Knotty Gestures: Subtle Traces to Support Interactive Use of Paper

Theophanis Tsandilas and Wendy E. Mackay

INRIA Saclay, Ile-de-France, INSITU and MSR-INRIA Lab

LRI, Building 490, Université Paris-Sud, Orsay Cedex, France

fanis@lri.fr, mackay@lri.fr

ABSTRACT

We introduce the *knotty gesture*, a simple yet powerful technique for interacting with paper. Knots are tiny circles that can be added to any gesture. Users can leave subtle marks that permit both immediate interaction in the flow of writing and create rich opportunities for future interaction. We identify diverse applications of knotty gestures and explore alternative techniques for interacting with their traces. We conducted two experiments to evaluate the design and recognition heuristics and demonstrated that people can successfully execute knotty gestures, even without feedback. Knotty gestures provide users with a subtle, in-the-flow-of-writing technique for tagging information and subsequently interacting with the paper.

Categories and Subject Descriptors

H.1.2 [**User/Machine Systems**]: Human Factors, H.5.2 [**User Interfaces**]: Evaluation/methodology, Theory & methods, Prototyping, User-centered design.

General Terms

Design, Experimentation, Human Factors.

Keywords

Interactive paper, pen gestures, tangible interfaces.

1. INTRODUCTION

Technologies such as Anoto® offer a wealth of new opportunities for users to interact with and via paper. Audio and visual feedback supported by the new generation of digital pens, e.g., Livescribe pens, allow users to turn pens into highly interactive tools. This poses a number of design issues: how can we help users to smoothly integrate computational power into the act of writing? Currently, users have two basic options for accessing computer functionality: *paper buttons* and *hand-written marks*. Paper buttons offer a more robust form of interaction, since recognition is rarely an issue. They support interaction in the moment, offering users a pre-specified palette of functionality that is available to them as they write. Buttons can be self-explanatory, requiring recognition, rather than recall. However, they do not visually communicate the state of interaction, may be located far from the actual writing and take up a great deal of space on the paper.

AVI'10, May 25–29, 2010, Rome, Italy.

The other alternative, hand-written marks, have the advantage that they do not distract the user from the task at hand. The writer can tag information for later indexing, such as [todo], or insert symbols, such as @ for an email address or an underline to link to a web page. The writer might also draw notes on a pre-printed musical score to facilitate recognition. Hand-written marks are contextual, used in close proximity to the actual writing.

We are interested in how to support interaction both at the time of writing and in the future, combining qualities of both paper buttons and hand-written marks. This paper presents *knotty gestures*, a simple yet powerful technique for interacting with paper. Knotty gestures are subtle: visible, but not obtrusive (Figure 1). Unlike pre-printed paper buttons, which take up space on the paper, and hand-written tags and symbols that clutter up the content, knots are easy for the reader to either detect or ignore. Knotty gestures reside on top of other handwritten strokes and activating them does not require switching to a command mode. They can serve as stroke delimiters, interaction entries, value registers and line connectors. They can be hierarchically nested, allowing for powerful interactions over handwritten data.

We begin by reviewing the literature and describing situations that motivate this work. We explore the design space of knotty gestures and discuss issues concerning its use and recognition. We then present the results of two experiments that explore the effectiveness and ease-of-use of knotty gestures. We conclude with a discussion of our findings and directions for future research.

Figure 1. Drawing knotty gestures over handwritten notes about a series of lectures. Here, knotty gestures (dots on lines & characters) activate audio recordings and link them with the notes for future reuse.

2. RELATED WORK

The advent of light-weight, inexpensive technologies for detecting hand-written ink on paper has triggered both research and commercial applications. *Anoto*[1] technology made interactive paper practical: a pen with a tiny embedded camera captures traces of ink with respect to a unique dot pattern printed on the paper. The system detects the date and time, as well as the precise gesture executed by the user, and makes the time-stamped gestures available on the computer, so that the user can read the notes that were written on the paper. Researchers have explored novel applications for scientists, particularly lab and field notebooks [21, 16] and have explored multi-media indexing [1, 11], copy-paste techniques between paper and documents [8, 18] and editing on physical 3D models [15]. Each approach proposes a specific pen-based set of gestures that are linked to pre-defined computer functions. Users can use these gestures to perform a command, e.g., copying a picture from one page to another, replacing a word, or editing a physical model. Mus*ink* [17] takes these ideas even further, offering users a rich continuum from non-defined to highly structured symbols that can be defined over time.

A limitation of the above approaches is the fact that they do not handle paper as an interactive medium. Gestures are “rigid” commands and instructions, interpreted by applications that reside on the computer. Earlier approaches have succeeded in making paper interactive by using special equipment such as a projector and cameras [19] or a graphics tablet and a PDA [10]. Recently, Song et al. [14] explored paper interaction with pens equipped with miniature projectors. Interaction using these approaches takes place on a separate layer that is not physically linked to the actual act of writing on paper. Liao et al.’s [9] strategy is to study simple forms of pen-top feedback (visual, auditory and tactile) provided directly while drawing gestures. More recently, the Livescribe[2] Pulse pens go beyond simply capturing ink data and offer audio i/o and live feedback via a tiny screen located on the pen.

Since pens are present in both paper interfaces and graphics tablets, techniques originally designed for one medium are often transferred to the other. For instance, Zeleznik et al. [22] introduced Fluid Inking gestures for pen-based computers. The form of these gestures is based on forms of annotations commonly found on handwritten paper documents. Pigtails were initially proposed as delimiters and menu selectors for applications on graphics tablets [4] that were later transferred to paper [8, 9, 15].

Even so, paper has its unique properties and its use poses several constraints. For example, gestures on paper leave a permanent trace. This means that extending powerful interaction techniques from pen-based computer applications to paper is not straightforward. For example, Octopocus [2] offers a rich combination of feedforward and feedback that helps users learn arbitrarily shaped gesture commands with pen input. However, this technique does not work with traditional paper. Similarly, InkSeine [5] allows for rich in-context interactions with virtual objects that are closely associated with handwritten notes. Such interactions take place in “ink-free” modes, permitting extensive use of graphical interface components such as popup menus and windows that are not available in ordinary paper interfaces.

3. CONCEPTUAL DESIGN

3.1 Motivation and Goals

Knotty gestures were born of years of experience studying diverse groups of users using paper. For users like composers of contemporary music and biology researchers, paper continues to play an important role. Composers of contemporary music use paper at different stages of the composition process [17], from sketching initial ideas to working on final scores. Biology researchers, on the other hand, rely on paper notebooks to record research protocols, data and results [10, 16, 21]. Clarity is important for both groups of users, and writing is performed with great discipline. At the same time, writing involves reflection, calculation and experimentation. Several composers make decisions on rhythms, silences and other musical entities while working on paper. Similarly, biologists use paper to reflect on data, make calculations on top of them, identify patterns, and comment on them.

Our design efforts took these needs into account by concentrating on forms of gestures that could be easily integrated into existing writing activities, while providing opportunities for fluid in-context interaction. We sought a design that balances expressive power and simplicity and supports user interaction with paper at two different phases of the writing process: (1) at the time of writing; and (2) in the future, when users return to what they have written in the past to rethink and re-interact with it.

3.2 Command Modes and Pen Feedback

Previous approaches [8, 9] make a hard distinction between regular writing and gestures that represent commands. Users must switch modes, e.g., by pressing a button, before drawing command gestures. Mode-switching mechanisms can be problematic, disturbing the flow of writing. Users must not only remember to switch modes, but pressing a button while writing may also force users to change the way they hold their pens.

In our design, we tried to eliminate the need for mode switching. We also decided to rely on forms of feedback afforded by light-weight pens. We chose to work with Livescribe pens as they are programmable and support both visual feedback and high-quality audio. Livescribe pens are equipped with a 96 x 18 pixels OLED display, which was adequate for the scope of our research. They are programmable through a Java ME API and can store data. Note that although the Livescribe’s API does not provide event classes for capturing runtime movements while writing, we created our own classes that capture events with a frequency of 1 event per 10 ms in a resolution of 677 dots per inch. This allowed us to support reliable runtime feedback.

3.3 Knots

Circular dots are common elements in handwriting. They can be found in punctuation, letters such as “i” and “j”, bulleted lists, and diagrams and to represent points. Their form is not consistent among individuals. For instance, the form of the dot on the “i” can vary greatly: it can be round, a stroke, a tick, or a small “v” drawn upright or horizontally [6]. Despite such variations, “pure” dots are naturally drawn by circularly moving the pen around a point, either clockwise or counterclockwise.

We chose dots as the basis of our design for four main reasons:

1. They are natural and easy to draw.
2. They take a small amount of space and can be attached everywhere, without creating messing writing.

---

[1] www.anoto.com

[2] www.livescribe.com

3. They are distinguishable and easily recognizable by both humans and computers.
4. Their form is circular, periodic and symmetric. A user can keep infinitely circling around a point in any direction, without destroying the shape of the dot.

Knotty gestures are activated on top of other handwritten strokes, such as characters, symbols, lines and curves. They act as "parasites", as they cannot exist on their own. Their traces resemble "knots" (Figure 2) which differentiates them from free dots in punctuation and bulleted lists.

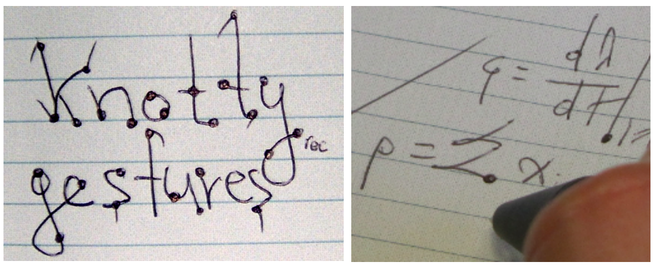

Figure 2. Knots.
***Left*: Traces of knotty gestures activated over handwritten text. Some knots have tails; others are annotated.**
***Right*: A knotty gesture activated at the end of a mathematical symbol, defining a new interaction point.**

Like pigtails [4], knotty gestures can be used as delimiters and command selectors. Knotty gestures, however, are more powerful than pigtails. They can be activated at any position of a stroke, not only at the end. Multiple knots can fit on a single stroke, and their role can be either hidden (regular knots) or revealed (see tailed and annotated knots in Figure 2). More important, knots do not interfere with the recognition of common handwritten strokes such as letters in handwritten text. Unlike knots, pigtails can be only used at the end of command strokes and only after switching to a command mode. This makes them unsuitable for the scenarios presented in Figure 2 and scenarios that we explore later in the paper.

The role of knots does not end upon their creation. They define points of interaction that can be revisited in the future. Unlike paper buttons, knots have local memory, storing their last state of interaction. This aspect of knots borrows properties found in InkSeine's breadcrumbs [5], applied to paper applications.

4. INTERACTION DESIGN

The paper buttons found in commercial notebooks use icons and labels to communicate their functions and they are usually large enough to facilitate selection by tapping. Knots are tiny so using them to support user interaction was a real challenge.

4.1 Basic Interactions

We began by exploring different techniques for interacting with a knot. The most promising four are:

1. *Tapping* on top of the knot. *Tapping* is commonly used in paper applications to activate buttons and menus.
2. *Holding* the pen over the knot. This is comparable to holding down a button to repeat an action, e.g., holding the "down" key on the keyboard to move down in a menu.
3. *Circling* around the center of the knot, clockwise or counterclockwise, to see a list of options and select one of them.
4. *Marking*. The user adds a mark like a marking-menu [7] (see Figure 3) over a knot, to activate a command or assign it a role. The effect of the mark depends upon the orientation of the knot's endpoint relative to its center.

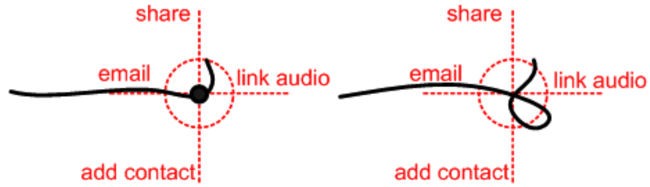

Figure 3. *Left*: Activating a selection with a marking knot *Right*: A pigtail [4]. Dotted lines show the frame of reference.

All four techniques can be used to select a value or command within a set of available options. Tapping is the most generic and allows for simple actions, such as tapping to read a value associated with a knot or tapping to end an audio recording. Marking is the only technique that leaves a visible trace, making it more appropriate when the effect of the action is permanent, in particular when this effect should be visually communicated to the user. The first three techniques are more appropriate when actions have a transient effect, as when information must be hidden or when it is implied by contextual data, e.g., annotations.

Circling occurs naturally while drawing a knot, but using it to control navigation in lists of selectable values proved to be more challenging than we expected. Representative examples of interaction through rotational movements include the wheel in Apple's iPod, earPod [23], and radial scrolling [12, 13]. Our problem was more difficult because rotation occurs within a tiny area, making motor control sensitive to speed variations and noise produced by accidental movements. Our goal was to encourage fine rotations within a small radius (1 - 1.5 mm), ensuring, at the same time, that navigation is both natural and precise. After early experimentation, we observed that people tend to draw dots with discrete rotational movements (Figure 4).

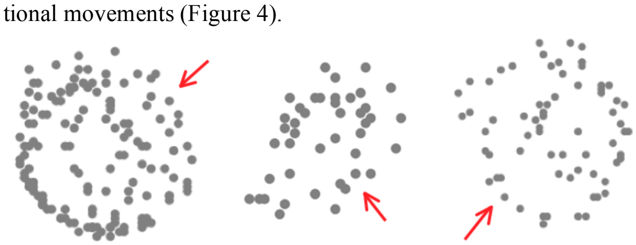

Figure 4. Samples of dots drawn by different users. Arrows show the side of the dot where rotation is regularly fast (low-density areas).

Based on this observation, we made the technique sensitive to discrete, oriented rotations, independently of their size. We detect only full or half rotations (Figure 5).

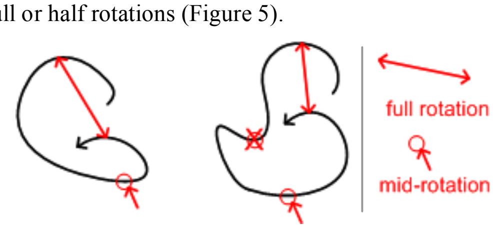

Figure 5. Detection of rotations with the *Circling* technique.

Detecting and controlling shorter rotations with precision is particularly difficult. We observed that full rotations better support motor memory, because the user need not always depend upon the

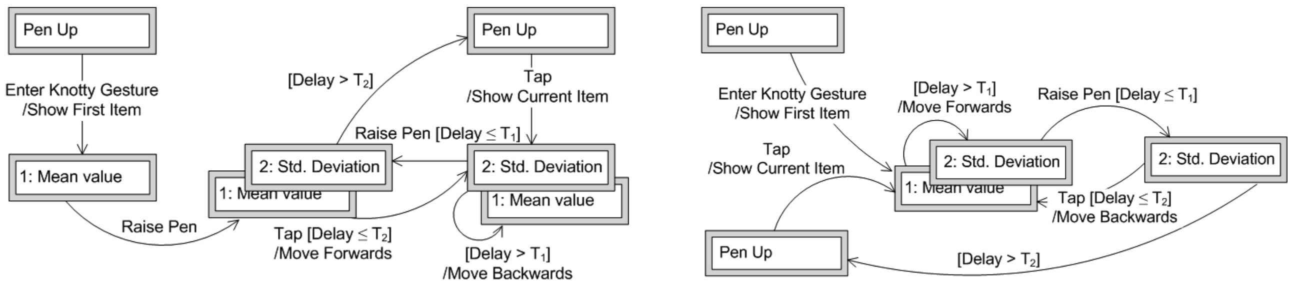

Figure 6. State diagrams (and screens) modeling navigation in a list of mathematical functions. Left: *Tapping* is used to move forwards and *holding* to move backwards. Right: *Holding* is used to move forwards and *tapping* to move backwards.

pen's feedback. We expected that our design would encourage users to draw knots within a small radius, which should theoretically be faster, according to Cao and Zhai [3].

4.2 Combining Basic Interactions

These four techniques can be combined to create richer interactions. For example, the user makes a circle when creating a knot to assign it a role, e.g. adding a timestamp, and taps it later to view the earlier timestamp. Combining holding and tapping is particularly interesting. Both techniques involve placing the pen down on paper: differentiating between them relies solely on a threshold delay over which the action is considered to be a "hold" rather than a "tap". In Figure 6, we demonstrate two symmetric implementations of navigation within linear lists by combining these two techniques. The figure shows how the techniques can be used both upon the creation of the knot to assign a value or at a later time, to review its value.

4.3 Scoping Strokes and Nested Knots

Knotty gestures can assign special roles to the strokes on which they are drawn. For example, they can convert a line to a link anchor, a separator, a text selector or an interactive slider. Alternatively, knots can define a scope over a line, making it the home of other specialized knots. These knots must be drawn at the beginning or end of a line, to reduce ambiguity and avoid conflicts between incompatible knots.

Figure 7 illustrates how a user can nest knots to attach audio recordings to his or her notes. The knot at the end of the line identifies it as an audio recording and marks its scope. Users can then create audio clips by drawing knots anywhere along the length of the recording line, shown here as knots r1, r2 and r3.

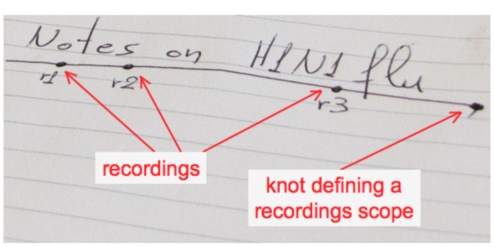

Figure 7. Nesting knots to link recordings to a line

Users can later replay their recordings by interacting directly with the knots. Figure 8 shows the state transitions and associated screens for recording and playing an audio clip. We implemented this functionality using the audio recording capabilities of Livescribe pens.

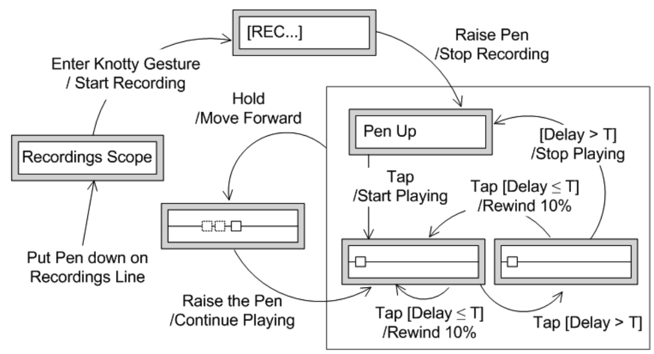

Figure 8. Recording and replaying an audio clip: state transitions and corresponding screens.

Knot nesting offers a powerful encapsulation mechanism. As with hierarchical menus, nesting reduces the number of available functions at each level. Nesting at multiple levels takes advantage of the ability of knots to act as line connectors. A user can draw a knot over a line and, without lifting the pen, continue with a new line. Figure 9 shows two levels of nesting: the user has drawn a nested knot (Level 1) over a line, to define a table. The nested knot is then used to connect a second line, which hosts other knots that serve as mathematical functions (Level 2).

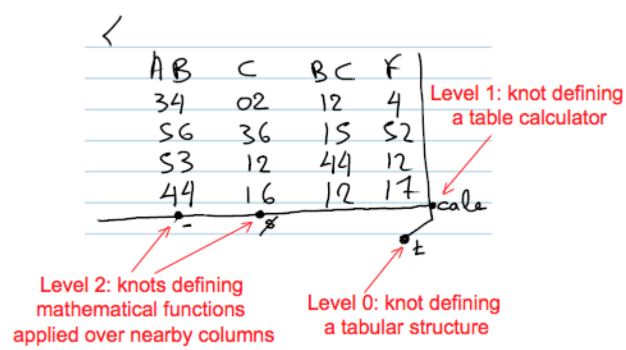

Figure 9. Tabular data

5. RECOGNITION

Circular dots have a simple form that people can easily identify. Even so, we expected a certain variability in how they are drawn. We decided to study these patterns and integrate the findings into our design, to ensure that knotty gestures are as usable as possible. We needed to find the right balance between precision and recognition, ensuring that (1) hand-drawn knots are easily recognized by the pen's processor and (2) recognition does not interfere with character and symbol recognition for regular writing.

5.1 Exploratory Study

To investigate how people draw dots on paper, we conducted a small exploratory study with seven participants (five right-handed and two left-handed). Results from this study were used, first, to explore natural patterns of drawing dots, and second, to refine and evaluate our designs and recognition heuristics.

5.1.1 Procedure

Participants used a Livescribe pen to write on paper. Sessions were brief, lasting approximately 10 minutes. Each session had two parts. Participants were first asked to write a small set of punctuated words and sentences on paper: “Hello world!”, “This is a butterfly.” “help!!!”, “five...”. They also wrote individual characters and punctuation marks that contained dots (“;”, “?”, “:”, and “?”). Each participant copied a total of 77 alphabetic characters, including four i’s, a single j and five tiny o’s, and 25 dots that served as punctuation marks. Participants then drew a total of 32 dots in different ways: on top of lines, as start-points and endpoints of lines, dots drawn naturally and dots drawn in a particular direction (clockwise or counterclockwise). We asked participants to write and draw normally, without rushing.

5.1.2 Dot Patterns

Analysis of dots drawn by participants was performed with a custom-made visualization tool. We focused on identifying the following patterns: *drawing direction*, *circularity*, *diameter*, *number of rotations,* and *density* of data points. A local drawing direction in a stroke can be assessed easily from a series of points $p_1(x_1, y_1)$, $p_2(x_2, y_2)$ and $p_3(x_3, y_3)$ by calculating the determinant *det* of the following matrix:

$$M = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}$$

When the drawing direction is clockwise, *det(M)* is greater than zero. When the direction is counterclockwise, *det(M)* is lower than zero. A value equal to zero corresponds to a straight line. To avoid the capture of accidental direction variations in our sampled measurements, we used a popularity-based filter to remove noise. The filter selected the most popular direction taken from the last seven measurements. Figure 10 illustrates examples from participants 1, 2 and 3. As shown in the figure, the left-handed participant consistently followed a clockwise direction (in red) to draw dots. The two right-handed participants usually drew in the opposite direction (in blue), but not always. Directions varied even for the same participant, indicating that knotty gestures must permit drawing in both directions.

Figure 10. Representative examples of dots drawn by Participants 1-3, as captured by the pen. Clockwise patterns are shown in red, counterclockwise patterns in blue.

Circularity, like direction, can be seen as a local property within a stroke. Circular points follow the same direction and have a common center. In characters like “o” or “d”, circles are open and clear, whereas dots have a spiral form, which usually closes towards the center. A spiral form also implies that more than one rotation around the center may exist. We used both the minimal circle diameter within a dot and the number of rotations to differentiate between dots and other circular forms that appear in writing. Finally, density depends on the length of a dot’s stroke but also on drawing speed. Dots that serve as punctuation are usually drawn instantly and leave a trace with a small number of points. Explicit knots are denser.

5.2 Profile of Recognizable Knotty Gestures

We based the recognition profile of knotty gestures on data captured from the first three participants. Data collected from the next four participants were used for validation. Although the sample size was relatively small, it served our purpose well. Our goal was to identify a set of simple and easily replicated heuristics, rather than base the recognition of knotty gestures on complex classification techniques that require expensive computational resources. Recognition had to be performed in real time while moving the pen and, ideally, be updated after the arrival of every new pen event.

Our initial experimentation with the lightweight $1 recognizer [20] running on Livescribe pens resulted in poor response times. We built the profile by combining the geometric properties presented earlier: drawing direction, circularity, diameter, number of rotations and density. We gave precision higher priority than recall, to reduce false positives when recognizing knotty gestures within text. Although we also sought to eliminate the number of false negatives, this was less important, since several dots in our data did not match the desired profile, e.g., they were too large. Even so, we expected that users would be able to easily learn to activate knotty gestures accurately, given appropriate feedback.

Table 1 presents our recognition results. Results are almost perfect for characters in terms of precision. Table 1 shows that 27% of punctuation dots were recognized as knotty gestures. However, this should not be considered as a limitation. Dots in punctuation appear in free space and their distinction from knots is trivial. If we add an additional requirement, i.e., that knots must be attached to strokes, precision becomes perfect.

Participant (hand)	false positives (characters)	false positives (punctuation)	false negatives (explicit dots)
1 (R)	0/77	0/25	9/32
2 (R)	0/77	19/25	1/32
3 (L)	0/77	2/25	7/32
Total (1-3)	0/231 (0%)	21/75 (28%)	17/96 (17.7%)
4 (L)	1/77	7/25	16/32
5 (R)	0/77	5/25	4/32
6 (R)	0/77	9/25	0/32
7 (R)	0/77	5/25	0/30
Total (4-7)	1/308 (0.32%)	26/100 (26%)	20/126 (15.9%)
Total (1-7)	1/539 (0.19%)	47/175 (27%)	37/222 (16.6%)

Table 1. Recognition results from participants 1-3 formed the recognition profile; participants 4-7 tested the profile.

Figure 11 illustrates the only character sample where our recognition algorithm falsely identified a knotty gesture (*P4, col.2*):

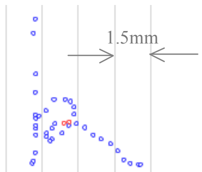

Figure 11. The one case where our recognition algorithm failed. Red points show where a knotty gesture was identified.

6. EVALUATION

We conducted two experiments to evaluate our design. The first tests the accuracy of knotty gesture recognition and the role of runtime feedback. The second evaluates the interaction techniques presented in Section 4.

6.1 Recognition Accuracy and Pen Feedback

Our exploratory study showed that users had an average of 16-17% recognition errors when drawing explicit dots, which rose to 50% for participant 4. This could be considered a significant obstacle for the usability of our design, unless users can easily learn to draw knotty gestures when given appropriate feedback. Experiment 1 tests how pen feedback affects recognition accuracy.

6.1.1 Participants

Four women and four men, 24 to 43 years old, participated in this experiment. Five of these participants had also participated in the exploratory study (Participants 1-5). Participants 6 & 7, for whom recognition recall had been perfect, were not included in this experiment.

6.1.2 Procedure

Training: Before starting the experiment, participants were shown the Livescribe pen and allowed to practice forming knotty gestures. They received feedback when knots were correctly recognized. Participants completed 24-30 practice trials in approximately two minutes.

Experiment: A within-subjects design with two conditions:

1. *Feedback* condition. A brief sound was produced as soon as a knotty gesture had been identified. An "OK" message was also displayed on the pen's screen.
2. *No Feedback* condition. No feedback was provided during the task or after its completion.

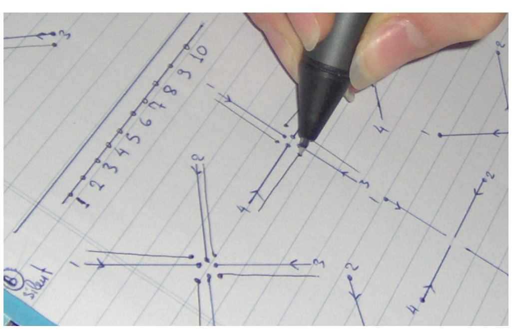

Figure 12. Experiment 1: Participants drew knots on top of preprinted lines or in parallel with oriented guide lines.

The conditions were counter-balanced for order. Participants were asked to activate knotty gestures by drawing dots: (1) over lines, (2) at the start of a line and (3) as line endpoints. Lines were drawn in eight different directions. Groups of numbered and oriented lines had been previously drawn on the experimental sheets to guide the participants throughout the experiment (Figure 12).

Each participant completed 52 tasks within each condition: 20 knots over lines, 16 knots as start points and 16 knots as endpoints. The whole experiment lasted about 15 minutes.

6.1.3 Results

Accuracy results are presented in Table 2. These results are low estimates, as they include errors not relevant to the recognition process, e.g., errors caused when a line stroke was interrupted and then resumed. As we had hoped, accuracy was over 96-97% for both conditions: participants clearly learned how to activate knotty gestures correctly. Accuracy was similar or even higher in the No Feedback condition, implying that feedback is no longer necessary once they have learned how to form knots.

Condition	Knot Type	Accuracy	Std. dev.
Feedback	dots over lines	96.3% (154/160)	4.4%
	line start points & endpoints	97.3% (249/256)	2.6%
No Feedback	dots over lines	100% (160/160)	0.0%
	line start points & endpoints	96.9% (248/256)	2.4%

Table 2. Accuracy results

We examined whether feedback had any effect on the time needed to draw knots, considering only knots drawn on top of lines. The effect of feedback on time was not significant ($F_{1,7} = 1.38$, $p = .28$), despite the fact that a mean difference of 120 ms was observed. Figure 13 (right) shows that this difference can be attributed entirely to Participant 1. Participants followed different strategies in the absence of feedback. Some, like participant 1, tried to ensure that a knotty gesture would be correctly recognized by drawing an increased number of spiral circles. Others drew knots faster, since they did not have to pay attention and respond to the feedback.

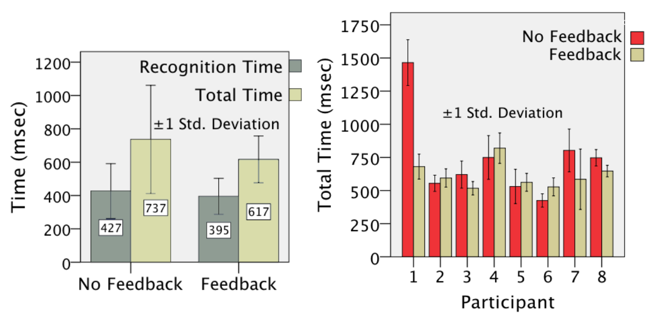

Figure 13. *Left*: Recognition and total time for both conditions *Right:* Distribution of total time across participants

6.2 Selection through Linear Lists

Experiment 2 tested two list navigation techniques (illustrated in Figure 6) and compared them to a variation of list navigation based on the *circling* technique. We refer to the former techniques as *tapping* (tap to move forwards) and *holding* (hold to move forwards).

6.2.1 Participants and Apparatus

12 volunteers (seven men and five women), 21 to 30 years old, participated in this experiment. Two were left-handed. As in the previous studies, participants interacted with a Livescribe pen and were given audio feedback within the experiment.

6.2.2 Task and Experimental Conditions

Participants performed a series of selection tasks by activating knotty gestures over horizontal lines. For each task, the target item was shown on the pen's display. Users were not required to reach the target value with the first attempt; they were allowed to return to the knot and refine its value. Audio feedback was provided as soon as a knotty gesture had been recognized. The three techniques were tested on two different lists of items: (1) an ordered list of numerical values from 1 to 10, and (2) a randomly ordered list of fruit names (in French): *kiwi, ananas, banana, prune, orange, figue, mangue, cerise, citron, fraise, pomme.*

All three techniques that we tested allowed for bidirectional movements. The dominant (forward) direction for *circling* was the direction, either clockwise or counterclockwise, that the user had followed to activate the knotty gesture. Selection transitions in the lists were triggered by full rotations. We found that faster transitions were particularly difficult to control.

The time thresholds T_1 and T_2 (Figure 6) for *tapping* and *holding* were selected empirically as 550 ms and 800 ms, respectively. The active range of *tapping* and *circling* was limited to 1.24 mm around the knot's center.

6.2.3 Procedure

We used a [2x3] within-subjects design, with *List Type* (Sorted, Unsorted) and *Technique* (*circling, holding, tapping*) as primary factors, and Item Position as a secondary factor. Each block of 20 trials was based on one of the six conditions, with two replications of each item position. Participants were exposed to two blocks per condition, or 240 total trials. Participants had a practice session of 15-20 trials before each condition. The order of presentation of the three techniques and the two list conditions was balanced using a Latin square design. The unsorted list changed between techniques, but stayed constant while the same technique was tested by a participant.

Participants were asked to perform tasks as quickly as possible, trying to be precise. After the completion of the experiment, participants were given a questionnaire to answer questions about their experiences with the three techniques and to rate them. Each session lasted about 50-60 minutes.

6.2.4 Results

Before performing our analysis, we removed outliers that appeared three standard deviations away from within-cell means. Outliers accounted for 2.3% of the total measurements. Figure 14 illustrates the overall results for the *total selection time* and the *number of visits* of a value before its final selection.

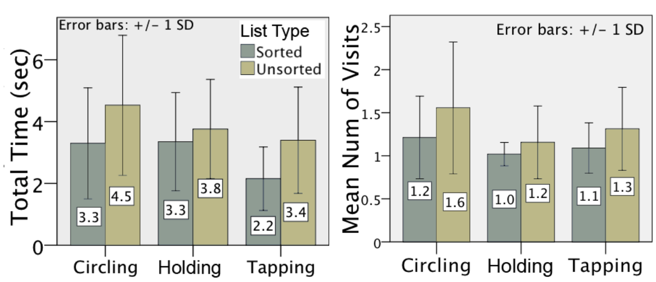

Figure 14. Results for Total Time and Number of Visits

Results show a clear advantage for the *tapping* technique for both unsorted and sorted lists. They also show an advantage of *holding* over *circling* for unsorted menus. An ANOVA repeated-measures analysis showed a significant main effect for both *List Type* ($F_{1,11} = 96.3$, $p < .0001$) and *Technique* ($F_{2,22} = 65.0$, $p < .0001$) on selection time. A pairwise comparison with Bonferroni's adjustment showed a significant difference between *circling* and *tapping* ($p < .0001$), *holding* and *tapping* ($p < .0001$), and *circling* and *holding* ($p = .027$) as well as a significant interaction effect between *Technique* and *List Type* on selection time ($F_{2,22} = 16.8$, $p < .0001$).

We observed learning effects. More specifically, the effect of *Block* on Total Time was significant ($F_{1,11} = 7.5$, $p = .019$). Note, however, that no significant interaction between *Technique* and *Block* was found ($F_{2,22} = 0.55$, $p = .59$). Finally, the effect the position of the target on Total Time was significant ($F_{9,99} = 247.1$, $p < .0001$). Figure 15 shows how the target's position affected user performance for each of the three techniques.

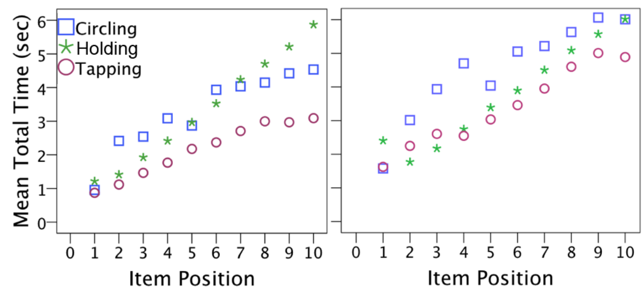

Figure 15. User performance with respect to the position of the target for Sorted (left) and Unsorted Lists (right)

6.2.5 Subjective User Feedback and Discussion

Subjective user answers and comments were consistent with the quantitative results. Most participants felt that tapping was the fastest technique. Holding was also highly rated, and some participants commented that it required the least effort. On the other hand, several participants found that the delay (550ms) between transitions was too long in the case of sorted numerical lists. We had selected this delay so that selection was precise in both conditions. However, we recommend that designers adapt its value to the type of the list or let users adjust it on their own.

Participants easily learned how to switch between forward and backward movements when using the holding and tapping techniques. Contrary to our expectations, *Circling* was less effective. Several participants complained that they could not easily control the technique, as transitions did not always follow their rotational movements. We observed that when participants' attention was on the pen's display, the pen's tip often moved out of the active range of the knot (radius of 1.24 mm). The problem was more frequent in the case of unsorted lists. The active range was set to discourage the creation of large knots. Nevertheless, this constraint was not communicated to the user with any feedback. Considering previous work on pen-top feedback [9], we are working on feedback techniques that could address this problem.

Besides user performance, each technique has its own advantages. Holding is particularly suitable for controlling non-discrete parameters, where continuous visual feedback is required. Our audio recording application (Figure 8) uses a variant of holding to control audio playback, allowing users to move forward with continuous transitions.

7. CONCLUSIONS AND FUTURE WORK

This paper introduces the *knotty gesture*, a simple yet powerful technique for interacting with paper. Our goal was to create a technique that fits easily into the flow of writing, is detectable by both people and recognizers, but does not distract from or clutter the content. We also wanted to provide users with interaction as they write and later, when they return to the page or operate on the data embedded in the page.

A knotty gesture is essentially a tiny circle or knot that can be added to any hand-written gesture. Users can draw barely visible knots that simply indicate that some form of interaction is available, without revealing the details. The user must interact directly with the knot in order to discover and activate its functionality. This is useful for formal documents or those containing private information. Users can also make knots more visible, either by drawing them larger or by adding tiny tails whose direction indicates different alternatives, like an inked pie-menu. Knots live as parasites over other ink strokes, such as characters in text, underlines, tags, and symbols, assigning meaning to them and making them interactive. Thus, a hand-written sigma can be turned into a summation function; a hand-drawn line can be designated as the frame for other knots that link to audio recordings.

We conducted an exploratory study to investigate how people naturally draw dots and two experiments to explore different questions about knotty gestures. We demonstrated that people can easily learn to create knotty gestures, even without feedback from the pen. We explored different mechanisms for interacting with knotty gestures, including *tapping*, *holding* and *circling*. These techniques have different advantages and disadvantages, which should be taken into account when designing applications that incorporate knotty gestures. Our future work will further improve their usability based on the findings of our studies and explore them in a range of paper applications. We are particularly interested in studying how knotty gestures can be used by groups of users who frequently switch between paper and computers.

Knotty gestures form a specific answer to the problem of how to provide users with a subtle, in-the-flow-of-writing technique that allows them to interact both as they write and when they later return to the printed page. Our approach emphasizes lightweight interaction, providing users with maximum control.

8. REFERENCES

[1] Avrahami, D., Hudson, S., Hudson, S., Moran, T., & Williams, B. (2001) Guided gesture support in the paper PDA. In Proc. UIST'01 ACM, pp.197-198.

[2] Bau, O. & Mackay, W. (2008) Octopocus: a dynamic guide for learning gesture-based command sets. In Proc. UIST'08, 37–46. ACM, pp. 173–182.

[3] Cao, X. & Zhai, S. (2007) Modeling human performance of pen stroke gestures. In Proc. CHI'07 ACM. pp. 495-1504.

[4] Hinckley, K., P. Baudisch, G. Ramos, & F. Guimbretière. Design and analysis of delimiters for selection-action pen gesture phrases in Scriboli. In Proc. CHI 2005, ACM (2005), 451-460.

[5] Hinckley, K., S. Zhao, R. Sarin, P. Baudisch, and E. Cutrell (2007). InkSeine: In Situ Search for Active Note Taking. In Proc. CHI 2007, pp. 251-260.

[6] Huber, R. & A.M. Headrick, Handwriting identification: facts and fundamentals. 1999: CRC Press.

[7] Kurtenbach, G. & Buxton, W. (1993) The limits of expert performance using hierarchic marking menus. In Proc. CHI '93 ACM. p. 482-487.

[8] Liao, C., F. Guimbretière & K. Hinckley. PapierCraft: a command system for interactive paper. In Proc. UIST 2005, ACM Press (2005), 241-244.

[9] Liao, C., F. Guimbretière, & C. Loeckenhoff. Pen-top Feedback for Paper-based Interfaces. In Proc. UIST 2006, ACM Press (2006), 201-210.

[10] Mackay, W. E., Pothier, G., Letondal, C., Bøegh, K., & Sørensen, H. E. (2002). The missing link: augmenting biology laboratory notebooks. In Proc UIST 2002, ACM, pp. 41- 50.

[11] Mankoff, J., Hudson, S., & Abowd, G. (2000) Interaction techniques for ambiguity resolution in recognition-based interfaces. In Proc. UIST '00 ACM. pp. 11-20.

[12] Moscovich, T. and Hughes, J. F. (2004). Navigating documents with the virtual scroll ring. In Proc. UIST '04. ACM Press, pp. 57-60.

[13] Smith, G. M. and m. c. schraefel (2004). The radial scroll tool: scrolling support for stylus- or touch-based document navigation. In Proc. UIST '04, ACM Press, 53-56.

[14] Song, H., Grossman, T., Fitzmaurice, G., Guimbretiere, F., Khan, A., Attar, R., and Kurtenbach, G. (2009) PenLight: combining a mobile projector and a digital pen for dynamic visual overlay. In Proc. CHI '09. ACM, pp. 143-152.

[15] Song, H., F. Guimbretière, C. Hu and H. Lipson. ModelCraft: capturing freehand annotations and edits on physical 3D models. In Proc. UIST 2006, ACM Press (2006), 13-22.

[16] Tabard, A., Eastmond, E. & Mackay, W. From Individual to Collaborative: The Evolution of Prism, a Hybrid Laboratory Notebook. In Proc. CSCW '08, ACM Press (2008), 569-578.

[17] Tsandilas, T., C. Letondal & W.E. Mackay. Musink: composing music through augmented drawing. In Proc. CHI 2009, ACM Press (2009), 819-828.

[18] Weibel, N., Ispas, A., Signer, B., and Norrie, M. C. 2008. Paperproof: a paper-digital proof-editing system. In Extended Abstracts CHI '08. ACM, pp. 2349-2354.

[19] Wellner P. Interacting with the DigitalDesk. Communications of the ACM ,1993. 36(7), p. 87-96.

[20] Wobbrock, J.O., A.D. Wilson & Y. Li. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In Proc. UIST 2007, ACM Press (2007), 159-168.

[21] Yeh, R., C. Liao, S. Klemmer, F. Guimbretière, B. Lee, B. Kakaradov, J. Stamberger, & A. Paepcke. ButterflyNet: A Mobile Capture and Access System for Field Biology Research. In Proc. CHI 2006, ACM Press (2006), 571-580.

[22] Zeleznik, R., Miller, T., Holden, L., LaViola, J., Fluid Inking: Using Punctuation to Allow Modeless Combination of Marking and Gesturing. 2004, Brown University, TR CS-04-11.

[23] Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R., and Baudisch, P. 2007. EarPod: eyes-free menu selection using touch input and reactive audio feedback. In Proc. CHI '07. ACM, pp. 1395-140.