed with root mean square analysis. Thus, in subsequent analysis only the quadratic model will be used. Improved results might be achieved with some more sophisticated machine learning algorithms like Gaussian process priors (Rasmussen & Williams, 2006), as used in (Weir, Rogers, Murray-Smith & Löchtefeld, 2012). They come at higher computational cost, which might not be justified by the increased accuracy compared to the quadratic linear regression used in our case.

Next, it is interesting to examine how errors behave when an offset model constructed with a different input technique is implemented. This phenomenon is not widely analyzed in the literature. If a significant deviation appears, it suggests that offset models are sensitive to the input technique and approach angle. Figure 21 presents these results. From the figure, it is evident that the error rate increases depending on which input technique is used for model building, although it was still lower than without any compensation, with the exception of the landscape techniques, where accuracy was best to begin with. It seems that, unsurprisingly, the largest increases are present when a different phone orientation is used, although there are some deviations (LIFP in cases of PIFP and P2TP). Error rates increased in all cases, compared to Figure 20, with increases ranging from 7.3% to 13.9% for P1TP, from 6.1% to 16.5% for PIFP, from 3.2% to 11.6% for P2TP, from 8.1% to 10.7% for LIFP and from 6.1% to 13.7% for L2TP. These data emphasize the importance of correctly inferring pose and input technique.

(Figure 21 about here)

Next, we examine how a model built in a static condition behaves when a person starts to walk, and how that compares to a model built with data from the dynamic measurement condition, e.g. speed for a particular input method. This is depicted in Figure 22a. From the figure it can be concluded that performance degrades significantly for portrait-based methods while the degradation is somewhat smaller for landscape-based methods. This implies that when the accuracy of the underlying method is good it matters less that the model was built using static or dynamic data, although differences do exist. It should be noted that use of a model built using static data is still an improvement over cases where no offset compensation is performed.

(Figure 22 about here)

However, a reduction in error difference is observed in Figure 22b, when using an offset model built using 0.75PWS, since 75% of PWS is reported in the literature (Bergström-Lehtovirta, Oulasvirta & Brewster, 2011) as the speed at which users spontaneously walk when using a mobile phone. Here again, the improvement is more dramatic for portrait-based methods (ranging from 5.3% to 11.9%) while for landscape-based methods the improvement is somewhat less impressive (ranging from 0.05% to 5.3%). This was confirmed with a paired t-test comparing portrait-based methods across all walking speeds (which was later also done for landscape-based methods). The following values were obtained: for the portrait case ($p < 0.001$, $df = 8$, $t = 10.22$, common language effect size ($CLES$) = 0.99); for the landscape case ($p = 0.52$, $df = 5$, $t = 0.67$, $CLES$ = 0.60). It can be concluded that if nothing else were known about the current input style, it might be better to build an offset model using 0.75PWS data, rather than static data, in order to reduce the error rate over every input technique and walking speed.

For accuracy as a second performance parameter, similar trends were observed. In the static condition the accuracy gain was lower (from 3.1 to 4.2 pixels depending on input technique) than in the dynamic case (from 5.6 to 12.7 pixels, depending on measurement condition). Implementing offset models built on corresponding model data therefore has a higher benefit in terms of accuracy in the dynamic case than in the static case. The same trends in the data can be observed in terms of how an offset model constructed on dynamic data outperforms a model based on static data. Although the difference is not so dramatic in terms of sheer amplitude (up to 4 pixels) it translates into significant results in terms of error rate, as was demonstrated earlier. An example of changes in model generation for all input methods, for one test subject, is presented in Figure 23. The figure was generated using the amplitude of the output of both x and y offset models (for particular on-screen location) for calculation of the total (in a Euclidian sense) offset amplitude. It would be interesting to change the walking speed in more precise increments and record what happens to the offset vector at particular screen locations and possibly infer an approximate analytical equation which describes these changes.

(Figure 23 about here)

Finally, the benefit of inclusion of speed in offset modeling was tested by including walking speed as a feature in the regression model. Walking speed was filtered with a 10-point moving average filter before inclusion in the model, in order to provide smoother input. In the training phase 80% of all posture-speed combinations (across all subjects) were used, while the remaining 20% was used for testing (this was done in 10-fold cross validation manner). In order to provide a fair comparison with other conditions, their results were sampled in similar way: random 20% of all posture-speed pairs (across all subjects) were selected for comparison with the new model. For all cases, the mean value and standard deviation were calculated as presented in Figure 24.

(Figure 24 about here)

These results suggest that the inclusion of walking speed decreases the error rate (10.88%) compared with the no compensation case. This we believe further emphasizes the speed dependency of offset models. However, it performs worst when compared to other compensation approaches: the corresponding speed model (5.36%) and 0.75PWS model (3.38%). It is worth noting that variation also increases significantly in the model with speed variable. One possibility is that the degraded performance could mainly be attributed to the variations in the speed estimate and that a more robust speed estimate or sensor would improve the performance of the model but more testing is needed.

5. CONCLUSIONS

In this paper we analyzed the influence of walking speed and input technique on standard performance parameters like error rate, accuracy, tapping time and throughput. The paper provides the most comprehensive analysis to date of finger tapping accuracy on capacitive touch screens, while mobile. This is done from the perspective of a particular input pose with all of its particularities, including different target sizes in portrait and landscape orientations.

The error rate in the static case (i.e. the subject was stationary) varied between a mean of 11.96% for the landscape-based index finger pointing technique to 30.12% for the portrait two thumbs pointing technique. Regardless of input technique, this error significantly increased when users started to move, with error rate minima for 75% or 100% of preferred walking speed (depending on technique).

In terms of accuracy in the static scenario, all methods demonstrated similar mean values, with the landscape index finger pointing being the most accurate and portrait two-thumbs pointing, least accurate. Similar trends to the case of error rate in the walking scenario were obtained. The results for tapping speed showed that mean values are again similar, with landscape two-thumb pointing being the fastest (0.517 s between taps) and portrait one thumb pointing the slowest (0.558 s between taps). Tapping speed increased when users started to move at speeds of 75% and 100% of preferred walking speed, while it dropped when 50% and 125% of preferred walking speed was used. Tapping speed was similar across all techniques, suggesting that users seemed to trade maintaining tap rate against worsening error rates.

Combining tapping accuracy and tapping speed in one parameter we analyzed the throughput, with the caveat that, due to the specific nature of the two-finger based input techniques, direct comparison of the results with similar cases studied in the literature (for example, for the computer mouse) are not straightforward. The largest throughput was obtained for landscape, index finger pointing (5.586 bit/s), while the lowest value was measured for the portrait, one-thumb pointing technique (4.968 bits/s). This result is somewhat surprising, since it shows that the portrait, one-thumb input technique is the slowest in transmitting user intent to the device but 60% of all subjects stated in their pre-measurement interviews that it was their preferred interaction technique.

In summary, these results lead to several conclusions which might be of interest to designers of mobile phone interfaces and/or applications:

1. Users performed better, in terms of throughput and error rate, with index finger pointing, while holding the phone with the other hand, than with thumb pointing. Also, landscape modes generally performed better (in terms of error rate) than portrait-based ones.
2. All parameters for all input techniques worsened when the user started to move (even with slower speeds), with best performance in general being measured at preferred walking speed.
3. Tapping time measurements suggested that users seemed to make a tradeoff, maintaining tapping speed against worsening error rates.

Implications for design: Such quantification of variations in performance in different postures might be useful for designers when deciding whether to support particular posture modes for mobile use. Furthermore, the quantification of worsening performance for different conditions can be used to recommend upper limits on accuracy for interfaces you want people to be able to use the system on the move, or to provide a principled basis to support adaptive simplification of interfaces as users walk more rapidly. Any such context-sensitive simplification of the interface would be visible to users, and would need to be communicated using carefully chosen design metaphors.

Context-dependent offset models are better than static offset models: We also analyzed how pose and walking speed influence the accuracy of offset models used to infer finger position. Offset models were built using simple regression with a quadratic model. The results confirmed earlier conclusions from the literature, that the use of offset models can significantly reduce error rates. In our measurements, the error rate was reduced by between 6-17% in the static case, while the improvement was even more dramatic in the mobile scenario, where the reduction was between 14-30% (depending on the input technique). This reduction was based on offset models built on data from the corresponding walking speed-input technique combination, but if a different walking speed and/or input technique was used (even in static condition) the correction rate degraded significantly. This demonstrated the walking speed and input technique variability of offset models. We demonstrated that building a global model on 75% preferred walking speed (and the appropriate input technique) is a good compromise,

although context-adaptive methods which blend models learned at different speeds would probably improve on this further.

The use of such context-dependent offset models, in conjunction with decisions based on the influence of walking speed, and the input technique, has the potential to both improve the user experience and allow for smaller-sized controls in a manner invisible to the user.

Gait phase effects for capacitive touch differ from earlier stylus results: Gait phase analysis demonstrated that the tapping distribution is dependent on the input technique being used, as well as the current walking speed. The existence of peaks in gait phase distributions with respect to touch events suggests that subjects are likely to tap during certain gait phases, demonstrating tap alignment with steps. In our analysis there was no strong correlation between preferred regions of gait phase for tapping and lower error rates, in contrast to earlier work (Crossan, Murray-Smith, Brewster, Kelly & Musizza, 2005). Although regions of higher accuracy were detected they did not correspond to the peaks in the gait phase distribution. This suggests that the different hardware and input styles used in these studies (PDA with resistive screen and physical stylus vs. capacitive touch screen on mobile phone with index finger pointing) make a significant difference to user behavior. One result of the analysis in this paper suggests that interfaces which encourage a rhythm of tapping might not have as high improvements in finger actuated capacitive touch screen performance as expected, given the results of (Crossan, Murray-Smith, Brewster, Kelly & Musizza, 2005).

Future research directions: In this study, users were selecting abstract targets (blank virtual keys) which might have some influence on performance parameters and offset models, as users aim at a visual target in an arbitrary location. It would be interesting to compare this with the results when using standard interface elements, like keyboards where the visual targeting is eliminated by learning effects (e.g. when typing a message, users know which key to press next and the general area where that key is located) and where actions are affected by co-articulation effects related to key presses before and after. Further learning effects include providing feedback information on targeting success which would allow users to adapt their behavior during the experiment, to improve performance. The conservative assumptions of no precision feedback in our study simplify the experiment design and analysis, but might make the results overly pessimistic.

The experiment design can also be generalized to other popular gestural inputs, and could be repeated to test the usability of more complex interactions such as swipe gestures, pinch to zoom, gesture typing vs. tap typing etc. while in motion and with a range of phone postures. Similarly, it could be applied to touch interaction on other wearables, such as smart watches.

The walking behavior chosen here was relatively simple, indoors, on an even surface and with minimal external disturbances. Future developments might use inertial sensing to log user tapping behavior 'in the wild', subject to more complicated scenarios.

REFERENCES

Accot, J., & Zhai, S. (2003). Refining Fitts' law models for bivariate pointing. *Proceedings of the CHI 2003 Conference on Human Factors in Computer Systems*. New York: ACM.

Apple Inc. (2010). iOS Human Interface Guidelines, <https://developer.apple.com/library/ios/navigation/> (Last accessed 07.11.2013.)

Azenkot, S., & Zhai, S. (2012). Touch behavior with different postures on soft smartphone keyboards. *Proceedings of the MobileHCI 2012 Conference on Human-Computer Interaction with Mobile Devices and Services*. New York: ACM.

Baldwin, T., & Chai, J. Y. (2012). Towards online adaptation and personalization of key-target resizing for mobile devices. *Proceedings of IUI 2012 Conference on Intelligent User Interfaces*. New York: ACM.

Barnard, L., Yi J. S., Jacko J. A., & Sears A. (2005). An empirical comparison of use-in-motion evaluation scenarios for mobile computing devices, *International Journal of Human-Computer Studies*, 62 (4), 487-520.

Bergström-Lehtovirta, J., Oulasvirta, A., & Brewster, S. (2011). The effects of walking speed on target acquisition on a touchscreen interface, *Proceedings of the MobileHCI 2011 Conference on Human-Computer Interaction with Mobile Devices and Services*. New York: ACM.

Bertram, J. E. A., & Ruina, A. (2001). Multiple walking speed-frequency relations are predicted by constrained optimization, *Journal of Theoretical Biology*, 209 (4), 445-453.

Böhm, W., & Hornik, K. (2012). A Kolmogorov-Smirnov test for r samples, *Fundamenta Informaticae*, 117 (1-4), 103-125.

Breiman, L. (2001), Random Forests, *Machine Learning* 45(1), 5-32.

Brewster, S. A., (2002). Overcoming the lack of screen space on mobile computers, *Journal Personal and Ubiquitous Computing*, 6 (3), 188-205.

Buschek, D., Rogers, S., & Murray-Smith, R. (2013). User-specific touch models in a cross-device context. *Proceedings of the MobileHCI 2013 Conference on Human-Computer Interaction with Mobile Devices and Services*. New York: ACM.

Clark, J. (2010). *Tapworthy: Designing great iPhone apps* (1st ed.). Sebastapol: O'Reilly Media

Conover, W. J., (1999). *Practical nonparametric statistics* (3rd ed.). New York: John Wiley & Sons Inc.

- Crossan, A., Murray-Smith, R., Brewster, S., Kelly, J., & Musizza, B. (2005). Gait phase effects in mobile interaction. *Proceedings of the CHI EA 2005 Conference on Human Factors in Computer Systems*. New York: ACM.
- Findlater, L., & Wobbrock, J. O. (2012). Personalized input: Improving ten-finger touchscreen typing through automatic adaptation. *Proceedings of the CHI 2012 Conference on Human Factors in Computer Systems*. New York: ACM.
- Flemisch, F., Adams, C., Conway, S., Goodrich, K., Palmer, M., & Schutte, P. (2003). *The h-metaphor as a guideline for vehicle automation and interaction* (Technical Report NASA/TM-2003-212672). Hampton, Virginia: Langley Research Center.
- Genuer, R., Poggi, J.-M., & Tuleau-Malot, C. (2010). Variable selection using random forests, *Patter Recognition Letters*, 31 (14), 2225-2236.
- Goel, M., Findlater, L., & Wobbrock, J. O. (2012). WalkType: Using accelerometer data to accommodate situational impairments in mobile touch screen text entry. *Proceedings of the CHI 2012 Conference on Human Factors in Computer Systems*. New York: ACM.
- Goel, M., Jansen, A., Mandel, T., Patel, S. N., & Wobbrock, J. O. (2013), ContextType: Using hand posture information to improve mobile touch screen text entry. *Proceedings of the CHI 2013 Conference on Human Factors in Computer Systems*. New York: ACM.
- Goel, M., Wobbrock, J. O., & Patel, S. N. (2012). GripSense: Using built-in sensors to detect hand posture and pressure on commodity mobile phones. *Proceedings of the UIST 2012 Symposium on User Interface Software and Technology*. New York: ACM.
- Gunawardana, A., Peak, T., & Meek, C. (2010). Usability guided key-target resizing for soft keyboards. *Proceedings of the IUI 2010 Conference on Intelligent User Interfaces*. New York: ACM.
- Henze, N., Rukzio, E., & Boll, S. (2011). 100,000,000 taps: Analysis and improvement of touch performance in the large. *Proceedings of the MobileHCI 2011 Conference on Human-Computer Interaction with Mobile Devices and Services*. New York: ACM.
- Himberg, J., Häkkinen, J., Kangas, P., & Mäntyjärvi, J. (2003). On-line personalization of a touch screen based keyboard. *Proceedings of the IUI 2003 Conference on Intelligent User Interfaces*. New York: ACM.
- Hoggan E., Brewster, S., & Johnston, J. (2008). Investigating the effectiveness of tactile feedback for mobile touchscreens. *Proceedings of the CHI 2008 Conference on Human Factors in Computer Systems*. New York: ACM.

- Holz, C., & Baudisch, P. (2010). The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. *Proceedings of the CHI 2010 Conference on Human Factors in Computer Systems*. New York: ACM.
- Holz, C., & Baudisch, P. (2011). Understanding touch. *Proceedings of the CHI 2011 Conference on Human Factors in Computer Systems*. New York: ACM.
- Johnson, P. (1998). Usability and Mobility; Interactions on the move. *Proceedings of the MobileHCI 1998 Conference on Human-Computer Interaction with Mobile Devices and Services*. New York: ACM.
- Kane, S. K., Wobbrock, J. O., Harniss, M., & Johnson, K. L. (2008). TrueKeys: Identifying and correcting typing errors for people with motor Impairments. *Proceedings of the IUI 2008 Conference on Intelligent User Interfaces*. New York: ACM.
- Kane, S. K., Wobbrock, J. O., & Smith, I. E. (2008). Getting of the treadmill: evaluating walking user interfaces for mobile devices in public spaces. *Proceedings of the MobileHCI 2008 Conference on Human-Computer Interaction with Mobile Devices and Services*. New York: ACM.
- Kjeldskov, A. K., & Stage, J. (2004). New techniques for usability evaluation of mobile systems, *International Journal of Human-Computer Studies*, 60 (5-6), 599-620.
- Lee, S., & Zhai, S. (2009) The performance of touch screen soft buttons. *Proceedings of the CHI 2009 conference on Human Factors in Computing Systems*. New York: ACM.
- Lin, M. K., Goldman, R., Price, K. J., Sears, A., & Jacko, J. (2007). How do people tap when walking? An empirical investigation of nomadic data entry, *International Journal of Human-Computer Studies*, 65 (9), 759-769.
- MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction, *Human-Computer Interaction*, 7, 91-139.
- MacKenzie, I. S., & Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. *Proceedings of the CHI 1992 Conference on Human Factors in Computer Systems*. New York: ACM.
- Mizobuchi, S., Chignell, M., & Newton, D. (2005). Mobile text entry: Relationship between walking speed and text input task difficulty. *Proceedings of the MobileHCI 2005 Conference on Human-Computer Interaction with Mobile Devices and Services*. New York: ACM.
- Murray-Smith, R., Ramsay, A., Garrod, S., Jackson, M., & Musizza, B. (2007). Gait alignment in mobile phone conversation. *Proceedings of the MobileHCI 2007 Conference on Human-Computer Interaction with Mobile Devices and Services*. New York: ACM.

- Nicolau, H., & Jorge, J. (2012) Touch typing using thumbs: understanding the effect of mobility and hand posture. *Proceedings of the CHI 2012 Conference on Human Factors in Computing Systems*. New York: ACM.
- Nokia (2011). Nokia N9 Icon Guidelines, <http://developer.nokia.com/Devices/MeeGo/> (Last accessed 30.10.2013.)
- Parhi, P., Karlson, A. K., & Bederson, B. (2006). Target size study for one-handed thumb use on small touchscreen devices. *Proceedings of the MobileHCI 2006 Conference on Human-Computer Interaction with Mobile Devices and Services*. New York: ACM.
- Park, Y. S., Han, S. H., Park, J., & Cho, Y. (2008). Touch key design for target selection on a mobile phone. *Proceedings of the MobileHCI 2008 Conference on Human-Computer Interaction with Mobile Devices and Services*. New York: ACM.
- Pirhonen, A., Brewster, S., & Houlguin, C. (2002). Gestural and audio metaphors as a means of control for mobile devices. *Proceedings of the CHI 2002 Conference on Human Factors in Computer Systems*. New York: ACM.
- Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning* (1st ed.). Cambridge: MIT Press.
- Renaudin, V., Susi, M., & Lachapelle, G. (2012). Step length estimation using handheld inertial sensors, *Sensors*, 12 (7), 8507-8525.
- Ribeiro, J., & Carvalhais, M. (2012). Web design patterns for mobile devices. *Proceedings of the PLOP 2012 Conference on Pattern Languages of Programs (Writer's Workshop)*. Berlin: Springer.
- Rogers S., Williamson, J., Stewart, C., & Murray-Smith, R. (2010). Fingercloud: uncertainty and autonomy handover in capacitive sensing. *Proceedings of the CHI 2010 Conference on Human Factors in Computer Systems*. New York: ACM.
- Rogers S., Williamson, J., Stewart, C., & Murray-Smith, R. (2011). AnglePose: Robust, precise capacitive touch tracking via 3D orientation estimation. *Proceedings of the CHI 2011 Conference on Human Factors in Computer Systems*. New York: ACM.
- Schildbach, B. & Rukzio, E., (2010). Investigating selection and reading performance on a mobile phone while walking. *Proceedings of the CHI 2010 Conference on Human Factors in Computer Systems*. New York: ACM.
- Sears, A. (1991). Improving touchscreen keyboards: Design issues and a comparison with other devices, *Interacting with computers*, 3 (3), 253-269.

Vogel, D., & Baudisch, P. (2007). Shift: A technique for operating pen-based interfaces using touch. *Proceedings of the CHI 2007 Conference on Human Factors in Computer Systems*. New York: ACM.

Weir D., Buschek D., & Rogers, S. (2013). Sparse selection of training data for touch correction systems. *Proceedings of the CHI 2013 Conference on Human Factors in Computer Systems*. New York: ACM.

Weir, D. Pohl, H., Rogers, S., Vertanen, K. Kristensson, P. O. (2014). Uncertain text entry on mobile devices. *Proceedings of the CHI 2014 Conference on Human Factors in Computing Systems*. New York: ACM.

Weir, D., Rogers, S., Murray-Smith, R., & Löchtefeld, M. (2012). A user-specific machine learning approach for improving touch accuracy on mobile devices. *Proceedings of the UIST 2012 Symposium on User Interface Software and Technology*. New York: ACM.

Wobbrock, J. O., Cutrell, E., Harada, S., & MacKenzie (2008). An error model for pointing based on Fitts' law. *Proceedings of the CHI 2008 Conference on Human Factors in Computer Systems*. New York: ACM.

Yamabe, T., & Takahashi, K. (2007). Experiments in mobile user interface adaptation for walking users. *Proceedings of 2007 International Conference on Intelligent Pervasive Computing*. Jeju City:IEEE.

Yatani, K., & Truong, K. N. (2007). An evaluation of stylus-based text entry methods on handheld devices in stationary and mobile settings. *Proceedings of the MobileHCI 2007 Conference on Human-Computer Interaction with Mobile Devices and Services*. New York: ACM.

Yesilada, Y., Harper, S., Chen, T., & Trewin, S. (2010). Small-device users situationally impaired by input, *Computers in Human Behavior*, 26, 427-435.

Yin, Y., Ouyang, T. Y., Partridge, K., & Zhai, S. (2013). Making touchscreen keyboards adaptive to keys, hand postures, and individuals – A hierarchical spatial backoff model approach. *Proceedings of the CHI 2013 Conference on Human Factors in Computer Systems*. New York: ACM.

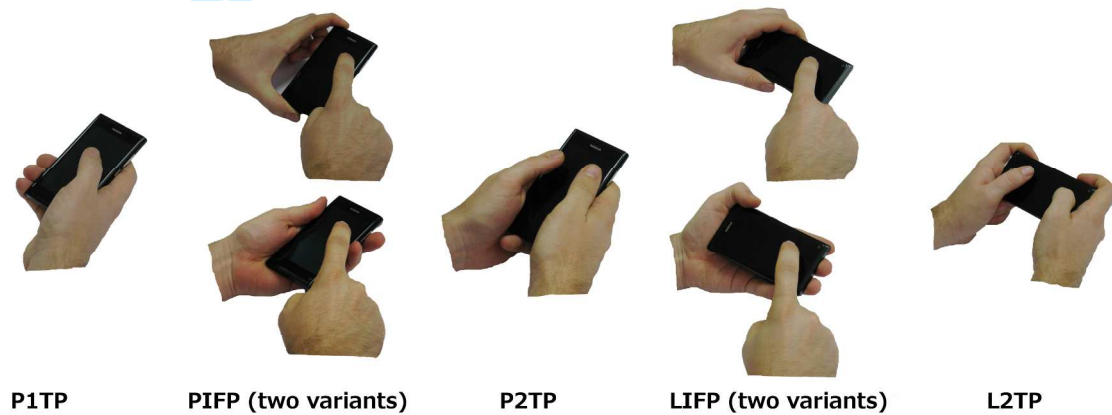


Figure 1 – Input techniques used in the experiment (P1TP – portrait one thumb pointing, PIFP – portrait index finger pointing, P2TP – portrait two thumbs pointing, LIFP – landscape index finger pointing, L2TP – landscape two thumbs pointing)

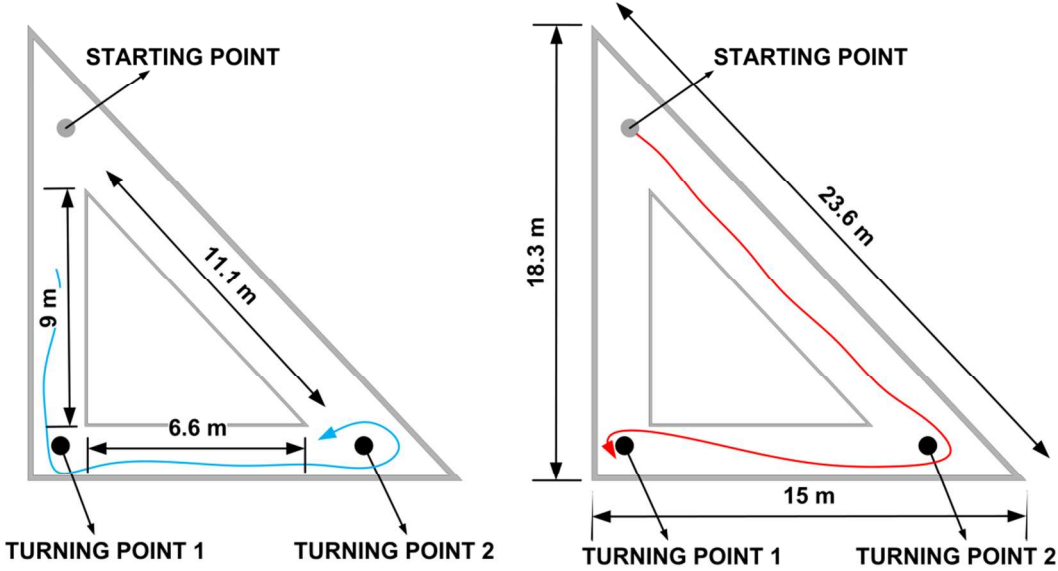


Figure 2 – Sketch of walking path used in the experiment along with dimensions. Left: transition from counterclockwise to clockwise walking direction. Right: transition from clockwise to counterclockwise walking direction. Subjects always started the measurement in the clockwise direction.

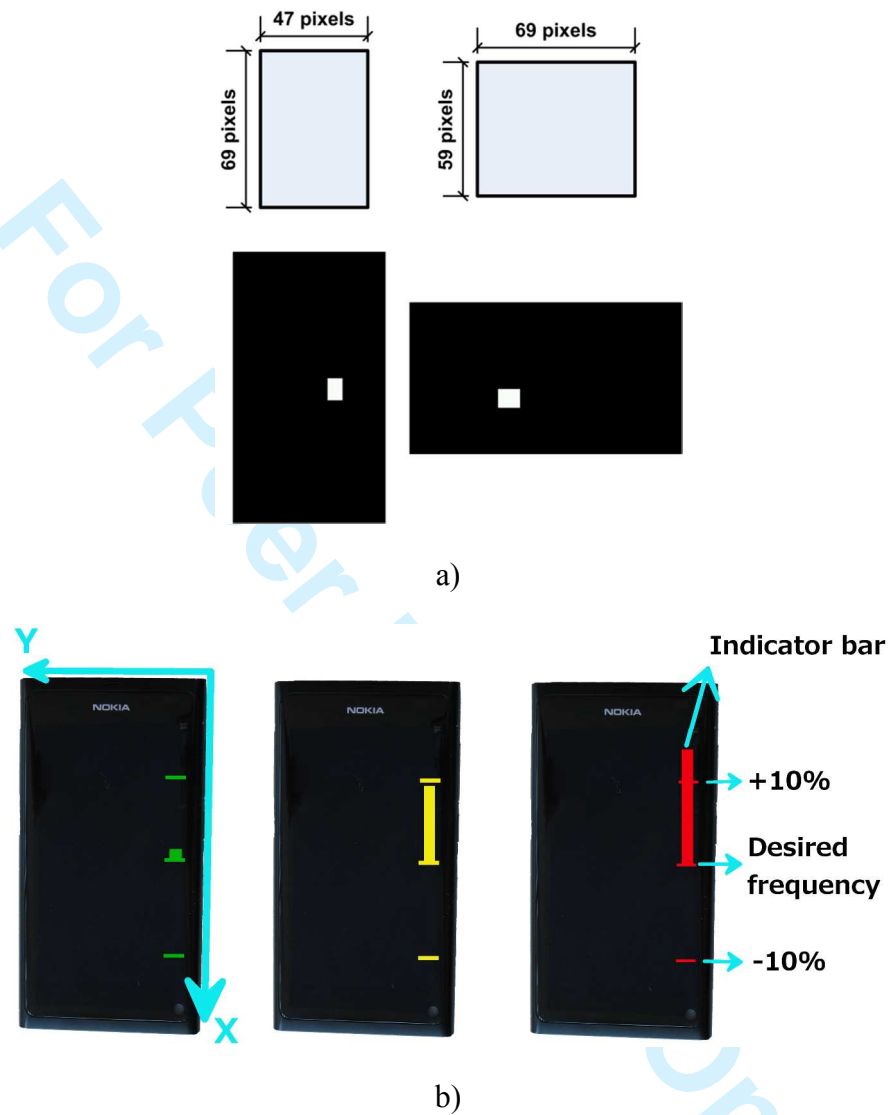


Figure 3 – Mobile phone used in experiment: a) dimensions of target squares used in the experiment (upper row) and screenshot of the mobile phone screen (not the entire phone) during measurements (lower row); b) examples of user feedback on gait frequency. The changing color of frequency ticks and indicator bar, depends on the current gait frequency. Yellow was used to indicate the within-range frequency, close to $\pm 10\%$ limits.

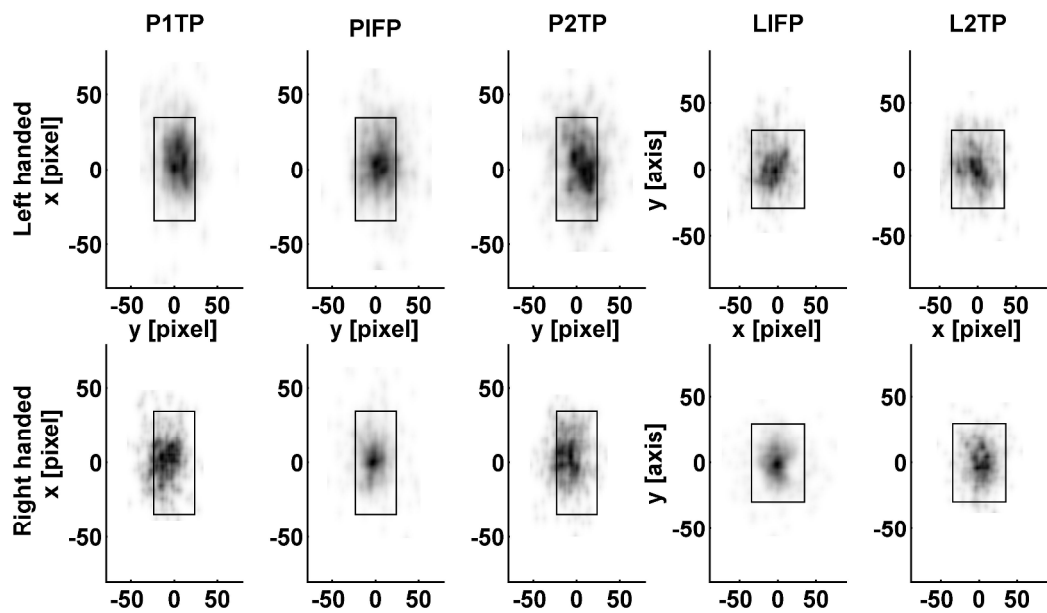


Figure 4 – Smoothed scatter plot of normalized (with respect to 0,0) touch locations for all input techniques in static conditions of 3 left handed (upper row) and 3 right handed (lower row) test subjects. The central square represents the actual target for both phone orientations (note the changed coordinate frame orientation).

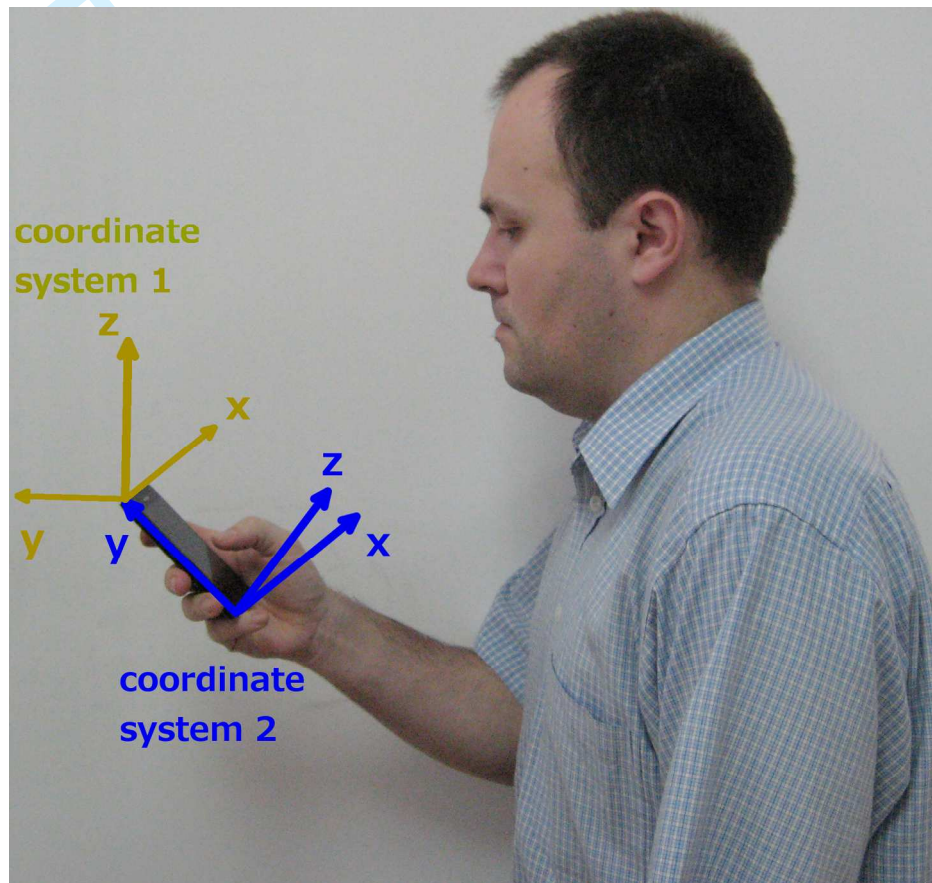


Figure 5 – Tilt and pitch compensation of the mobile phone coordinate system (note that the figure depicts an accelerometer coordinate system which is different from the screen coordinate system – see Figure 3)

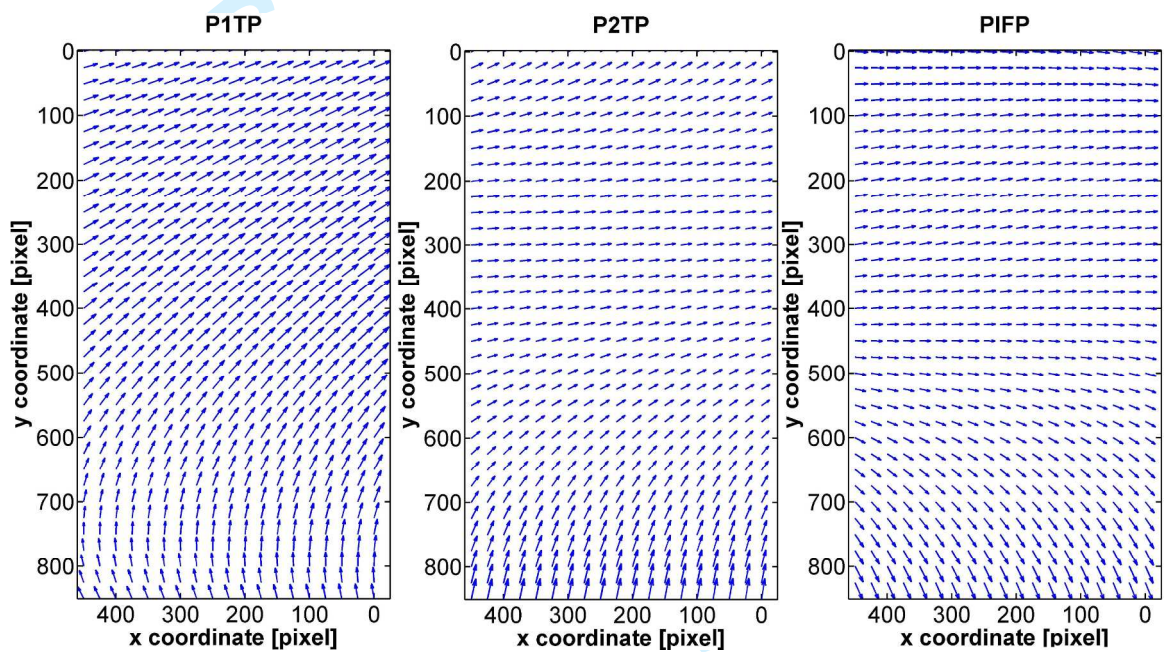


Figure 6 – Offset model vector fields for one test subject in portrait mode trained on static data. Each arrow corresponds to the offset vector in x and y direction associated with a 25x25 pixel area.

Input technique	Error rate [%]		Total error (accuracy) [pixel]		Tapping time [s]		Throughput [bits/s]	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD
P1TP	27.74	16.21	24.71	6.33	0.5583	0.1644	4.977	1.498
PIFP	25.49	18.6	23.55	8.10	0.5367	0.1490	5.457	1.631
P2TP	30.12	16.03	23.75	6.69	0.5407	0.1625	5.161	1.607
LIFP	11.96	13.72	21.75	7.18	0.5367	0.1546	5.586	1.612
L2TP	17.95	15.17	24.38	7.16	0.5172	0.1330	5.446	1.795

Figure 7 – Mean values and associated standard deviations (STD) for static condition for four performance parameters

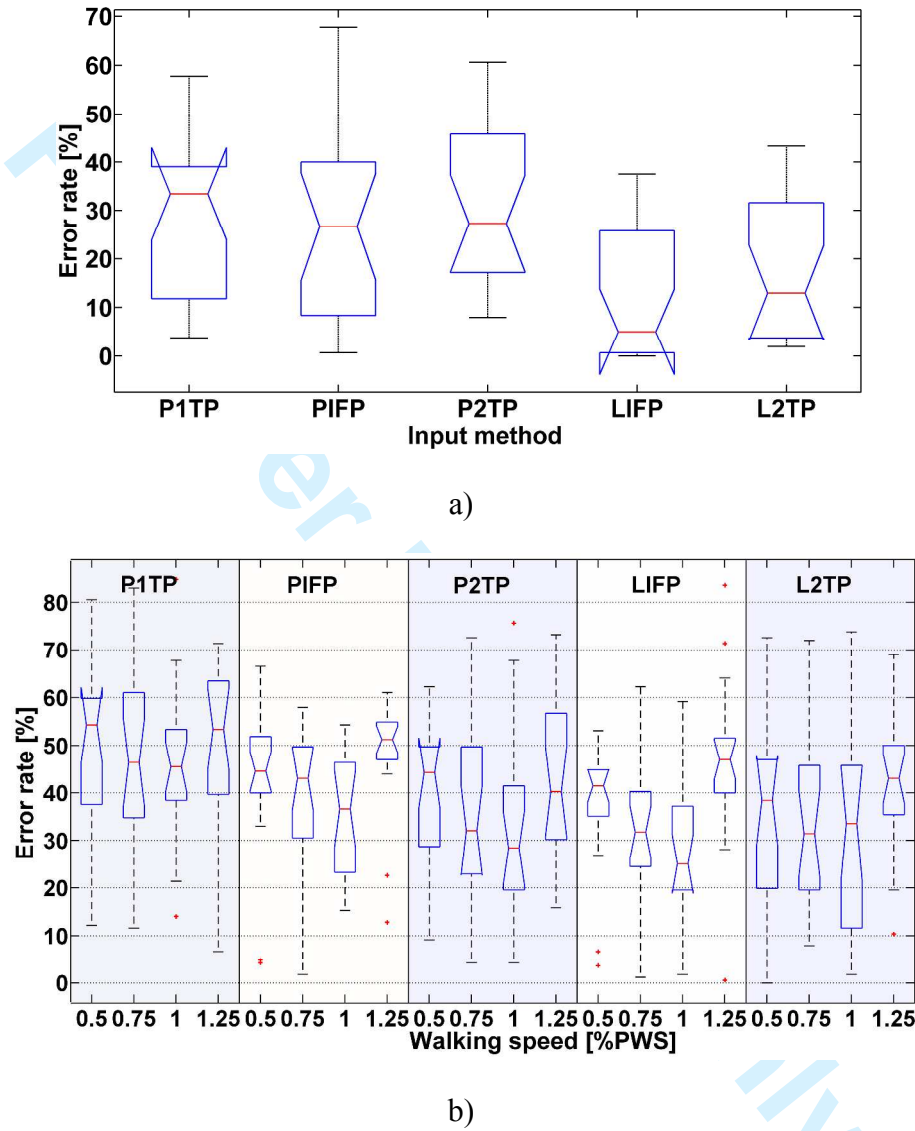


Figure 8 – Box-and-whiskers plot of error rate for: a) static conditions across all test subjects for all input techniques; b) dynamic conditions across all test subjects for all input techniques

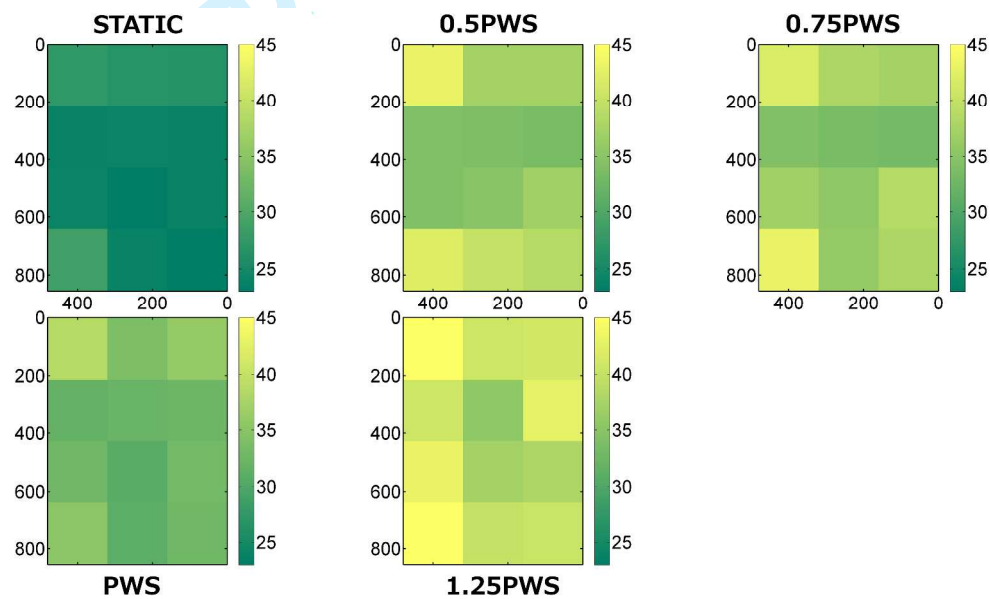


Figure 9 – Heat map plot of error rates at screen divided into 12 sections during P1TP interaction (first row: static, 0.5PWS, 0.75PWS; second row: PWS, 1.25PWS). Note that values are representative of error rates for screen grid parts and not the overall error rate.

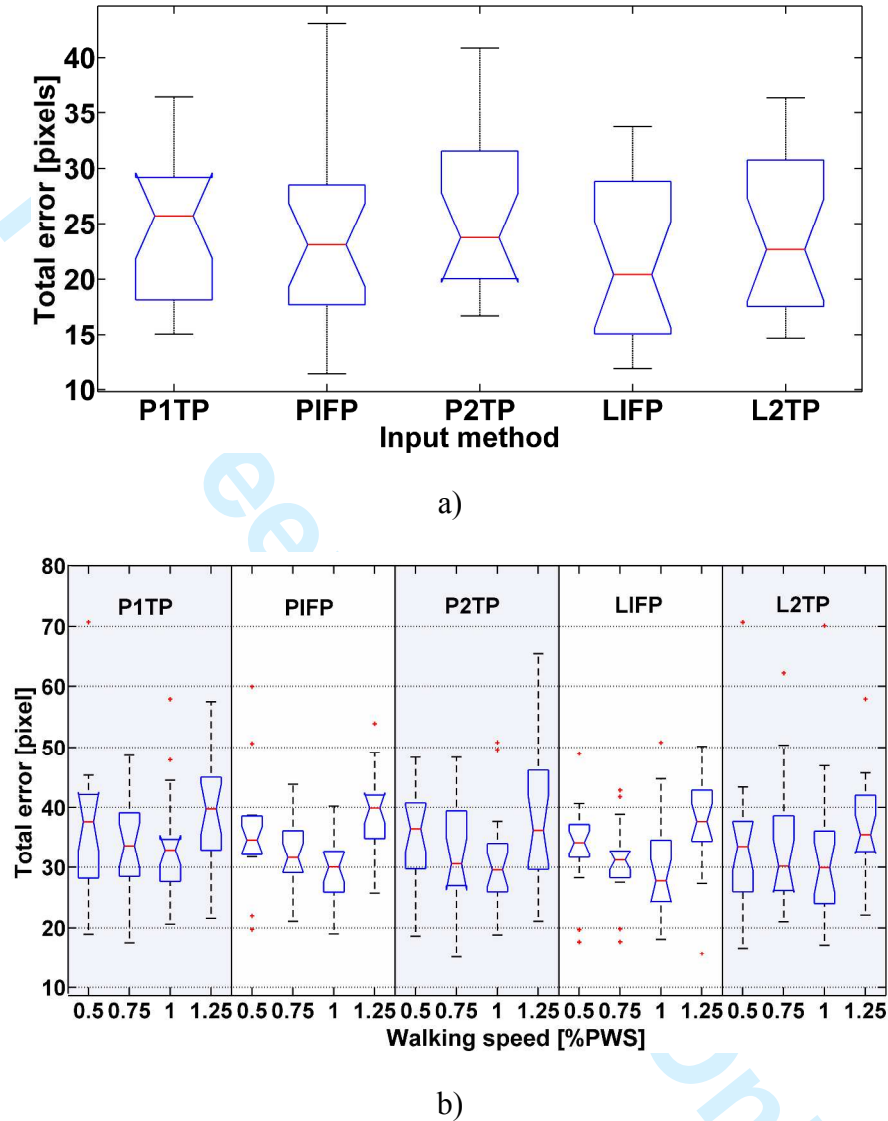


Figure 10 – Box-and-whiskers plot of total error for: a) static conditions across all test subjects for all input techniques; b) dynamic conditions across all test subjects for all input techniques.

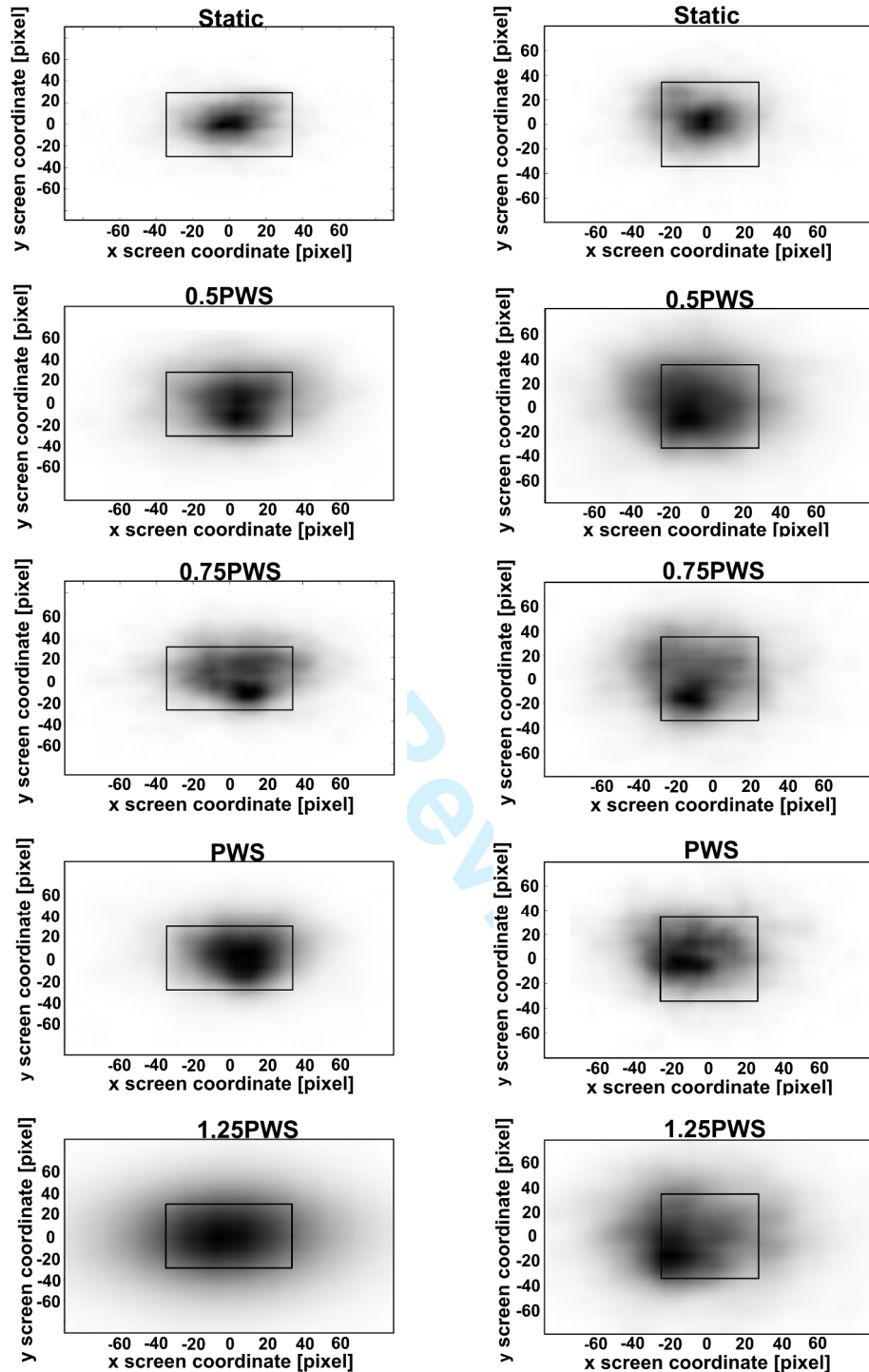


Figure 11 – Smoothed scatter plot of normalized (in respect to 0,0) touch locations for the PIFP input technique (left figure) and LIFP input technique (right figure) in all measurement conditions across all test subjects. The central square represents the actual target in portrait/landscape mode.

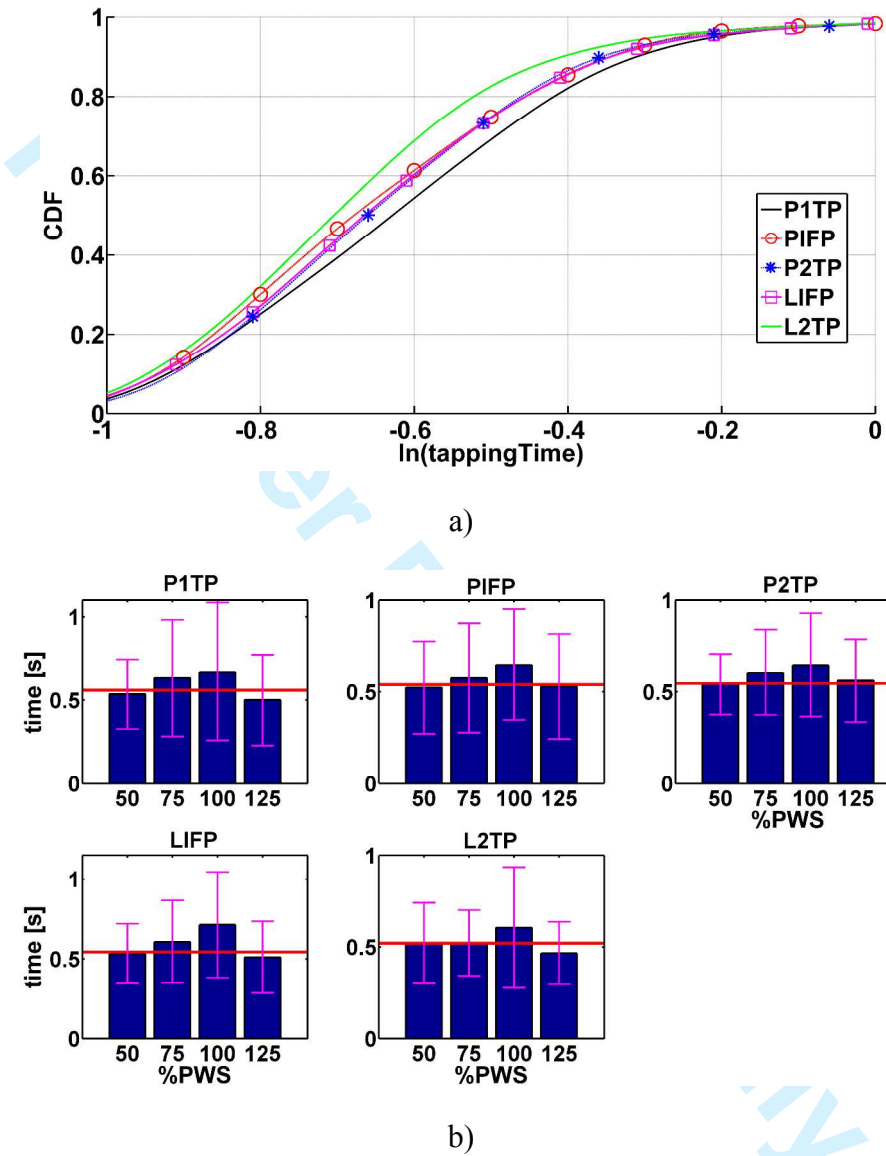
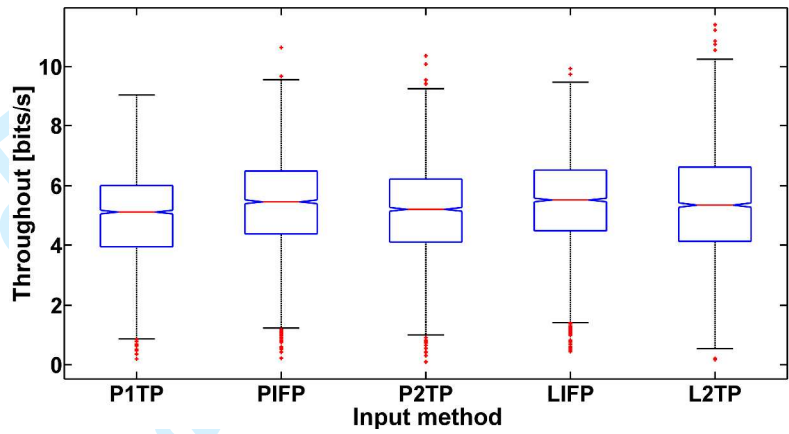


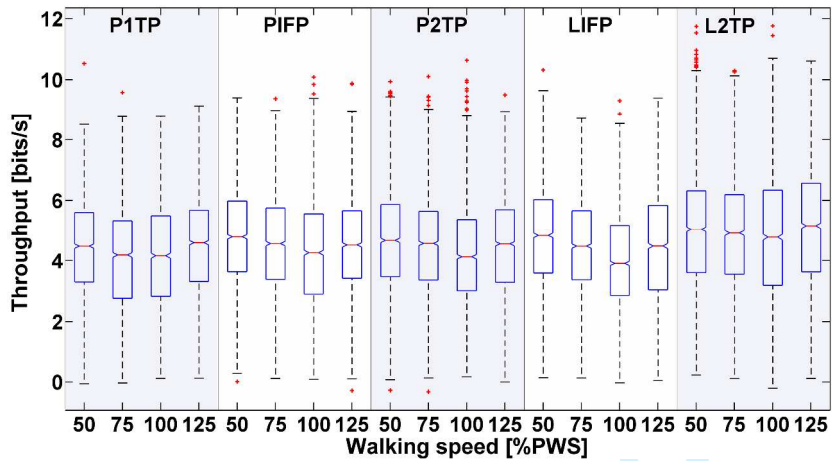
Figure 12 - Tapping speed results: a) Cumulative distribution function (CDF) plot of tapping time for different input methods in the static condition, over all test subjects, b) comparison of tapping time means over all test subjects in different dynamic conditions and for different input methods. The horizontal red line corresponds to the mean tapping time in the static condition.

Input technique	Correlation coefficient	95% confidence interval		Statistical significance (p value)
		lower bound	upper bound	
P1TP	-0.19	-0.23	-0.15	p<0.001
PIFP	-0.11	-0.15	-0.08	p<0.001
P2TP	-0.09	-0.13	-0.06	p<0.001
LIFP	-0.18	-0.21	-0.14	p<0.001
L2TP	-0.13	-0.16	-0.09	p<0.001

Figure 13 – Correlation analysis of tapping speed and total error in static condition



a)



b)

Figure 14– Comparison of throughput values obtained for: a) all input methods in the static condition, b) all input methods in the dynamic condition.

Input technique	Walking speed	Throughput [bits/s]	
		Mean	STD
P1TP	0.5PWS	4.525	1.624
	0.75PWS	4.354	1.717
	PWS	4.319	1.761
	1.25PWS	4.306	1.724
PIFP	0.5PWS	4.732	1.628
	0.75PWS	4.663	1.616
	PWS	4.558	1.689
	1.25PWS	4.517	1.636
P2TP	0.5PWS	4.508	1.534
	0.75PWS	4.555	1.645
	PWS	4.481	1.727
	1.25PWS	4.469	1.699
LIFP	0.5PWS	4.960	1.467
	0.75PWS	4.78	1.559
	PWS	4.640	1.618
	1.25PWS	4.558	1.748
L2TP	0.5PWS	5.202	1.895
	0.75PWS	5.001	1.842
	PWS	5.035	2.108
	1.25PWS	5.007	2.045

Figure 15 – Throughput means and standard deviations (STD) for all input techniques and all walking speed

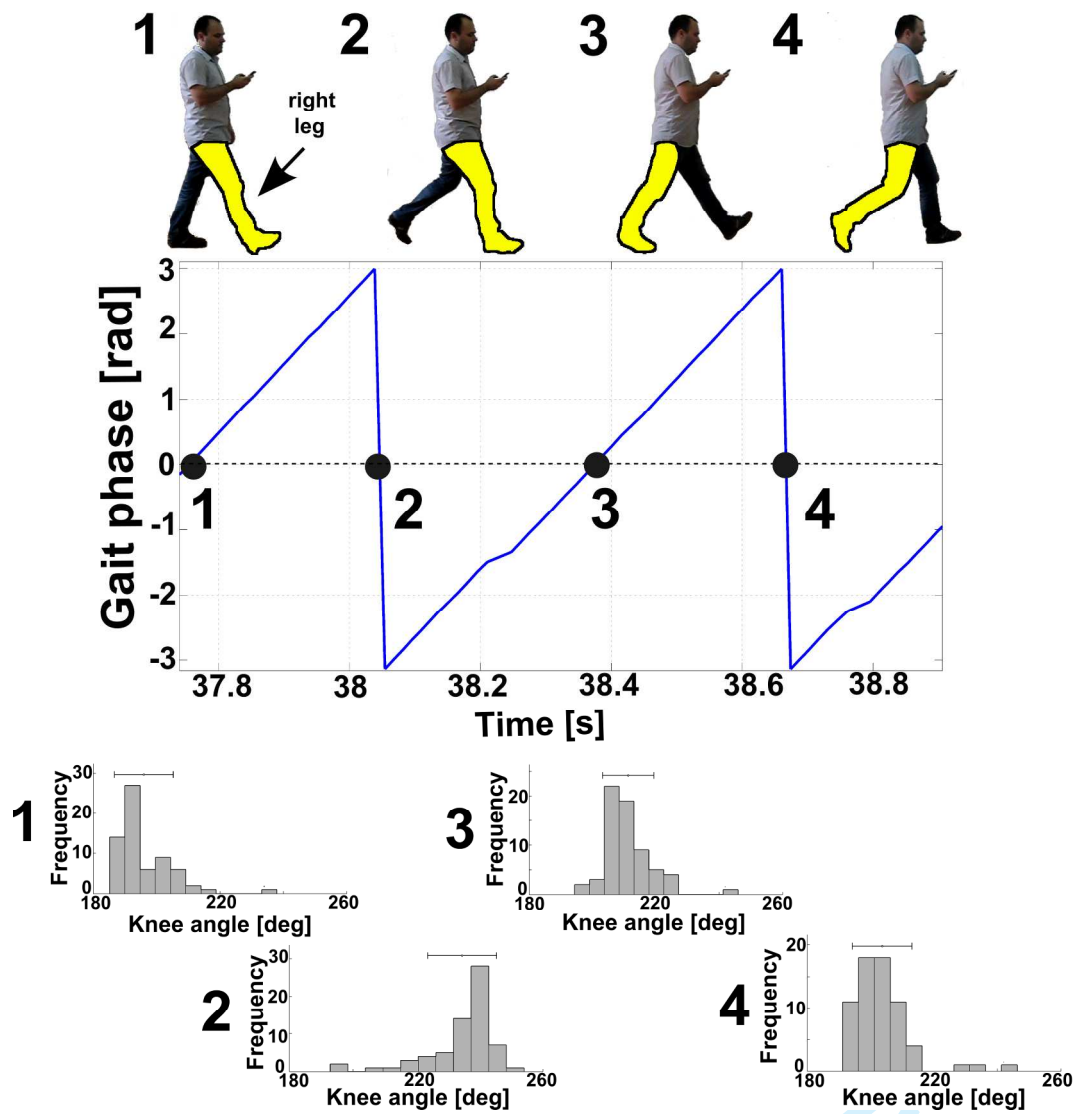


Figure 16– Illustration of the relationship between actual subject gait and gait phase zero crossings. Histograms depict distribution of right knee angle values for time instances when zero-crossing occurred. Four key points are marked with corresponding numbers in all subplots.

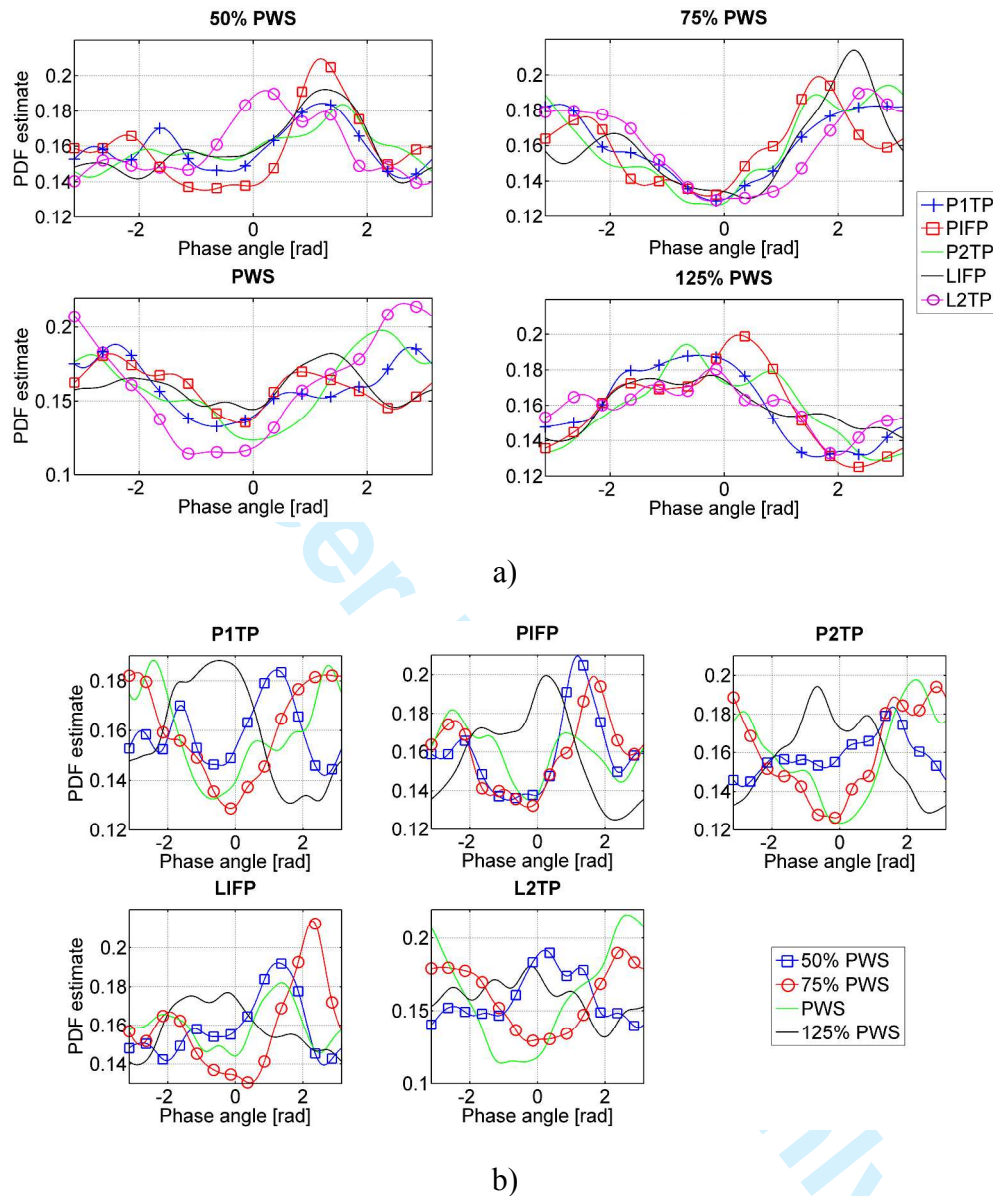
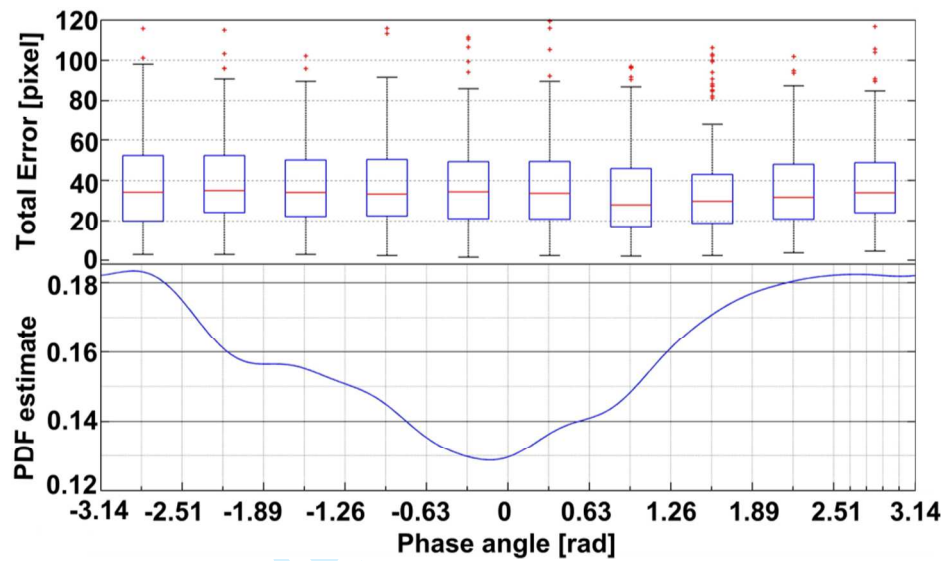
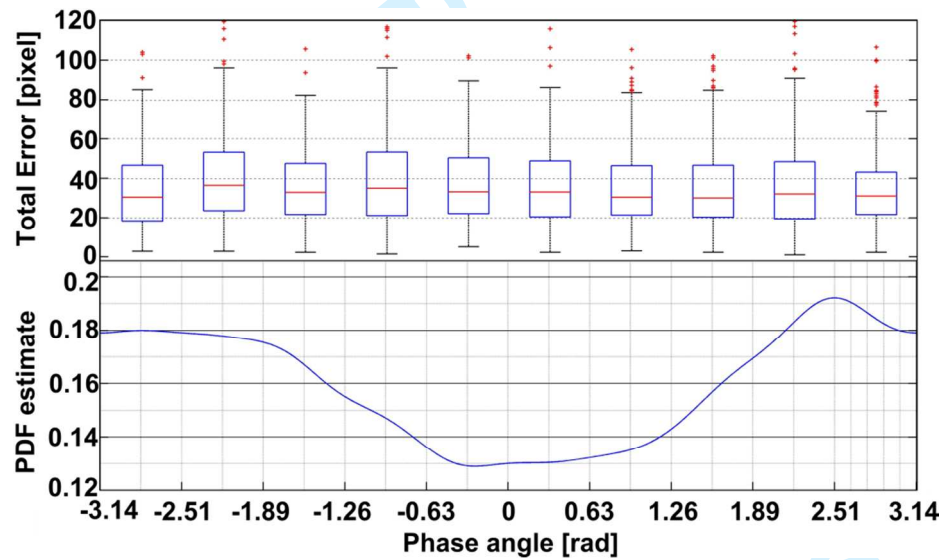


Figure 17– Comparison of gait phase angle distributions along z axis. Figure 17a depicts a comparison for each walking speed over all input methods while Figure 17b depicts a comparison of each input method over all walking speeds.



a)



b)

Figure 18– Examples of gait phase/total error analysis along the z axis for two input method-walking speed scenarios: a) figure depicts the P1TP-0.75PWS scenario; b) figure depicts the L2TP-0.75PWS scenario. Gait phases have been binned into 10 groups. For comparison reasons, the PDF estimate is also plotted.

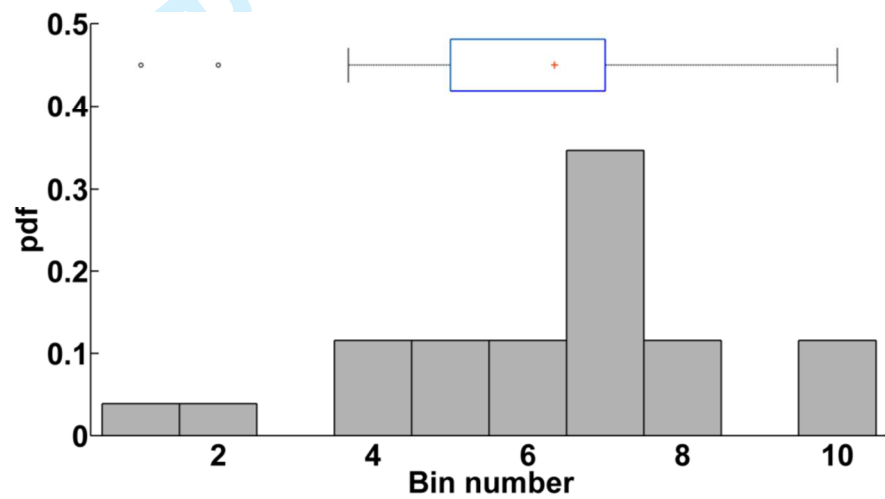
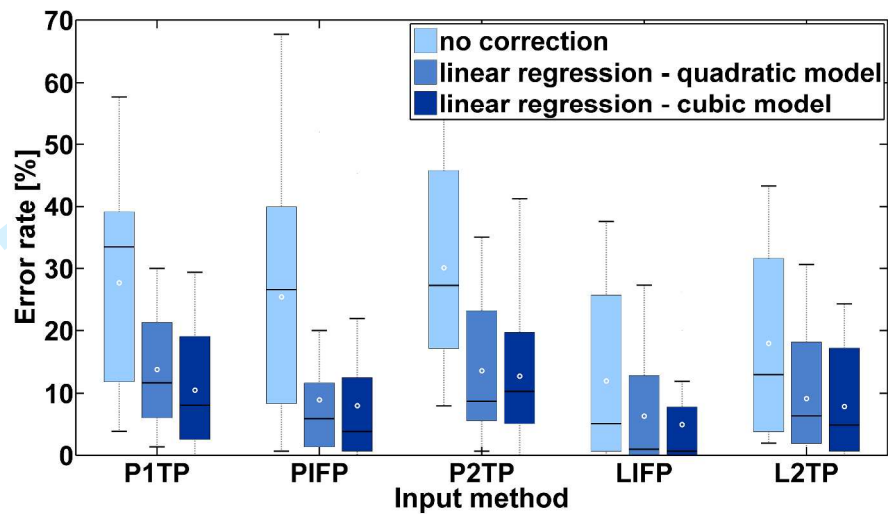
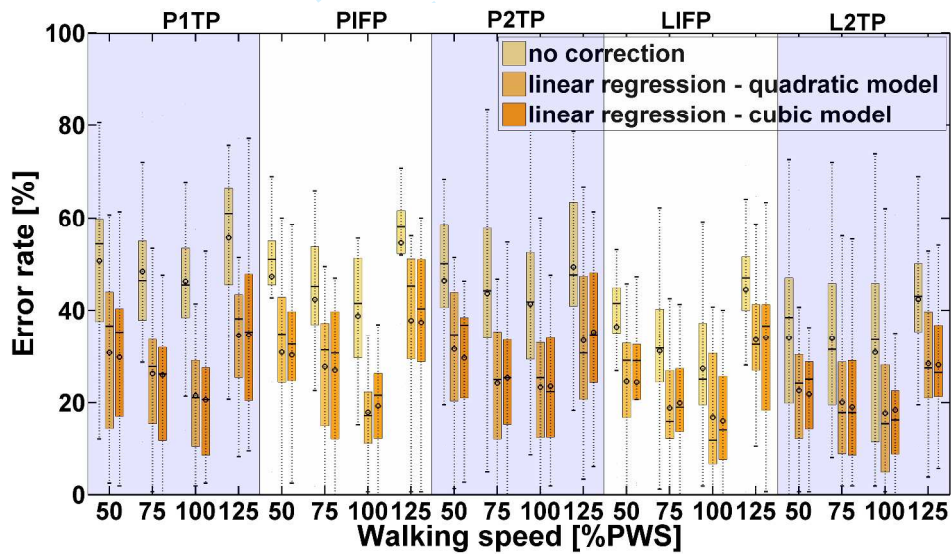


Figure 19 – Histogram of gait phase bins with highest accuracies across all pose-speed combinations and subjects.



a)



b)

Figure 20 – Error rate box plots for corrected and uncorrected bias in different conditions for the a) static and b) dynamic cases (black and white circles represent means). Data from the condition was used to build the model.

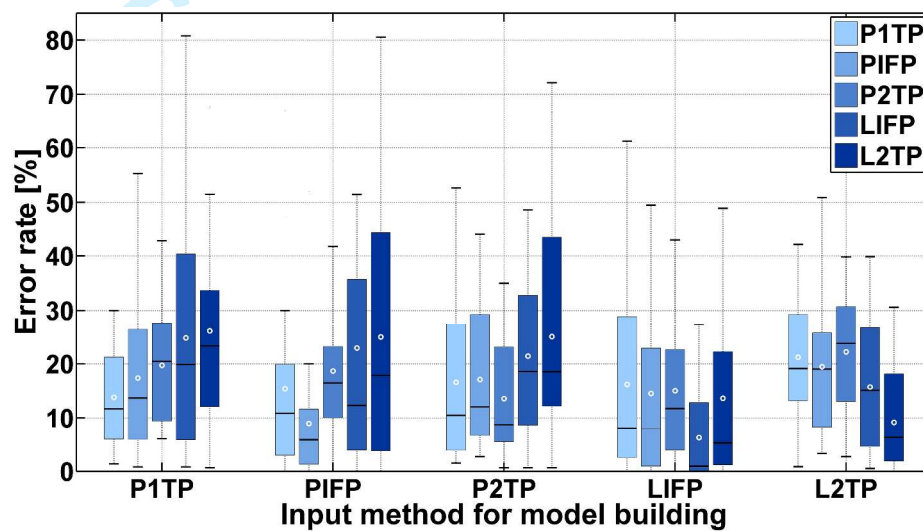
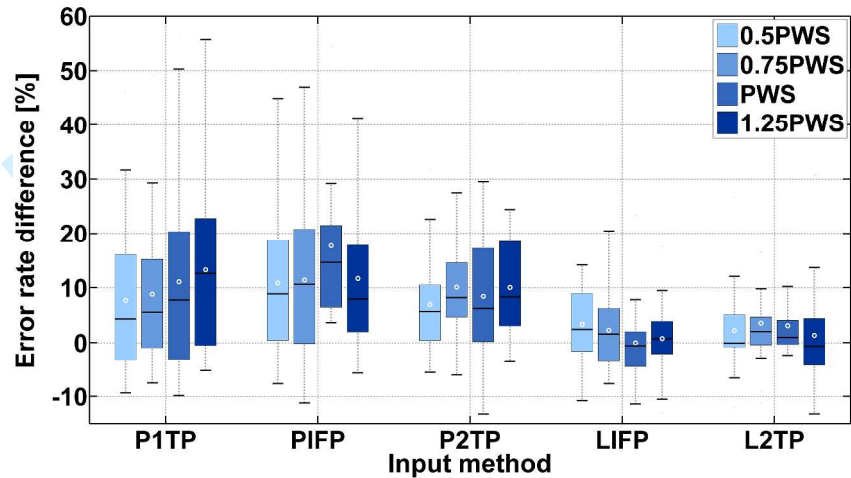
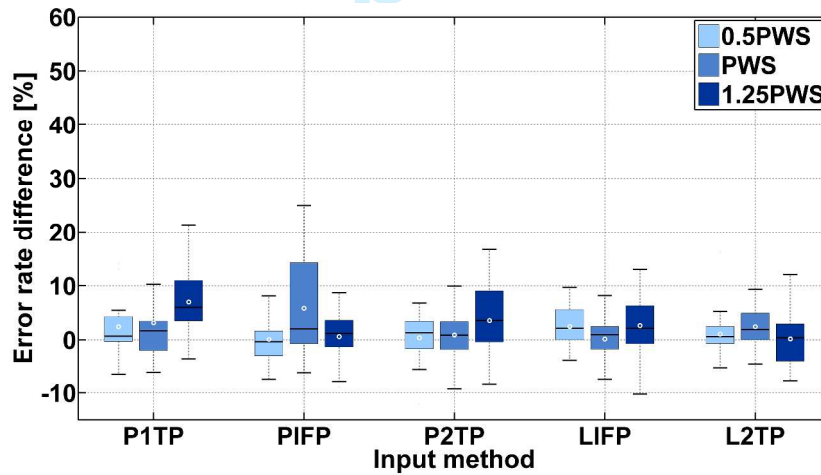


Figure 21 – Error rate box plots in the static condition for an offset model built using data collected on same subject but with different input techniques (circles represent means).



a)



b)

Figure 22– Error rate differences between offset model errors (white circles are means): a) difference between dynamic conditions and a model built for the static condition, and b) difference between dynamic conditions and a model built on 0.75PWS data.

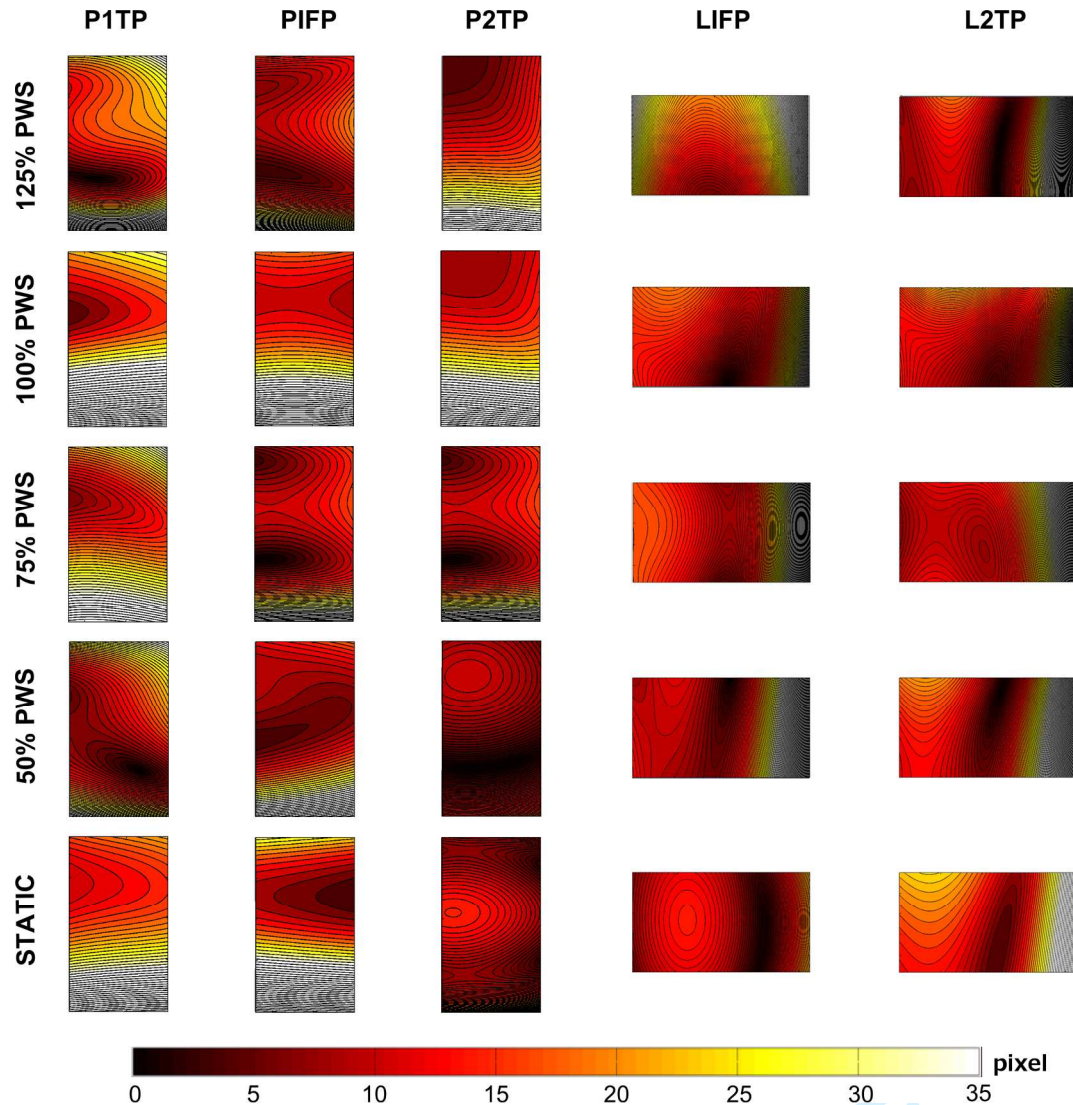


Figure 23– Comparison of contour plots for amplitudes of offset models for all measurement scenarios for one subject (subject No. 16). Note that, for comparison reasons, the scale has been limited to the $[0,35]$ interval.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Approach	Mean error rate	Standard deviation
No compensation	41.67%	7.69%
Compensation with corresponding speed model	25.43%	1.85%
Compensation with 0.75PWS speed model	27.41%	3.56%
Compensation with speed in statistical model – linear quadratic regression	30.79%	8.05%

Figure 24 – Comparison of mean error rate and standard deviation of regression model with speed variable with baseline and other compensation approaches