From Smart Light Dimmers to the IPOD: Text-Input with Circular Gestures on Wheel-Controlled Devices

Frode Eika Sandnes
Faculty of Engineering, Oslo University College
frodes@hio.no

Yo-Ping Huang
Department of Computer Science and Engineering, Tatung University
yphuang@ttu.edu.tw

Abstract

A uni-stroke text input strategy for wheel input controls is proposed, such as the wheel controls found on the 5th generation ipod multimedia devices. The popularity of the ipod indicates that the wheel input paradigm is intuitive and usable. The proposed uni-strokes are based on circular motions that follow the contour of the wheel. Spatial mnemonics based on the shape of the alphabetic characters are used to minimize the time and effort learning the uni-strokes. An approximate distance based method is used for robust character recognition of uni-stroke patterns.

1. Introduction

Text input on small mobile devices such as mobile phones, smart home appliances and multimedia devices, including mp3 players, is challenging because such devices are rarely equipped with keyboards. A large number of novel input strategies for mobile devices have been proposed in recent years. These input techniques can be classified according the type of physical input controls available on the device and include numeric keypad-based [1], keystroke input [2], chording [3], stylus-based touch pads [4-6], joysticks [7] and tilt [8]. Tradeoffs often exist between speed of operation/productivity and easy of learning. Other factors considered are error [9], divided attention and limited visual feedback [10], richness of input alphabets, abilities to edit text, etc.

1. 1. The ipod

The ipod has received much media attention and has become a consumer favourite [11-13]. The ipod has a number of characteristics that can help explain its popularity. The 5th generation ipod is equipped with a touch sensitive virtual wheel that allows users to make menu selections from lists, adjust the volume, navigate through songs and videos, etc. The wheel is virtually rotated by moving the finger in either clockwise or anticlockwise direction. Furthermore, there are five buttons beneath the wheels that are activated by pushing, or clicking, the wheel in the centre, north, east, south or west. These buttons have a consistent single mode, where pressing the centre means select, north means back, south means play/pause and east and west mean next and previous track, respectively.

The wheel paradigm has strong perceived affordance as most users know how to operate the wheel without instruction. The limited number of modes and consistency of the buttons results in a steeper learning curve. Another feature is simplicity – the set of functions are streamlined and imposes a reduced cognitive load on the users. The ipod can be used with limited visual feedback. For instance, if the device is lying in the users pocket the device can be operated through touch without taking the device out of the pocket. The user can easily sense the orientation of the device through touch due to the rectangular shape and the wheel that is located at the bottom half of the device. The ipod is symmetric and can therefore be operated with either hand and the spatial skill set acquired by users is conserved across hands. Moreover, the device can be operated either using one hand, where the thumb is used to operate the wheel (see Figure 1), or using two hands where one hand holds the device and the index finger of the other hand operates the wheel (see Figure 2). However, note that the ipod is not a bimanual device although two hands are used.

Figure 1. One-handed operation of the ipod, using the thumb to interact with the wheel.

Figure 2. Two-handed operation of the ipod, holding the device in one hand and using the index finger of the other hand to interact with the wheel.

1.2. Text input on the ipod

Despite its widespread use there is surprisingly little research on HCI aspects of the ipod and its wheel paradigm. One reason for this lack of attention is that the ipod is protected by various patents and until recently the ipod was only capable of running the manufacturer's own proprietary software and therefore not a suitable platform for doing HCI research. However, recent developments such as the IpodLinux Project with its podzilla user interface has changed this situation [14, 15] as researchers can develop their own applications for the ipod. For instance, IpodLinux allows researchers to implement logging of user behaviour which is essential for empirical HCI studies.

1. 3. Podzilla and text input

The original ipod software system has a digit input system for entering dates and times which is based on scrolling through the sequence of digits. No facilities are provided for inputting text. However, several text input techniques has been implemented for podzilla [16] and other techniques are under consideration such as Dasher [17]. Podzilla allows text to be input through physical keyboards connected through the serial port on older ipods. However, external keyboards are impractical and not always easy to attach in a mobile setting. Another technique is scroll through, where the wheel is used to select a character and the activation button is used to make the selection. This intuitive method has been discussed in the literature [2, 18]. Text can be input with little training and there is room for error. However, this technique is slow and dependent on visual feedback. All the symbols of the alphabet cannot be displayed simultaneously for large symbol palettes such as Kana (Japanese phonetic alphabet).

An on-screen keyboard input technique is also provided where the activation button is used to select characters and the north, south, east and west buttons are used for navigating the keyboard. This technique is intuitive, but visual attention is required and the wheel is not exploited. Furthermore, the up, down, left right navigation is inconsistent with the intended modes of the four keys. More exotic and less feasible input techniques include morse code, where the middle button is used to input the codes (the wheel and the other keys are not utilised) and unicode hex input, where the unicode value of a symbol is input in hex and the hex digit selected with the scroll wheel. Few users remember unicode values and this input technique intended special purposes only. In the four button keyboard technique the keys are used to direct the user to the desired characters through a menu structure. In the telephone keyboard technique the wheel is segmented into eight sectors and each sector is activated by a gentle tap (not button click). The centre button is also used. Both multitap and predictive text input is implemented.

The Podzilla text input technique that is the most similar to the one proposed in this paper is cursive, where the wheel and the action button are used to approximate English cursive script. Characters are input as strokes where the wheel is used to input vertical line and curved strokes and the action button is used to input circular strokes. The play button is used to accept an input sequence. One obvious problem is that the cursive input technique relies on mixed mode input, i.e., both the wheel and the activation button are used for inputting strokes. The push of the activation button cannot strictly speaking be classified as a stroke as a keystroke is a motion in the z-dimension as wheel interactions are motions in the x-y plane which is perpendicular to the z dimension. Generally, a shortcoming of all the Podzilla text input techniques is that they all rely on the five buttons, which breaks the consistency of the original ipod user interface..

1.4.Gesture-based text input

There is a significant body of research on gesture based text input, i.e., text input where characters are input according to some gesture, usually a shape that resembles the letter. Most studies focus on touch pads or touch sensitive displays where the user can use the finger or a stylus for input and the gestures are represented in the (two-dimensional) plane. Various studies include unistroke [19], Graffiti [20], Cirrin [21], SHARK [22], EdgeWrite [7]..

1.5. Objectives

The objective of the strategy proposed in this study is to solely rely on the wheel for textual input, such that the five buttons can be used for other purposes to maintain mode consistency, or the strategy can be used on wheel controls without buttons. Next, the technique should be easy to learn and require minimal retention as the strategy relies on a set of spatial stroke mnemonics [23, 24]. Moreover, the strategy is less dependent on visual feedback than for instance the scroll through technique. Consequently, one should be able to expect text input speeds to approach that of normal handwriting. Finally, this study is not limited to the ipod device but address the wheel input paradigm for text input in general as the paradigm could be applied to any device with a touch sensitive pad or display such as PDAs.

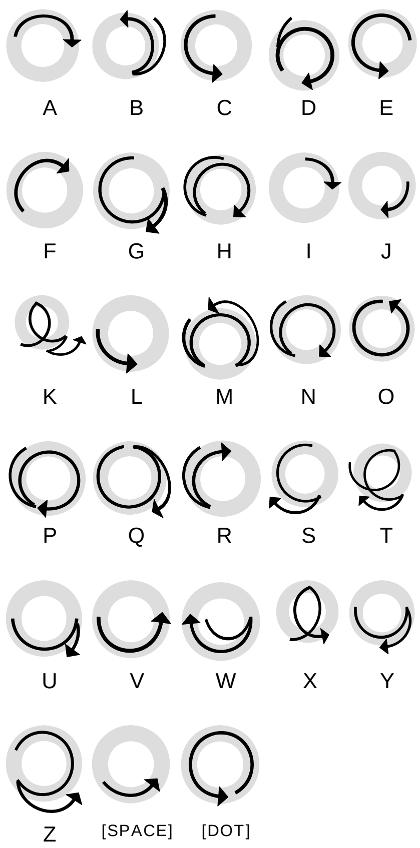


Figure 3. An alphabet of circular gestures based on spatial mnemonics.

1.6. Circular uni-stroke text-input gestures

Figure 3 shows the gestures adopted in this study for the characters of the English alphabet, space and full-stop. The gray circles symbolise the wheel and the arrows the path of the finger as it travels along the wheel, where the arrowhead indicates the end position of the gesture. The gestures are based on the physical appearance of the characters. The physical shapes of the characters are therefore used as spatial mnemonics with the purpose to accelerate the learning of the circular stroke palette. For example, A is represented by a clockwise motion from east to west, symbolising the “roof” of the capital letter A. B is represented by to continuous half-circles – first clockwise, then anticlockwise, to symbolise the two bumps on the capital letter B. C is represented by a anticlockwise half-circle from north to south just as the C would be written in one single stroke. Most of the mnemonics are based on uppercase letters such as A, B and C, while E, F, H, R, T and U are based on lower-case letters. Others could be either upper or lower case, for instance C, K, M, N, O, U, I, J, P, S, V, W, X, Y and Z. Space is simply a gentle motion from southwest to southeast that has the appearance of the underscore character that is often used to represent space (space is obviously a void shape).

Another rationale behind the proposed palette of mnemonics is uniqueness – each gesture needs to be unique and distinguishable from the other gestures. For instance, C, E and F are similar as these gestures are all half-circles in clockwise direction. However, C is a perfect half circle from south to north, E is a ¾ circle from south to east and F is in between, i.e., a half circle from southwest to northeast. Clearly, humans introduce perturbations and error into this model and an approximate strategy is therefore employed to resolve ambiguities in a robust manner.

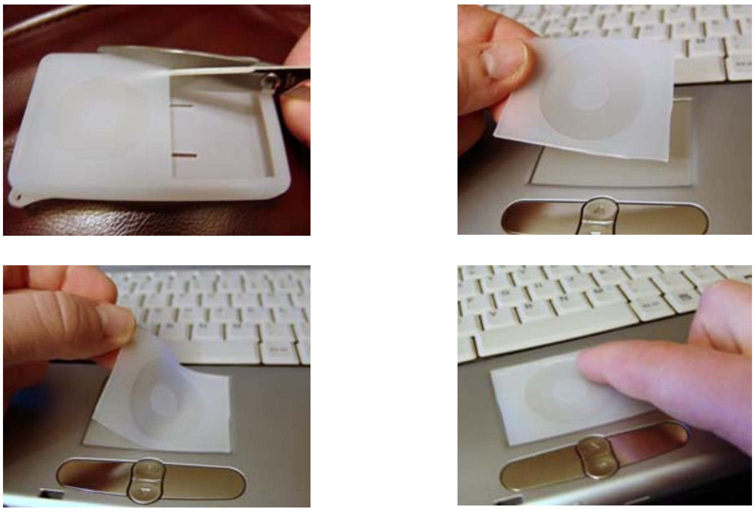


Figure 4. A DIY cut-out ipod wheel template overlaid on a laptop computer touch pad.

2 . Implementation

2.1. Physical realisation

To demonstrate the feasibility of the proposed approach, and for the sake of simplicity, a simulator running on a notebook computer was implemented. The notebook computer has a touchpad that can be used as an alternative pointing device. The touchpad was turned into a virtual ipod wheel by using an ipod wheel rubber guide. The rubber guide was obtained by taking a suitable piece (cut-out) from an ipod video rubber casing (see Figure 4). This rubber casing has a thin ridge along each side of the wheel guiding the finger along a circular track. This strategy gives the user a similar tactile feedback as using the ipod wheel both in terms of surface texture and physical dimensions.

2.2.Calibrating the wheel

It is necessary to calibrate the wheel as different devices may have different touch pad resolution and a motion results in different measurements. The user calibrates the system by rotating the wheel at least once. Then a set of measurements (x_1,y_1), (x_2,y_2), .., (x_n,y_n). The radius of the wheel is then:

$$r = \frac{(x_{\max} - x_{\min}) + (y_{\max} - y_{\min})}{4} \quad (1)$$

Here, x_{max}, y_{max}, x_{min} and y_{min} are the maximum and minimum values in the dataset, respectively. Moreover, the centre of the wheel is given by:

$$x_0 = x_{\min} + r$$
$$y_0 = y_{\min} + r \quad (2)$$

2.3. Updating the centre

The absolute position of the wheel centre (x_0,y_0) is shifted whenever a user starts a new uni-stroke from a new angle on the wheel. The strategy is based on the assumption that each uni-stroke is represented by the set of measurements (x_1,y_1), (x_2,y_2), .., (x_n,y_n). For each uni-stroke the wheel is contained within the bounding box defined by ($x_0 - r$,$y_0 - r$) and ($x_0 + r$,$y_0 + r$). Clearly, all measured coordinates (x_i,y_i) should fall inside this bounding box. To check whether a measurement is a wheel motion, i.e., that the motion follows the path of the circle, the following condition must be satisfied:

$$r_{\min}^2 \le (x_i - x_0)^2 + (y_i - y_0)^2 \le r_{\max}^2 \quad (3)$$

Where $r_{min} = r – r_{error}$, $r_{max} = r + r_{error}$ and r_{error} is the error margin or half the width of the wheel. This criterion is illustrated in Figure 5.

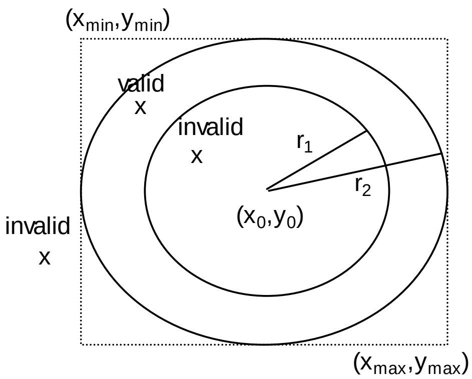


Figure 5. The valid path around the wheel.

The centre of the wheel is updated for each gesture as follows. First, if the set of measurements satisfy Eq. (4), then the uni-stroke comprises at least one full circular motion and the centre can be recalculated as for the calibration.

$$x_{\max} - x_{\min} \approx 2r$$
$$y_{\max} - y_{\min} \approx 2r \quad (4)$$

If this condition is not satisfied the uni-stroke is not a full circle and the centre is computed by first finding the midpoint between the start and the end point.

$$x_m = \frac{x_1 + x_n}{2}$$
$$y_m = \frac{y_1 + y_n}{2} \quad (5)$$

Next, the measurement i_{min} closest to and i_{max} the most far away from (x_m,y_m) is found. Note that the middle point on the curve is not necessarily in the middle – this situation is especially the case for uni-strokes with one or more turning points.

$$i_{\min} = \arg\left[\min_j\left((x_j - x_m)^2 + (y_j - y_m)^2\right)\right] \quad (6)$$

and

$$i_{\max} = \arg\left[\max_j\left((x_j - x_m)^2 + (y_j - y_m)^2\right)\right] \quad (7)$$

Then, if i_{min} = 0 or n then i_{max} is used, otherwise i_{min} is used.

The centre can then be computed.

$$x_0 = x_i + r\left(\frac{x_m - x_i}{d}\right)$$
$$y_0 = y_i + r\left(\frac{y_m - y_i}{d}\right) \quad (8)$$

Where d is the Euclidean distance between (x_m,x_m) and (x_i,y_i).

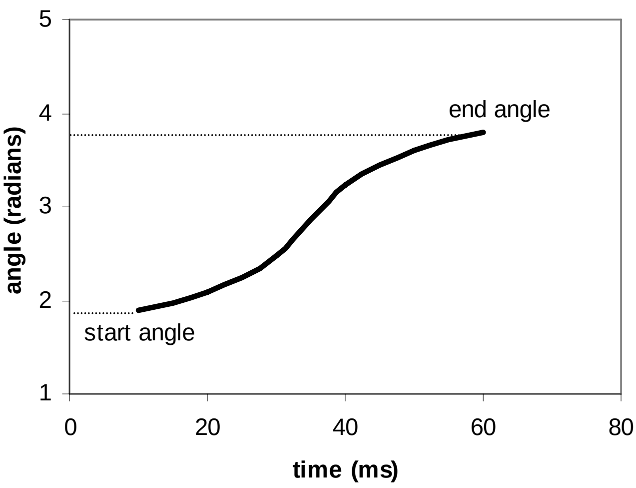


Figure 6. Detecting the motion of the wheel.

2.4. Detecting angles

It is straightforward to calculate the angle a_i of a position (x_i,y_i) given the centre of the wheel (x_0,y_0) as (see Figure 6)

$$a_i = \begin{cases} \arctan\left(\frac{y_i - y_0}{x_i - x_0}\right) & if \quad x_i \neq x_0 \\ \frac{\pi}{2} & if \quad y_0 - y_i > r \\ \frac{3\pi}{2} & if \quad y_i - y_0 > r \end{cases} \tag{9}$$

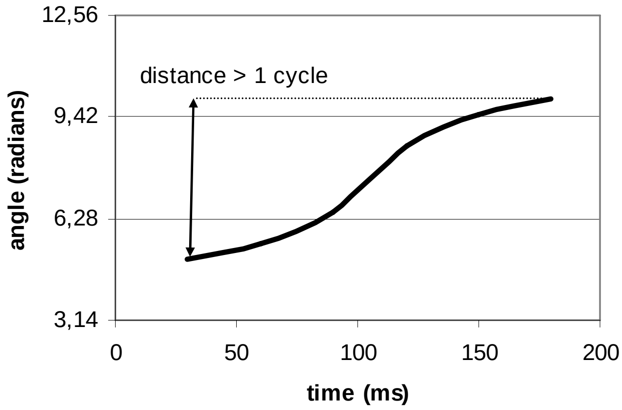


Figure 7. Measuring distance.

Distance travelled

To measure the total distance travelled in terms of rotations using the wheel, the total angular distance moved is used (see Figure 7). The angular distance d_{ij} from angle a_i to a_j, where $i<j$ and the angles are ordered in chronological order, can be defined as:

$$d_{ij} = \sum_{k=i+1}^{j} \cos^{-1}\left(\frac{(x_k - x_0)(x_{k-1} - x_0) + (y_k - y_0)(y_{k-1} - y_0)}{\sqrt{(x_k - x_0)^2 + (y_k - y_0)^2}\sqrt{(x_{k-1} - x_0)^2 + (y_{k-1} - y_0)^2}}\right) \tag{10}$$

Then, the distance travelled in terms of cycles c_{ij} is:

$$c_{ij} = \frac{d_{ij}}{2\pi} \tag{11}$$

2.5. 2.5 Detecting directional turning points

Some gestures involve a combination of clockwise and anticlockwise motions. Directional turning points are discovered by inspecting the sign of the first derivative of the angular path of motion (see Figure 8). Therefore, there is a directional turning point at a_i if

$$\mathrm{sign}(a_i - a_{i-1}) \neq \mathrm{sign}(a_{i+1} - a_i) \tag{12}$$

where $1<i<n$. However, small noise-perturbations are low-pass filtered using a threshold of 3, i.e., the shift in direction has to hold for at least three consecutive measurements for the shift to be recorded. This value was determined through experimentation. Furthermore, when the rotation crosses the zero degree mark, the above condition will trigger a false direction change. However, this false trigger is discarded by the low pass filter.

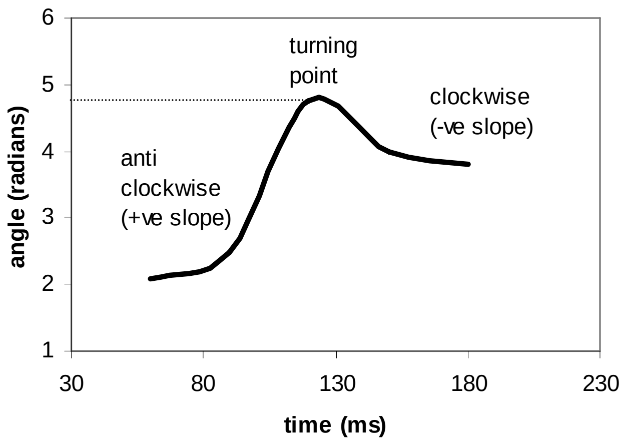


Figure 8. Detecting turning points.

Table 1. The characteristics of the gestures provided in Figure 3.

char	start pos	end pos	start direction	Turning points	Distance
A	W	E	clockwise		0.50
B	N	N	Clockwise	S	1.00
C	N	S	Anticlockwise		0.50
D	NW	S	Anticlockwise,	SW	1.25
E	E	S	Anticlockwise		0.75
F	SW	NE	Clockwise		0.50
G	N	S	Anticlockwise	E	1.00
H	NW	SE	Anticlockwise	SW	1.25
I	N	E	Clockwise		0.25
J	E	S	Clockwise		0.25
K	SW	SE	Anticlockwise	SE, SW	2.00
L	W	S	Anticlockwise		0.50
M	NW	NW	Anticlockwise	SW, SE	1.75
N	NW	SE	Anticlockwise	SW	1.25
O	N	N	Anticlockwise		1.00
P	NW	SW	Anticlockwise		1.40
Q	N	S	Anticlockwise	N	1.40
R	W	N	Clockwise		1.00
S	N	N	Anticlockwise	S	0.80
T	SW	SW	Anticlockwise	SE	2.00
U	W	SE	Anticlockwise	E	0.75
V	W	E	Anticlockwise		0.50
W	W	W	Anticlockwise	E	1.00
X	SW	SE	Anticlockwise		1.30
Y	W	S	Anticlockwise	E	0.75
Z	N	N	Clockwise	S	1.20
_	SW	SE	Anticlockwise		0.25
.	S	S	Anticlockwise		1.00

2.6. Circular gesture recognition

The characteristics of the gestures outlined in Figure 1 are listed in Table 1. The features listed are start position s_i, end position e_i, start direction r_i, total distance travelled d_i, number of turning points t_i, turning point positions $t1_i$ and $t2_i$. Given a set of input measurements in the form of start-angle *s*, end-angle *e*, start-direction *r*, total-distance *d*, number of turning points *t*, and turning point positions *t1* and *t2*, the corresponding character is found by selecting the character that has the most similar characteristics. The measure of similarity used herein is:

$$\begin{aligned} f(s,e,d,r,t,t1,t2,i) \;=\; & \left(\frac{s-s_i}{2\pi}\right)^2 + \left(\frac{e-e_i}{2\pi}\right)^2 \\ & + \left(d-d_i\right)^2 \\ & + D(r,r_i) + 100(t-t_i)^2 \\ & + \left(\frac{t1-t1_i}{2\pi}\right)^2 + \left(\frac{t2-t2_i}{2\pi}\right) \end{aligned} \quad (13)$$

where

$$D(x_1,x_2) = \begin{cases} 1000 & if \quad x_1 \neq x_2 \\ 0 & if \quad x_1 = x_2 \end{cases} \quad (14)$$

3. Conclusions

A text input strategy for wheel controlled devices such as the ipod or virtual wheel controls on smart home appliances was presented. The strategy is based on circular uni-stroke gestures. Spatial mnemonics are used to reduce the learning time and effort. A distance based strategy is used for robust pattern matching.

4. References

[1] M. Ingmarsson, D. Dinka, and S. Zhai, "TNT: a numeric keypad based text input method," in *Proc. SIGCHI conference on Human factors in computing systems*, Vienna, Austria, ACM Press, 2004.

[2] I. S. MacKenzie, "Mobile Text entry using three keys," in *Proc. NordCHI'02*, ACM Press, 2002, pp. 27-34.

[3] K. Lyons, T. Starner, D. Plaisted, J. Fusia, A. Lyons, A. Drew, and E. W. Looney, "Twiddler typing: one-handed chording text entry for mobile phones," in *Proc. CHI 2004*, ACM Press, 2004, pp. 81-88.

[4] S. Zhai, M. Hunter, and B. A. Smith, "Performance optimization of virtual keyboards," *Human Computer Interaction*, vol. 17, no. 2, pp. 229-269, 2002.

[5] D. Goldberg and C. Richardson, "Touch typing with a stylus," in *Proc. ACM Interchi'93*, ACM Press, 1993, pp. 80-87.

[6] I. S. MacKenzie, R. B. Nonnecke, S. Riddersma, C. McQueen, and M. Meltz, "Alphanumeric entry on pen-based computers," *International Journal of Human-Computer Studies*, vol. 41, pp. 775-792, 1994.

[7] J. O. Wobbrock, B. A. Myers, H. H. Aung, and E. F. LoPresti, "Text entry from power wheelchairs: edgewrite for joysticks and touchpads," in *Proc. ASETS'04*, Atlanta, Georgia, ACM, 2004, pp. 110-117.

[8] K. Partridge, S. Chatterjee, and R. Want, "TiltTYpe: Accelerometer-supported text entry for very small devices," in *Proc. CHI'01*, ACM Press, 2001, pp. 201-204.

[9] R. W. Soukoreff and I. S. MacKenzie, "Metrics for text entry research: An evaluation of MSD and SPC, and a new unified error metric," in *Proc. CHI'2003*, ACM Press, 2003, pp. 113-120.

[10] J. Lumsden and A. Gammell, "Mobile note taking: investigating the efficacy of mobile text entry," *Springer Lecture Notes on Computer Science LNCS*, vol. 3160, pp. 156-167, 2004.

[11] A. Serwer, "It's iPod's revolution: We just live in it," *Fortune*, vol. 151, no. 13, pp. 43-43, 2005.

[12] P. Marks, "Fashion industry covets 'iPod factor'," *New Scientist*, vol. 186, no. 2502, pp. 26-26, 2005.

[13] B. Dipert, "Striking back against the iPod empire," *EDN*, vol. 50, no. 12, pp. 52-52, 2005.

[14] "IpodLinux," 2006.

[15] B. Hayes, "Using an iPod in Linux," *Linux Journal*, vol. 2005, no. 135, pp. 8, 2005.

[16] J. Bettencourt, "Text Input Methods," 2006. (???)

[17] D. J. Ward, A. F. Blackwell, and D. J. C. MacKay, "Dasher - a data entry interface using continuous gestures and language models," in *Proc. UIST 2000*, ACM Press, 2000, pp. 129-137.

[18] F. E. Sandnes, "Evaluating mobile text entry strategies with finite state automata," in *Proc. Mobile HCI'05*, Salzburg, Austria, ACM Press, 2005, pp. (in press).

[19] P. Isokoski, "Model for unistroke writing time," in *Proc. The SIGCHI Conference on Human Factors in Computing Systems*, Seattle, Washington, United States, ACM Press, 2001, pp. 357-364.

[20] I. S. MacKenzie and S. X. Zhang, "The immediate usability of graffiti," in *Proc. The Conference on Graphics interface '97*, Kelowna, British Columbia, Canada, 1997, pp. **129-137**.

[21] J. Mankoff and G. D. Abowd, "Cirrin: a word-level unistroke keyboard for pen input," in *Proc. The 11th Annual ACM Symposium on User Interface Software and Technology*, San Francisco, California, United States, ACM Press, 1998, pp. 213 - 214.

[22] P. O. Kristensson and S. Zhai, "Gestures: SHARK2: a large vocabulary shorthand writing system for pen-based computers," in *Proc. The 17th Annual ACM Symposium on User Interface Software and Technology*, Santa Fe, NM, USA, ACM Press, 2004, pp. 43-52.

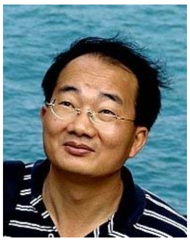
[23] F. E. Sandnes and Y.-P. Huang, "Chording with spatial mnemonics: automatic error correction for eyes-free text entry," *Journal of Information Science and Engineering*, vol. 22, no. 5, pp. 1015-1031, 2006.

[24] P. U.-J. Lee and S. Zhai, "Top-down learning strategies: can they facilitate stylus keyboard learning?," *International Journal of Human Computer Studies*, vol. 60, pp. 585-598, 2004.

Authors

Frode Eika Sandnes

received a B.Sc. in computing science from the University of Newcastle Upon Tyne, England, and a Ph.D. in computer science from the University of Reading, England. He has several years of experience from the space industry developing communications and onboard systems for low-earth orbit environmental satellites. He is currently a professor in the Department of Computer Science at Oslo University College, Norway. Dr. Sandnes' research interests include multiprocessor scheduling, error correction and mobile human computer interaction.



Yo-Ping Huang

received a Ph.D in electrical engineering from Texas Tech University in 1992. Dr Huang is currently a professor in the department of computer science and engineering at Tatung University, Taiwan. His research interests include intelligent information retrieval, data mining, artificial intelligence and application systems for handheld devices.

INTENTIONAL BLANK