May 2021

BRAILLE EXPERIENCE ON UNMODIFIED SMARTPHONES

D Shin

Anupam Pathak

BRAILLE EXPERIENCE ON UNMODIFIED SMARTPHONES

ABSTRACT

Braille is a set of 3-by-2 tactile patterns that have a one-to-one mapping with the English alphabet utilized by the visually impaired to read and write. Techniques of this disclosure enable a computing device, such as a smartphone, tablet, wearable device (e.g., smart watch), laptop, etc., to activate particular patterns of haptic motors of the computing device that correspond to each character. The computing device may digitally encode the text shown on the display into braille dots using an on-device character recognition system. When the user raster-scans their finger over different pixels of the screen, the computing device determines the particular braille dots at the location of the user's finger and activates a pattern of haptic motors corresponding to the particular pattern of braille dots at the location of the user's finger. In this way, the user may perceive a haptic pattern that corresponds to the pattern of the braille dots that the user is "touching" on a computing device without requiring physical morphing the display and without requiring physical modification of the computing device.

DESCRIPTION

Braille is a set of 3-by-2 tactile patterns that have a one-to-one mapping with the English alphabet utilized by the visually impaired to read and write. However, modern smartphones do not support conventional Braille as there is no straightforward way to mechanically morph smartphone displays. While visually impaired users may utilize a screen reader or other software to read aloud information shown on the display, other people in proximity to the visually impaired user may also be able to hear the contents of the display being read aloud, which may have privacy implications.

Rather than relying upon audio that may introduce privacy concerns or require wearing headphones, which may limit what the visually impaired person may be able to hear, techniques described here enable a computing device to activate patterns of haptic motors that correspond to Braille dot patterns. As the user moves their finger over the screen, the computing device may activate the haptic motors based on the braille dot pattern associated with the character displayed at the location at which the computing device detects their finger. The user may feel the particular pattern of haptic motors that are activated and recognize the pattern as a particular alphanumeric character. In this way, a computing device may utilize existing hardware to provide a privacy-aware technique to enable visually impaired users to read information displayed by the computing device.

FIG. 1 illustrates an example computing device 100 that implements the technique of this disclosure. Computing device 100 may include a display 102 and haptic motors 104A-104F

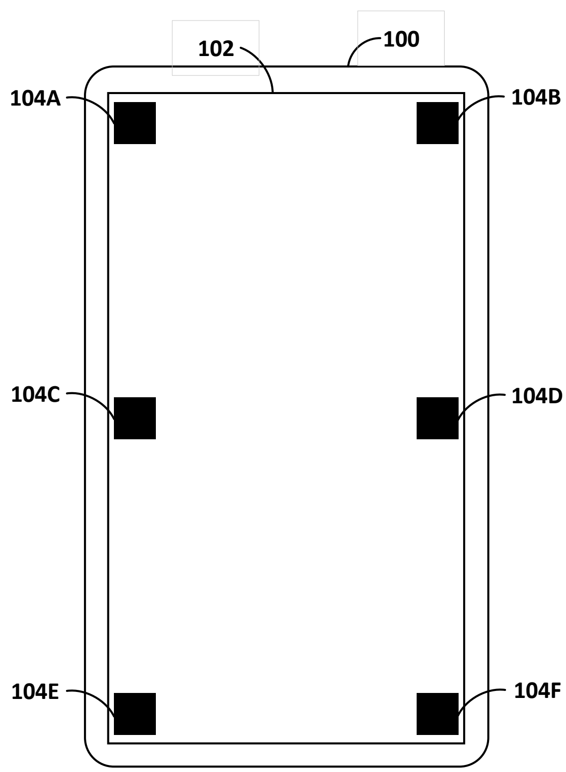

FIG. 1

(collectively "haptic motors 104"). Computing device 100 may be any mobile or non-mobile computing device, such as a cellular phone, a smartphone, a desktop computer, a laptop computer, a tablet computer, a portable gaming device, a portable media player, an e-book reader, a watch (including a so-called smartwatch), a gaming controller, and/or the like.

Display 102 may be implemented using various display hardware. In some examples, display 102 may function as an input device using a presence-sensitive input component, such as a presence-sensitive screen or touch-sensitive screen, which receives tactile input from a user of computing device 100. The presence-sensitive input component may receive indications of the tactile input by detecting one or more gestures from the user of computing device 100 (e.g., the user touching or pointing to one or more locations of the presence-sensitive input component with a finger or a stylus pen). The presence-sensitive input component may determine a location (e.g., an (x,y) coordinate) of the presence-sensitive input component at which the object was detected. As one example range, the presence-sensitive input component may detect an object, such as a finger or stylus that is within two inches or less of the presence-sensitive input component.

Haptic motors 104 may include one or more haptic actuators, such as linear resonant actuators, eccentric rotating mass vibration motors, piezoelectric transducers, electromechanical devices, and/or other vibrotactile actuators, and drive electronics coupled to the one or more actuators. The drive electronics may cause the one or more haptic actuators to induce a selected vibratory response into at least a portion of the computing device 100, thereby providing haptic feedback in the form of a tactile sensation to a user of computing device 100.

A user may cause computing device 100 to output information using display 102 (e.g., by powering on computing device 100, activating display 102, etc.). For example, a user may cause computing device 102 to begin executing an application, which, in turn, may cause display 102 to output a graphical user interface of the application. The graphical user interface may include content that is a mixture of graphic and textual content, such as a web page. If the user is visually impaired, the user may not be able to visually perceive the content shown on display 102.

In accordance with techniques of this disclosure, computing device 100 may identify alphanumeric information included in the graphical user interface and may cause a pattern of haptic motors to vibrate in a particular pattern that corresponds to a character shown on display 104. For example, while not shown in FIG. 1, computing device 100 may include an on-device character recognition engine that analyzes the content shown on display 102, identifies any alphanumeric characters included in the content, and generates braille dot patterns associated with the identified alphanumeric characters. That is, the on-device character recognition engine may determine the braille dot patterns without sending any information to a remote server. Further, the on-device character recognition engine may not maintain any history of information shown on display 102. By performing the character recognition and braille dot pattern fully on computing device 100 and without maintaining a history of recognized characters, the techniques described herein may further ensure the privacy of any sensitive information that may be output by display 102.

In one example, the on-device character recognition engine analyzes the content included in the graphical user interface and identifies the characters “H”, “I”, and “#”. The on-device character recognition engine identifies the braille dot patterns associated with one or more of those characters. In particular, the braille dot patterns for “H”, “I”, and “#” are shown in FIG. 2. In some instances, the on-device character recognition engine may cause display 102 to output the braille dot patterns for each recognized character in place of the corresponding recognized character. Whether the particular braille dot patterns are shown on display 102 or not, the on-device character recognition engine may associate the location of display 102 at which each recognized character is displayed with the corresponding braille dot pattern for that character.

The user may use their finger to read what is shown on display 102. For example, the user may touch any location of the screen, may sequentially move their finger over the screen (e.g., in a left-to-right and top-to-bottom pattern (i.e., a raster scan pattern), in a right-to-left, top-to-bottom pattern, etc.), or any combination thereof. A presence-sensitive component of display 102 may detect the presence of the user's finger in proximity to display 102. The presence-sensitive component may detect the user's finger over an area of display 102 and may provide an indication of the area of display 102 at which the user's finger is detected to a finger morphology detection module. The finger morphology detection module may process raw capacitive multi-touch data generated by the presence-sensitive component of display 102 by applying a connected components detection algorithm to localize islands of potential finger touch regions.

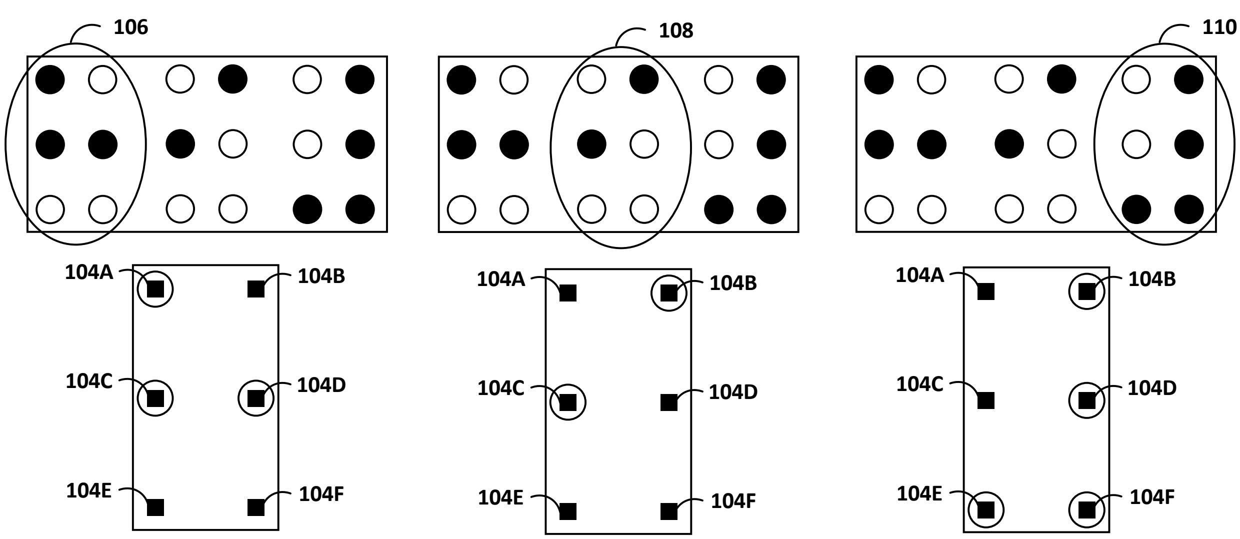

FIG. 2

FIG. 2 includes three different finger touch regions, 106, 108, and 110, of display 102 over which the user's finger is detected. Continuing the example above where the characters "H", "I", and "#" are included within the graphical user interface, computing device 100 may determine that finger touch region 106 is associated with a location of the braille dot pattern for the letter "H". That is, computing device 100 may determine finger touch region 106 intersects with the area associated with the braille doc pattern for the letter "H". Responsive to determining that finger touch region 106 is associated with the location of the braille dot pattern "H", computing device 100 may use a spatial motor driver to activate the pattern of haptic motors 104 corresponding to that braille dot pattern. In particular, as the braille dot pattern for the letter "H" includes raised dots as shown by the filled in dots within area 106, computing device 100 causes haptic motors 104A, 104C, and 104D to vibrate (as shown by the circle around each of haptic motors 104A, 104C, and 104D).

As the user continues to move their hand across display 102, computing device 100, using the presence-sensitive component and the finger morphology detection module, detects the user's finger at finger touch region 108 and then at finger touch region 110. Responsive to the user's finger being detected over area 108, computing device 100 determines that area 108 is associated with the braille dot pattern for the letter "I" and activates haptic motors 104B and 104C, which correspond to the braille dot pattern for the letter "I". Similarly, in response to the user's finger being detected over area 110, computing device 100 determines that area 110 is associated with the braille dot pattern for the character "#" and activates haptic motors 104B, 104D, 104E, and 104F, which corresponds to the braille dot pattern for the character "#".

Areas 106, 108, and 110 all encompass all six dot locations for a particular braille dot pattern. However, a user's input is not typically as well aligned with the areas of display 102 associated with each braille dot pattern. For example, as shown in FIG. 3, the area of display 102 at which the user's finger is detected may be an irregular area and may only be associated with a portion of a braille dot pattern.

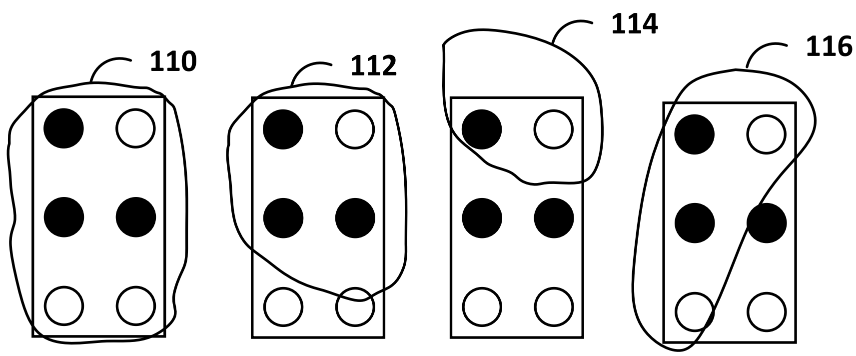

FIG. 3

FIG. 3 includes finger touch regions 110, 112, 114, and 116, which illustrate different areas of display 102 at which the user's finger may be detected. To account for finger touch regions that do not directly align with areas associated with particular braille dot patterns, computing device 100 may apply an intersection constraint. For each braille dot pattern, computing device 100 may determine a number of nodes (i.e., raised dots within the dot pattern that are represented by filled circles in the example of FIG. 3) and non-nodes (i.e., flat dots within the dot pattern that are represented as unfilled circles in the example of FIG. 3) covered by the finger touch regions. To be considered covered by the finger touch region, computing device 100 may require that at least a threshold portion (e.g., 50%, 70%, 90%, etc.) of each node and non-node be covered by the finger touch region.

As one example, computing device 100 determines that finger touch region 110 intersects with all nodes and non-nodes of the braille dot pattern for the letter "H" as all of the nodes and non-node are covered by finger touch region 110. Responsive to determining that finger touch region 110 satisfies the intersection constraint, computing device 100 may activate haptic motors 104A, 104C, and 104D.

As shown in FIG. 3, finger touch region 112 does not intersect all of the nodes and non-nodes for the braille dot pattern for the letter “H”. In such an example, computing device 100 may determine that finger touch region 112 does intersect with three nodes and one non-node. As finger touch region 112 does not cover any portion of the bottom left non-node, computing device 100 determines that finger touch region 112 does not intersect with that node. Finger touch region 112 does cover at least a portion of the bottom right non-node. Computing device 100 may determine a percentage (i.e., ratio) of the bottom right non-node covered by finger touch region 112 and determine whether that percentage satisfies a threshold percentage required to consider the node as being intersected by finger touch region 112. In this example, finger touch region 112 intersects less than 50 percent of the bottom right node and computing device 100 determines that such a percentage does not satisfy the threshold. Accordingly, computing device 100 determines that the bottom right non-node is not intersected by finger touch region 112.

Even though finger touch region 112 does not intersect with all 6 nodes and non-nodes of this particular braille dot pattern, computing device 100 may still determine if there is sufficient intersection to determine that the input is an intentional input and activate the appropriate haptic motors 104. As one example, computing device 100 may apply an intersection constraint that requires that the number of nodes and non-nodes intersected by finger touch region 112 is at least four and that the four nodes must define a two-by-two square. As finger touch region 112 does intersect with at least four nodes and non-nodes and those intersected nodes define a two-by-two square, computing device 100 may determine that such an input is an intentional input (i.e., the user is attempting to read that particular braille dot pattern). Responsive to determining that the input is an intentional input, computing device 100 activates haptic motors 104A, 104C, and 104D.

With respect to finger touch region 114, computing device 100 may determine that finger touch region 114 only intersects with one node and one non-node. Applying the same intersection constraint, computing device 100 determines that fewer than four total nodes and non-nodes are covered by finger touch region 114 and, as such, determines that the input is not an intentional input. Because the input is determined not to be an intentional input to read the character associated with this particular braille dot pattern, computing device 100 does not activate the particular pattern of haptic motors 104 for this particular braille dot pattern in response to detecting this user input.

In some examples, the detected user input may cover a total of four nodes and non-nodes but the covered nodes and non-nodes may not form a two-by-two square. Finger touch region 116 illustrates one example finger touch region that partially satisfies the intersection constraint. In this example, computing device 100 determines that finger touch region 116 covers a sufficient portion of all three nodes and non-nodes in the left side of the braille dot pattern and the non-node in the top right of the braille dot pattern. As such, computing device 100 determines that a minimum number of nodes and non-node are covered by finger touch region 116, satisfying the first part of the intersection constraint. Next, computing device 100 determines whether the four total nodes and non-nodes form a two-by-two square. In this example, the nodes and non-nodes covered by finger touch region 116 do not form a two-by-two square. Computing device 100 determines that finger touch region 116 fails the second requirement for the intersection constraint and, thus, determines that the user input is not an intentional user input trying to read this particular braille dot pattern and does not activate the particular pattern of haptic motors 104 for this particular braille dot pattern.

It is noted that the techniques of this disclosure may be combined with any other suitable technique or combination of techniques. As one example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Publication 2014/0181722 by Yu-Na Kim and Sang-Hyup Lee. As another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Publication 2012/0218193 by Arnett Weber and Raymond Michael Dikun. As another example, the techniques of this disclosure may be combined with the techniques described in Shokat, S., Riaz, R., Rizvi, S.S. *et al.* "Deep learning scheme for character prediction with position-free touch screen-based Braille input method," *Hum. Cent. Comput. Inf. Sci.* **10,** 41 (2020), which is available online at https://doi.org/10.1186/s13673-020-00246-6.

.