

Using Mobile Phones to Write in Air

Sandip Agrawal
Department of ECE
Duke University

Ionut Constandache
Department of CS
Duke University

Shravan Gaonkar
Department of CS
University of Illinois

Romit Roy Choudhury
Department of ECE
Duke University

Kevin Cave
Speech Pathology & Audiology
Duke Medical School

Frank DeRuyter
Speech Pathology & Audiology
Duke Medical School

ABSTRACT

The ability to note down small pieces of information, quickly and easily, can be useful. This paper proposes a system called *PhonePoint Pen* that uses the in-built accelerometer in mobile phones to recognize human handwriting. By holding the phone like a pen, an user should be able to write short messages or draw simple diagrams in the air. The acceleration due to hand gestures can be translated into geometric strokes, and recognized as characters. The geometric images and/or characters can then be sent to the user's email address for future reference. We implemented the *PhonePoint Pen* on the Nokia N95 platform, and evaluated it through real users. Results show that english characters can be identified with a median accuracy of 83%, if the users conform to a few constraints. Our ongoing and future work is focused on iteratively eliminating these constraints, with the prospect of developing a new input technology for personal devices.

1. INTRODUCTION

Imagine the following scenario. While driving to office, Leslie stops at a traffic light. As she mentally sifts through her tasks for the day, she remembers that she needs to call her friend, Jane, very soon. Since Leslie tends to forget her personal commitments, she decides to make a note of this task. Therefore, while keeping her gaze on the traffic lights, she reaches for her phone from the pocket, and by holding it like a pen, she writes "JANE" in the air. She also gestures a checkmark to email the written note to herself. She does not look at any of these hand-gestures she makes. Once in her office, she finds an email in her mailbox that reads "PhonePoint Pen – JANE". Leslie calls Jane, talks to her, and deletes the email. The figure below shows the output of writing Jane using PhonePoint Pen.



The above is a fictional scenario, however, represen-

tative of a niche in pervasive computing applications. In particular, we believe that there is a class of applications that will benefit from a technology that can quickly and effortlessly "note down" short pieces of information. Although existing technologies have made important advances to meet the needs, the quality of user-experience can perhaps be improved. We discuss some avenues of improvement, and motivate the potential of PhonePoint Pens.

Typing an SMS, while popular among the youth, has been unpopular among a moderate section of society. Studies report user dissatisfaction with mobile phone typing [1, 2, 3]. The major sources of discomfort arise from small key sizes, short inter-key spacings, and the need for multi-tapping in some phone keyboards. With increasingly smaller phones, keyboard sizes may decrease, exacerbating the problem of physical typing.

Even if future keyboards [4] improve the typing experience, some problems may still persist. While walking, or with one hand occupied, typing in information may be inconvenient. Using the mobile phone accelerometer to capture hand gestures, and carefully laying them out in text or image, can improve the user experience. The ability to write without having to look at the phone keypad may offer an added advantage.

One may argue that voice recorder applications on mobile phones may be an easy way to input short pieces of information. However, searching and editing voice-recorded content is difficult (unless processed through a separate speech-to-text software). Further, browsing through multiple voice messages is time-consuming. Writing in air, and converting them to typed text, may alleviate these problems.

Current approaches are largely ad hoc. People use whatever is quickly reachable, including pen-and-paper, sticky notes, one's own palm, etc. None of these

scale because they are not always handy, and more importantly, not always connected to the Internet. Thus, hurriedly noted information gets scattered, making information organization and retrieval hard.

This paper proposes to use the in-built accelerometer in modern mobile phones as a quick and ubiquitous way of capturing (short) written information. The problem definition bears similarity to known problems in gesture recognition. However, as we will see later, recognizing actual alphabets in air (using the phone processor, a noisy accelerometer, and no software training), raises a number of new challenges. For instance, as a part of writing the alphabet “A” on paper, one must write “/\” first, lift and reposition the pen on the paper, and then write the “—”. When writing in air, the phone cannot easily say which part of the hand-movement is intended to be the “re-positioning” of the pen. The problem is further complicated by the inherent noise in mobile phone accelerometers, the user’s involuntary wrist-rotation, and practical difficulties in deriving displacement from noisy acceleration. The PhonePoint Pen addresses majority of these challenges by treating the accelerometer readings as a digital signal, and successively refining it through simple numerical and signal processing algorithms. The simplicity is important to ensure that the operations can be performed on the phone processor. Once individual geometric movements have been tracked, their sequence of occurrence is matched against a decision tree (a simple grammar). The outcome of the matching operation yields the english character.

The PhonePoint Pen is not yet like a true pen-in-the-air, and requires the user to get used to a few soft constraints. Users that do not rotate their wrists while moving, do not write too fast, and write 15-inch sized capital letters, achieve an average accuracy of 83% with english alphabets. The geometric representation of the characters (shown as 2D images) are quite legible, except in 23% cases. The performance degrades as these constraints get violated, such as with new users. However, after writing around 20 characters, most users observed greater than 70% accuracy. Surveys and verbal feedback from random student users, as well as from speech-impaired patients in Duke Hospital, were positive. The absence of visual feedback while writing did not appear to be a concern at all. While more research is certainly necessary, our current findings give us confidence that the PhonePoint Pen could become a publicly-usable technology in the near future.

The conception of the ideas and a preliminary design of the PhonePoint Pen (P3) was published in MobiHeld 2009 [5], a workshop collocated with ACM Sigcomm.

Besides a mature design, full implementation, and a real-user based evaluation of the system, this paper adds a number of functional capabilities:

1. The workshop version was only capable of geometric representations of characters; this paper allows for actual character recognition leading to (editable/searchable) text.
2. The workshop version focused on identifying a single character, while this paper attempts to recognize transitions from one character to another, forming words.
3. The workshop version used a back-end server for processing; this paper is capable of on-phone analysis, and can display the results on the phone’s screen with 2-3 seconds latency.
4. Finally, this paper adds a few miscellaneous features such as character deletion, spaces between characters, digit recognition, and the ability to email with a gesture (a check-mark in the air).

The overall system is implemented on the platform of Nokia N95 phones using Python as the programming platform. The PhonePoint Pen demo video, and other related information, is available at:

<http://synrg.ee.duke.edu/media.htm>

The rest of the paper is organized as follows. Section 2 discusses the potential use-cases for the PhonePoint Pen. The core design challenges are discussed in Section 3, followed by the system design and algorithms in Section 4. The implementation and evaluation is presented in Section 5. Section 6 discusses some of the remaining research challenges, and the related work is visited in Section 7. Finally, the paper closes with a summary in Section 8.

2. USE CASES

We present some use-cases for the PhonePoint Pen (P3). These are not necessarily to express the utility of current system; instead they are a vision of the future. However, where applicable, we show examples from our current system.

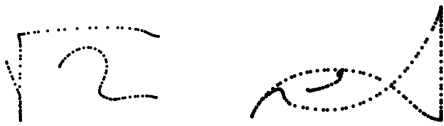
Assistive Communications for Impaired Patients: The Speech Pathology and Surgery division of Duke Medical School expressed keen interest in using the PhonePoint Pen as an assistive technology for impaired patients. Several patients suffer from inherent speech impairments, or experience similar conditions after surgeries. War veterans may have lost fingers, while others may lack finger-dexterity for typing on keypads. Yet, these patients are often capable of broad hand gestures, such

as in sign-languages. PhonePoint Pen can prove to be of assistance to such patients. It may permit a small degree of impromptu communication between a speech/hearing-impaired patient and someone who does not understand sign-languages. We have performed 15-minute experiments with 5 real patients at the Duke Hospital, and discussed the applicability of the system with surgeons, care-givers, and healthcare advisors. As discussed later, the PhonePoint Pen was met with high enthusiasm.

One-handed Use: People often have one of their hands occupied, perhaps because they are carrying a suitcase, a baby, or holding onto the rails in a moving train. P3 allows one-handed actions, approximating the experience of a pen/pencil. Besides, users can write words without looking at their hands. Even if the characters overlap in space, P3 can identify them individually and lay them out as regular text. The word below is written in air without looking at the hand movement.



Equations and Sketching: One of the students who volunteered to test the system suggested the possibility of quickly writing equations in the air. Equations are difficult to write with regular phone keyboards, and P3 may be convenient. Other use-cases involve sketching simple diagrams. While explaining an idea over the phone, a person could quickly draw a simple figure and send it to her caller. On similar lines, one may sketch driving directions, or draw out a desired food item (e.g., fish) in a foreign country's restaurant. At present, P3 is unable to draw figures with high reliability – the following fish and equation were drawn in 2 attempts each.



Mashing with Cameras: While attending a seminar, imagine the ability to take a picture of a particular slide, and write out a quick note in the air. The phone can superimpose the text on the slide, and email it to the user. Photos in a party or picnic, may similarly be captioned immediately after they are taken.

Emergency Operations and First Responders: The Department of Homeland Security (DoHS) has expressed interest in PhonePoint Pen as a quick input method that does not require the user's visual focus. Emergency scenarios are often unsuitable for typing, or even talking on the phone, because the observer may be engaged in looking at the events around her. The ability to observe and gesture at the same time is anticipated to have

value in these situations.

While the above use-cases are specific to phone-based applications, the basic idea of writing-in-air can be generalized to other devices and applications (e.g., a TV remote control could allow users to write "17" in the air to switch to channel 17). With this range of applications in mind, we visit the design of the PhonePoint Pen, next.

3. CORE CHALLENGES

Existing devices, such as the Wii remote [6], have the ability to identify hand gestures with reasonably good accuracy. However, most of these devices are more resourceful in terms of hardware and battery, and several offer visual cues to their users (perhaps through a monitor or a TV). Commodity mobile phones are embedded with low-cost sensors and constrained by limited battery power. This magnifies the inherent problems in air-writing. We begin with a discussion of these core research challenges, and our approaches alongside each of them. Thereafter, we assemble these blocks into a functional prototype.

(1) Filtering Hand Rotation without Gyroscope

Issue: Nokia N95 phones are equipped with a 3-axis accelerometer that detects acceleration in the X, Y, and Z directions. Figure 2(b) shows an example of raw accelerometer readings on each of the 3 axes. The accelerometers measure linear movement along each axis, but cannot detect rotation. Hence, if the human grip rotates while writing, the reference frame of acceleration gets affected. Existing devices like "Wii Motion Plus" and Airmouse employ a gyroscope to discriminate rotation [7, 8]. In the absence of gyroscopes in phones, preventing the rotational motion is a problem.

Proposed Approach: We begin with a brief functional explanation of the gyroscope. Consider the position of a gyroscope-enabled phone (GEP) at time $t = t_0$ in 2D space (shown in the left side of Figure 1). At this initial position, the figure shows that the GEP's axes are aligned with the earth's reference axes (i.e., gravity is exactly in the negative Y direction). The accelerometer reading at this position is $\langle I_x(t_0), I_y(t_0) - g \rangle$, where $I_x(t_0)$ and $I_y(t_0)$ are the instantaneous acceleration along the x and y axis at time t_0 respectively, and g is the gravity. Now, the phone may rotate at the same physical position at time t_1 also shown in Figure 1 (right). The phone now makes an angle θ with the earth's reference frame, and the accelerometer readings are $\langle I_x(t_1) - g\sin(\theta), I_y(t_1) - g\cos(\theta) \rangle$. However, it is possible that the phone moved along the XY plane in a manner that induced the same acceleration as caused by the rotation. This leads to

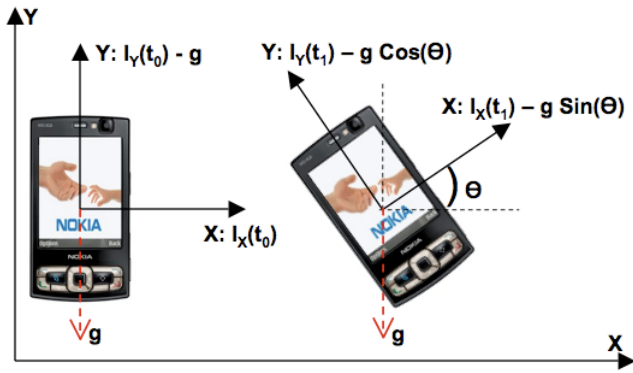


Figure 1: Earth's gravity projected on the XY axes; the axes are a function of the phone's orientation.

an ambiguity that gyroscopes and accelerometers can together resolve (using angular velocity detection in gyroscopes). However, based on the accelerometer readings alone, linear movements and rotation cannot be easily discriminated.

We have two plans to address this issue. (i) The simpler one is to pretend that one of the corners of the phone is the pen tip, and to hold it in a non-rotating grip (shown in Figure 2(a)). Some users also found it easier to hold it like a white-board eraser – this grip also reduced wrist-rotation. (ii) Alternatively, while writing an alphabet, users may briefly pause between two “strokes”. The pause is often natural because the user changes the direction of movement (from one stroke to another). For example, while writing an “A”, the pause after the “/” and before starting the “\” can be exploited. An accelerometer reading at this paused time-point can identify the components of gravity on each axis, and hence, the angular orientation θ can be determined. Knowing θ , the phone's subsequent movement can be derived. To be safe, the user may be explicitly requested to pause briefly between two strokes. Of course, we assume that the phone rotates only in between strokes and not within any given stroke (i.e., while writing each of “/”, “\”, or “-”). When writing too quickly, or for certain movement-impaired patients, this assumption gets violated, resulting in distortions in the geometric domain.

(2) Coping with Background Vibration (Noise)

Issue: Accelerometers are sensitive to small vibrations. Figure 2(b) reports acceleration readings as the user draws a rectangle using 4 strokes (around 350 units on the Z-axis is due to earth's gravity). A significant amount of jitter is caused by natural hand vibrations. Furthermore, the accelerometer itself has measurement errors. It is necessary to suppress this background vibration (noise), in order to extract jitter-free pen gestures.

Proposed Approach: To cope with vibrational noise, we apply two noise-reduction steps. First, we smooth the accelerometer readings by applying a moving average over the last n readings (in our current prototype, $n=7$). The results are presented in figure 2(c). Next, we mark as noise all acceleration values less than $0.5m/s^2$. We chose this threshold based on the average vibration caused when the phone was held stationary. All acceleration values marked as noise are set to 0. Figure 3(a) shows the combined effect of smoothing and vibration filtering.

(3) Computing Displacement of Phone

Issue: The phone's displacement translates to the size of the air-written character, as well as their relative positions (such as in equations, figures, etc.). The displacement δ is essentially computed as $\delta = \int (\int a dt) dt$, where a is the instantaneous acceleration. In other words, the algorithm first computes the velocity (the integration of acceleration), followed by the displacement (the integration of velocity). Noise in the acceleration readings will reflect on the velocity computation, and will get magnified in the computation of displacement. For instance, an erroneous short positive impulse in the accelerometer (i.e., acceleration becoming positive and then returning to zero), results in a positive velocity. Unless a negative impulse compensates for the positive impulse, the phone would continue to be in a state of constant velocity. When this velocity is integrated, the displacement error will be large.

Proposed Approach: In order to reduce the velocity-drift errors, we look at consecutive accelerometer readings marked as noise in the previous step. We reset the velocity to zero, if n ($=7$) consecutive readings have been filtered out as vibrational noise. This is because a continuous sequence of noise vibration alone is a good indicator of a pause, or a statically held phone; hence, it is an opportunity to suppress residual error. Figure 3(b) shows the effect of resetting the velocity. Even if small velocity drifts are still present, they have a tolerable impact on the displacement of the phone. As seen in figure 3(c) the amount of displacement and the shape drawn are represented reasonably well. The direction of movement is inferred from the signs of the acceleration along the X, Y, and Z axes.

(4) Differentiating an “A” from a Triangle

Issue: The imaginary slate in the air has no global reference frame for position. While writing character “A”, assume the writer has already drawn the “/” and “\”, and now lifts the pen to draw the “-”. Observe that the phone has no idea about the global position of

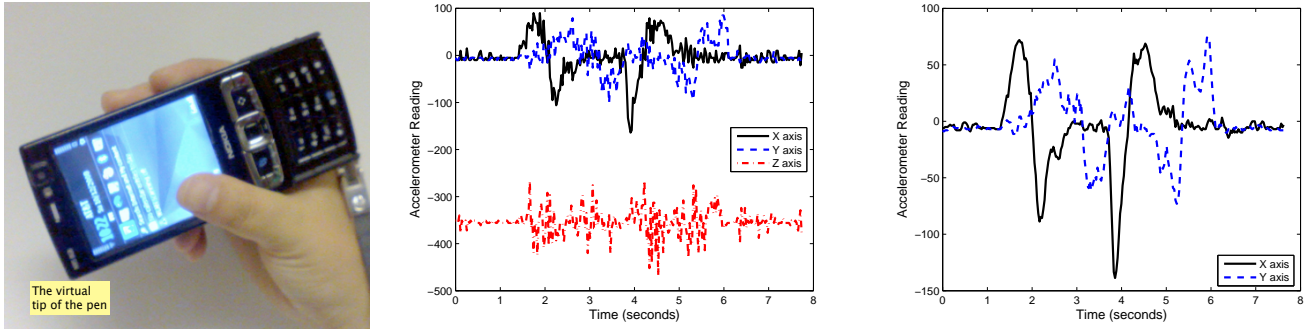


Figure 2: (a) Pretending the phone’s corner to be the pen-tip reduces rotation. (b) Raw accelerometer data while drawing a rectangle (note gravity on the Z axis). (c) Moving average computation.

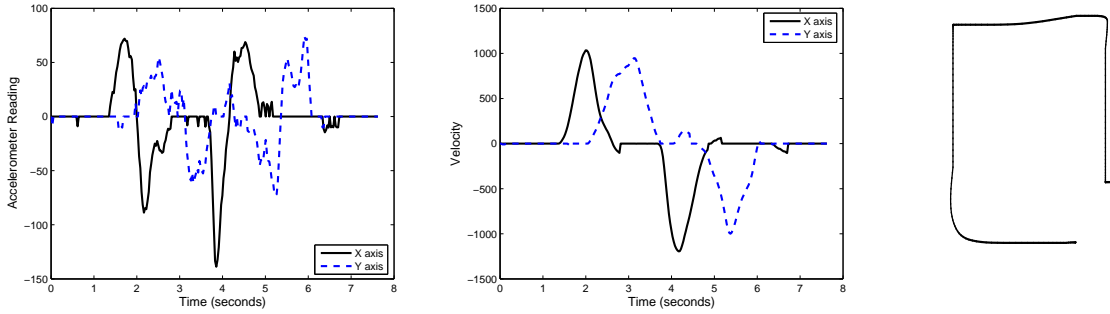


Figure 3: (a) Final processed acceleration readings (b) Computing velocity as an intermediate step towards measuring displacement. (c) The approximate rectangle as the final output.

“/”. Hence, upon drawing the “-”, the pen does not know whether it is meant to be added in the center (to indicate an “A”), or at the bottom (to indicate a triangle, Δ). This ambiguity underlies several other characters and shapes.

Proposed Approach: This is a difficult problem, and we plan to jointly exploit the accelerations along the X, Y, and Z axes. Consider the intent to write an “A”. Also assume that the user has just finished writing “/”. The pen is now at the bottom of the “\”. The user will now lift the pen and move it towards the up-left direction, so that it can write the “-”. The lifting of the pen happens in 3D space, and generates an identifiable impulse on the Z axis. When the acceleration in Z axis is above a certain threshold, we label that stroke as a “lifting of the pen”. This pen-lifting can be used as a trigger for the user going off the record. User movements in the XY plane are still monitored for pen repositioning, but do not get included in the final output. When the phone is in position to write “-”, a small pause can be used as an indication for going back on the record.

(5) Identifying Character Transitions

Even if pen-lifts are recognized, certain ambiguities remain. For instance, “B” and “13” may have the ex-

act same hand-movement, including the pen-lift. The user’s intention is difficult to recognize, making character distinction hard.

Proposed Approach: We rely on a combination of multiple heuristics to mark character separations. The simplest approach is to require the user to include a special gesture between characters, like a “dot” or a relatively longer pause. Thus, “13” should be written as “|”, “.”, “⌋”, “⌋”, while the “B” should be “|”, “⌋”, “⌋”. While this may be inconvenient, the phone pen also employs additional methods of delimiting characters. These methods rely on understanding what the user has written till now, and what the next “stroke” is likely to be. We will discuss this in detail in the next section, after we have discussed stroke-detection and a simple stroke-grammar to identify characters.

4. SYSTEM DESIGN AND ALGORITHMS

The above building blocks provide for a geometric representation of air-written characters. While the geometric version can be displayed or emailed as an image, conversion to text is likely to be more useful (for browsing and searching). This section develops the algorithmic components towards this goal.

4.1 Stroke Detection

Characters can be viewed as a sequence of strokes. The alphabet “A”, for instance, is composed of 3 strokes, namely “/”, “\”, and “—”. If the discrete strokes can be pulled out from the seemingly continuous movement of the hand, it would be possible to infer the characters. To this end, we have analyzed the english alphabets and characterized the basic set of strokes, as in Figure 4.

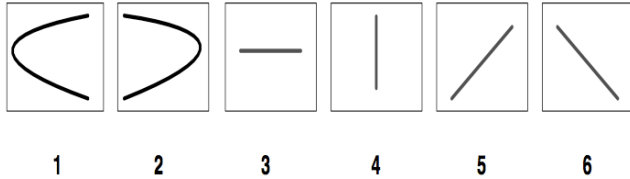


Figure 4: Basic strokes for English characters.

To identify the strokes, P3 computes a running variance of the accelerometer readings. When this variance falls below a threshold, P3 marks those regions as a pause of the hand. The pauses demarcate the human-strokes, allowing P3 to operate on each of them individually. For stroke-detection, our basic idea is to correlate the human-strokes against each of the basic strokes. This form of correlation is not new, and has been used as standard primitives in classification and matching techniques [9, 10, 11]. The correlation is performed over a varying window size of accelerometer readings. This is because the hand often rotates towards the end of the stroke, and the samples corresponding to the rotations should ideally be pruned out. In such cases, a shorter window size offers better correlation, in turn yielding the exact size of the human-stroke. Besides, even when the pauses are short between strokes, varying the window-size identifies the stroke boundaries. The intuition is that two consecutive strokes are typically different in the English alphabet, and thereby, correlating across the boundaries of the strokes reduces the correlation value. Performance results, reported later, show reasonable reliability in stroke detection. The natural question, then, pertains to combining the strokes into a character.

4.2 Character Recognition

The PhonePoint Pen observes the logical juxtaposition of strokes to deduce the character that the human is trying to write. For this, we adopt a stroke grammar for english alphabets and digits. Figure 5 shows a pruned down version of this grammar for visual clarity. The grammar is essentially a tree, and expresses the valid sequence of strokes to form an alphabet. Moreover, the grammar also helps in stroke-recognition because it provides P3 with an ability to anticipate the next stroke. For instance, observing strokes “| \ /” in succession, P3 can anticipate an “M” and expect the next stroke to be a

“|”. Thus, by correlating “|” to the stream of accelerometer readings (and ensuring a high correlation), the system can better identify the end-points of the next stroke. This helps in identifying the residual samples, which in turn helps in tracking the re-positioning of the hand in-between strokes. The benefits are cascading.

In certain cases, the user’s hand movement may be falsely classified as an incorrect stroke. A frequent example is “\” and “⊃”. Because the user’s hand has a natural rotational motion (pivoted at the elbow), moving diagonally for a “\” results in an arc, which then gets classified as “⊃”. Thus “N” cannot be classified correctly. To account for such possibilities, we have updated the grammar tree. For example, if stroke 6 (“|”) is followed by stroke 2 (“⊃”), we call it a “D” or “P”; however, if this is again followed by a stroke 6 (“|”), we infer an “N”. We observe that similar opportunities are numerous in the stroke grammar, adding to the robustness of the system. We do not include this updated grammar in the paper and only show a small example in Figure 6.

Interestingly, the stroke grammar presents a number of ambiguities. For instance, “O” and “S” are composed of the same strokes, namely, “⊃” and “⊂”. P3 resolves this by simply observing the direction of movement in the second stroke. If the hand is moving upwards, computed from the sign of the Y-axis acceleration, the alphabet is declared as an “O”, and the vice versa. “X” and “V” can also be solved similarly. Another ambiguity is between “D” and “P”. In this case, P3 computes the range of Y-axis movements and compares them. If the ranges are comparable (second stroke being greater than 0.75 of the first), the alphabet is deemed as a “D”, else a “P”. Finally, some kind of ambiguities are relatively harder. For instance, “X” and “Y” have the same strokes, and only differ in how the user repositions her pen. Since hand-repositioning does not have any preset movement, they are more prone to error. Thus, even though the “\” in “Y” is smaller than that of “X”, P3 often makes a mistake. We intend to further improve on the hand re-positioning algorithms in future (through some learning), and better resolve these issues. For this paper, we assume that the human user will be able to identify what was the intent (like a typo), or a spell checker would perform the right substitution.

4.3 Word Recognition

Recognizing the juxtaposition of characters, to recognize a word, adds to the ambiguity. For instance, “B” and “13” are identical in terms of the strokes used, and so are “H” and “IT”. Unless we find a signature to demarcate characters, the PhonePoint Pen will yield false positives. Towards this goal, we consider a combination

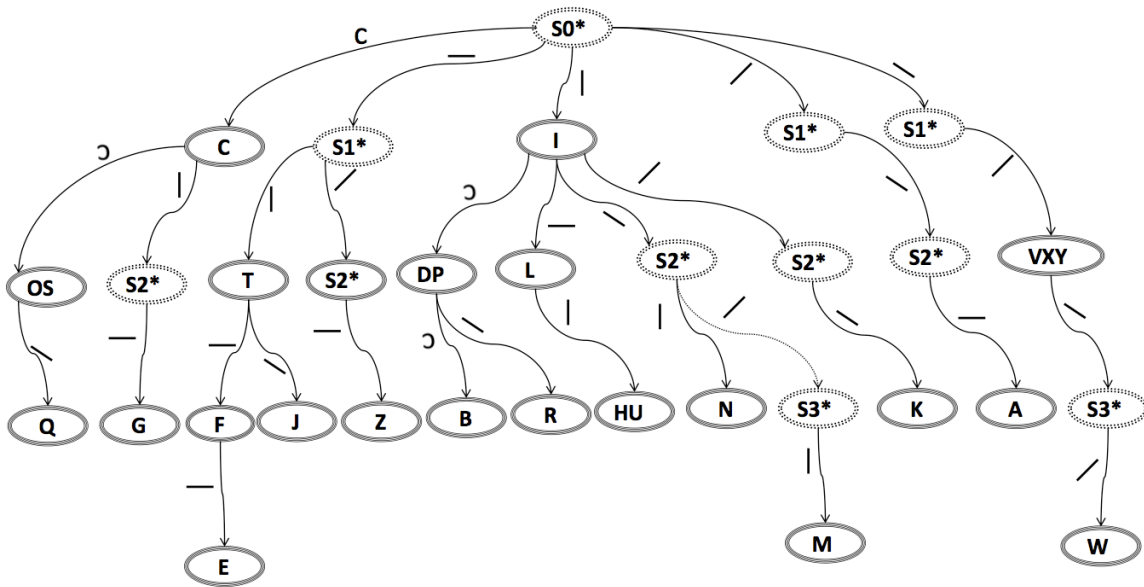


Figure 5: Developing the stroke set, and creating the grammar for character recognition.

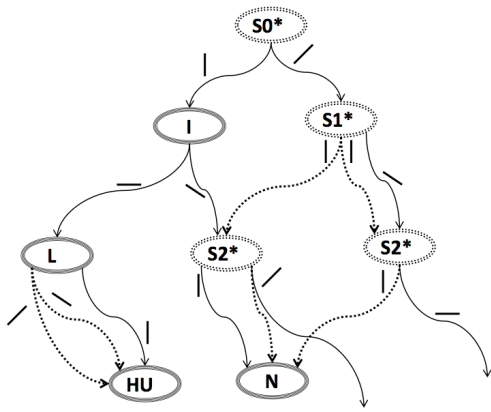


Figure 6: Incorporating tolerance into the grammar tree, translating it into a graph. Alphabet “N” can be reached via multiple paths. The ambiguity between H and U remains.

of multiple heuristics. None of these heuristics are individually reliable, but may be reasonable when used in conjunction. First, we make the observation that while transitioning from one alphabet to the next, users have a naturally longer pause (especially with upper case alphabets). Second, we observe that in some cases, if the hand moves in a *leftward horizontal* direction, it may be a hint about the start of a new character. This happens, for example, when a user has written across the imaginary plane in front of her, and moves back in space (towards left) to write more. Typically, since none of the strokes (except “G”) has a horizontal stroke from right to left, such a movement, *when observed*, can be a segregator of characters. Third, we ask the users to gesture a “dot” between characters whenever they

can remember. Thus, “13” should be written as “|. ㄩㄩ”, while the “B” should be “| ㄩㄩ”. Drawing such a dot presents a unique signature to delimit characters, but of course, slows down the user while writing. Thus, the user can only use it if she remembers or wishes to. If the delimiter is not used, the accuracy of character recognition may decrease.

We note that not all cases are like “B” and “13”. Even without the delimiter, some characters will present a natural separation by an application of the stroke grammar. In other words, given a sequence of strokes, the pen anticipates the next stroke to be from a specific set of strokes. If the next stroke is not in this anticipated set, then it implies the start of a new character. For example, given “|” and “—”, the phone can anticipate a “—” assuming that the user is trying to write an “F” or an “E”. However, if the next stroke is “/”, then the phone immediately infers that the prior alphabet was an intended “T”. Even if the delimiter is not present, such character transitions can be recognized to form words. Of course, we acknowledge that all the above approaches are heuristics. Nevertheless, they are very simple and amenable to on-phone processing. Also, when used in conjunction, we find a reasonable rate of false positives.

4.4 Control Gestures

To write a short phrase, the words need to be separated by spaces. In certain cases, the characters may need to be deleted. Further, the user should be able to email the written/drawn content to her email address. These are a few control operations that are vital to improve

the user’s experience. The PhonePoint Pen assigns a unique gesture to each of these, and recognizes them without difficulty. Specifically, the space is denoted by a long horizontal movement or two dots. The deletion is like using an eraser – the users shakes her hand at least four times briskly. To email, the user must draw a check mark in the air. With these functionalities in place, we present the implementation details of the PhonePoint Pen, followed by performance evaluation.

5. IMPLEMENTATION AND EVALUATION

We prototyped the PhonePoint Pen on a Nokia N95 mobile phone, equipped with a software accessible 3-axes accelerometer. The phone accelerometer was programmed to obtain 30-35 instantaneous acceleration readings per second. We developed a server side implementation first on MATLAB. Basic MATLAB libraries allowed us to implement some signal processing techniques (filtering) and simple statistical analysis. We ported this code on Python for on-phone processing. Some of the techniques were simplified (filtering operations were modified to running averages and subtractions). The results from Python and MATLAB are different in rare occasions.

The remainder of this section is organized in three parts: (1) evaluation methodology, (2) PhonePoint Pen evaluation with Duke students, and (3) user-study and experiences from patients with cognitive/motor impairments conducted at Duke University Hospital.

5.1 Evaluation Methodology

The P3 evaluation is centered around *character and word recognition accuracies*. We define accuracy as the fraction of successful recognitions, when a user writes individual alphabets/characters or words¹. Alongside P3’s character recognition accuracy, we also capture P3’s quality of geometric representation. For this, we display the geometric characters to a human, and ask her to recognize them. We call the correctly identified fraction, Human Character Recognition (HCR) accuracy. The HCR accuracy somewhat evaluates the quality of P3’s geometric output.

To compute word-recognition accuracy, we randomly generated words from a dictionary and requested test users to write them. Our main observation was that long words are more prone to errors because users have a higher chance to miss pauses between character strokes or forget to mark the character end. As we see later, word-recognition accuracy will degrade with word length. Nevertheless, since PhonePoint Pen outputs typed-text we can apply automatic dictionary

¹We use the words “alphabet” and “character” interchangeably.

correction and improve the final accuracy. For words, we will report the accuracy of P3, Dictionary-Assisted P3, and P3-with-HCR.

We conducted PhonePoint Pen tests mainly with Duke University students from computer science, and engineering. The test group comprised of 10 students in three categories: expert, trained, and novice. The expert set comprised of 2 users who extensively pilot-studied with P3 (wrote around 75 characters) before beginning the tests. Four users from the trained set wrote 26+ characters (each English alphabet approximately once), and four users from the novice set practiced less than 10 characters. We also performed a user study with a small population of 5 patients from the Duke Hospital – the primary purpose was to gain insights into P3’s applications into assistive technology. According to our IRB approval, the patients were allowed to record up to 8 characters. The patients had no previous experience with our prototype, and performed the experiments under the supervision of care-givers. Although P3 broadly failed in the tests, we will report the valuable experience and feedback we gained from neurosurgeons, physicians, and speech pathologists.

5.2 Performance Evaluation

We present the evaluation results from the Duke University student group. We start by showing a few written samples (both geometric and text), followed by a metric-based evaluation.

Air-Written Samples

Figure 7 shows the geometric version of alphabets M, O, B, I, S, Y and S, written by a trained user (each alphabet written separately). The acceleration readings for M and Y are presented alongside. P3 correctly converts the accelerations to alphabets, while the geometric versions are human legible (at least when the same person writes and reads the text). Figure 8 shows some examples of air-written words – these small words were written in one attempt. Evidently, the lack of a reference frame degrades the sense of proportion and relative placement of characters.

Even though characters are quite distorted, the stroke grammar proved tolerant to the distortions, and yielded correct results for all the words in Figure 8. However, mistakes occurred in several other cases, especially when the user wrote too quickly, or forgot to mark the character transitions with a longer pause or a “dot” in the air. The mistakes were naturally pronounced with longer words. For instance, the words PEACE, GAME, and MINUS were decoded as PEMCE (no pen-lift for the A resulting in an M), GAMF (P3 was unable to detect the quick “_’ for E) and MWUS (the lack of the

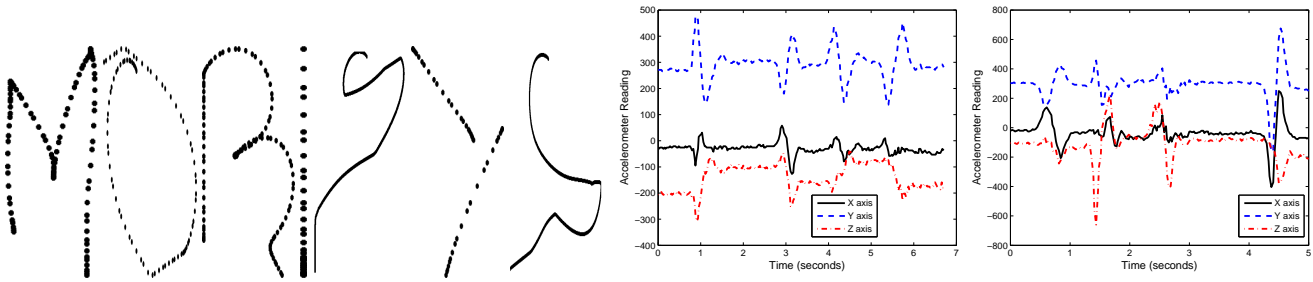


Figure 7: Alphabets M, O, B, I, S, Y, S, as outputs of the PhonePoint Pen. Although distorted, the characters are legible. Crossing the Y requires a pen lift and repositioning in the absence of a reference frame.

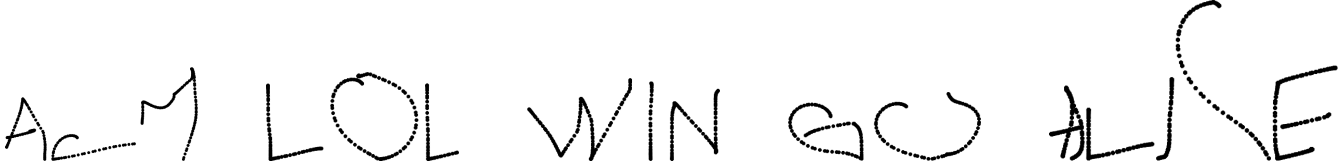


Figure 8: Words written in air: ACM, LOL, WIN, GO, and ALICE.

“dot” after I made I and N coalesce into a W). We will visit this aspect again in the next subsection.

Systematic Evaluation

Towards an upper bound on P3’s performance, we asked 2 expert users to draw each English alphabet 10 times. Human users were also asked to recognize each of the 260 alphabets. Figure 9 shows the P3 and HCR accuracies. The performances are comparable, except in some cases (D, G, Q, and Y), where P3’s character recognition was better than its geometric representation. Averaging over all alphabets, P3 achieved a 90.15% accuracy while HCR attained 77%.

To test the system with inexperienced users, we requested the trained and the novice groups to write all 26 alphabets. We requested only one trial, so that users do not gain experience over time. In total we tested 4 trained and 4 novice users. Figure 11 shows the average accuracy per user. In general, the trained writers performed better than the novice users, achieving an accuracy of 83.6% and 60.5% respectively. However, two of the novices performed well, matching and even surpassing the accuracy of a trained user. One of the novice users found it hard to write, and was able to achieve only 40% accuracy. His performance did not improve even with reasonable training.

To understand the problem, we requested the two weakest novice users to write with P3 on a surface (a table-top). The results improved appreciably (Figure 11), offering reason to believe that the users had difficulty staying on the (imaginary) vertical plane while writing in air. If one assumes that flat surfaces may often be accessible, P3 may still be useful to such users.

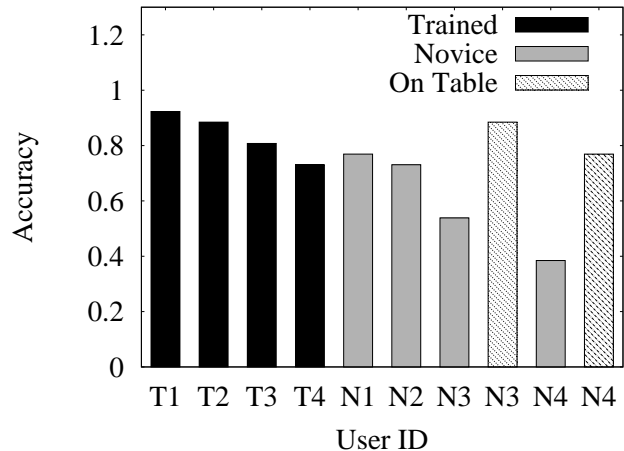


Figure 11: Avg. accuracy per (novice/trained) users.

Figure 10 shows the per-alphabet accuracy for trained and novice users. Some characters (like C, M and O) were decoded correctly for all the users. On the other hand, alphabets that require pen lifting (A, E, F, H, K, T, X and Y) proved to be consistently better for trained users. This is a result of more hands-on experience with the P3; trained users realized that lifting the pen needs to be performed explicitly, as opposed to moving the phone on the same plane. Character X seemed the most demanding for the novice users. For some, the alphabet was often mistaken for Y, while for the rest “going off the record” was unsuccessful. Alphabet H also registered a similar problem.

We wanted to understand the usability of the system with respect to alphabet sizes. Therefore, we asked users to choose the smallest size alphabet at which they

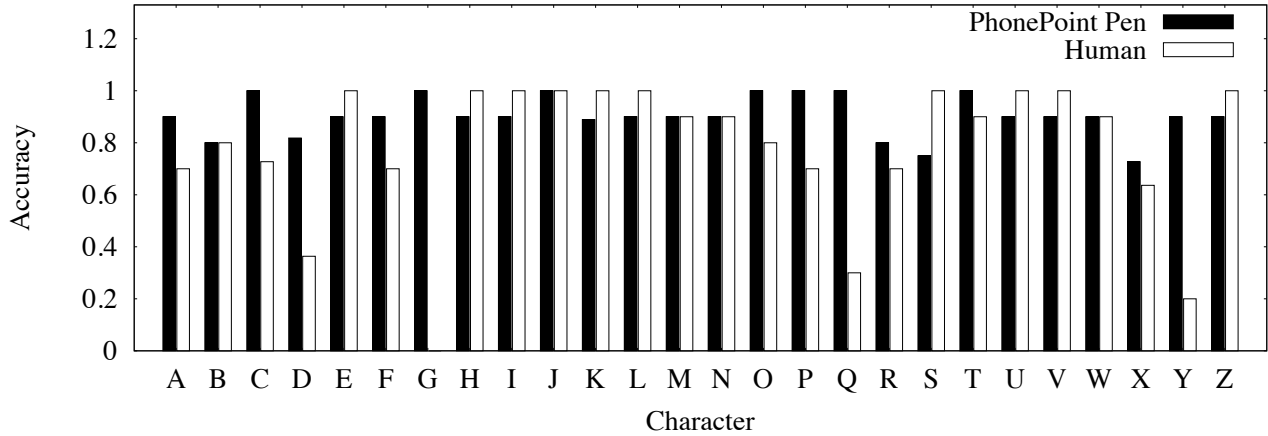


Figure 9: Per-character accuracy for expert users.

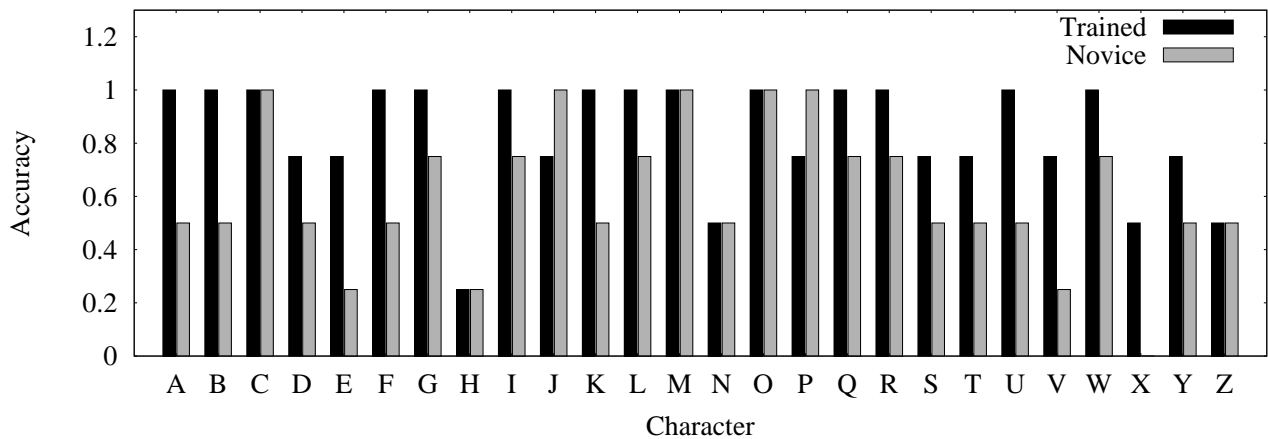


Figure 10: Per-character average accuracy for trained and novice users.

can air-write comfortably. Most users expressed ease while writing up to roughly 10-inch sized characters (the earlier experiments were with approximately 15-inched alphabets). Figure 12 shows the recognition accuracy for most of the users. Interestingly, one of the novice users (N3 in Figure 11) experienced a sharp accuracy improvement at these relatively smaller sizes. Through more tests, we realized that smaller strokes may result in less velocity drift, in turn curbing the accelerometer based errors. The phone also stays in the same vertical plane, often improving the user experience and ease of writing. Since accuracy degraded with less than 6-inch letter, we believe that 10-inch sizes can be currently supported with P3. As we discuss later, this may not be sufficient, especially for the medical applications we have in mind.

Recall that stroke-grammar exhibits inherent ambiguity. For example, D and P are written with the same strokes “|” and “⊃”. Similarly, characters {V, X, Y} and {O, S} use common sets of strokes. We disambiguate between

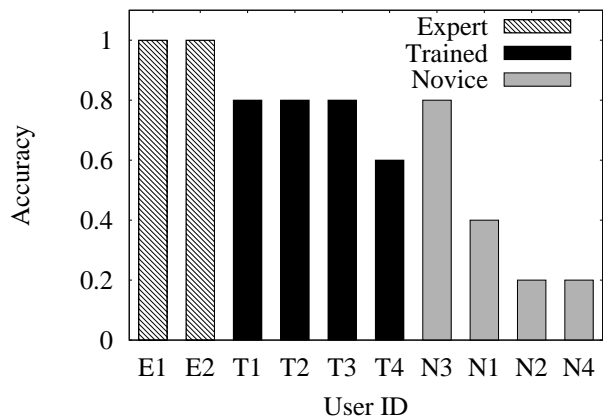


Figure 12: Accuracy with 10-inch characters.

these characters by computing the phone displacement during the off-the-record movement, as well as the direction of the strokes. Figure 13 presents the percentage of trials that were disambiguated correctly together with the percentage of incorrect outcomes. This graph

is computed over all the ambiguous characters recorded during our test trials. Results show that our repositioning system is reasonably robust. Mistakes consist mostly of characters being decoded incorrectly and not as a result of ambiguity. Ambiguous alphabets were decoded incorrectly in less than 12% of the test cases for characters D, P and S. Alphabet X showed a higher ambiguity with Y (around 33% false positives). This is mainly due to the lack of a reference frame to draw “\” and “/” of similar length.

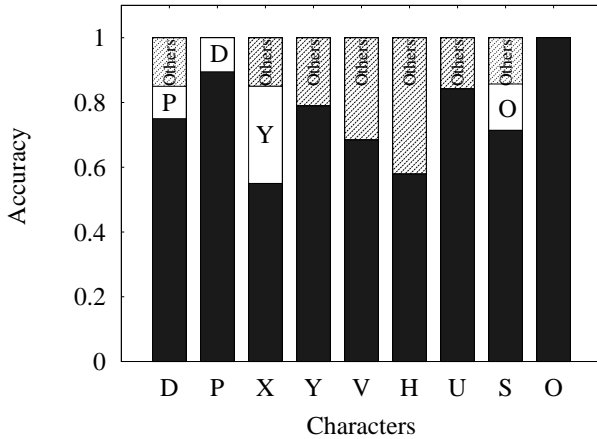


Figure 13: Disambiguation performance.

To understand the speed of air-writing, Figure 14 shows a CDF of the writing-duration for expert, trained, and novice users. On average, correctly decoded characters were written in around 4.5 seconds. P3 certainly requires users to write slow, so that adequate number of accelerometer samples can be collected for capturing the strokes. With improved support for accelerometers, this problem may get alleviated.

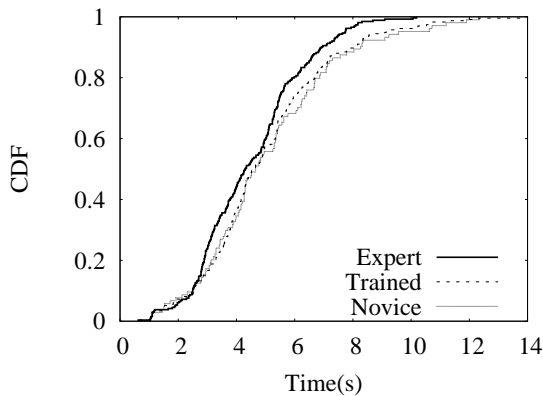


Figure 14: Recognition accuracy as a function of the time to write the character.

We asked expert users to also write full words in air. The words ranged from 2 to 5 characters and were chosen randomly from a dictionary. The users wrote

Table 1: Word recognition.

Word Length	PhonePoint Pen	Spell Check	Human
2	9/10	9/10	5/10
3	9/10	10/10	5/10
4	6/10	9/10	5/10
5	7/10	7/10	9/10

10 words for each length. Table 1 reports the P3’s recognition accuracy. Dictionary assistance improves the accuracy in some cases, while randomly selected humans were able to read the words with 40% error.

Energy Measurements

We ran experiments to compute the energy footprint of the Nokia N95 accelerometer. We sampled the accelerometer at the same rate as PhonePoint Pen on a fully charged Nokia N95 8GB phone. The phone exhibited an average battery-lifetime of 40 hours. We conclude that P3’s impact on battery lifetime is marginal, especially in light of its occasional usage pattern.

Informal User Feedback

We performed informal survey and usability tests with more than 40 Duke University students, and 5 faculty. In addition, the early prototypes of PhonePoint Pen featured in Slashdot and EnGadget and received more than 100 comments. A video-demo on Youtube received more than 97,800 views. From the real user surveys, and from the online forums, we received valuable feedback. In general, people found P3 exciting; one user commented that “*even though I may not use it frequently, I would like to have this as an iPhone App*”. Some users were critical, and commented that they may look “*silly*” while writing in the air. The large character-size was received with hesitation in the YouTube demo; however, currently we are able to write smaller. A few users raised concerns about privacy, while others responded by saying that when observing a user air-writing, “*decoding the alphabets backwards is difficult*”. Many users liked the prospect of using the phone as the TV remote, and writing channel-numbers in the air. Finally, one online comment was enthusiastic about being able to sign in air, perhaps when a FedEx package gets delivered to the door.

5.3 Experience with Duke Hospital patients

In collaboration with Physicians from the Surgery/Speech Pathology department of Duke Health Center, we carried out PhonePoint Pen tests with live patients suffering from various forms of cognitive disorders and motor impairments. Based on IRB approval, five patients were requested to write 8 randomly chosen alphabets. The patients were selected with varying degree of motor-impairments (e.g., a hydrocephalus lumbar drain trial

patient exhibited good cognition but weak motor skills; a patient from a car accident had a right side paralysis and spinal injury, but was able to write with his right hand; a 72-year old stroke patient had weakness on both limbs with sever tremors). The tests were carrier under the supervision of medical practitioners and care-givers, who first learnt to use P3 from us. We were not allowed to observe the patients, however, we interacted closely with the care-givers to receive feedback. Importantly, P3 generated interest in the hospital, drawing neurosurgeons, speech therapists, and care-givers to witness the tests and comment on the potential applications and additional requirements. The overall experience proved to be invaluable. We report the main lessons from it, here.

(1) The P3 design requires users to press a button before writing, and to re-press the button at the end. This proved to be a bad design choice – patients found writing very intuitive, but were unable to press the button. One patient pressed the buttons many times, another pressed the wrong one, and yet another found it hard to press. The unanimous feedback was to “replace the button with gestures” that will mark the start and end of air-writing.

(2) A neurosurgeon criticized that the P3 prototype required “shoulder, elbow, and wrist coordination”, a constraint that can hardly be satisfied with such classes of patients. His recommendation was to reduce the size of the letter so that it can be written with elbow movements alone. Moreover he suggested developing filters that would cancel the tremor in people’s hands, and thereby recognize the characters.

(3) One particular advantage of P3, even in light of specialized medical gadgets, is familiarity with cell phones. Physicians and care-givers emphasized the difficulties patients face in adopting new technological gadgets, particularly at the higher age group. Using the patient’s own phone to communicate “made a lot of sense”. They said, with a good degree of reliability, they envision a wide range of applications. Interestingly, care-givers showed enthusiasm event to the prospect of the patient changing her TV channel by writing in the air.

Table 2: Patient performance.

Patient ID	1	2	3	4	5
Accuracy	1/8	1/8	1/8	5/8	could not press button

6. LIMITATIONS

In its current form, the PhonePoint Pen is not ready for wide-scale use. Several limitations need to be resolved to bring the system to the standards of a product. This

section discusses the key limitations, and some ideas to resolve them in future.

Drawing and Writing Long Words still Primitive

The sketching capabilities are primitive with our current system. The primary problem stems from the inability to track the phone movement while the pen is being repositioned on the imaginary plane. Thus, although the actual written words/shapes are identified, their relative placements are often incorrect. The problem is greater as the figure involves multiple pen-lifts. Long words and sentences also have the problem. Our ongoing work is exploring the possibility of using the camera to determine hand-movements. By correlating the camera view over time, the hand motion may be better characterized. This may even help in detecting hand-rotation.

Writing while Moving

If a person writes in the air while moving, the accelerometer readings will reflect that movement. The prototype described in this paper does not attempt to filter out these background movements. We believe that some of these movements, such as walking or jogging, exhibit a rhythmic pattern that can be removed. However, others are less recognizable. A person may rotate her body while writing, or a car may bounce while passing through a pothole – the induced noise may be difficult to eliminate. We plan to address this problem in our future work.

Cursive Handwriting

The current prototype supports uppercase alphabets only. Lower case alphabets may be possible to support with a different set of strokes and a modified stroke-tree. However, supporting cursive writing is significantly more difficult. The problems of stroke-detection and character-recognition are exacerbated due to the continuous movement of the hand. One approach would be to apply pattern recognition algorithms on the entire window of (noise-suppressed) accelerometer readings. However, such a scheme will not only require complex computation, but may also need a reasonable degree of training. We have traded off these functionalities for simplicity.

Smaller Alphabets and Quicker Writing

With the current prototype, alphabets need to be around 10 inches in size. Moreover, users may need to write them somewhat slowly (no quicker than 1 alphabet per second) so that the accelerometer readings can be captured, and the pauses between strokes/characters identified. Ideally, the system should permit the user to write small and quick. This would require more frequent sampling of the accelerometer. We expect future

phones to support such sampling rates.

Realtime Character Display

Ideally, the strokes could be displayed in real time on the phone display. This might offer a visual feedback to the user that could be useful while writing several words. Perhaps this could be useful for deleting characters as well. Our current system does not support this realtime capability. However, this may be feasible with more powerful processors in future phones.

Comment on Survey and Testing Population

Students who have tested P3, and those that have participated in our survey, are mostly students from the Computer Science and Engineering departments. This students are likely to have an understanding of accelerometers, and could have adapted to P3's behavior. In that sense, our accuracy results could be biased; a lay person may not necessarily achieve the reported accuracy.

7. RELATED WORK

Gesture recognition has been widely studied in the past [10, 12]. Majority of the work has focused on tracking the movement of the hand, either through specialized wearable sensors, or through vision based techniques. For example, variants of "smart glove" based systems [13, 14, 15] have recognized hand and finger movements with impressive granularity. While useful for specific applications, such as in augmented reality, the need to wear these gloves preclude spontaneous use-cases. Moreover, the gloves are equipped with a number of sensors, and the gestures are not converted to alphabets. Recognizing alphabets using noisy mobile phone accelerometers present a different set of challenges.

Cameras have also been used to track an object's 3D movements in the air [6]. Microsoft Research recently demonstrated a project, titled "write in air" [16], that uses an apple in front of a camera to air-write alphabets. Computer vision based algorithms can precisely discern the movement of the apple (or any other object) to create both geometric and textual representations of the alphabets. Noisy accelerometers and limited processing in mobile phones lack several advantages present in computer-connected cameras. Moreover, the system does not recognize words, avoiding the problems of transition between characters. Signal processing based techniques are useful, but not sufficient [11, 17, 18]

Authors in [9] have used mobile phone accelerometers to perform gesture recognition [12]. They show the possibility to detect gestures with an impressive

99% accuracy, with as few as one training. While the control gestures in PhonePoint Pen are inspired by this work, we emphasize that character recognition entails an additional set of problems. Specifically, gestures are significantly tolerant to error – as long as the errors repeat across all gestures, the gesture can be identified. PhonePoint Pen on the other hand, must track the phone's movement in all the 3 axes without any training. Moreover, issues like pen-lifts, character transition, stroke-grammar, rotation avoidance, and character disambiguation are unique to character/digit recognition.

A popular device capable of tracking hand movement is the Wii remote (or "Wiimote") used by the Nintendo Wii console [6]. The Wiimote uses a 3-axes accelerometer to infer forward and backward rapid movements. In addition, optical sensors aid in positioning, accurate pointing, and rotation of the device relative to the ground. The optical sensor is embedded on the Wiimote and relies on a fixed reference (and a sensor bar) centered on top of the gameplay screen. The "Wiimote" can be augmented with the "Wii Motion Plus", a pluggable device containing an integrated gyroscope. Using this feature rotational motion is captured. These three sensors – the accelerometer, the gyroscope, and the optical sensor – can reproduce motions similar to real arm-motion. The Nokia N95 consists of only a (low-cost) accelerometer, and limited processing capabilities, in comparison to the Wii. Developing the pen on this platform entails a variety of new challenges.

The Logitech Air Mouse [8] targets people who use computers as multimedia devices. The Air Mouse provides mouse-like functionalities but the device can be held in air similar to a remote control. Accelerometers and gyroscopes together allow for linear and rotational motion of the pointer on the screen. Unlike the Air Mouse, the proposed phone-based pen does not have a screen on which one may see the pen movement in real time. The absence of visual cues makes positioning of the pen a difficult problem.

A series of applications for the Nokia N95 use the built-in accelerometer. The NiiMe [19] project transformed the N95 phone into a bluetooth PC mouse. The PyAcceleREMOTER [20] project developed a remote control for the Linux media player MPlayer. By tilting the phone, the play, stop, volume, fast-forward, and rewind functions of the player are controlled. Inclinator provides car inclination values while Level Tool allows measurement of the inclination of different surfaces by placing the phone on that surface. Lastly many video games for the N95 phone make use of the accelerometer, e.g., to guide a ball through a maze. Being able to write in the air, we believe, is a more

challenging problem than the ones in existing systems.

Livescribe Smartpen [21] is a pen-like device capable of tracking a person's writing. The device requires a special finely dotted paper to monitor movement of the pen. The pen recognizes the alphabets and numbers, and hence, can be downloaded to a PC. However, the dotted paper may not be always accessible, making ubiquitous note-taking difficult. Tablet PCs also suffer from this problem of ubiquitous accessibility.

8. CONCLUSIONS

This paper attempts to exploit the accelerometer in mobile phones to develop a new input technology. While today's users are mostly used to keyboards and touch-screens, we propose to mimic a pen. By holding the phone like a pen, the user should be able to write short messages in the air. The phone identifies the hand gestures as one of multiple strokes, compares the sequence of strokes against a grammar, and recognizes the air-written alphabets. The entire process requires no training, and can run entirely on the phone's processor. The written message is displayed on the phone-screen, and may also be emailed to the user if she desires. We believe that in the age of microblogging and tweeting, such input devices may be effective to note down information on the fly. Moreover, the pen may offer an intuitive user-experience, adding to the menu of current input methods. We call this system PhonePoint Pen, and demonstrate its feasibility through a Nokia N95 prototype and real user studies. The performance results are promising, while the user feedback (from the student community) is highly positive.

9. REFERENCES

- [1] Christine Soriano, Gitesh K. Raikundalia, and Jakob Szajman, "A usability study of short message service on middle-aged users," in *OZCHI '05: Proceedings of the 19th conference of the computer-human interaction special interest group (CHISIG) of Australia*, Narrabundah, Australia, 2005, pp. 1–4.
- [2] Vimala Balakrishnan and Paul H.P. Yeow, "A study of the effect of thumb sizes on mobile phone texting satisfaction," in *Journal of Usability Studies, Volume 3, Issue 3*, 2008, pp. 118–128.
- [3] Vimala Balakrishnan and Paul H.P. Yeow, "Sms usage satisfaction: Influences of hand anthropometry and gender," in *Human IT 9.2*, 2007, pp. 52–75.
- [4] Nokia, "Virtual keyboard," <http://www.unwiredview.com/wp-content/uploads/2008/01/nokia-virtual-keyboard-patent.pdf>.
- [5] Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, and Romit Roy Choudhury, "Phonepoint pen: using mobile phones to write in air," in *MobiHeld Workshop on Networking, systems, and applications for mobile handhelds*, 2009.
- [6] Nintendo, "Wii console," <http://www.nintendo.com/wii>.
- [7] Nintendo, "Wii motion plus," <http://www.nintendo.com/whatsnew>.
- [8] Logitech, "Air mouse," <http://www.logitech.com>.
- [9] J. Liu, Z. Wang, and L. Zhong, "uWave: Accelerometer-based Personalized Gesture Recognition and Its Applications," Mar. 2009.
- [10] Thomas Baudel and Michel Beaudouin-Lafon, "Charade: remote control of objects using free-hand gestures," *Commun. ACM*, 1993.
- [11] "AiLive LiveMove pro," AiLive Inc, <http://www.ailive.net/liveMovePro.html>.
- [12] Xiang Cao and Ravin Balakrishnan, "Visionwand: interaction techniques for large displays using a passive wand tracked in 3d," *ACM Trans. Graph.*, 2004.
- [13] John Kangchun Perng, Brian Fisher, Seth Hollar, and Kristofer S. J. Pister, "Acceleration sensing glove (asg)," in *In ISWC. 1999*, IEEE.
- [14] I.J. Jang and W.B. Park, "Signal processing of the accelerometer for gesture awareness on handheld devices," in *IEEE Workshop on Robot and Human Interactive Communication*, 2003.
- [15] Paul Keir, John Payne, Jocelyn Elgoyhen, Martyn Horner, Martin Naef, and Paul Anderson, "Gesture-recognition with non-referenced tracking," in *3DUI '06: Proceedings of the 3D User Interfaces*, 2006.
- [16] "Microsoft Research," Write in The Air, TechFest 2009, <http://www.youtube.com/watch?v=WmiGtt0v9CE>.
- [17] Juha Kela, Panu Korpipää, Jani Mäntyjärvi, Sanna Kallio, Giuseppe Savino, Luca Jozzo, and Di Marca, "Accelerometer-based gesture control for a design environment," *Personal Ubiquitous Comput.*, 2006.
- [18] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li, "Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes," in *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, 2007.
- [19] Asier Arranz, "Niime," <http://www.niime.com/>.
- [20] "Pyacceleremoter project," <http://serk01.wordpress.com/pyacceleremoter-for-s60/>.
- [21] LiveScribe, "Smartpen," <http://www.livescribe.com/>.