Tender – Smartphone Application for Braille Reading

Shoichi Ito
National Institute of Technology, Nagano College
Nagano, Japan
shoichi@nagano-nct.ac.jp

Yoshinori Fujisawa
National Institute of Technology, Nagano College
Nagano, Japan
fujix@nagano-nct.ac.jp

ABSTRACT

Visually impaired people use Braille in their daily life. Braille text is printed in many ways like a handrail of stairs, guide maps, and so on. Visually impaired people read these Braille guides to navigate themselves. However, misprinted Braille may leads visually impaired people to wrong or dangerous direction. Misprinted Braille should be corrected by sighted people. On the other hand, most of sighted people can not read Braille directory. To solve this problem, we developed a Braille reading application for sighted people. User can read about 70% of Braille text from this application just by taking a photograph.

CCS CONCEPTS

• **Human-centered computing** → *Accessibility systems and tools.*

KEYWORDS

Braille, Visually Impaired People

ACM Reference Format:
Shoichi Ito and Yoshinori Fujisawa. 2019. Tender – Smartphone Application for Braille Reading. In *7th ACIS International Conference on Applied Computing and Information Technology (ACIT 2019), May 29–31, 2019, Honolulu, HI, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3325291.3325368

1 INTRODUCTION

Braille is a tactile writing system used by people who are visually impaired. It consists of 6 raised dots and each of them is about 1 millimeter in diameter. Visually impaired people read Braille text by tracing it using their fingertips. Figure 1 shows the size and spacing of Braille characters. Detailed dimensions in the figure are shown in Table 1. Japanese Braille and English one have slightly different physical size, but they share almost similar structure with one another.

To read such very small dot patterns, visually impaired people need a delicate sense of touch and training. In addition to these difficulties, Braille has its own unique grammar different from any spoken languages because of 6 dots in Braille can express 2^6 patterns only. Therefore, Braille is hard to read not only for sighted people but also for people who are visually impaired. In Japan, less than 20% of visually impaired people use Braille in daily life [2].

ACIT 2019, May 29–31, 2019, Honolulu, HI, USA

ACM ISBN 978-1-4503-7173-5/19/05…$15.00
https://doi.org/10.1145/3325291.3325368

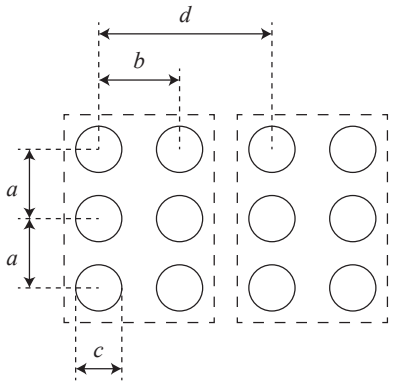

Figure 1: Size and spacing of Braille characters

Table 1: Dimensions in millimeter

Parameter	Japanese	English
a	2.2–2.5	2.3–2.5
b	2.0–2.5	2.3–2.5
c	1.3–1.7	1.5–1.6
d	5.1–6.3	6.1–7.6

In the last decade, guides for visually impaired people described in Braille have been widely used in the whole city, especially in public areas. If the Braille guide was misprinted, visually impaired people may be guided to wrong or dangerous direction. However, since many sighted people are not able to read Braille text, misprinted guides have been avoided for a long period of time.

We developed a smartphone application to solve this problem. Our application is called "Tender". If sighted people took a photograph of Braille text in the city, then Tender analyses the image and translate the Braille text into Japanese text. By using this application, sighted person can correct misprinted Braille guide easily.

2 SCENARIO AND TECHNOLOGY

In this section, we describe how Tender works along the general scenario that sighted people use Tender. The user tap Tender's application icon on the screen of the smartphone, and Tender boots up. Figure 2 shows home screen of Tender. First of all, users choose [Image] or [Keyboard] at the bottom of Figure 2. These selections are how to input Braille text into Tender. Users can choose the following three methods:

(1) take a photograph of the Braille
(2) choose a photograph of the Braille from a camera roll of the smartphone

Figure 2: Home screen of Tender

(3) input Braille patterns directory from a Tender's software keyboard for Braille

[Image] is selected as default. Processing flow from three types of input to Japanese text is described in Figure 3. Tender's Braille analysis module extracts dot pattern of Braille text from Braille using image processing algorithms. Dot pattern of Braille is also generated by users from Braille Keyboard of Tender. Translation module accepts dot patterns of Braille text and the translated text is output.

2.1 Photography

Users take a photograph of Braille text printed on some materials with their own smartphone's back camera. Figure 4 shows Tender's camera mode. Users can zoom in/out, choose a trimming region of one line text to translate. This process is the same as users choose a photograph of Braille from camera roll of the smartphone.

In the real world, brightness of the environment or material color may cause detection errors. After taking a photograph of Braille text, users can correct the image manually to remove some reasons of detection error. Users can change the contrast of the image by parameter a in Eq.(1). Eq.(1) is reffered to as a sigmoid function, where x is a pixel value of original image, x' is a pixel value of contrast-controlled image, and M is a maximum pixel value of 8 bit gray scale image (i.e. 255). Both of x and x' are 8 bit gray scale value.

$$x' = M\left[1 + e^{-\frac{a}{M}(x-\frac{M}{2})}\right]^{-1} \quad (1)$$

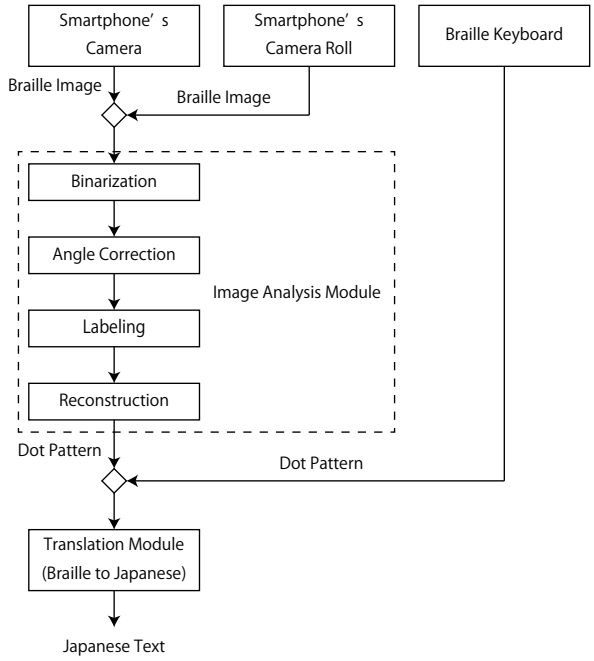

Figure 3: Processing flow

Figure 4: Taking a photograph of Braille and hand-operated regulation

Figure 5 shows an example of contrast control by Eq.(1). A center image is an original one. The upper image is processed by Eq.(1) with $a = 3$, the lower image is similar but $a = 40$. We can extract a luster of dots from Braille image very easily for $a = 40$ image.

2.2 Binarization

Image of Braille text is converted into 8 bit gray scale image and contrast of the image is adjusted by users as explained above. Figure 6 is an example of a gray scale image of Braille text. Black patterns in the upper part of the image is a text which is printed with black ink for sighted people.

Tender internally makes a histogram of each pixel value as shown in Figure 7. The small peak in the figure labeled A means a black ink described above. The highest peak labeled B is a paper color and C is the cut-off value to binarization. Tender calculates the middle of B and maximum pixel value (i.e., pixel value of the most bright pixel in Figure 6) as C. Each pixel is assigned 0(=black) if the pixel value is larger than C, assigned 1(=white) if the other. Note that

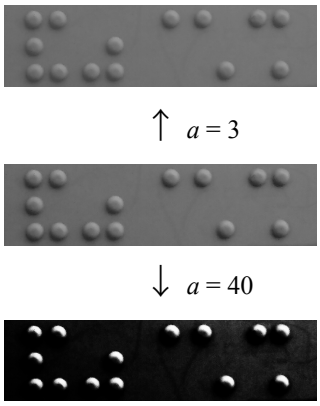

Figure 5: Contrast control by Eq.(1)

Figure 6: An example image of Braille text in 8 bit gray scale

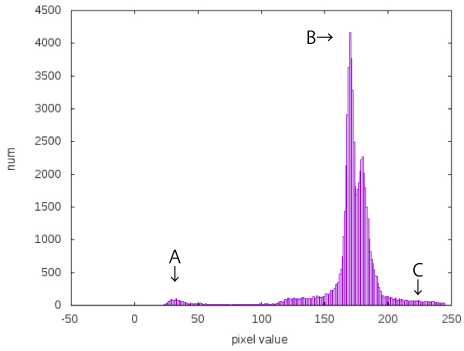

Figure 7: Histogram of pixel value

Figure 8: Binarized image of Braille text

black and white reversal is operated in this step. After this process, Figure 6 is binarized into Figure 8. In Figure 8, a bright luster of each Braille dot is extracted well, and black ink text is automatically disappeared.

Note that this binarization process works well if bright lusters of Braille dot and paper color can separate clearly.

2.3 Angle Correction

Let the size of the binarized image processed above section is $W \times H$ pixel. Figure 9 is the same image as Figure 8, but horizontal guide lines are added. It seems to slightly slant to the lower right.

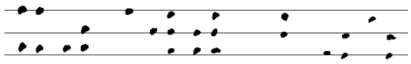

Figure 9: Slightly slnated image of Braille

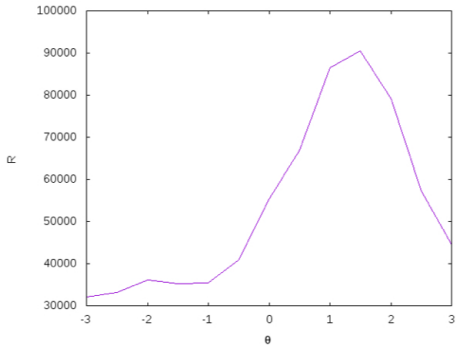

Figure 10: Rotation angle vs. R in Eq.(2)

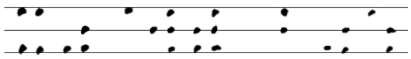

Figure 11: Binarized Braille image (slant corrected)

This slant should be corrected to improve the accuracy of Braille translation. Tender corrects this slant automatically.

First, Tender counts the number of black dots on i-th line as C_i. Then, all the square value of C_i is summed up in R defined in Eq.(2). If we rotate Figure 8 with small rotation angle, R changes its value. In other words, a value of R in Eq.(2) is a function of rotation angle, and it describes a degree of the slant. Figure 10 shows the value of R as a function of rotation angle.

$$R = \sum_{i=1}^{H} C_i^2 \tag{2}$$

Tender rotates the image and calculates R between -3 to $+3$ degree every 0.001 degree, and finds the best rotation angle θ_{max} to maximize R. Original image (Figure 8) is rotated in θ_{max} and we get Figure 11. In contrast with Figure 9, Braille dots in Figure 11 are well aligned horizontally.

2.4 Labeling

Next, we decompose dots in the image of Braille. Tender extracts the sets of key coordinates of Braille dots from an image of Braille. We start from a binarized Braille image. In Figure 12 , every Braille dot in the image has distorted shape. It is difficult to process such patterns directly with high accuracy. Therefore, we have to regularize these sets of distorted shapes into those of proper coordinates.

We adopt the labeling algorithm to the Braille image. By using this algorithm, connected pixels are grouped into a same cluster. Figure 13 shows a result. In this figure, each pixel is colored with

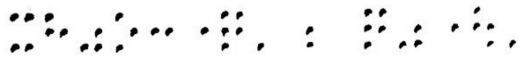

Figure 12: Braille image before labeling

Figure 13: Braille image after labeling

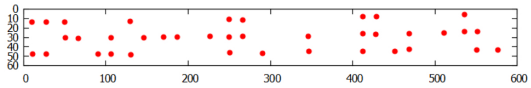

Figure 14: Sets of coordinates of the center of gravity

different colors. Pixels belonging to the same cluster are colored with same color. In this step, each Braille dot is cleary separated.

Then, coordinates of the center of gravity for every cluster is calculated. We denote a coodinate of the center of gravity for i-th cluster as (X_i, Y_i). Index i takes a range of $1 \leq i \leq N$, where N is a total number of clusters. We also denote the set of these coordinates as $\{(X_i, Y_i)\}$.

Figure 14 plots sets of coordinates of the center of gravity from Figure 13. We successfully extracted proper coordinates of every Braille dot from Braille image.

2.5 Reconstruction of Braille from dots

Then, these coodinates of dots will be grouped and reconstructed into individual Braille character. It is very difficult to detect how many blank columns are there, because blank dots do not appear in a camera image.

As a first step, we arrange the set of coordinates of the center of gravity in certain range. The acquired coordinates of the center of gravity are scattered in a broad range depending on the size of inputted image. We normalize all the value of the center of gravity by multiplying with factor r. Normalization factor r is defined in Eq.(3), where Y_{min} and Y_{max} are minimum/maximum value of Y_i in $\{(X_i, Y_i)\}$, respectively.

$$r = \frac{100}{Y_{max} - Y_{min}} \tag{3}$$

In this normalization step, we got new sets of the center of gravity $\{(\hat{X}_i, \hat{Y}_i)\}$, where $\hat{X}_i$, $\hat{Y}_i$ are normalized coodinates defined in Eq.(4) and (5). Normalized coodinate values take a range $0 \leq \hat{X}_i \leq \hat{X}_{max}$, $0 \leq \hat{Y}_i \leq 100$, respectively.

$$\hat{X}_i = r\,(X_i - X_{min}) \tag{4}$$

$$\hat{Y}_i = r\,(Y_i - Y_{min}) \tag{5}$$

Now, we can classify $\{(\hat{X}_i, \hat{Y}_i)\}$ into three groups corresponding to three lines in Braille dots. If $\hat{Y}_k$ takes a value within $0 \leq \hat{Y}_k < 33.3$, Tender interprets k-th dot as a dot in the first line. Similarly,

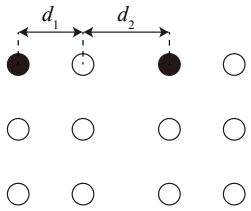

Figure 15: Example of one blank column between two dotted columns

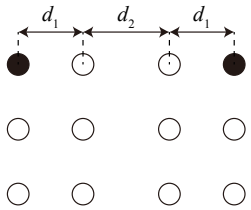

Figure 16: Example of two blank columns between two dotted columns

$33.3 \leq \hat{Y}_k < 66.6$ is the second line, $66.6 \leq \hat{Y}_k \leq 100$ is the third line.

Coodinates from an image usually have some fluctuation originated from distortion of each dot image. Tender measures average value and distribution of these coodinates. If the gap between x coordinates of two dots is within the error range, Tender regards these two dots as in the same column. Tender sorts all the value of $\hat{X}_i$ in ascending order. With this sorting process, dots which belong to the same column in the same Braille character are collected nearby. Tender checks the difference between the two neightbor sorted coordinates and numbers the column number to each dots.

As we explained above, blank columns do not appear in a camera image. But blank columns in Braille character is as important as dotted columns. Therefore, we need to interpolate blank columns into detected dots. In Figure 15, d_1 and d_2 correspond to b and d in Figure 1, respectively. As shown in the figure, if the distance between black dots equals to $d_1 + d_2$, Tender interpolates a blank column between these black dots. Note that the white dots do not exist in the binalized image, which implies they are treated as blank. As shown in Figure 16, if the distance between black dots equals to $2d_1 + d_2$, Tender interpolates two blank columns between these black dots. Similarly, Tender can detect and interpolate up to four blank columns.

If the first distance between two black columns equals to d_2 as shown in Figure 17, Tender interpolates a blank column into the head of Braille text. Braille text should have even number of columns because one Braille character is made up of two columns. Tender interpolates a blank column into the tail of Braille text if the text has odd columns.

In the final step of reconstruction, all dots are replaced with the value 1 if dot exists, 0 if dot does not exist. These values are grouped into every 6 dots, then encoded to 6 bit values. For example, Braille

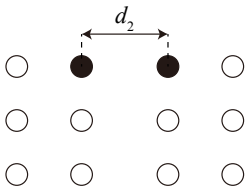

Figure 17: Example of one blank column at the head of Braille text

of Figure 6 is encoded into 101101, 001011, 000100, 010111, 011111, ..., a set of 6 bit values.

2.6 Translation

As the final step, Tender translates Braille dot patterns into Japanese based on Japanese Braille grammar. Our translation module of Braille to Japanese is able to translate almost all Japanse Braille text in daily life correctly, except for URL string and some special characters like é or ä.

Figure 18 shows an example of translation result. First image of the figure is the original Braille image taken from smartphone's back camera. Second image of the figure is a binarized image using algorithms described in Section 2.2 and 2.3. Third image of the figure is Braille pattern detected using the algorithm described in Section 2.4 and 2.5. This Braille image is printed with Unicode Braille character font set. In this step, Braille is now recognized as not as images but as proper characters by Tender. The last Japanese text describes a translation result. This shows that first Braille image is translated into Japanese exactly.

Note that the physical form of Braille character is very identical in the whole world, namely one Braille character consists of 6 dots (2 columns and 3 lines). Therefore, image processing algorithm described above sections can be adopted not only to Japanese Braille but also to the Braille in other country. Only the grammar to translate Braille to a proper language are not identical. In Tender, "Translation Module" in Figure 3 can be replaced with a module for another language. For example, using Braille to English translation module, English users can read English Braille. Unified English Braille (UEB) module is currently under development.

2.7 Braille Keyboard

Braille text is printed in many different kinds of materials such as papers, plastics, steels, films and so on. In addition to materials, non-white colors of the material and brightness of the environment causes error translations in many cases.

In such a case, users can input Braille pattern directory from Tender's Braille keyboard with touch controls. In Figure 2, Tender's Braille keyboard appears with [Keyboard] button (Figure 19). 6 circles in Figure 19 corresponds to 6 dots of one Braille character. Each dot changes alternately dotted and blank by one tap. Tender translates Braille text to Japanese text type-by-type. This keyboard works only on Tender, it can not use generic virtual keyboard for the operating system.

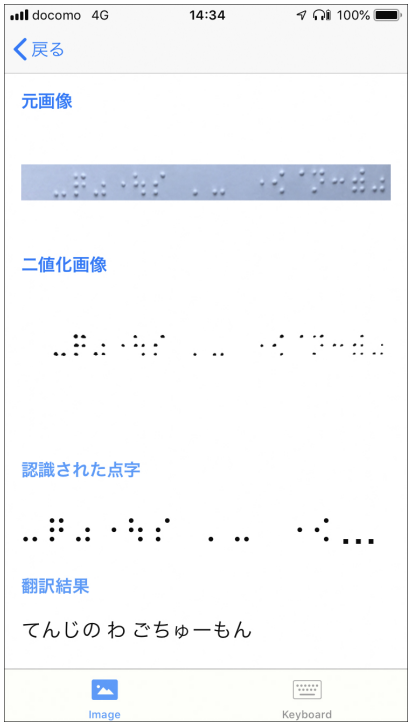

Figure 18: Translation results

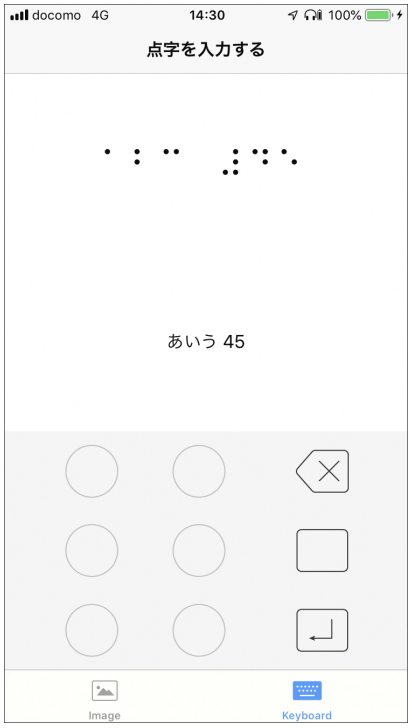

Figure 19: Braille keyboard

3 PERFORMANCE EVALUATION

We tested this application. We can read about 70% of Braille text in our daily life. For example, Braille texts printed on white papers or white stickers are able to be read by Tender. It can also read Braille text printed in dark gray colored plastics, red plastics, flat stainless plate and so on. However, Tender can not read Braille text when their dots are printed on luster materials or surrounding light conditions. In this case, we need to correct translation results using Braille keyboard to input Braille dot pattern into Tender.

Note that some algorithms to detect Braille in prior reseach have strong dependency on the size of input Braille image. In other words, some algorithms can not detect Braille dots if input image size would not take a proper size for the algorithm. As explained in Section 2.5, all coodinates of Braille dots are normalized in Tender by Eq.(4) and (5). This normalization process enables to solve a size dependency of input image. For example, Figure 12 is 612× 75 pixel image which contains 15 Braille characters. Tender can detect Braille dots exactly not only in Figure 12, but also in 30% reduced (180×22 pixel) image.

4 SUMMARY

In this paper, we explained our software, Tender, a smartphone application to translate Braille for sighted people.

Tender works well under the limited and desirable environment. However, Tender can not read all Braille text in our daily life due to many variations in printed materials and so on. We need to improve image processing algorithm in the next version of Tender. English version of this software has also been developed. Translation engine for UEB Grade 1 is almost complete, some parts of Grade 2 [1] have already been developed.

We are also working on a wearable-type Braille reader device for visually impaired people based on same technology with Tender. In this device, we use a micro size pressure sensor attached on the tip of a finger to detect real Braille dots. Users slide their finger on the Braille text, then this device reads translated text with voice.

REFERENCES

[1] R. J. CLARKE. 2016. *Learn Braille – Uncontracted (Grade 1) & Contracted (Grade 2).* CreateSpace Independent Publishing Platform.

[2] Employment Measures for Persons with Disabilities Division Employment Development Department. 2008. *Survey on persons with pyhsical disability 2006.* Technical Report. Ministry of Health, Labour and Welfare of Japan.