A system for tracking braille readers using a Wii Remote and a refreshable braille display

Inthraporn Aranyanak & Ronan G. Reilly

Behavior Research Methods

e-ISSN 1554-3528

Behav Res
DOI 10.3758/s13428-012-0235-8

ONLINE FIRST

Behavior Research Methods

VOLUME 42, NUMBER 2 ■ MAY 2010

BRM

EDITOR
Gregory Francis, *Purdue University*

ASSOCIATE EDITORS
Ira H. Bernstein, *University of Texas Southwest Medical Center*
Mark W. Greenlee, *University of Regensburg*

A PSYCHONOMIC SOCIETY PUBLICATION
www.psychonomic.org
ISSN 1554-351X

Springer

Behav Res
DOI 10.3758/s13428-012-0235-8

A system for tracking braille readers using a Wii Remote and a refreshable braille display

Inthraporn Aranyanak ·
Ronan G. Reilly

Abstract This article describes a cheap and easy-to-use finger-tracking system for studying braille reading. It provides improved spatial and temporal resolution over the current available solutions and can be used with either a refreshable braille display or braille-embossed paper. In conjunction with a refreshable braille display, the tracking system has the unique capacity to implement display-change paradigms derived from sighted reading research. This will allow researchers to probe skilled braille reading in significantly more depth than has heretofore been possible.

Keywords Braille · Hand tracking · Reading · Blind · Wiimote

Sighted reading has been investigated scientifically for well over 100 years (e.g., Huey, 1968). In the last 30 years or so, the precision with which sighted reading could be studied, at least at the perceptual level, has undergone a revolutionary change with the development of modern eyetracking technology (Rayner, 1998). Research on tactile reading was also first conducted around the turn of the last century (Bürklen, 1917/1932). The dominant tactile writing system has, for nearly two centuries, been braille. However, technological developments have not made a significant impact on the study of braille reading. Much of the research over the last 30 years has entailed the laborious analysis of video recordings, collected typically at around 25 frames per second (FPS). In general, the sampling rates of most of the video recording techniques used to study braille reading have been too low to capture any subtle alterations in reading speed that might, for example, reflect online language processing.

Therefore, to study braille reading with close to the sampling rate of some of the low-end modern eyetrackers (e.g., the Tobii X60 and X120 from Tobii Technologies AB, Danderyd, Sweden), we designed a high-resolution finger-tracking system utilizing affordable components, yet providing high temporal and spatial resolution of finger movements. The aim of this article is to describe this new hand-tracking system in detail and to highlight some of the information about braille reading behavior that we have obtained by using it.

The braille writing system

The braille writing system was invented around the 1820 s by Louis Braille. Standard braille consists of sets of cells, each allowing for the display of 3 rows × 2 columns of raised dots. The height of a dot is approximately 0.5 mm; the space between dots is approximately 2.5 mm; and the space between cells is approximately 3.75 mm. From cell to cell, the maximum width is 6.25 mm.

The writing system is divided into three levels: (1) Grade 1 braille, in which the characters stand for normal print letters on a one-to-one basis; (2) Grade 2 braille, which is a terser form involving a systematic set of contractions to reduce the lengths of words; (3) Grade 3 braille, which is a comprehensive shorthand version. This level has more

---

This report was completed while the first author was a Visiting Fellow and the second author a Visiting Professor at the MARCS Institute, University of Western Sydney.

Electronic supplementary material The online version of this article (doi:10.3758/s13428-012-0235-8) contains supplementary material, which is available to authorized users.

I. Aranyanak · R. G. Reilly (✉)
Department of Computer Science,
National University of Ireland, Maynooth,
Maynooth, Co. Kildare, Ireland
e-mail: ronan.reilly@nuim.ie

abbreviated words and is more challenging to learn than Grade 2. All braille beginners start learning Grade 1 and then progress to Grade 2. There is also an extension of the encoding from six dots (64 possible combinations) to eight (256 possible combinations) that was initially developed for special purposes, such as music notation and a special shorthand code used by blind stenographers in Austria and Germany. More recently, eight-dot braille has been used in conjunction with computers, where the extra dots are used to identify highlighted items and the position of the computer cursor, and so forth.

Braille tracking devices

The first attempts at recording the hand movements of braille readers were described in Bürklen (1917/1932). The measurement device, referred to as a "Tastschreiber" or touch writer, was quite crude and consisted of a pen-like instrument clamped to the braille reader's finger, the tip of which scratched a trace of the hand movements on smoked paper. The Tastschreiber was reportedly uncomfortable to wear and would quite likely have interfered with fluent reading.

In more recent times, Mousty and Bertelson (1985) used two video cameras to record braille readers' hand movements. The frame rate for both cameras was 25 FPS. The braille text to be read was supported on a tablet. One of the cameras was placed above the tablet, so that the left-to-right hand movements could be recorded. The other one was set up at the same level as the tablet facing the reader, in order to record finger movements. A digital clock with a resolution of 100 ms was mounted on the tablet, and its output was recorded by the cameras while subjects read.

A higher-resolution recording system was used by Bertelson and his colleagues in the early 1990s (Bertelson, Mousty, & Radeau, 1992; Mousty & Bertelson, 1992). The tracking component had originally been developed by Noblet, Ridelaire, and Sylin (1985). Although Noblet et al.'s tracking solution was arguably the best available until recently, it required considerable expertise in the area of circuit fabrication and video technology to construct a workable device. In Bertelson et al.'s use of the Noblet et al. tracker, the braille text was presented using an Active Line braille display controlled by an Apple II microcomputer. The recording device consisted of a solid-state video camera (Hitachi KP120U) located above the display. A small light-emitting diode (LED) attached to the nail of the reading finger flashed every 40 ms. The location of each flash was recorded as *x*–*y* coordinates by the camera, allowing finger movements to be captured with reasonable accuracy.

Millar (1997) described a videotaping technique to collect data from braille readers that involved a transparent surface (66 × 30 cm) supporting a transparent sheet of plastic embossed with braille text. The readers' hands were recorded from below. The video camera and a monitor were connected to each other. The monitor could display text and hand movements in real time. In each frame, the cumulative time, at 10-ms resolution, was digitally displayed on the monitor. The data were collected at a rate of 40 FPS and analyzed by hand, frame by frame (Millar, 1997).

Breidegard et al. (2008) introduced an automatic finger-tracking system (AFTS) for identifying finger location in video recordings of braille readers. Previous users of video-based systems (e.g., Millar, 1997; Mousty & Bertelsen, 1985) had had to analyze each video frame manually to determine finger positions. The AFTS used an algorithm based on template matching and filtering to detect and track the readers' fingertips. The system had two cameras. The first camera was placed underneath a semitransparent braille sheet mounted on a transparent glass plate. Its spatial resolution was 768 × 576 pixels, and the resolution of a second camera, located at an angle above the fingers, was 320 × 240 pixels. The frame rates of both cameras were 25 FPS.

Hughes (2011) constructed a device for examining the movement kinematics of braille readers. The equipment comprised a digitizing tablet about 30 cm square with a spatial resolution of 0.1 mm. The tip of a digitizing pen was attached to the end of the dominant reading fingertip and connected to the pen's electronics, which were mounted separately on the reader's forearm. The tip of the pen was 3–10 mm distant from the reading finger's center, and the reading surface was not touched by the pen. The sampling rate of the system was 100 Hz.

While Hughes's (2011) system is innovative in its use of hardware and has a good sampling rate, it appears to be limited to tracking just one finger. For example, in our study, we found that single-handed readers, some of whom used more than one finger, accounted for eight out of our sample of 20, with the remainder utilizing various styles of two-handed reading. In addition, the arrangement of the pen tip at the end of the reading finger precludes the use of certain finger positions favored by some readers. For example, we found one reader who used the very tips of the fingers rather than the bulbous pads. Also, some readers we have studied prefer to angle their fingertips several degrees off vertical, where vertical is with respect to the orientation of the braille characters. This is presumably to increase the fingertip surface area in contact with the text. However, Hughes's pen arrangement would be likely to give erroneous cell location information, in this case. Correcting for any offset in location would be complicated by the tendency for readers to dynamically alter their finger orientation as they sweep across the cells.

A Wii remote–based tracking system

As can be gathered from the foregoing brief review, modern recording techniques have been based on either the manual or automatic analysis of video recordings or from the active generation of signals by LEDs (or other devices) attached to readers' fingers. The latter approach is more suitable for a precise, in-depth analysis of braille reading. However, until recently, signal generation approaches have had the disadvantage of either requiring custom-built hardware (e.g., Noblet et al., 1985) or limiting the number of hands tracked and the mode of reading (Hughes, 2011).

Our aim in building the system described in this article was to overcome these limitations and to create a high-resolution finger-tracking system approximating the temporal and spatial resolution of modern eyetrackers, but constructed from off-the-shelf components. For example, entry-level eyetracking solutions start with a sampling rate of 60 Hz (e.g., the ASL H6 from Applied Science Laboratories, or the Tobii x60 from Tobii Technology AB). At the top end, eyetrackers of course can sample at much higher rates. For example, the EyeLink 1000 from SR Research Ltd can sample from one eye at 2000 Hz. A sampling rate of 100 Hz is acceptable for a braille tracking application, given that hand movements are significantly slower than saccadic eye movements.

We opted for a tracking solution similar to that of Noblet et al. (1985) by using trackable infrared LEDs with a high-resolution camera. However, instead of having to construct an LED tracking camera from scratch, as they did, we took advantage of developments in the gaming industry and used a relatively cheap, ready-made input device in the form of Nintendo's Wii Remote (or Wiimote, to use its more popular designation). The Wiimote is a new-generation input device for Nintendo gaming software and is relatively cheap, at less than €40. From the perspective of our application, its most appealing feature, apart from cost, is its built-in infrared camera, which can calculate the positions of up to four infrared light sources. The camera sensor provides location data with a resolution of 1,024×768 pixels at a 100 Hz sampling rate. The sampling rate is greater than a standard webcam, which is typically in the 25–40 FPS range, and is comparable to the entry-level eyetrackers discussed.

Our tracking system comprises a refreshable braille display (Handy Tech Elektronik GmbH, Horb-Nordstetten, Germany), a Wiimote, four infrared LEDs, and a laptop that integrates the components, provides an easy-to-use interface, and logs the reading data. Hand tracking is accomplished by attaching an infrared LED to one or two of a subject's reading fingers and using feedback from the Wiimote's infrared camera to determine the locations of the two light sources. The vast majority of readers use at least one of their index fingers for reading. In our sample of 20 readers, all of them used at least their index fingers, although a small number of readers used another finger in addition to their index. Note that there is no reason, in principle, why one could not track the positions of more than two fingers. Our code (supplied as a supplement to this article) could be readily modified to do so. However, careful consideration of the fingers that multifingered one-handed readers use should allow for the determination of which finger to track. In the case of someone using two or more, the rightmost finger is the most reasonable one to track. Thus, even when one-handed readers use multiple fingers, in principle we could recover the positions of the other fingers from a single tracking LED on the reading hand.

To display braille characters under computer control, we used the Easy Braille refreshable display (version 3.1). This is an electromechanical device that comprises 40 braille cells (6.5 mm wide, cell to cell, for a six-dot cell). The dimensions of the display are 30.5 cm wide by 9.0 cm deep by 2.9 cm high. To display the braille dot patterns, small rounded pins are raised and lowered through holes in the cell surface. However, like the majority of refreshable braille displays currently on the market, the Easy Braille display is only able to present text one line at a time. Nonetheless, the big advantage of this type of display is its potential for dynamically altering text contingent on finger position. Why this is important will be discussed in a later section.

We should add that the system described here could be readily adapted for use with a multiline refreshable display. Moreover, we have also implemented a paper mode in which our system can study multiline braille reading in its more typical form. As already mentioned, the main point of integrating a refreshable display with our tracker was the potential offered for more powerful experimental paradigms than are possible with currently available systems.

Equipment configuration

Figure 1 is a schematic representation of how each of the hardware components is integrated into the tracking system. Text is presented using the refreshable braille display. The fingers to be monitored are tagged with infrared LEDs and tracked by a Wiimote. A control program running on a laptop is used to read the finger coordinates from the Wiimote camera and to control the text displayed. The system described supports the tracking of one or two reading fingers, depending on the reader's reading style.

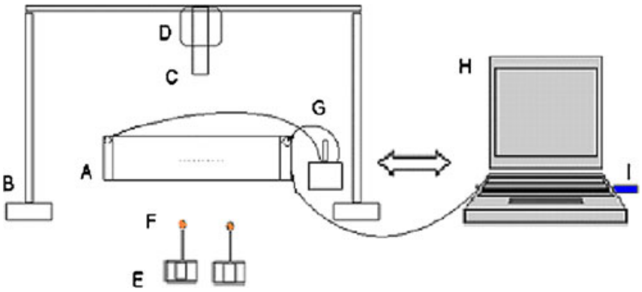


Fig. 1 Schematic representation of the tracking system: **A** refreshable braille display, **B** retort stand for supporting cameras, **C** Wiimote, **D** video camera, **E** elastic wrist straps, **F** light-emitting diodes (LED), **G** switch box, **H** laptop, and **I** Bluetooth receiver

The tracker components are configured as follows. First, a horizontal bar approximately 50 cm in length is clamped to the top of two retort stands (B in Fig. 1) at approximately 45 cm from the table surface. The braille display is placed exactly between the two stands. The Wiimote is clamped to the middle of the horizontal bar, with the infrared camera aimed at the braille display. The height above the surface of the braille display is approximately 40 cm. It is important to ensure that the lens of the Wiimote's camera is parallel to the surface; otherwise, the mapping between braille cells and camera pixels can be uneven across the camera's input range. A video camera is also attached by a clamp to the horizontal bar and pointed toward the braille display. It was found useful to have a complementary video recording of the readers' performance to record any idiosyncratic reading styles that the subjects might have used.

Two LEDs are attached to the braille display just to the left of the first cell and the right of the last cell, in order to facilitate calibration. They are linked to a switch box, which allows them to be switched on one at a time during the calibration process and switched off during tracking. The braille display and the laptop are connected via a USB cable, while communication between the Wiimote and the laptop is via a Bluetooth connection. We used a standard infrared LED with a round cross-section of 3 mm, 850 nm peak wavelength, and an emission angle of ±13º.

The infrared camera maps the light sources onto a 1,024 × 768 planar grid. Given the configuration of the Wiimote and display described above, a single 6.25-mm-wide braille cell subtends approximately 20 pixels. In theory, 1,310 braille cells can fit into a 1,024 × 768 grid. However, if the braille display is placed near the boundary of the grid, tracking accuracy declines. We have determined that the optimal location for the display is between 100 and 900 pixels on the *x*-axis and centered around pixel 380 on the *y*-axis.

To track readers using braille paper, the physical configuration of the system is the same as for the refreshable display. The paper should be anchored to the table with tape and oriented square-on to the Wiimote camera. It will help to mark out the location of the page on the table in order to maintain calibration consistency. The main difference when using paper is that calibration requires three points rather than two: (1) just above the first line at the leftmost cell, (2) just above the first line at the rightmost possible cell position, and (3) just below the last line at the first cell. For Points 1 and 2, we use the LEDs controlled by the switch box (see Fig. 1), and for Point 3 we use an LED attached to the end of a wand. These LEDs are switched on sequentially by the experimenter under direction from the calibration routine.

There are two common sizes for printed braille pages 8.5 × 11 inches, which can be embossed with up to 34 characters per line and 25 lines per page, and 11.5 × 11 inches, which can be embossed with up to 40 characters per line and 25 lines per page. The number of characters per line and lines per page are adjustable in the program in order to reflect this variation. The optimal region for positioning the braille page is between 200 and 825 pixels along the *x*-axis and between 45 and 740 pixels on the *y*-axis. Therefore, the three calibration points need to be inside the rectangle defined by this boundary.

The limitations of using braille paper are that the camera can track braille precisely only for 23 lines per page (just short of the standard number per page), but can track up to 40 cells per line. Furthermore, the system needs calibration every time there is a new page. Therefore, data collection is a little more time-consuming in paper mode than in display mode.

Data collection methodology

During reading, subjects are asked to place elastic straps on their wrists that carry the batteries to power the LEDs (see Fig. 2). The LEDs themselves were generally placed at the center of a reader's index fingernails using Blu-Tack (a putty-like glue). Care should be taken so that the LEDs are placed along the vertical axis of the nail so as to get accurate cell location readings, particularly for readers who read with their fingertips, causing the nail to be held almost vertically to the plane of the display. Because of the nipple-like shape of the LEDs (see Fig. 2), the tracker can cope well with these variations in finger orientation. Note that the camera must

Fig. 2 Photograph of a reader with LEDs attached to the index fingers and batteries attached to the wrists

have an unobstructed view of the LEDs for accurate recording. However, the system is tolerant of transient loss of signal.

Subjects are first asked to read practice sentences from the display in order to confirm the configurations of hands and fingers that they prefer. The results of the practice trials are then used to determine how many fingers to track (one or two) and which is the dominant braille reading finger. As already mentioned, in some cases readers will use more than just their index finger on one hand. In one particular case, we had a reader use the three central fingertips of their right hand. In this case, we tracked the ring finger. In general, if more than one finger on the one hand is involved, we will track the leading, rightmost finger.

Before starting the actual experiment, a calibration validation line is presented on the braille display, consisting of the letters ABC at the beginning and CBA at the end of the line. The participants are asked to place their index fingers on the letters A at either end and then to sequentially move the fingers to B and to C. We then adjust the LEDs if the fingers are not registered as being on the correct cells, as indicated by the user interface. We usually validate calibration every ten sentences and recalibrate when necessary. We generally find calibration to be stable and only in need of adjustment if the readers have accidentally nudged the display or if one of the LEDs has come loose. However, in order to verify the stability of the calibration, we ran a series of calibration checks every minute over a 30-min period and found that the reported positions of two static LEDs shifted for the last two measurements by just one pixel. Since calibration accuracy is checked every 10 sentences during experiments (approximately every 3 min) and one braille cell subtends approximately 20 pixels, we consider this level of calibration drift unlikely to cause problems in tracking accuracy.

In the case of paper reading, an ABC–CBA validation line is printed on the first and last lines of the page. In this case, validation of calibration is done at the beginning and end of each page reading.

An important methodological consideration is the degree to which the recording apparatus interferes with natural reading. We asked participants in a pilot study of our system to rate on a 5-point scale the impact on their reading of wearing the LEDs and batteries (1 = *no impact*, 5 = *significant impact*). Out of the sample of 20, 70 % gave a response of 1 (*no impact*), while the remainder gave a response of 2 (*some impact*).

Tracker control program

The code controlling the tracker was written in Visual C# running on Microsoft's .NET platform. This system was constructed using Microsoft's Visual Studio 2005. The system uses two main external libraries: (1) a library to read LED source coordinates from the Wiimote (Peek, 2008) and (2) a library for writing to the braille display (Handy Tech Electronik GmBH, Horb-Nordstetten, Germany).

Listings 1 and 2 are examples of the C# code used to control the tracker. They illustrate methods involved in the calibration of the Wiimote (Listing 1) and in the initialization and output of text to the braille display (Listing 2).

We developed a simple user interface for configuring, calibrating, and recording reading data. The flow of control of the tracking software is driven by a simple forms interface. There are two versions of the program, one for use with the refreshable display, and one for use with braille-embossed paper.

When either version of the program starts up, the first form presented (Fig. 3) gathers data about the subject: his or her name, the experimental condition to be received, the dominant reading hand, and so on. This is then followed by a calibration window (Fig. 4), which facilitates the calibration of the braille display with respect to the camera and allows hand positions to be associated with specific cells in the display.

For the paper version, the calibration process is a little more complicated, involving the use of three landmark points (see Fig. 5).

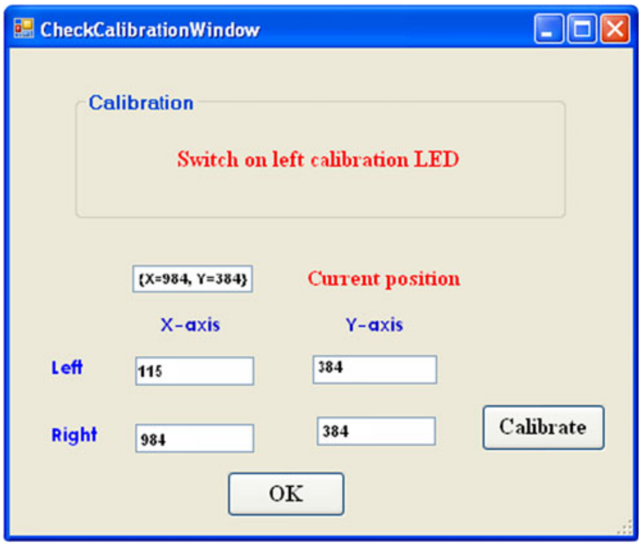
Once calibration is complete, the experimenter makes the choice of reading mode. The control program supports the tracking of one or two fingers. The form in Fig. 6 is used to indicate the reader's preferred reading mode.

Finally, the system is ready to record data. Figure 7 shows the line of text that is being read. The braille display can only present a single line at a time, but sentences can be presented as a series of single lines, where the reader steps through the lines using a space bar.

Subject Name: Lily
File Name: S1-40
Condition: Easy question
Handedness: Right

Fig. 3 Form interface for recording subject and experimental information. This includes the subject's name, the file name in which the recorded data will be stored, the experimental condition under which the data are being collected, and the dominant reading hand of the subject

In Fig. 7, the ASCII codes corresponding to the braille translation of the text are displayed. One ASCII code corresponds to one braille cell. However, there is not a one-to-one mapping between these codes and regular text, because the braille encoding includes various contractions for common letter sequences (e.g., in Fig. 7, the character pair “,:” represents the word “Which,” where the comma indicates capitalisation and the colon is a contraction for the word “which”). This form also allows the experimenter to progress to the next line of text by means of the Next Line button. However, the usual mode of progression is for the reader to press a bar on the braille display. The Calibrate button on this form allows the system to be recalibrated if the experimenter detects some variation in tracking accuracy. When reading is complete, the Exit button is clicked in order to close the application and save the recorded data to file.

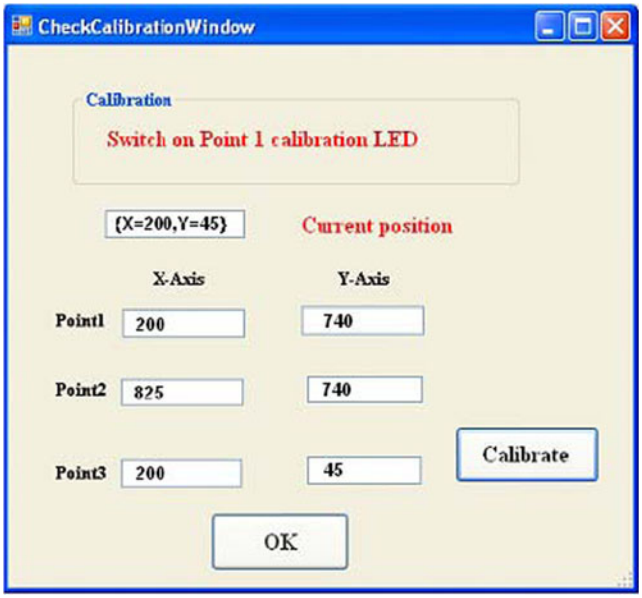


Switch on left calibration LED

{X=984, Y=384} Current position

	X-axis	Y-axis
Left	115	384
Right	984	384

Fig. 4 Calibration form for refreshable display mode. The top left box displays the *x*–*y* coordinates of the current position of the LED. The fields labeled “Left” and “Right” contain the *x*–*y* coordinates of the left and right calibration points at either end of the cell array on the braille display. These points are set using the Calibrate button on the form



Switch on Point 1 calibration LED

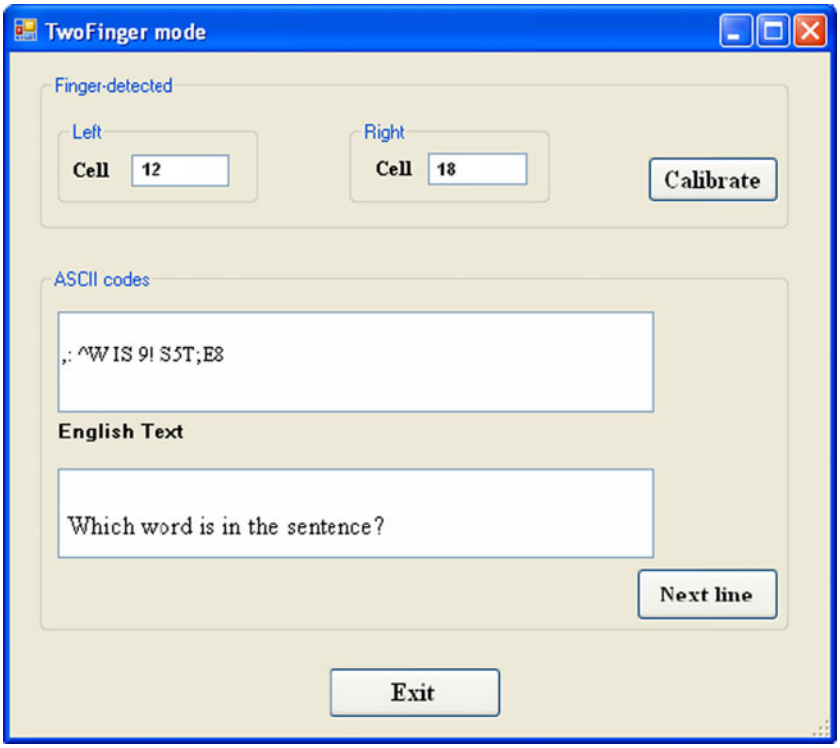
{X=200,Y=45} Current position

	X-Axis	Y-Axis
Point1	200	740
Point2	825	740
Point3	200	45

Fig. 5 Calibration form for braille paper mode. The top left box displays the *x*–*y* coordinates of the current position of the LED. The fields labeled “Point1” and “Point2” contain the *x*–*y* coordinates of the left and right calibration points at either end of the first line on a braille page. Point 3 contains the *x*–*y* coordinates of the left calibration point below the first character of the last line on the page. The points are registered in turn by using the Calibrate button on the form

○ 1 Finger Mode
● 2 Finger Mode

Fig. 6 Control program that supports the tracking of one or two index fingers, depending on the reader's style



Left Cell 12

Right Cell 18

ASCII codes: ,: ^W IS 9! S5T;E8

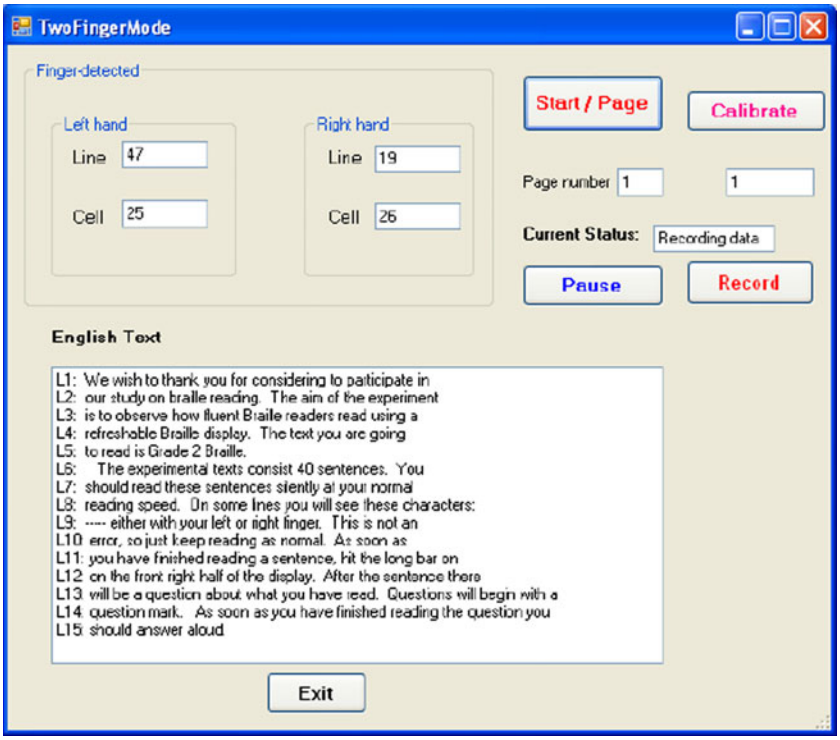
English Text: Which word is in the sentence?

Fig. 7 Display of the text line currently being read using the refreshable display in two-finger mode. The one-finger form has a similar layout. The ASCII codes in the box are the Grade 2 braille encodings of the text and include various contractions for common letter sequences

Figure 8 is the corresponding form for the paper mode of the program. Note that the interface provides dynamic feedback about both the line and cell number for each hand tracked.

The output data

The output data file records, in ASCII format, both subject details and the position of the finger(s) every 10 ms.



Left hand: Line 47, Cell 25

Right hand: Line 19, Cell 26

Page number 1 1

Current Status: Recording data

English Text

L1: We wish to thank you for considering to participate in
L2: our study on braille reading. The aim of the experiment
L3: is to observe how fluent Braile readers read using a
L4: refreshable Braille display. The text you are going
L5: to read is Grade 2 Braille.
L6: The experimental texts consist 40 sentences. You
L7: should read these sentences silently at your normal
L8: reading speed. On some Ines you will see these characters:
L9: ---- either with your left or right finger. This is not an
L10 error, so just keep reading as normal. As soon as
L11: you have finished reading a sentence, hit the long bar on
L12 on the front right half of the display. After the sentence there
L13 will be a question about what you have read. Questions will begin with a
L14 question mark. As soon as you have finished reading the question you
L15 should answer aloud

Fig. 8 Form that shows cells and lines being read in braille and paper mode by the left hand and the right hand. The English text is shown in the textbox. The Calibrate button initiates the calibration sequence, and the Start/Page button is for starting to record the cell and line numbers in an output file

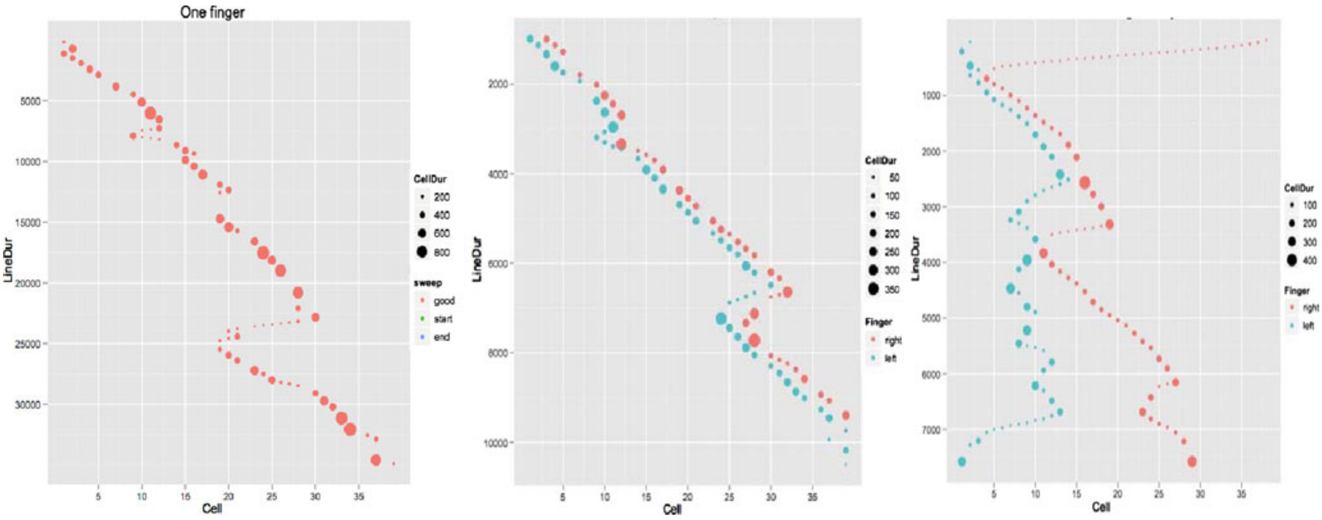


Fig. 9 Examples of three styles of braille reading: (left) single-handed, (middle) two-handed conjoint, and (right) two-handed disjoint. The x-axis represents cell location, and the y-axis represents cumulative time, going from top to bottom. Note that the sizes of the dots indicate transit time through the braille cell

Following an initial header giving subject details, each line in the file represents the location of one or both fingers at 10-ms intervals. If two fingers are being tracked, the output will consist of two sets of location information. In the case of paper mode, this location information includes both line and cell numbers.

Software is provided with the system to aggregate the raw output time spent transiting through each cell. These data can

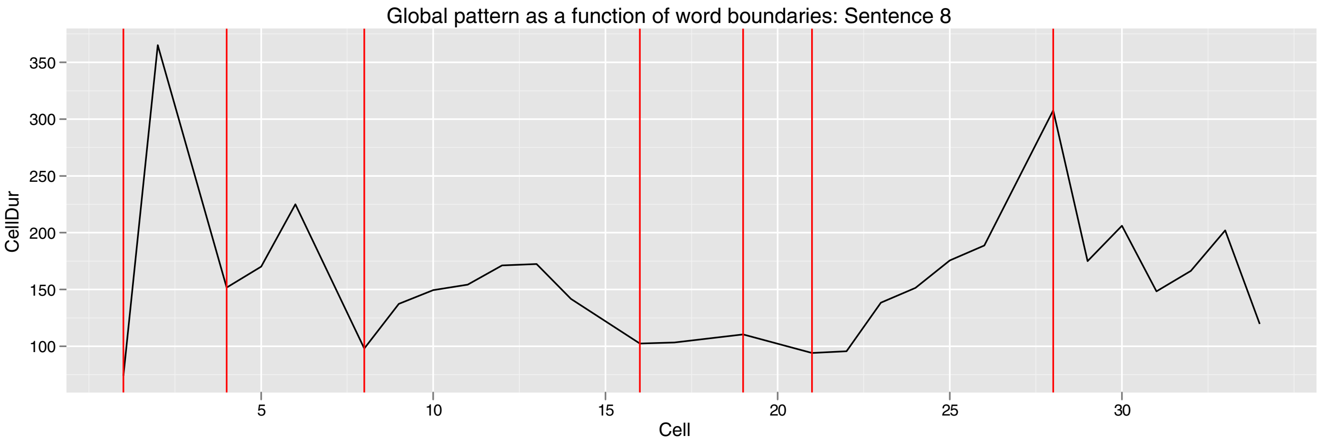


Fig. 10 Averaged reading times for a single pass of eight one-handed readers for the sentence "The new curtains had the smoothest texture", which maps to the ASCII coding sequence ",! NEW CURTA9S _H ! SMOO!/ TEXTURE", where each character represents a six-dot pattern. Note that "The" is represented by a single-cell contraction, and that at the beginning of the sentence it is preceded by a comma to denote capitalization. Note also the inverted-U shape for some of the longer words

then be processed with a standard statistical analysis package. It is, of course, possible to analyze the raw data in other ways, such as from a kinematics perspective (Hughes, 2011). However, software for this type of analysis is not provided with our system.

Display change

We believe an especially useful feature of our system is its capability to make changes to the displayed text contingent on the position of one or both fingers. For example, the tracker control program implements a boundary paradigm similar to that commonly used in sighted reading (Rayner, 1975). This could be useful, for example, for the study of which of several fingers is the most sensitive in readers who use more than one finger on a single hand. The latter style tends to occur among fast one-handed readers.

Another application of the display change capacity might be to provide different inputs to the left and right hands, in the case of two-handed readers. This would be especially helpful in answering questions about the division of labor between the hands of two-handed readers. For example, do the hands process the text in a complementary way, or is one hand more dominant? One possibility is that processing effort is spread over both hands and that the allocation of effort alters dynamically in response to the changing demands of the reading task. Another scenario is that the roles of the hands are relatively fixed throughout the reading process. One way to explore this issue would be to split the braille input stream across the two hands so that each hand only picked up, say, every alternate word. We can achieve this by dynamically masking words depending on whether they are being read by one or the other hand. This forces the reader to integrate inputs from both hands to make sense of the text. By comparing the impact that masking has as compared to a nonmasked condition, we can gauge the relative contribution of each hand. Furthermore, by altering the task demands on the reader by varying the text difficulty, we can probe whether the distribution of processing across hands is static or dynamically dependent on the task demands.

The display change feature is only feasible due to the relatively high-speed infrared camera of the Wiimote in conjunction with the real-time control program that supports the integration of the camera and the braille display. We believe that ours is currently the only tool available for answering, in a relatively direct way, questions concerning the relative roles of multiple fingers in braille reading.

System evaluation

An evaluation of the system was carried out before conducting more large-scale experiments, the results of which have already been alluded to in earlier parts of this article. Note that the focus of the evaluation was on the refreshable display system rather than on the paper one.

A group of 20 Grade 2 braille readers were recruited to participate in the study, ten male and ten female. Their ages ranged from 17 to 70 years, with the average being around 38. Sixteen of the subjects had been blind from birth, and four were late blind. None of them had any residual vision. The subjects were asked to read 80 English sentences and to answer questions about their content.

The hand movement patterns that we found accorded well with those from previous studies (Eatman, 1942; Mousty & Bertelson, 1985). The overall reading patterns followed three styles: eight of the subjects used one hand, and 12 used two hands. Of the two-handed readers, nine used a conjoint style, with the two index fingers only ever about one or two cells apart. The remaining three used disjoint hand movements involving their two index fingers.

While readers using one finger adopted a more or less homogeneous approach to reading, those using two fingers used a variety of different reading styles. These can be broadly categorized as (1) using both hands conjointly to read lines, followed by a return sweep involving both hands; (2) using both hands to read from the beginning of the line to near the end, and then parting their left and right hands, with the right continuing, the left returning to the beginning, followed eventually by the right when it reaches the end of the line; (3) using the left hand to rescan text that has already been traversed by the right hand.

A sample of results from our study are shown in Fig. 9, which gives examples of the different reading styles encountered in the sample. The sizes of the dots in the graphs represent transit time through the cell. It should be kept in mind here that, with skilled braille readers, the hand does not “fixate” or pause on a cell; it is continually in motion. So, the variations that we see in the sizes of the dots in Fig. 9 are due to relatively slower or faster transit times through cells.

Figure 10 is an example of the pattern of movement over one sample sentence, with results averaged from single passes (i.e., where there was no rereading) made by eight one-handed readers. Note the inverted-U shape over some of the words (e.g., “new” and “curtains”). This subtle slowing down in midword is a common pattern found with high-speed readers (Aranyanak & Reilly, 2012). These speed variations can be on the order of 50 ms and would be hard to detect in devices using lower sampling rates, such as those reviewed in the preceding sections.

The data visualizations adopted in Figs. 9 and 10 are influenced to some degree by a sighted reading perspective. Hughes (2011), in contrast, focused on the kinematics of braille reading (e.g., finger velocity). Some work is still to be done to find a suitable way to represent and analyze braille reading data in a way that does justice both to the dynamics of the information pickup process and to its lexical and textual nature. It is our view that each approach provides a complementary perspective on the braille reading process.

Caveats and limitations

We maintain that the system proposed here represents a significant advance on what is currently available to braille researchers. Nonetheless, there are already a number of aspects of the system requiring fine-tuning.

While one of the goals of our system was to afford as normal a reading environment as possible, without sacrificing accuracy, our tracking hardware does impose some constraints on the reader. We have elected to track one finger per hand at most, which means that we have to select the main reading finger when a reader uses more than one per hand. In our sample, the number of such readers was small. Nonetheless, they were also among the fastest readers that we encountered. While we are satisfied that a judicious choice of tracked finger will give reliable information in these situations, there may be a case for modifying the system optionally to use more than one LED per hand.

Related to the preceding point, a useful additional feature for our system would be an extra video camera to monitor the mode of contact of the readers’ fingers with the braille text. Our current configuration uses a video camera mounted above the reading plane. A camera that viewed the reader’s hands from the front would help in identifying which fingers to track in the case in which several fingers on one hand are used.

Crucial to much of the effective use of our system is the reliability and accuracy of our infrared sources. The LEDs that we use are smaller than the average finger width, so this leaves open the possibility that we are underestimating the transit time of the finger through a cell. We believe that this is less of a problem when looking at aggregated transit times through words, since it would be a constant error. The issue becomes more important in the case of display change manipulations, in which one could change a displayed cell while the reader’s finger had not left it or had already encroached upon it. Again, this issue could be dealt with by finer tuning of the system’s code to take account of individuals’ finger widths.

Conclusions

This article has described a cheap and easy-to-use finger-tracking system for studying braille reading, either from a refreshable display or from embossed paper. It provides spatial and temporal resolution close to that found in modern entry-level eyetracking systems. With its capabilities to implement display change paradigms on a refreshable display, it has the potential to allow researchers to probe in significantly more depth the remarkable achievement that is skilled braille reading. While we have had to makes some compromises in designing our system, overall we believe that it has better performance, greater generality, and significantly better ease of use than any other system currently available.

The code for the current version of the tracking system can be downloaded as supplementary materials for this article.

Author note This research was supported in part by a Pat and John Hume Scholarship to the first author. The authors gratefully acknowledge the extensive help and support provided by the National Council for the Blind in Ireland. An earlier version of the software interface for the Wiimote and braille display was developed by Qing Zhang, which greatly helped in the writing of the current version of the software. We thank Barry Hughes and an anonymous reviewer for their helpful and insightful comments on an earlier draft of the manuscript. Finally, we are grateful to the subjects in our experiment for their time, patience, and helpful comments.

Appendix

Listing 1 Code excerpt to illustrate interfacing with the Wiimote and the Handy Tech braille display: Sample C# code that initializes communication with the Wiimote and updates the infrared sensor locations (this example is used as part of the calibration routine for the tracking system)

```
private void FirstCalibrationWindow_Load(object sender, EventArgs e){
      Wiimote m_mote;
      int sb = 0;
      float x, y;
      float x1, y1, x2, y2;
      String a, b, c, d;
      // mWC holds the list of all wiimotes
      WiimoteCollection mWC = new WiimoteCollection();
      m_mote = null;

      try {
            mWC.FindAllWiimotes();
      }
      catch (WiimoteNotFoundException ex){
            MessageBox.Show(ex.Message,
                  "Wiimote not found error",
                  MessageBoxButtons.OK, MessageBoxIcon.Error);
      }
      catch (WiimoteException ex){
            MessageBox.Show(ex.Message, "Wiimote error",
                      MessageBoxButtons.OK, MessageBoxIcon.Error);
      }
      catch (Exception ex){
            MessageBox.Show(ex.Message, "Unknown error",
                      MessageBoxButtons.OK, MessageBoxIcon.Error);
      }

      foreach (Wiimote wm in mWC){
            m_mote = wm;
            break;
      }

      if (m_mote != null){
            // connect it and set it up as usual
            m_mote.WiimoteChanged +=
                  new EventHandler<WiimoteChangedEventArgs>
                        (m_mote_WiimoteChanged);
            m_mote.Connect();
            if (m_mote.WiimoteState.ExtensionType !=
                  ExtensionType.BalanceBoard) {
                  m_mote.SetReportType(InputReport.IRAccel, true);
            }
      }
}

delegate void UpdateDelegate(IRState irSensor);  //update sensor

void m_mote_WiimoteChanged(object sender, WiimoteChangedEventArgs e){
      try {
      this.BeginInvoke(new UpdateDelegate(UpdateUI),
            e.WiimoteState.IRState);
      }
      catch (Exception) {
      }

}

void UpdateUI(IRState e){
      UpdateIR(e.IRSensors[0]);
}

private void UpdateIR(IRSensor irSensor){   //get data from Wiimote
      if (irSensor.Found) { //Wiimote detects infrared
            // get X and Y coordinates
            x = irSensor.RawPosition.X;
            y = irSensor.RawPosition.Y;
            textBox3.Text = irSensor.RawPosition.ToString();
      }
}
```

Listing 2 Code excerpt to illustrate interfacing with the Wiimote and the Handy Tech braille display: C# code that initializes the braille display, reads keypresses from it, and displays a line of text

```
Wiimote newGetWii;
public Thread thrd;

//declare braille class
HtBrailleDriverClass braille = new HtBrailleDriverClass();
private void _Fingers_Load(object sender, EventArgs e)
{
      braille.initialize(); // Initialize the braille display
      Array a = new byte[] { };
      braille.displayText(ref a);
      braille.onKeysPressed += new
      IHtBrailleDriverSink_onKeysPressedEventHandler
      (braille_onKeysPressed);
}

public delegate void brailleRead(ref System.Array keys);

void braille_onKeysPressed(ref System.Array keys, int
            cursorRoutingPosition){
      this.BeginInvoke(new brailleRead(GetKey), keys);
}

void GetKey(ref System.Array keys){
      int keyCount = keys.GetUpperBound(0);
      String key = "NULL";
      HtBrlKeys a = (HtBrlKeys)(keys.GetValue(0));
      key = a.ToString();
      if (key == "KEY_RIGHT") {
      if(line ==1){
            end = 2; // starting recording the data
            }
            ShowSent(); // calling method ShowSent
      }
}

private void ShowSent(){
      // byte[] SH = new byte[40];
      for (int i1 = 0; i1 < 40; i1++){
            SH[i1] = 0;
      }
      for (int i1 = 0; i1 < 40; i1++){
            SH2[i1] = 0;
      }
      for (int i1 = 0; i1 < 40; i1++){
            SH3[i1] = 0;
      }
      for (int i1 = 0; i1 < 42; i1++){
            St1[i1] = 0;
      }
      if (wrBrl.Peek() != -1){ // not the end of the file
            string Sent = wrBrl.ReadLine();
            string Text = wrEng.ReadLine();

            textBox3.Text = Sent;
            textBox4.Text = Text;
            char[] s = Sent.ToCharArray();//cut the text into characters
            for (int i = 0, i1 = 0; i < Sent.Length && i < 40; i++){
                  PutIntoSH(s, ref SH, ref i1, i);
            }//get the ascii code

            Array st = new byte[40];
            // put the characters into the array
            for (int i = 0; i < 40; i++){
                  st.SetValue(SH[i], i);
            }
            // show the characters on the display
            braille.displayText(ref st);
      }
      else { // the end of the file
            textBox3.Text = "The End of File!!";
            textBox4.Text = "The End of File!!";
            Array st = new byte[] { };
            braille.displayText(ref st);
      }
}
```

References

Aranyanak, I., & Reilly, R. G. (2012). *Cohort effects in high-speed braille readers*. Manuscript in preparation.

Bertelson, P., Mousty, P., & Radeau, M. (1992). The time course of braille word recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18,* 284–297.

Breidegard, B., Eriksson, Y., Fellenius, K., Jonsson, B., Holmqvist, K., & Stromqvist, S. (2008). Enlightened: The art of finger reading. *Studia Linguistica, 62,* 249–260.

Bürklen, K. (1932). *Touch reading of the blind* (F. K. Merry, Trans.). New York, NY: American Foundation for the Blind, Inc. (Original work published 1917).

Eatman, P. F. (1942). *An analytic study of braille reading.* Unpublished doctoral dissertation, University of Texas.

Huey, E. B. (1968). *The psychology and pedagogy of reading.* Cambridge, MA: MIT Press. (Originally work published 1908).

Hughes, B. (2011). Movement kinematics of the braille-reading finger. *Journal of Visual Impairment and Blindness, 105,* 370–381.

Millar, S. (1997). *Reading by touch*. New York, NY: Routledge.

Mousty, P., & Bertelson, P. (1985). A study of braille reading, 1: Reading speed as a function of hand usage and context. *Quarterly Journal of Experimental Psychology, 37A,* 217–233.

Mousty, P., & Bertelson, P. (1992). Finger movements in braille reading: The effect of local ambiguity. *Cognition, 43,* 67–82.

Noblet, A., Ridelaire, H., & Sylin, G. (1985). Equipment for the study of operating processes in braille reading. *Behavior Research Methods, Instruments, & Computers, 17,* 107–113. doi:10.3758/BF03200904

Peek, B. (2008). *Wiimote library.* Retrieved March 30, 2012, from www.codeplex.com/WiimoteLib

Rayner, K. (1975). The perceptual span and peripheral cues in reading. *Cognitive Psychology, 7,* 65–81.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin, 124,* 372–422.