Development of TARS Mobile App with Deep Fingertip Detector for the Visually Impaired

Yoichi Hosokawa, Tetsushi Miwa, and Yoshihiro Hashimoto
Nagoya Institute of Technology, Nagoya, Japan
{y.hosokawa.456, t.miwa.528}@nitech.jp, hashimoto@nitech.ac.jp

Abstract. We propose TARS mobile applications that uses a smartphone with a camera and deep learning fingertip detector for easier implementation than using a PC or a touch panel. The app was designed to recognize the user’s hand touching the images with the rear camera and provide voice guidance with the information on the images that the index finger is touching as a trigger. When performing gestures with either the index finger or thumb, the app was able to detect and output the fingertip point without delay, and it was effective as a trigger for reading. Thumb gestures are assumed to have reduced detection variances of 68% in the lateral direction because they rarely move the other four fingers compared to index finger gestures. By performing multiple detections in the application and outputting the median, the variances of detection can be reduced to 73% in the lateral direction and 70% in the longitudinal direction, which shows the effectiveness of multiple detections. These techniques are effective in reducing the variance of fingertip detection. We also confirmed that if the tilt of the device is between −3.4 mm and 4 mm, the current app could identify a 12 mm difference with an accuracy of 85.5% as an average in both of the lateral and longitudinal directions. Finally, we developed a basic model of TARS mobile app that allows easier installation and more portability by using a smartphone camera rather than a PC or a touch panel.

Keywords: Tactile graphics with an audio response system · TARS · Voice guidance gesture · Fingertip detector · Visual impairment

1 Introduction

As one way for visually impaired people to understand figures and maps, they use tactile graphics with raised shapes on the paper. While this has the advantage of allowing understanding of the shape and position of objects by touching it, it also has the disadvantage of being difficult to distinguish objects among points, lines, and description Braille on the tactile graphics. Sometimes it requires others to explain the images and the users cannot learn on their own.

Kwok et al. [1] showed that the format of the drawn images, such as surface and contour lines, size, and height, improves the readability of the tactile graphics.

Yamamoto et al. [2], on the other hand, embedded audio descriptions in the tactile graphics so that users can focus on the image search, instead of relying on the
description Braille. Talking Tactile Tablet [3] is a similar device that makes it easy to read tactile graphics.

We [4] developed tactile graphics with an audio response system (TARS) which enables visually impaired people to hear an audio description of the image on the tactile graphics by using a touch panel to set the tactile graphics and a computer with a screen reader and tapping the image on the tactile graphics set on the touch panel. The device was made at a cost less than those of previous research with the advantage of portability. TARS allows visually impaired people to focus on the tactile graphics, gain a lot of information from them, and learn independently.

Simon et al. [5] revealed that deep learning from the camera images of a computer can provide fingertip detection. Miwa [6], a co-researcher, showed that the fingertip detection is 98% accurate, regardless of the type of tactile graphics, under conditions of sufficient brightness by using a smartphone camera.

We propose TARS mobile applications that uses a smartphone with a camera and deep learning fingertip detector for easier implementation than using a PC or a touch panel. The app was designed to recognize the user’s hand touching the images with the rear camera and provide voice guidance with the information on the images that the index finger is touching as a trigger. Figure 1 shows the installation of the device and the tactile graphics. Initially, it was triggered by a user talking to the device, but in this research, we decided to use gestures as triggers based on the assumption that a malfunction would occur when multiple users simultaneously talk to the device. In the preliminary tests, the user’s fingertip points detected by the app deviated from the actual fingertip point and voice guidance could not be made.

We assume that the following factors caused the problem:

- The fingertip detection outputs varied because the users moved their hands extensively to perform gestures.
- Since the fingertip detection logic includes inference, its output could vary every time even if the user touches the same point and sometimes could vary widely.
- The device could be set tilted and cause the fingertip detection to be misaligned with the images.

In this study, we will identify solutions to these factors and examine the development of the TARS mobile app, which can be used by visually impaired people on their own.

![Fig. 1. Device installation](image-url)
2 Methods

2.1 Application Development

According to a study by Watanabe et al. [7], 53.1% of Japanese visually impaired people own a smartphone, and 91.9% of all blind people own an iPhone. Therefore, development of the application targeted the iPhone.

Development conditions: iPhone SE (April 2020 model), 128 GB, iOS 13.4.1, Xcode 11.3. We developed a hand tracking system using Google’s MediaPipe v0.5 [8]. Visually impaired people use both hands to search for images and finally obtain the information of the target with the fingertips of one hand. Therefore, we decided to use one hand for the fingertip detection. We designed an app that provides voice guidance for the user’s index fingertip coordinates by activating AVSpeechSynthesizer when the user performs voice guide gestures (hereinafter “gestures”). We developed a total of four methods to activate the app, two based on the difference in fingertip detection times and two based on the difference in the types of gestures.

Methods Based on the Fingertip Detection Times. Single detection: Every time fingertip detection is performed, it outputs the coordinates. Multiple detections: After fingertip detections are performed seven times, it outputs the median value.

Methods Based on the Gesture Types. Gesture 1: A motion to fold the fingers other than the index finger from the condition of an open hand while touching the target image with the index finger. Figure 2 (left) shows the motion. Gesture 2: A motion to bend the thumb from inside to outside or outside to inside while touching the image with the index finger with an open hand. Figure 2 (right) shows the motion.

![Fig. 2. Left: gesture 1 (index finger); right: gesture 2 (thumb)](image-url)
2.2 Assessing the Accuracy of Fingertip Detection

We observed that when the user’s fingertip touched a point, released, and touched it again, the fingertip detection did not output the same coordinates but sometimes output largely deviated coordinates. The reason was assumed to be because the fingertip detection logic includes inference. For this problem, we conducted tests with the participants to assess the degree of variance in detection.

Test participants: Seven visually impaired persons who have used Braille for more than 3 years. The test participants’ data are shown in Table 1. The participants were informed of the test, understood the details, and agreed to participate.

| Table 1. Test participants’ data |
|---------------------------------|
| Age | Mean: 39.7 S.D.: 11.5 |
| Sex | Male: 5 Female: 2 |
| Hand to touch images | Left hand: 3 Right hand: 4 |
| Width of palm (mm) | Mean: 83.0 S.D.: 7.0 |
| Length from fingertip to wrist (mm) | Mean: 172.1 S.D.: 13.4 |

After the participants practiced the gestures with voice guidance sufficiently, they conducted the test by touching one point in the image and performing the gestures to activate voice guidance. The test was conducted five times, the coordinates were recorded every time, and the variance in detection was analyzed. Also, we used js-STAR9.8.4 for the statistical analysis.

2.3 Evaluation of Deviation Between the Tactile Graphics and the Fingertip Detection Coordinates

2.3.1 Deviation Calculation

To calculate the deviations between the tactile graphics and the fingertip detection coordinates, we set two calibration points (CPs) on the image and calculated the distance ratios between the CPs and target points.

The top left of an A4 size tactile graphics in the longitudinal direction is set as the origin (0, 0). For example, CP1 (25, 25), CP2 (190, 230), and P (105, 130) are on the tactile graphics. Assume that the app detects the fingertip coordinates and output CP1 (195, 389), CP2 (810, 1156), and P (486, 798) coordinates.

- Detection coordinates: \((486 - 195)/(810 - 195) = 0.47\)
- Tactile graphics coordinates: \((105 - 25)/(190 - 25) = 0.48\)
- Coordinate deviation: \((0.47 - 0.48) * (190 - 25) = -1.93\)

From this calculation, P was detected with a deviation of −1.93 mm from the tactile graphics in the lateral direction. In the same way, P was detected with a deviation of 4.32 mm in the longitudinal direction.
Adjustments for the Deviations. When using the app, it is expected that the above-mentioned deviations in fingertip detection and the deviations caused by device misalignment will occur. We evaluated the matching rate between tactile graphics coordinates and detection coordinates by conducting tests with participants. We selected three participants who achieved smaller detection coordinate deviations than the others in the previous test, the multiple detections and gesture test. The participants were informed of the test, understood the details, and agreed to participate. After the participants practiced the gestures sufficiently, they searched for the points of CP1 and CP2, and P1 to P9 in the image shown on the Fig. 3 and performed gestures to activate voice guidance. Based on the mean coordinates of P1 to P9 gained by the three participants, the matching rates between the tactile graphics coordinates and the fingertip detection coordinates were calculated. The matching rate was evaluated using two different adjustment methods:

- Adjustment 1: Method to calculate the detection points based on the angle difference between the line connecting CP1 and CP2 and the line connecting fingertip detection CP1 and CP2.
- Adjustment 2: Method to calculate the detection point using a constant coefficient to adjust the keystone distortion on the screen.

![Sample figure](image)

Fig. 3. Sample figure

3 Results

3.1 Functions of the Application

When the entire hand image was on the screen, 21 key points were detected from one hand. The key points are shown in Fig. 4 (left). The device was set with a distance to allow the rear camera to show both the image and the user’s hand even when the user searches for the edge of the A4 size images in the longitudinal direction, Fig. 4 (middle
and right) show the fingertip detection conditions when the device is installed at a height of 35 cm and 40 cm respectively. As a result, we decided to use the app with the device on a 40 cm height stand. The responses to voice guidance by multiple detections showed no delay compared to single detection.

3.2 Voice Guidance Gesture Responses

Gesture responses were assessed by identifying the detected key points. The criteria for the assessment are as follows:

- Gesture1: \( P_{12}(y) < P_{11}(y) \) & \( P_{16}(y) < P_{15}(y) \)
- Gesture2: \( (P_{17}(x) - P_{05}(x))/3 > P_{05}(x) - P_{03}(x) \)

Both gestures were successfully recognized. When the gestures were recognized, the voice guidance of the index fingertip coordinates was made.

3.3 Fingertip Detection Accuracy

A variance analysis for the two factors; gesture types and multiple detections was performed based on the results of the fingertip detection test by the 7 participants. Fingertip detection variances in the lateral direction are shown in Table 2. In the lateral direction, the main effect of the gesture type was significant \( F(1, 6) = 9.79, p < 0.05, \eta_p^2 = 0.62 \), and gesture 2 (average 5.85 pixels) had less variance than gesture 1 (average 8.61 pixels).

The main effect of multiple detections was also significant \( F(1, 6) = 26.12, p < 0.01, \eta_p^2 = 0.81 \). Multiple detections (average 6.10 pixels) were less varied than single detection (average 8.36 pixels). The interaction effect between gesture types and multiple detections was not significant \( F(1, 6) = 1.36, p = 0.29, \eta_p^2 = 0.18 \).
Fingertip detection variances in the longitudinal direction are shown in Table 3. In the longitudinal direction, the main effect of gesture type was not significant \( F(1, 6) = 0.12, p = 0.74, \eta^2_p = 0.02 \).

The main effect of the multiple detections shows a significant trend \( F(1, 6) = 4.96, p < 0.07, \eta^2_p = 0.45 \). The multiple detections (average value 6.39 pixels) show a trend that has less variances than single detection (average value 9.00 pixels.) The interaction effect between gesture types and multiple detections was not significant \( F(1, 6) = 0.17, p = 0.69, \eta^2_p = 0.03 \).

### Table 2. Two-factor variance analysis in the lateral direction (pixel)

| Gesture 1 | Single detection | Multiple detection |
|-----------|------------------|--------------------|
| N         | 7                | 7                  |
| Mean      | 10.25            | 6.98               |
| S.D.      | 2.24             | 1.96               |

| Gesture 2 | Single detection | Multiple detection |
|-----------|------------------|--------------------|
| N         | 7                | 7                  |
| Mean      | 6.47             | 5.22               |
| S.D.      | 1.86             | 1.69               |

### Table 3. Two-factor variance analysis in the longitudinal direction (pixel)

| Gesture 1 | Single detection | Multiple detections |
|-----------|------------------|---------------------|
| N         | 7                | 7                   |
| Mean      | 8.61             | 6.35                |
| S.D.      | 2.58             | 2.24                |

| Gesture 2 | Single detection | Multiple detections |
|-----------|------------------|---------------------|
| N         | 7                | 7                   |
| Mean      | 9.39             | 6.44                |
| S.D.      | 3.01             | 1.70                |

### 3.4 Evaluation of the Deviations Between the Tactile Graphics and the Fingertip Detection Coordinates

The tilt of the device was calculated based on the difference in angle between the line connecting CP1 and CP2 of the tactile graphics and the line connecting CP1 and CP2 of the fingertip detection. Since the tilt angle of the device is a small amount, it was converted to the length based on the device length of 138.4 mm. For example, −3.4 mm indicates that the device is tilted 3.4 mm to the right. We set the device at 5 different angles and calculated the length differences for tilt angles and calculated the deviation between the tactile graphics coordinates and the fingertip detection coordinates for each test point at every tilt angle. Then we obtained the matching rates by classifying the test point deviations into 5 ranges from 4 mm to 12 mm and dividing the number of test points in each range by the total number of test points. Tables 4 and 5 show the matching rate in the lateral direction and the match rate in the longitudinal direction.
under no adjustment. The matching rates increased as the tilt of the device decreased. If the device tilted by 0.2 mm, the matching rate within 6 mm in both the lateral and longitudinal directions was about 90%. The matching rate within 10 mm is as high as 100%. Also, the matching rate within 12 mm was 91% on an average in the lateral direction and 90% on an average in the longitudinal direction at any tilt angle from −3.4 to 4 mm. The matching rates by adjustment method are shown in Table 6. In the lateral direction, within 12 mm, the matching rate with no adjustment was higher than those with adjustments. In the longitudinal direction, within 8 mm, the matching rate with adjustment1 was higher than that with no adjustment by 2 points on average. In the longitudinal direction, within 6 mm, the matching rate with adjustment2 was higher than that with no adjustment by 4 points on average. Other than these cases, the matching rates with no adjustment were higher than those with adjustments.

Table 4. Matching rate in the lateral direction with no adjustment (%)

| Device tilt angle (mm) | −3.4 | −0.3 | 0.2 | 1.2 | 4.0 |
|-----------------------|------|------|-----|-----|-----|
| Range 4 mm            | 22   | 56   | 56  | 11  | 44  |
| 6 mm                  | 33   | 56   | 89  | 11  | 67  |
| 8 mm                  | 44   | 89   | 89  | 44  | 67  |
| 10 mm                 | 56   | 100  | 100 | 89  | 89  |
| 12 mm                 | 67   | 100  | 100 | 100 | 89  |

Table 5. Matching rate in the longitudinal direction with no adjustment (%)

| Device tilt angle (mm) | −3.4 | −0.3 | 0.2 | 1.2 | 4.0 |
|-----------------------|------|------|-----|-----|-----|
| Range 4 mm            | 22   | 67   | 78  | 44  | 0   |
| 6 mm                  | 33   | 89   | 89  | 67  | 11  |
| 8 mm                  | 56   | 100  | 100 | 67  | 11  |
| 10 mm                 | 89   | 100  | 100 | 78  | 11  |
| 12 mm                 | 89   | 100  | 100 | 100 | 11  |

Table 6. Matching rate by adjustment method (%)

| Direction | Lateral | Longitudinal |
|-----------|---------|--------------|
|           | No adj. | Adj. 1 | Adj. 2 | No adj. | Adj. 1 | Adj. 2 |
| Range     |         |        |        |         |        |        |
| 4 mm      | 38      | 31     | 33     | 42      | 42     | 49     |
| 6 mm      | 51      | 49     | 49     | 58      | 58     | 60     |
| 8 mm      | 67      | 64     | 62     | 67      | 71     | 64     |
| 10 mm     | 87      | 80     | 67     | 76      | 76     | 71     |
| 12 mm     | 91      | 89     | 71     | 80      | 80     | 73     |
4 Discussion

4.1 Variations in Fingertip Detection

When performing gestures with either the index finger or thumb, the app was able to detect and output the fingertip point without delay, and it was effective as a trigger for reading. Fingertip detection recognizes the skeletal structure of the wrist and arm, infers the position of the palm, and then calculates 21 key point coordinates based on the information. As a result, if the user moves their hands or fingers extensively, it is expected to recalculate for fingertip detection, resulting in greater coordinate variances. Thumb gestures are assumed to have reduced detection variances of 68% in the lateral direction because they rarely move the other four fingers compared to index finger gestures.

Also, the fingertip detection coordinates vary every time the same point is touched, and sometimes a very deviating value is detected. By performing multiple detections in the application and outputting the median, the variances of detection can be reduced to 73% in the lateral direction and 70% in the longitudinal direction, which shows the effectiveness of multiple detections. These techniques are effective in reducing the variance of fingertip detection.

4.2 Evaluation of the Deviations Between the Tactile Graphics and the Fingertip Detection Coordinates

By setting the calibration points in the image, we could easily identify the deviation between the tactile graphics and the fingertip detection coordinates. When printing an image, the printer driver may adjust the size of the image automatically. Deviation amounts in size vary among printers depending on the specifications or performance. Since the calculation of the deviations is relative to the calibration points, it is an effective means of determining the deviations even if the image is scaled up, reduced, or shifted in any direction. Also, because it is difficult for a visually impaired person to put the image in a fixed position, it should be effective to use the calibration points and adjust the position of the image. In order to resolve the deviation between the tactile graphics coordinates and the fingertip detection coordinates, two methods were used to calculate the coordinates using angle adjustment and keystone adjustment in this test. We found that it was the most effective to reduce the tilt of the device itself. We also confirmed that if the tilt of the device is between $-3.4$ mm and 4 mm, the current app could identify a 12 mm difference with an accuracy of 85.5% as an average in both of the lateral and longitudinal directions. Figure 5 shows the location of Japanese cities in circles with a diameter of about 12 mm. This application becomes a learning tool that allows users to know a city name by holding their fingers at the city’s location.
The test was conducted with the help of people who use Braille on a daily basis, and it was confirmed that the device could be installed on their own. In the future, the challenge is to develop voice guidance and user interfaces to help the users reduce the tilt of the device by themselves in setting.

Finally, we developed a basic model of TARS mobile app that allows easier installation and more portability by using a smart phone camera rather than a PC or a touch panel.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Number 17H00146.

References

1. Kwok, M.G., Fukuda, T.: The new method for making tactile maps based on the human sense characteristics. IEICE technical report. Well-being Information Technology 103(590), 55–62 (2004)
2. Yamamoto, K., Izumi, K., Inaba, J., Takayoshi, D., Yoshie, K., Higuchi, N.: Authoring tool of tactile graphics with voice explanation for teachers of schools for the blind. IEICE technical report. Well-being Information Technology 111(424), 39–44 (2012)
3. Talking Tactile Tablet 2 (TTT). http://touchgraphics.com/portfolio/ttt. Accessed 10 June 2020
4. Hosokawa, Y.: Effect of tactile graphics with an audio response system on reading speed and accuracy of diagram. In: The 55th Conference of the Japanese Association of Special Education, Aichi, Japan, p. 83 (2017)
5. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: 2017 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1145–1153. (2017)
6. Miwa, T., Hosokawa, Y., Hashimoto, Y., Lisi, G.: TARS mobile app with deep fingertip detector for the visually impaired. In: Ahram, T., Karwowski, W., Vergnano, A., Leali, F., Taiar, R. (eds.) IHSI 2020. AISC, vol. 1131, pp. 301–306. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-39512-4_48

7. Watanabe, T., Kobayashi, M., Minatani, K.: A survey of ICT device usage by blind people 2017 (2020). http://altmetrics.ceek.jp/article/hdl.handle.net/10191/00051465. Accessed 11 June 2020

8. Mediapipe. https://google.github.io/mediapipe. Accessed 10 June 2020