GyroPen: Gyroscopes for Pen-Input With Mobile Phones

Thomas Deselaers, Daniel Keysers, Jan Hosang, and Henry A. Rowley

Abstract—We present GyroPen, a method to reconstruct the motion path for pen-like interaction from standard built-in sensors in modern smartphones. The key idea is to reconstruct a representation of the trajectory of the phone's corner that is touching a writing or drawing surface from the measurements obtained from the phone's gyroscopes and accelerometers. We propose to directly use the angular trajectory for this reconstruction, which removes the necessity for accurate absolute 3-D position estimation, a task that can be difficult using low-cost accelerometers. We connect GyroPen to a handwriting recognition system and perform two proof-of-concept experiments to demonstrate that the reconstruction accuracy of GyroPen is accurate enough to be a promising approach to text entry. In a first experiment, the average novice participant ($n = 10$) was able to write the first word only 37 s after the starting to use GyroPen for the first time. In a second experiment, experienced users ($n = 2$) were able to write at the speed of 3–4 s for one English word and with a character error rate of 18%.

Index Terms—Computer and information processing, pattern recognition, gesture recognition, handwriting recognition, text recognition.

I. INTRODUCTION

Small (touch)screen areas on mobile devices often limit their capabilities for user interaction. In this paper, we present GyroPen, a method that brings an experience similar to "drawing with a pen" to mobile devices without using a stylus and without the space restrictions typically imposed by a small form factor. The user can hold the mobile phone like a pen and "write" on any surface (see Fig. 1). The trajectory of the phone's "writing corner" is reconstructed from the phone's sensors: its gyroscopes and accelerometers. Because the proposed method does not require to use a touchscreen, it is appealing for small form factor devices or for devices lacking a screen.

A promising application of GyroPen's capabilities is as a method for text entry. Text entry on mobile devices is a topic of interest as mobile devices are getting more popular: Entering text on mobile devices is still considered inconvenient by many, although a large variety of input methods have been proposed since the first mobile phones. Twelve-button phones with predictive text-entry methods (see, e.g., [1]) have mostly been replaced by touchscreen devices with soft keyboards. These devices, however, suffer from the "fat finger problem," i.e., the fact that a human finger is thicker than a typical key on the virtual keyboard [2], and thus, methods for automatic prediction and correction were added, e.g., in the form of gesture-based
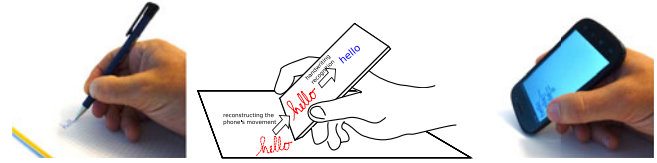
Fig. 1. GyroPen provides a similar experience for *writing with a phone* as *writing with a pen*.

systems for keyboard entry, e.g., SHARK [3], ShapeWriter [4], and SwiftKey [5]. One of the alternative means of text entry is handwriting, which works naturally with GyroPen input.

II. RELATED WORK

The idea to use inertial sensors for interaction with mobile devices and for text entry in particular has been discussed in the literature before. One of the first reports is [6] in which a pen device with two accelerometers constructed for the task is combined with a specifically developed writer-dependent HMM recognizer to recognize one of a fixed set of seven words. Zhang *et al.* [7] discuss a similar setup to distinguish ten numerals.

The TiltText approach [8] uses movements of the cell phone to disambiguate among multiple text candidates when entering text on a 12-button keypad using a low-cost tilt sensor. Similarly, Shrimp [9] aims at easing text entry, but instead of relying on a tilt sensor, it uses the phone's camera to estimate its motion using computer vision. Shrimp improves on the text prediction to allow entry of out-of-vocabulary words more easily.

WalkType [10] helps users to type on a touchscreen while walking. It uses accelerometer data to predict and account for misplaced touch events and corrects the user input with a language model and a model of common typing errors.

Gómez *et al.* [11] propose a Dasher-like [12] system for text entry controlled by accelerometers.

Approaches that are closely related to ours are PhonePoint-Pen [13] and Airwriting [14]. These systems allow a user to write in the air similar to the process of writing on a blackboard, which requires the user to make fairly large writing gestures in contrast with the small movements required in our approach. Both PhonePoint-Pen and Airwriting use dedicated handwriting recognition systems built for motion-based input. In contrast, our system uses an "off-the-shelf" handwriting recognition system leveraging the efforts of the handwriting recognition community over several decades without further modifications. This is possible because the reconstructed writing paths of our system match how users write with a pen on paper. Recognition can easily be extended to more languages, scripts, and symbols by replacing the recognition backend.

PhonePoint-Pen uses only the accelerometers of a mobile phone, and the recognizer is comprised of a manually

engineered decision tree and language-model-based correction to disambiguate likely character confusions. PhonePoint-Pen does not distinguish letter case but supports three editing gestures and numerals. Average character writing times between 3.0 and 4.3 s are reported ($n = 10$) and a character error rate (CER) of 8.1% for trained users ($n = 4$) and single character input (with the note that "recognition degrades with increasing word-length"). Agrawal *et al.* [13] also include a more in-depth discussion of further related work in this area.

In Airwriting, a specialized motion-sensing device with gyroscopes and accelerometers is attached to the user's wrist. An HMM-based system is trained on raw sensor data, i.e., the motion path is not reconstructed. A user-independent word error rate of 11% for a fixed vocabulary of 8K words is reported ($n = 9$) [14] for this specific data. The task-dependent training of the Airwriting system requires a sufficient amount of annotated motion data. In contrast, our approach can directly benefit from improvements in normal handwriting systems and thus is easy to extend to new languages.

Miyagawa *et al.* [15] present a basic system that uses both accelerometers and gyroscopes attached to a pen to reconstruct short writing paths in 3-D, but do not evaluate the results quantitatively or specifically for text entry.

An approach similar to ours is the Magic Wand system [16], [17]. The device in [16] uses both accelerometers and gyroscopes and is specifically constructed to allow 3-D trajectory reconstruction. In contrast with our approach, the gyroscopes are mainly used to compensate for gravity and attitude (compare Section IV), not directly for the estimation of the writing path (our approach to do so is described in Section V). The writing plane is estimated from the 3-D motion of the device, and the trajectory is projected onto that plane. In contrast, GyroPen supports the use of angular movement only, as it occurs when writing with a pen while resting the palm on a surface. In [16], no recognition is performed, but $26 + 10$ Graffiti-style symbols can be distinguished by a human observer from the reconstructed paths. Cho *et al.* [17] describe an extension in form factor and a direct integration into a specifically designed gesture recognition system. On a set of 13 gestures designed for easy discrimination by the system, error rates of below 1% are reported ($n = 15$).

The ImuPen [18] similarly estimates motion by double integration of acceleration information with elaborate additional signal processing stages. For writing on a surface, an error rate of 9.6% for a ten-class digit recognition problem is reported in comparison with 2.8% when a digitizer tablet is used. This error rate dropped to 5.4% for writing on an unrestricted surface.

Regarding mobile handwriting recognition in general, it was already available on the Apple Newton [19] but did not become popular until the simplified writing system Graffiti on Palm Pilots [20] and its extensions [21], [22] were available. In modern smart phones and tablets, several handwriting input methods exist [23]–[25].

III. SENSORS

Modern cellphones typically contain three types of motion and orientation sensors: accelerometers, gyroscopes, and a magnetometer. The *accelerometer* measures the acceleration of the phone in three dimensions (or, equivalently, the forces acting on the phone). One main component of the measured acceleration is caused by gravitational forces. Additionally, the accelerometer measures any force applied to the phone that results in acceleration. Accelerometers have been built into smart phones for several years, e.g., the first Android phone and the original iPhone already had accelerometers. Most accelerometers built into current devices use the piezoelectric effect. The measurements of an accelerometer are given in $\mathrm{m/s}^2$.

The *gyroscope* measures rotation about its own three axes. Gyroscopes only became popular in smart phones around the year 2010, e.g., the Nexus S and the iPhone 4 have built-in gyroscopes. Most gyroscopes built into current smart phones use an oscillator to measure the Coriolis effect. The gyroscope measures angular velocity in rad/s.

The *magnetometer* can be considered a 3-D compass that measures the direction of the magnetic field at the current location. Magnetometers have also been built into mobile phones since the first generation of smart phones. Most magnetometers in cell phones directly measure the magnetic field using the Hall effect. Unfortunately (for the use in low-latency applications), magnetometers are often slow. The measurements of a magnetometer are given in μT.

In our approaches described below, we generally poll the measurements of all sensors with a frequency of about 150 Hz, which we found to be sufficient to capture writing movements while balancing the tradeoff for low energy consumption.

IV. ACCELEROMETER-BASED APPROACH

The first approach to reconstructing the motion of a mobile phone that comes to mind uses the accelerometers and (double) integrates over their measurements.

The accelerometer measures the acceleration of the phone in three dimensions $a = (a_x, a_y, a_z)$. When the phone is held still with respect to the user in static conditions, the accelerometers only measure gravity $g = (g_x, g_y, g_z)$, which is typically $|g| = 9.81\,\mathrm{m/s}^2$. Once the gravity is estimated, the vector a can be rotated into a vector $a' = (a'_x, a'_y, a'_z)$, where the z-axis points in the opposite direction of the gravity, and thus, the directions of a_x and a_y are parallel to the desired horizontal writing plane. Thus, discarding the z component after rotation effectively eliminates g, which is usually the strongest force in the accelerometer measurements.

With these rotated observations, we start from a hypothetical starting point (x_0, y_0) and compute the position (x_N, y_N) of the phone after N sensor readings as

$$(x_N, y_N) = (x_0, y_0) + \sum_{n=1}^{N} \Delta t_n (v_{nx}, v_{ny}) \qquad (1)$$

with

$$(v_{nx}, v_{ny}) = (0,0) + \sum_{i=1}^{n} \Delta t_i (a'_{ix}, a'_{iy}) \qquad (2)$$

where Δt_n is the time that passed between measurements $n-1$ and n. While this approach is appealing in theory, it turns out that, in practice, the double integral (or sum) is very sensitive to noise, and the sensors built into current phones are often too noisy to give sufficiently precise readings for accurate
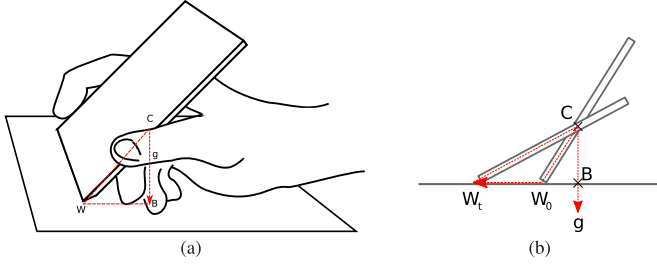
Fig. 2. Writing with the phone. (a) Hand is holding the phone like a pen. To determine which words the user writes, we measure the rotation around point C and estimate at which point W the phone touches the writing surface. (b) Simplified to two dimensions, the writing point W moves as the phone rotates and translates.
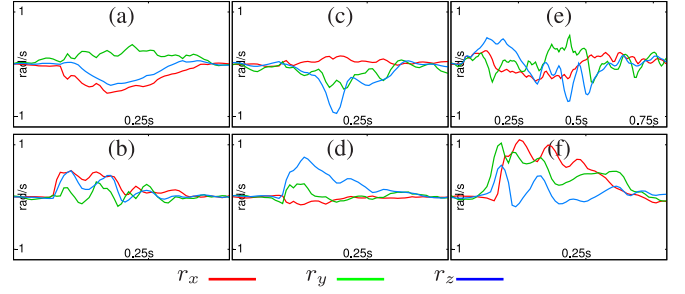

Fig. 3. Measurements from the gyroscope while performing different writing strokes in $\frac{rad}{s}$ over time. (a) "Left to right." (b) "Right to left." (c) "Bottom up." (d) "Top down." (e) "Circle." (f) "Pen up."

estimation of a writing path. Airwriting [14] avoids this problem by 1) not reconstructing the writing path explicitly but training the handwriting recognizer on sensor data directly and 2) using a high-accuracy sensor unit, which is more expensive than the manufacturing price of current smartphones.

V. GYROPEN: GYROSCOPE-BASED APPROACH

GyroPen allows for writing with the phone like a user normally would write with a pen.[1] Most movements of a pen while writing are performed without actually moving the entire hand. Instead, the writing motion of the pen tip results from moving the pen using mostly finger and wrist movements. This motion consists in large part of rotations. Then, the entire hand is moved by a few centimeters, e.g., to start a new word or line.

GyroPen uses the phone's gyroscopes to estimate the motion of the writing point. The gyroscopes directly measure angular *velocity* about three axes $r = (r_x, r_y, r_z)$, and thus, a single integration will be sufficient to determine how far the phone was rotated. Starting from an initial attitude of the phone $o_0 = (o_{0x}, o_{0y}, o_{0z})$, which is initialized from a gravity estimate while the phone is held still, we update the attitude estimate by rotating it according to the gyroscope measurements.

The motion of the phone's writing point W is approximated by assuming that the overall motion consists of a rotation around a fixed virtual center point C and a translation of the pen along the line C-W such that W continuously touches the surface (see Fig. 2). For different users, the point C may be at different locations depending on how they hold the phone. Considering the 2-D case in Fig. 2(b), the user rotates the phone such that the writing point W moves from W_0 to W_t and continuously shifts the phone such that W touches the writing surface. Under these assumptions, the phone's attitude (as determined from the gyroscopes) is sufficient to determine the writing motion trajectory from W_0 to W_t. During writing, the C's position may vary, but for smooth motion there is no large distortions.

g is the direction of gravity, and we assume the distance between points B and C (the distance of C from the writing surface) to be approximately 5 cm. Note that the exact distance between these points does not influence the nature of the recorded handwriting but only changes a fixed scaling factor that

is applied to the reconstructed trajectory. The handwriting system may later apply an extra size normalization (we do not pass on any writing area sizes), and thus, the recognition accuracy is invariant with respect to the writing size (Section V-F).

Note that this approach can be extended to larger movements typical for say writing with a laser pointer (see Section V-E). It also works well when restricting the writing to a very small area: Restricting the writing area to a band of height 5 mm was found to be unproblematic for recognition.

A. Initialization

For initialization, we determine the initial attitude o_0 of the phone, while it is not moving using the accelerometer. We estimate gravity g from N accelerometer measurements, while the phone is not moving, which gives us a reliable estimate of our initial attitude o_0 (up to a rotation about the vertical axis). During the implementation of the system, we experimented with different values for N and found that $N = 10$ is a suitable choice. This means that the initialization only takes a fraction of a second, and it is a good compromise between a stable estimate for gravity and waiting for the initialization.

B. Tracking Movements

To update the phone's attitude as it rotates, we use the gyroscope measurements. From the sequence of phone attitudes, we compute the trajectory of the phone's writing point W on the writing surface.

We use quaternions as a convenient way to describe and work with rotations in 3-D space. When we say that we update an attitude o_1 using an angular velocity r, we first determine the angle $Q(\Delta t \cdot r)$ by which the phone was rotated from r and the time Δt corresponding to this measurement. Then, we compute the updated attitude o_2 by rotating o_1 by that angle

$$o_2 = o_1 * Q(\Delta t \cdot r) \qquad (3)$$

where $*$ is the quaternion multiplication that corresponds to rotating o_1 by the angle $\Delta t \cdot r$.

This is performed for every measured angular velocity r_t successively. Then, we compute the location at which point W touches the writing surface for every time step t, assuming the writing surface is orthogonal to the gravity vector g and cutting it at point B.

---

[1]See the video in our supplementary material for a comparison how a user writes with a pen and with GyroPen.
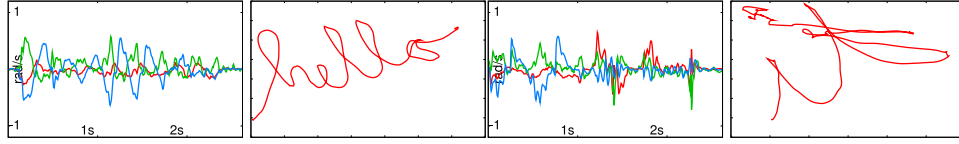
Fig. 4. Examples of writing the words "hello" and "it." (Left) Gyroscope measurements (cp. Fig. 3). (Right) Reconstructed writing path.

Fig. 3 shows the output of the gyroscope sensor, while different writing movements are performed. It can clearly be seen that writing into different directions creates distinct gyroscope measurements. However, it is also clearly visible that raising the writing point W above the writing surface [see Fig. 3(f)] has a very similar pattern to a "right to left" movement [see Fig. 3(b)]. In both cases, the r_x value goes up strongly and stays high for the entire movement; the r_y and r_z values show a wave pattern. The biggest difference between these two movements in our plots is the magnitude of the measurements, which is not intrinsic to the movement direction but only depends on the movement speed. Therefore, we decided not to handle "pen up" movements explicitly but trace the movements over the entire time and handle the "pen up" strokes (i.e., the reconstructed strokes not intentionally written) at a later stage.

Fig. 4 shows examples of reconstructed writing paths for the two words "hello" and "it," both of which are legible to humans. In the path for "it," the pen-up problem is clearly visible: The delayed strokes (i.e., the t-stroke and the i-dot) are connected to the previously written strokes.

C. New-Word Heuristic

With the method described above, it is possible to write in an area of about 7×5 cm without moving the hand on the writing surface. This area is sufficiently large to allow for writing sizes similar to results of writing with a pen on paper. When writing with a pen, a user occasionally moves their hand forward to be able to keep writing in a line. Here, moving the hand forward is not necessary. Instead, the hand can remain at the same spot while writing multiple words (or word-parts) on top of each other: When the end of a word is reached, the user lifts the writing corner, moves it to the left, and then writes the next word "over" the previous one.

To detect this restart gesture, we apply a straightforward decision rule based on the measurement r_x of the gyroscope of rotation around the phone's x-axis, where the combined signals for "pen up" [see Fig. 3(f)] and "right to left" [see Fig. 3(b)] leads to a characteristic pattern (see Fig. 5), which allows detection of the restarts and segmentation of the writing into three parts: w_i the first word, ξ_i the restart gesture which is discarded, and w_{i+1} the second word.

w_i contains all sensor measurements from the beginning (or from the end of ξ_{i-1}) until r_x exceeds a threshold θ_u. Starting from there, all observations that exceed a threshold θ_l are considered to be part of the restart gesture ξ_i, and once the r_x drops below threshold θ_l, the next word w_{i+1} starts. The thresholds were chosen to be $\theta_u = 0.6$ and $\theta_l = 0.2$ from analyzing multiple graphs similar to Fig. 5 during the initial implementation of this method and ahead of any of our user studies.
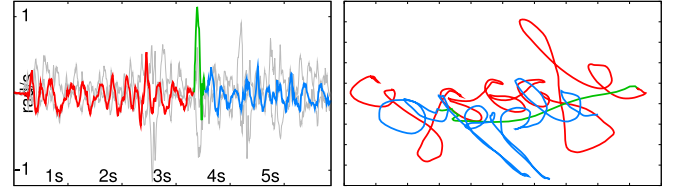

Fig. 5. Sensor output of the gyroscope r_x when writing the word "google," then moving the phone back to the beginning, and then writing "apple" at the same location and the reconstructed writing path. The colors encode the recognized words: w_1 in red, the restart gesture ξ_1 in green, and w_2 in blue. In gray, the gyroscope reading for r_y and r_z.
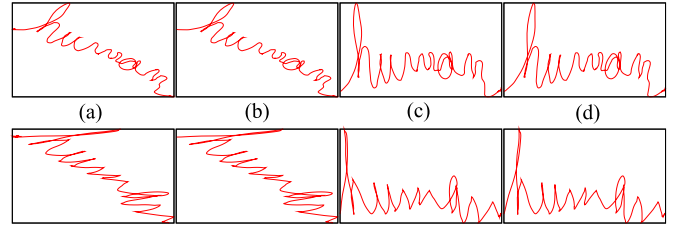

Fig. 6. Postprocessing steps. Writing paths from two different users going through the postprocessing pipeline. (a) Reconstructed writing path. (b) After start–stop detection. (c) After slope correction. (d) After slant correction.

D. Postprocessing

After the writing path has been reconstructed, we perform several postprocessing steps to make the recovered writing paths more similar to normal handwriting and to reduce differences between the writing styles of different users.

1) Start/Stop Detection: Our system requires the phone to be held still while the gravity vector is estimated, but holding the pen stationary is unusual in handwriting data. Similarly, the end of the input is signaled by holding the phone still for half a second. Removing these pauses in the phone motion improves handwriting recognition because it makes the output of GyroPen more like normal handwriting input for which the handwriting recognition system was optimized.

For start detection, we drop all observations until the distance between two consecutive points exceeds a threshold θ_s. For the stop detection, we wait until the phone has been still for half a second by detecting that no two consecutive points have a distance exceeding the threshold $\theta_s = 0.04$ for 500 ms.

While the difference is not visually apparent [see Fig. 6(a) and (b)], the start–stop detection removes between 10 and 50 similar points on most samples at the beginning and end.

2) Slope Correction: Due to different users holding the phone at different angles, the recovered writing paths have different slopes [see Fig. 6(b)].

Fig. 7.   (a) Laser pointer mode and (b) examples of uppercase letters written using the laser pointer mode.

To improve recognition, we normalize the slope by measuring the angle ϕ_w between the first and the last point of each written word w, averaging over the written words, and rotating the words toward the horizontal by this angle [see Fig. 6(c)]. This heuristic is very simple, and more elaborate techniques may improve the accuracy of the overall system (see Fig. 11).

3) Slant Correction: After the slope correction, we apply a slant correction to normalize the writing paths. In normal handwriting of Latin-script languages, the most common directions are vertical (or near-vertical) strokes, such as the letter "I" or the first and the last part of the letter "M." Also here, different users tend to write with different slants [see Fig. 6(c)]. In order to normalize for slant, we compute a histogram of directions for each stroke by computing the deltas between every pair of points between pen-down and pen-up events, determine the dominant direction, and then apply a shear transform to align it to the vertical [see Fig. 6(d)].

E. Laser Pointer Mode

One further advantage of the gyroscope-based approach is that it easily transfers to a "laser pointer"-like writing mode where we assume the writing point W to be on the extension of the phone's longer dimension [see Fig. 7(a)]. In this case, the hypothetical distance between the points C and the point W determines the scale factor of the trajectory.

This mode turns out to be useful for interaction with home cinema systems, similar to the touch pad on recent Sony settop boxes [26], and the phone motion can either be used to control a cursor or to handwrite characters. Handwriting can be activated by pressing and holding a button (physical or on the touchscreen), similar to the interaction model of a laser pointer. Fig. 7(b) shows one example of each of the 26 English uppercase letters written using the laser pointer mode.

F. Handwriting Recognition

To recognize the written items, we use an online handwriting recognition system similar to the one used in the Apple Newton [19] but using an extended feature set similar to the NPen++ recognizer [27]. In online handwriting recognition, the input to the system is a trajectory of (x, y) coordinates over time t. The handwriting recognition engine used was built for normal online handwriting data aiming to recognize stylus input and writing with a finger on a touchscreen. Its recognition accuracy is comparable with other state-of-the-art online handwriting recogni-

tion systems. Note that in contrast with other systems (e.g., 8000 words in [14], 13 gestures in [17]), the handwriting recognition system is an open vocabulary recognizer that can recognize any word that can be written with its alphabet independently of whether it is a proper word and whether the system has ever seen it in the past. To evaluate this, we tried to write the authors' last names and found this to be easily possible.

Above we mentioned that GyroPen does not handle pen-up movements explicitly but considers them to be part of the writing motion. To handle such strokes, the online handwriting recognition literature has been using pen-up strokes as part of the observation [27]. Pen-up strokes also have been used as a means to make models for Chinese handwriting recognition invariant to printed and cursive writing styles [28].

The handwriting recognition system performs size and writing-speed normalization, and therefore, the writing size and speed of a particular item have no impact on the recognition.

G. Calibration

As every sensor (even of the same type) behaves slightly differently, we perform a phone-specific calibration. This has to be performed once per phone and only takes a few minutes. With this calibration, we aim to account for sensor-specific noise and scale so that we can compute calibrated sensor readings s^c from the raw sensor readings s^r and offset O as $s^c = S \cdot s^r + O$.

For both the gyroscopes and the accelerometers, we want to compute the calibrated 3-D vectors as

$$\begin{pmatrix} s_x^c \\ s_y^c \\ s_z^c \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{pmatrix} \cdot \begin{pmatrix} s_x^r \\ s_y^r \\ s_z^r \end{pmatrix} + \begin{pmatrix} O_x \\ O_y \\ O_z \end{pmatrix} \quad (4)$$

In the following, we describe how the parameters S and O are determined for the accelerometers and gyroscopes.

1) Accelerometers: For the accelerometers, we can estimate the offset and the scale jointly. To estimate the offset and the scale, we record a total of I accelerometer measurements $a_1, \ldots, a_I$ in at least three different positions, while the phone is not moving. Using the assumption that the phone should measure an acceleration of $g = 9.81 \, \text{m/s}^2$ while the phone is still and given our error model [see (4)], we can compute the parameters S and O using a system of $3 \cdot I$ linear equations

$$[a_i^c = S a_i^r + O]_{i=1}^I \quad (5)$$

where we approximate the calibrated measurement as $a^c = 9.81 \cdot a^r / ||a^r||$.

Note that this assumes that the scale for each dimension is the same and that the offset is small. In practice, this is not always true, but we validated the assumption by estimating the scale of the axes when the entire gravity components was on a single axis, which suggested that the approximation is appropriate.

Then, we solve this system for the entries of the diagonal matrix S and the offset vector O using the pseudoinverse of the expanded equation system.

For the accelerometers of our experiment phones, we found the scale to be in the range between 0.9 and 1.1 and the offset to be between -0.1 and $0.1 \, \text{m/s}^2$.

2) Gyroscopes: To estimate the scale S and the offset O for the gyroscope, we apply a two-step procedure: First, we estimate O, then S. For convenience, we change the error model of the gyroscope to $r^c = S(r^r + O)$, which is equivalent to (4) but simplifies notation.

An ideal gyroscope will measure no angular velocity when not in motion. To estimate the offsets (O_x, O_y, O_z), we record M gyroscope measurements while holding the phone still. Then, we compute the average over these and set the offset to be

$$(O_x, O_y, O_z) = -\frac{1}{M} \sum_{m=1}^{M} (r_x, r_y, r_z). \qquad (6)$$

To compute the scale of the gyroscopes, we would ideally know the gyroscope output, while the phone performs a rotation for which we know the ideal output for every step. We approximate this by asking the user to move the phone in a series of rotations about each axis in turn, starting and stopping at the same spot. Then, we record the timestamp pairs for which the phone has the same attitude as detected using the phone's 3-D compass.

We record a sequence of I gyroscope r and compass m measurements $(r_1, m_1), \ldots (r_i, m_i)$. Then, we detect pairs (i, j) with $i < j$, where $m_i = m_j$ for a pair of time indices. These pairs represent points where the user has completed a rotation and is holding the phone still. We know that at these two time steps, pairs (o_i, o_j) should be identical. o_j can be estimated as [compare with (3)]

$$o_j = o_i \overset{j}{\underset{n=i}{*}} Q(\Delta t \cdot r_n). \qquad (7)$$

This allows us to assess how much our estimated attitudes o_j differ from their expected attitudes o_i on average:

$$\sum_{(o_i, o_j)} d(o_i, o_j) \qquad (8)$$

where $d(o_i, o_j)$ measures the angle between o_i and o_j.

Then, we use the Downhill–Simplex method [29] to minimize (8) starting with an initial identity matrix for S_r.

For the gyroscopes of our experiment phones, we found the scale to be in the range between 0.9 and 1.1 and the offset to be between -0.01 and 0.01 rad/s.

3) Evaluating the Calibration Parameters: We evaluate our calibration methods by computing the calibration criterion for two different sets of calibration measurements. For the accelerometer calibration, the criterion is the sum of squared error of the system of linear equations [see (5)]; for the gyroscopes calibration, the criterion is given in (8).

We recorded two calibration measurements for the accelerometer and the gyroscope on different days. Then, we estimated the calibration parameters S and O independently for the two measurements and evaluated the criterion on the respective other.

The experiments show that the calibration procedure works for both accelerometers and gyroscopes (see Table I). The estimated calibration parameters for both sequences are significantly better (lower) than using no calibration at all (top row).

TABLE I
QUANTITATIVE EVALUATION OF THE CRITERION

	accelerometer		gyroscope	
	1	2	1	2
$S = \begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix}, O = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	0.257	0.387	0.022	0.0118
calibration 1	0.179	0.225	0.019	0.0033
calibration 2	0.205	0.190	0.020	0.0021

The calibration criterion for both accelerometers and gyroscopes is given for two measurements. Calibration 1 and 2 were estimated on the measurements 1 and 2, respectively.

VI. EVALUATION

For the experiments, we used two different devices: a Samsung Galaxy Nexus and a Samsung Nexus S. These contain an Invensense MPU-3050 motion sensing unit and an EM-Tech EME1511AFRC module including a motion sensor, respectively. Both are low-powered motion-sensing units and during the experiments for the paper, we did not observe any additional power drain on the batteries of the used devices. Additionally, we performed informal experiments with four other popular Android smartphones (LG Nexus 4, Samsung Galaxy S3, Sony Experia LT26i, HTC One X) to verify that the proposed methods are not overoptimized for the test devices. On each of these devices, the GyroPen prototype worked without any changes.

A. Evaluation of Reconstruction Accuracy

To evaluate writing path reconstruction accuracy, we performed experiments recording the reconstructed writing path and the actual writing path of the phone's writing point simultaneously. To record the actual writing path, we used a Wacom Bamboo Pen & Touch graphic tablet and fixed its stylus to the phone using sticky tape. With this setup, we recorded a dataset of 13 words and drawings[2] and compared the reconstructed paths from GyroPen with the ground-truth paths from the tablet.

To measure the reconstruction error, we aligned both paths using a dynamic programming algorithm similar to the one used for stereo reconstruction [30] and measured the error between the reconstructed and the recorded path. The experiments showed an average deviation between our reconstruction and the ground truth of less than 4% relative (5% standard deviation) to the length of the path. The highest error was observed for one of our small drawings where slant correction went wrong (15%).

Four examples comparing the reconstructed paths are shown in Fig. 8. For the first three examples, the paths were aligned very well. In the bottom-right example, the slant correction built into GyroPen failed because it was designed with handwriting in mind, and thus, the stickfigure drawn with GyroPen is strongly slanted to the left. Thus, when using GyroPen for drawing, it is advisable to disable the slant correction.

---

[2]apple, C, d, google, hello (3x), programming, house drawing (3x), stick figure drawing, heart drawing.
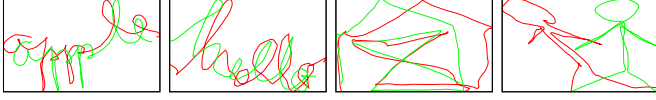
Fig. 8. Comparison of the reconstructed writing paths and the aligned ground-truth paths. Red: the reconstructed path from GyroPen, Green: the ground-truth path from the graphic tablet.
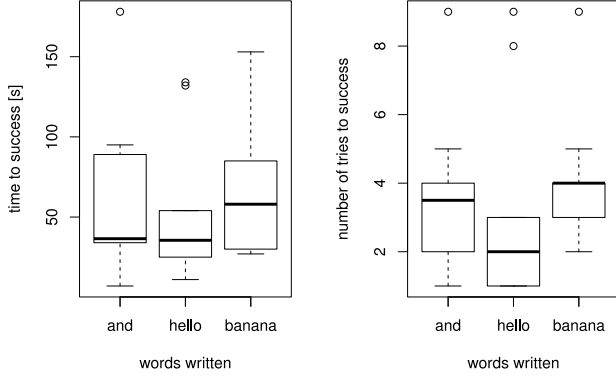

Fig. 9. Distribution of (a) writing times (in seconds) and (b) number of tries necessary until success as box-and-whisker plots. The bold line gives the median values, the lower and upper boundaries of the box mark the 25% and 75% quantiles. The whiskers give the minimal and maximal measured values and the circles mark outliers.

B. User Study: Learning to Use GyroPen

We assessed how quickly a novice user of GyroPen is able to use it for writing a few simple words in a proof-of-concept experiment. We conducted a small user study with ten volunteers who had never used the GyroPen prototype before and had no insight into how it works. The participants, all employed at the same company as the authors of this paper at the time of the study, were aged between 20 and 40 years; nine were males and one was female. There were two left-handed and eight right-handed participants. The study was conducted using the following script and lasted about 15 min per participant. The prototype was running on a Samsung Nexus S phone.

1) *Introduction:* We gave a short introduction how GyroPen can be used and demonstrated writing a few words.

2) *Writing:* We asked the users to try to write three strings of increasing length and difficulty: 1) "and," 2) "hello," and 3) "banana." To get an objective measurement of the entire system's performance, we asked the participants to continue trying to write these strings until the system recognized the string. We limited the number of tries to ten times and recorded the number of tries and the time it took each participant.

3) *Questions:* We asked the participants a series of fixed questions, starting by having them rate the experience with respect to "interestingness and novelty" on a scale from 1–10. Then, we asked them for positive and negative adjectives, whether they would install a GyroPen app, and if they saw specific circumstances in which the method could be particularly useful.

1) Quantitative Evaluation: Fig. 9 shows the distribution of writing times and number of tries necessary for the study participants to write each word. The variation in these measurements

is very large. This shows that different users have a varying degree of difficulty adapting their natural writing movements.

The average participant could write the first short word after half a minute, a longer one after about the same time, and about 1 min to write the six-letter word "banana." (The median times are 36.5, 35.5, and 58.0 s, respectively.)

The median number of tries for each test word were 3.5, 2, and 4, respectively. This suggests that the participants learned this new way of writing quickly and could write the second word with few attempts, but then needed more attempts for the third longer (and likely harder to recognize) word.

Note that these times are measured up to the point that the trajectory could be interpreted correctly by the handwriting recognition system—the reconstructed and visualized text was readable for a human typically much earlier. The median number of tries until a human could read the test words was 1.5, 1.0, and 2.5 (compared with 3.5, 2.0, and 4.0, which we report for machine recognition, Fig. 9). This means that when used for the purpose of just taking a note in a graphical format (just storing the sketch, not requiring recognition), the approach could deliver results much faster than the values above suggest (compare also the study in Section VI-C, which shows that experienced users need 3–4 s to write a word).

The outliers toward the longer times and number of tries show that this input method may not be the method of choice for fast text entry for everybody, but similar restrictions apply for other input methods, e.g., speech input or handwriting using a stylus.

On the other hand, some participants were able to input the desired word very quickly and with only few attempts. This is despite the fact that most of the participants started the study with stating that they had a "terrible handwriting."

In summary, we consider the results of this experiment to be very promising since they indicate that new users will be able to use GyroPen quickly and without much help. For more reliable results, it would be necessary to perform a large study with more users and more words.

2) Qualitative Evaluation: The participants rated the interestingness and novelty of the approach on average with a score of 8 out of 10. The source of participants makes it likely that there is some bias in these results, but we feel that they are an interesting summary of the voiced opinions.

Prompted for positive impressions, the participants responded with these and similar assessments: surprised it works and recognized my handwriting, amazing, awesome, new, interesting, cool idea, intuitive, interesting, has potential, cool and fun, innovative, natural, wonderful idea, much better than handwriting on a touchscreen, definitely better than voice recognition.

Prompted for negative impressions, the participants responded with these and similar assessments: smartphone is bulky/clunky/big, maybe as a standalone pen it could work better, need to improve moving the hand, needs a surface, awkward to hold the phone like this, difficult for long words, cannot beat speed of other input methods for me, there's a learning curve, quality needs to be improved, a bit difficult.

When asked about situations in which this input method could be particularly useful, the answers included: taking of digital notes quickly (with and without handwriting recognition), e.g., in school or for a quick note; when too noisy for speech recognition; when using the touchscreen is difficult (wet or sticky
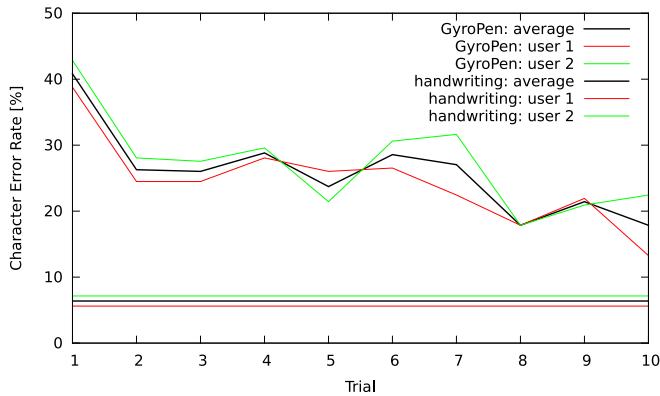
Fig. 10.   Average CER for the ten trials of writing the 52 words for the second user study.

Fig. 11.   Writing styles of two users using normal handwriting on a mobile phone touchscreen and using GyroPen. The recognition result of the handwriting recognition system is shown under the samples.

fingers); for storing sketches; just on a pen that is paired with a smart-phone; recording signatures; kids' drawings.

While the absolute writing area of the GyroPen approach is about the same size as a phone screen, the thickness of fingers makes it difficult to write as small as with GyroPen. Therefore, users had the impression that with GyroPen there is more space. The high accuracy of GyroPen easily allows writing with a letter height of about 5 mm. This is smaller than most people can write with their fingertip on a touchscreen.

In summary, while there is room for improvement, most participants viewed GyroPen as interesting and promising.

C. User Study: Writing the Full Alphabet

To evaluate more parameters of the proposed input method, we performed a second experiment with two users who had used the GyroPen system in the past and had a good understanding of how it works.

In this experiment, the task was to write 26 common English words,[3] one starting with each letter of the English alphabet, both starting in lower- and uppercase. The users were asked to write each of the 52 words ten times in sequence with GyroPen. During this experiment, we measured the CER (see Fig. 10) and writing speed. For comparison, we also asked the users to write each word once with a finger on a touchscreen for normal handwriting recognition.

Both users improved their writing skills: When writing a word for the first time, the users had about 40% CER. The CER dropped to about 28% in the second trial, and after ten trials, the users averaged at about 18% CER. For comparison, the same users obtained between 5% and 7% CER when using normal handwriting. This suggests that some of the errors that users make are due to an imperfect recognition system but that a certain amount of errors are introduced through using GyroPen. As the handwriting recognition system was trained on conventional handwriting input, we suspect that this gap could be closed further by 1) adding GyroPen training data to the recognition system or 2) adjusting the GyroPen reconstruction to let the output resemble conventional handwriting even more, or both.

---

[3]and; been; can; day; even; for; game; have; issue; job; know; long; more; new; one; play; quick; run; some; the; under; very; was; xenon; year; zone.

Regarding writing speed, user 1 needed an average of 3.6 s to write each word, and user 2's average was 3.1 s. In comparison, using normal handwriting input both users needed slightly more than 1 s to write the words on average. The measurements show no significant change in writing speed during the ten trials.

We also analyzed a potential dependence between the error rate and the writing speed by measuring the correlation between normalized writing speed (per word per user) and CER and found a correlation coefficient of 0.07, which indicates that the dependence is very small if there is a dependence at all. The same result is obtained when comparing the average CER on words that were written faster than the median speed (per word per user), which is the same as the CER on words that were written slower up to the third significant digit. Note, however, that these experiments have been performed with just two users, and thus, the results may not hold in a large user study.

Fig. 11 shows a comparison of normal writing styles of the two users with their writing styles using GyroPen for selected words. It is interesting to observe that some of the characteristics of the users' handwriting styles are preserved, e.g., the characteristic shapes of the loops in "g" and "y" seem to differ more between writers than between recording methods.

Further, the examples show how the handwriting recognition system makes some mistakes on GyroPen samples that are clearly legible for a human. We suspect the difference to be due to increased slant and slope of the recorded samples and also partially due to a missing reference frame.

D. Non-Latin Script Input

To evaluate if the GyroPen approach is applicable to other scripts, we performed the following experiment: We switched the handwriting recognizer language to Chinese and tried to write a few Chinese characters. Recognizing these characters worked surprisingly well using a cursive writing style. We did not need to apply any additional tuning except disabling the slope correction. Examples of writing trajectories and recognized characters are shown in Fig. 12.

VII. CONCLUSION AND OUTLOOK

We have presented GyroPen, a method that reconstructs a user's writing path from the inertial sensors of a phone, and
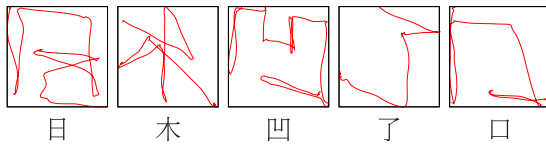
Fig. 12. Writing Chinese characters. Reconstructed writing paths (top) and recognition results of the Chinese handwriting recognition engine (bottom).

have shown that this is a promising approach toward handwriting with the phone. We describe the approaches used for high-accuracy writing-path reconstruction and show that the reconstructed writing paths are accurate enough to be recognized by an off-the-shelf handwriting recognition engine without the need for special tuning. In a first proof-of-concept experiment, the majority of the participants reacted positively to the approach and could use it after a learning period of just a few minutes. In a second experiment, we observed that the method can be used to write all letters with acceptable accuracy after some practice. We have shown that the system is an interesting prototype aiming to enable handwriting with the phone rather than on the phone for an intuitive experience for text entry into mobile phones, which will open unique applications in the future.

GyroPen could be improved by explicitly handling pen-up strokes, for instance, using a sensor-fusion approach of gyroscopes and accelerometers or by training a handwriting recognition system that is fully invariant to pen-up strokes.

The proposed method has the advantage that it works on smartphones without any modification to the hardware. If it was possible to add additional sensors to the phones, a similar user experience could be obtained, e.g., by building a small trackball or an optical mouse sensor into the phones writing corner. Another alternative to using the inertial sensors would be to use computer vision techniques with the built-in camera to reconstruct the phone movements similar to TinyMotion [31]. This would also work without additional sensors but might be problematic if the writing surface is very homogeneous, and thus, there would be no features to track.

REFERENCES

[1] D. L. Grover, M. T. King, and C. A. Kushler, "Reduced keyboard disambiguation system," U.S. Patent 5 818 437, 1998.
[2] N. Henze, E. Rukzio, and S. Boll, "Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices," in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2012, pp. 2659–2668.
[3] P.-O. Kristensson and S. Zhai, "SHARK2: A large vocabulary shorthand writing system for pen-based computers," in Proc. 17th Annu. ACM Symp. User Interface Softw. Technol., 2004, pp. 43–52.
[4] S. Zhai, P. O. Kristensson, P. Gong, M. Greiner, S. A. Peng, L. M. Liu, and A. Dunnigan, "Shapewriter on the iPhone: from the laboratory to the real world," in Proc. Extended Abstracts Human Factors Comput. Syst., 2009, pp. 2667–2670.
[5] SwiftKey, (2010). "Swiftkey app," [Online]. Available: http://www.swiftkey.net
[6] B. Milner, "Handwriting recognition using acceleration-based motion detection," in Proc. IEE Colloq. Document Image Process. Multimedia, 1999, pp. 5/1–5/6.
[7] S. Zhang, C. Yuan, and Y. Zhang, "Handwritten character recognition using orientation quantization based on 3D accelerometer," in Proc. 5th Annu. Int. Conf. Mobile Ubiquitous Syst.: Comput., Netw. Services, 2008, pp. 54:1–54:6.
[8] D. Wigdor and R. Balakrishnan, "TiltText: Using tilt for text input to mobile phones," in Proc. 16th Annu. ACM Symp. User Interface Softw. Technol., 2003, pp. 81–90.
[9] J. Wang, S. Zhai, and J. Canny, "SHRIMP—Solving collision and out of vocabulary problems in mobile predictive input with motion gesture," in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2010, pp. 15–24.
[10] M. Goel, L. Findlater, and J. O. Wobbrock, "Walktype: Using accelerometer data to accomodate situational impairments in mobile touch screen text entry," in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2012, 2687–2969.
[11] I. Gómez, P. Anaya, R. Cabrera, R. Octavio, and A. Molina, "Predictive system text entry controlled by accelerometer with any body part," J. Accessibility Design, vol. 2, no. 1, pp. 31–44, 2012.
[12] D. J. Ward, A. F. Blackwell, and D. J. C. MacKay, "Dasher: A gesture-driven data entry interface for mobile computing," Human-Comput. Interaction, vol. 17, no. 2/3, pp. 199–228, 2002.
[13] S. Agrawal, I. Constandache, S. Gaonkar, R. R. Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in Proc. 9th Int. Conf. Mobile Syst., Appl. Services, 2011, pp. 15–28.
[14] C. Amma, M. Georgi, and S. Tanja, "Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3D-space handwriting with inertial sensors," in Proc. Int. Symp. Wearable Comput., 2012, pp. 52–59.
[15] T. Miyagawa, Y. Yonezawa, K. Itoh, and M. Hashimoto, "Handwritten pattern reproduction using pen acceleration and angular velocity," IEICE Trans. Inform. Syst., Pt.1 (Jpn. ed.), vol. J83-D-1, no. 10, pp. 1137–1140, 2000.
[16] W.-C. Bang, W. Chang, K.-H. Kang, E.-S. Choi, A. Potanin, and D.-Y. Kim, "Self-contained spatial input device for wearable computers," in Proc. 7th IEEE Int. Symp. Wearable Comput., 2003, p. 26.
[17] S.-J. Cho, J. K. Oh, W.-C. Bang, W. Chang, E.-S. Choi, J. Yang, J. Cho, and D.-Y. Kim, "Magic wand: A hand-drawn gesture input device in 3-d space with inertial sensors," in Proc. Int. Workshop Frontiers Handwriting Recognit., 2004, pp. 106–111.
[18] J.-S. Wang, Y.-L. Hsu, and J.-N. Liu, "An inertial-measurement-unit-based pen with a trajectory reconstruction algorithm and its applications," IEEE Trans. Ind. Electron., vol. 57, no. 10, pp. 3508–3521, Oct. 2010.
[19] L. Yaeger, B. Webb, and R. Lyon, "Combining neural networks and context-driven search for on-line, printed handwriting recognition in the Newton," AAAI AI Mag., vol. 19, pp. 73–89, 1998.
[20] C. H. Blickenstorfer, "Graffiti: Wow!" Pen Comput. Mag., vol. 1, pp. 30–31, Jan. 1995.
[21] I. S. MacKenzie and S. J Castellucci, "Reducing visual demand for gestural text input on touchscreen devices," in Proc. Extended Abstracts Human Factors Comput. Syst., 2012, pp. 2585–2590.
[22] H. Tinwala and I. S. MacKenzie, "Eyes-free text entry with error correction on touchscreen mobile devices," in Proc. 6th Nordic Conf. Human-Comput. Interaction: Extending Boundaries, 2010, pp. 511–520.
[23] Diotek. (2013). Diopen. [Online]. Available: https://play.google.com/store/apps/details?id=com.diotek.ime.diopen
[24] PhatWare. (2013). Writepad. [Online]. Available: http://www.phatware.com/?q=product/details/writepad
[25] VisionObjects. (2013). Myscript. [Online]. Available: http://www.visionobjects.com/myscript
[26] Sony, NSZ-GS7. (2013). [Online]. Available: http://www.sony.co.uk/product/google-tv/nsz-gs7
[27] S. Jaeger, S. Manke, J. Reichert, and A. Waibel, "Online handwriting recognition: the NPen++ recognizer," Int. J. Document Anal. Recog., vol. 3, no. 3, pp. 169–180, 2001.
[28] T. Long and L.-W. Jin, "Hybrid recognition for one stroke style cursive handwriting characters," in Proc. Int. Conf. Document Anal. Recognit., 2005, pp. 232–236.
[29] J. A. Nelder and R. Mead, "A simplex method for function minimization," Comput. J., vol. 7, pp. 308–313, 1965.
[30] P. Belhumeur, "A Bayesian approach to binocular stereopsis," in Proc. Int. J. Comput. Vision, vol. 19, no. 3, pp. 237–260, 1996.
[31] J. Wang, S. Zhai, and J. Canny, "Camera phone based motion sensing: Interaction techniques, applications and performance study," in Proc. 19th Annu. ACM Symp. User Interface Softw. Technol., 2006, pp. 101–110.