https://doi.org/10.2352/issn.2168-3204.2025.22.1.19

A Hybrid 3D Laser Scanning and Machine Learning System: Paving the Way for Autonomous Braille Digitization

Lei He; Library of Congress; Washington, DC/USA

Abstract

The Library of Congress has been conducting the Braille digitization to preserve its tactile braille music collection electronically. In this project we scan braille papers with a 3D laser sensor to obtain the coordinates of the recto/front and verso/back dots, which are fed into our digitization program for dots pattern recognition. Our software groups the dots in different lines and further into different characters. Specifically, each dot in a character is identified according to the relative positions to its local neighbors in the same character, based on which the character is recognized and the corresponding ASCII glyph code is written into the final output file. Experiments on different collections show the robustness of our system.

Introduction

The Music Section at the National Library Service (NLS) for the Blind and Print Disabled at the Library of Congress (LOC) has been conducting the braille digitization in order to preserve its tactile braille music[1] collection electronically since over 20 years ago [1]. The request to increase the scanning rate has been rising significantly in recent years due to the high demand for the digitized braille music through the LOC online digital library. In addition, our braille digitization effort also helps reduce the physical storage space for the accumulated braille scores and save the fragile and aging scores.

In 1824, Louis Braille invented the tactile reading and writing system *braille* (named after the inventor), which uses raised dots to represent characters. Braille is a code and it can be transcribed into different languages[2]. Figure 1 shows the braille cell structure that consists of an array of three rows and two columns of dots. Any combination of dots may be raised at these six positions, which constructs a dictionary of sixty-four characters. In practice, multi-character sequences are used for some specific purposes, such as representing more characters or the pitch and rhythm for braille music.

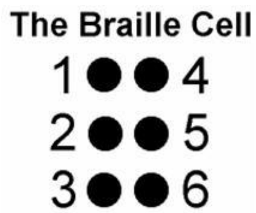

Figure 1*. Braille dot cell structure*

[1] https://blogs.loc.gov/nls-music-notes/2022/09/braille-music-basics-reading-notes-and-octave-signs/
[2] https://brailleworks.com/braille-resources/what-is-braille/

Braille digitization translates the dots embossed on papers to the corresponding braille ASCII code (glyph) that represents different characters. Existing braille scanning systems [2, 3, 4, 5, 6, 7, 8] utilize optical cameras to scan braille papers (see Figure 2 for examples) and conduct dots pattern recognition. Traditional optical braille recognition systems [2, 3, 4, 5] usually apply a sequence of image processing techniques to the dot images, such as image de-skewing, enhancement, segmentation, braille dots detection, and cell construction, to clean the image and extract the valid dots for character recognition. With such complicated procedures, those approaches usually are sensitive to the intermediate result from each step. Specifically, the image noise and distortions introduced in image acquisition severely affect the recognition accuracy. Later systems [6, 7] apply classical machine learning algorithms such as neural networks [6] and Support Vector Machine [7] to improve the recognition accuracy, which still requires a sequence of operations before the final character recognition of each braille cell respectively. Current systems [8, 9] improve overall performance by employing the state-of-art deep learning technologies to extract and recognize all the braille characters together from the whole image, which are robust to the imaging quality impairments. For example, with the training images of rotated papers, the deep learning networks may overcome the problem of misaligned dot lines, i.e., the character dots may still be recognized correctly even the image is rotated. The learning-based approaches requires a large set of training data with manual annotations. In some systems, the image acquisition quality remains an issue of the whole system, i.e., certain image preprocessing like image alignment and enhancement are still required for a better performance.

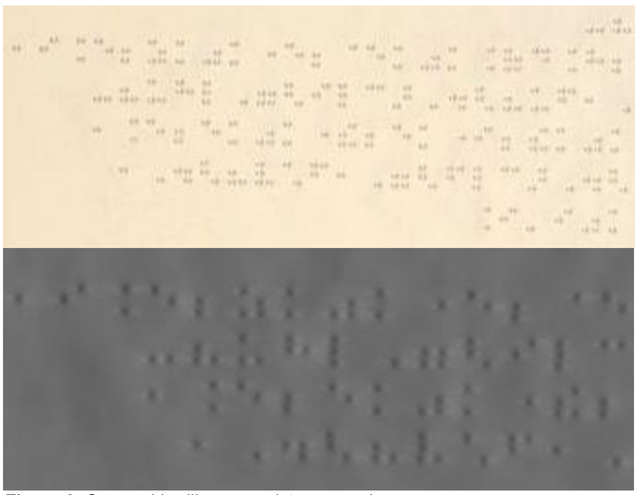

Figure 2*. Cropped braille paper picture samples*

After exploring various products over the last twenty years, the Music Section at NLS has been focusing on OBR[3] and DotScan[4] for a large-scale braille digitization [1]. However, both systems are still far from our expectation on the robust performance and friendly user interface, which involve intensive labor tasks and achieve low recognition accuracy. For example, the OBR always misses the dot lines at the margin of the paper and the results cannot be proofread without a visual presentation of the dots, all of which mandate over 20% working hours of the operators to solve these problems. On the other hand, the DotScan provides an interface for an easy proofread, but cannot handle the inter-point (double-sided) braille.

To overcome the above-mentioned limitations of the optics-based systems, we constructed a novel platform for braille digitization that utilizes laser sensors.

Braille Digitization System

In this project we conduct the braille digitization by scanning a braille paper through laser sensors that locate all the dots on the paper. The dots of the same character are then grouped together to identify the character, and finally the corresponding braille ASCII code is recorded.

The digitization starts with the depth information collection for the braille dots, which is implemented by a set of four 3D/line scan cameras (Keyence® XG-X2802[5]), with the positive depth values for the recto (convex) dots and negative depth values for the verso (concave) dots. The locations of these recto and verso dots are the input to our digitization software (see Figure 3 for the system setup and the software interface), in which we develop the machine learning program for the dots pattern recognition and conversion to the glyph code. Specifically, the laser cameras are mounted on a linear stage (Newport® IMS400CCHA[6]) which is driven by the motion controller (Newport® XPS-D2[7]). Our digitization software controls the motion of the stage through the controller, and triggers the laser sensor to scan the braille paper. The system can process both single-sided and inter-point (double-sided) braille papers. The scanner output is the coordinates of the recto (positive depth) and verso (negative depth) dots. Separately for the recto and verso sides, our machine learning program applies the clustering algorithms (e.g., DBSCAN, k-Medoids) to group the dots into different lines (i.e., a set of characters form a line of text) and rows (i.e., a braille character has 3 rows and 2 columns), and further into different characters according to the space/gap among dots. Figure 4 illustrates the sample dot cells with the space parameter settings for the clustering algorithms (GWC-H: Gap Within Characters - Horizontal, GWC-V: Gap Within Characters - Vertical, GBC-H: Gap Between Characters - Horizontal). Each dot in a character is further identified according to the relative positions to its local neighbors in the same character, based on which the character itself is recognized and the corresponding glyph code is written into the final output file. For those characters with a single dot or a single row/column of dots, their dots' positions again are identified according to their relative distances with respect to their nearest neighbors in the same line. For those lines with single rows, the characters are determined by the space between their nearest neighbor lines.

In practice those parameters are adjusted each time for a new set of papers. Once the first page of a book is digitized and proofread (including manual paper alignment, missing or noise dots inspection, scanning and conversion, and glyph code verification), the remaining papers can be placed on the board with the same position and orientation for automatic scanning and conversion. Manual operations are only needed to label the false positive (noise) dots caused by random shapes in papers. In the next section we present the results of different paper samples.

***Figure 3**. 3D braille digitization system and software interface*

*	*		*	*		*	*
*	*		*	*		*	*
*	*		*	*		*	*

GWC-V GBC-H GWC-H

***Figure 4**. A line of characters: space/distance of dots within and between characters*

---

[3] https://www.eyeway.org.in/?q=optical-braille-recognition
[4] http://dotscan.de/eng/index.php?page=systeme
[5] https://www.keyence.com/products/vision/vision-sys/xg-x/models/xg-x2802/
[6] https://www.newport.com/p/M-IMS400CCHA?srsltid=AfmBOook_pPWubJ0DgB8e7XBCwnYVX-ykxArh_5h5D09jNyv5qgsF8jw
[7] https://www.newport.com/p/XPS-D2?srsltid=AfmBOoo_ypr4bDu17XYPivGe-oc4LVFNjCFFwH6JYiqKuFDI1QostAF5

Results

Our digitization software implements braille digitization with two steps: *Scan* and *Preview & Conversion*. An operator first places the braille paper under the laser scanner and arranges the paper in the scanning direction (Figure 5), then he can click the *Scan* button in the user interface (Figure 3) to start the scanning. The stage motion speed and the duration of the pause after scanning can be edited in the interface window. A high-speed scanning may introduce distortion or missing dots in the result. After scanning, the operator may click the *Preview & Conversion* button to locate the scanning output files which record the coordinates of all the recto and verso dots. Then the clustering programs in the digitization software automatically separate the recto and verso dots for each character and save the corresponding glyph code. Figure 6 shows an example of the detected dots that are initially grouped into different lines (highlighted by different colors) for the operator preview. The operator may click *Yes* in the message box to proceed to the next steps of character clustering and conversion if everything looks fine, or click *No* to go back to rescan the paper if something is incorrect.

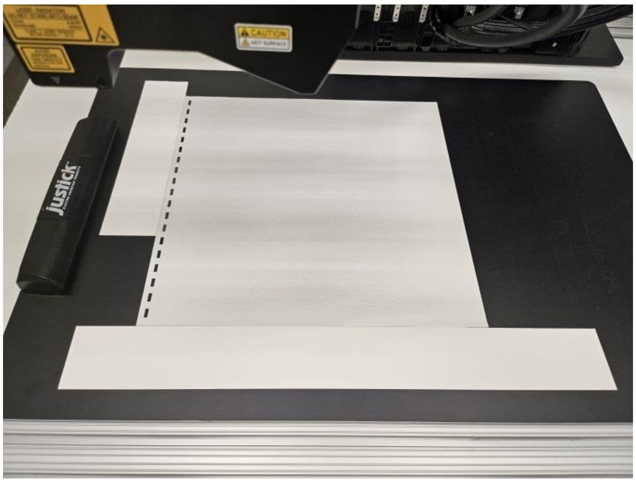

***Figure 5**. Alignment of the braille paper along the scanning direction*

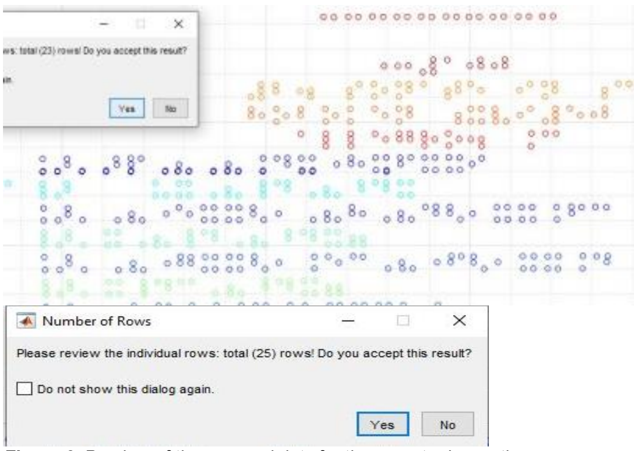

***Figure 6**. Preview of the scanned dots for the operator inspection*

If the operator confirms everything is fine, the software continues to group the dots in each line into different rows and then characters, i.e., the conversion is implemented line by line from the top to the bottom in a page. Figure 7 shows that the dots in the same line are further grouped into different rows and characters with our clustering programs. For example, the top image in Figure 7 shows the three rows of the braille dots in one line, which are highlighted by three colors respectively. The bottom image in Figure 7 shows the dots of fifteen characters separated by different colors. Each character is finally recognized by identifying all the dots of the character. After the conversion of all the lines in the page, Figure 8 presents the output glyph code of the recognized characters in both recto and verso sides, which may be reviewed and proofread by the operator. Throughout the digitization process, the software displays various error/warning messages to describe the possible reasons causing the conversion errors. For example, sometimes one or more dots may be grouped into incorrect rows due to the skew of the scanned paper, e.g., the top row dots of the beginning characters are at the same height as the bottom row dots of the ending characters, or vice versa.

***Figure 7**. Each dot in a line of characters should be correctly grouped into one of the Braille rows (top, middle, or bottom), and one of the characters in the line.*

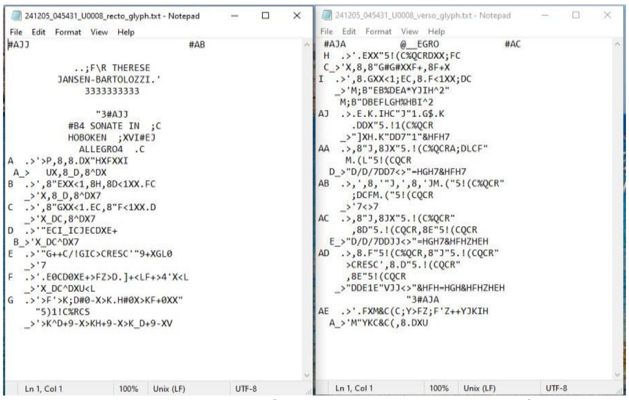

***Figure 8**. The output glyph code for both recto and verso sides of the paper*

In practice, most braille books may be digitized by following the procedures above. However, there are cases such as books with large size page or with random shapes embossed inside some pages, which requires special handling. For a large size paper, the operator may rotate it to scan the paper. Before clicking the *Preview & Conversion* button, the operator needs to check the box *Rotate Page* in the user interface. The rest of the procedures are the same as regular paper digitization. Figure 9 shows a sample with a guitar shape embossed on the paper, and the scanning result can be previewed by clicking the button *Noise Identification*, as shown in Figure 10. It can be seen that some false positive ("noise") dots are shown in the middle of the result. The operator may click these false positive dots to label them, which can then be removed by clicking the button *Noise Remove* button. After noise removal, the operator may repeat the conversion procedures by clicking the *Preview & Conversion*, and the software automatically updates the dots for the conversion.

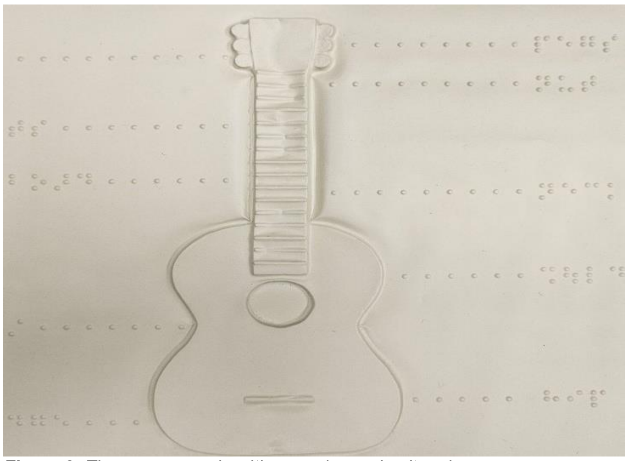

***Figure 9**. The paper sample with an embossed guitar shape*

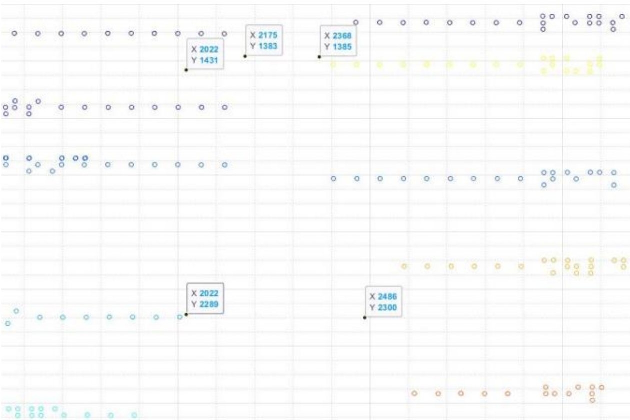

***Figure 10.** The labelled "noise" dots in the paper*

Conclusions

In this paper we present a novel automatic braille digitization system with 3D laser line cameras, which effectively overcomes the limitations (e.g., intensive labors, high sensitivity to image distortion and noise, and complex operations) of the existing optics-based systems. The system was built by integrating the hardware of laser sensors and a pair of motion controller and stage, based on which the system software was developed to control the hardware and to convert the scanned braille dots into character glyph code. Our system simplifies the digitization with two steps of scanning and glyph code conversion. Experiments on different braille samples show the robustness and high accuracy of system performance. For future work, we will upgrade the software to handle very large size paper which cannot be digitized in one scan. Specifically, we will conduct multiple scans of different areas of the paper and stich the results together for a final conversion.

The author would like to thank Ms. Katie Rodda and Ms. Juliette Appold from the Music Section in National Library Service for providing valuable insights and guidance during the system software test phase. We also acknowledge the financial support from the National Library Service and a generous gift from Susan D. Diskin[8], PhD, who enabled the acquisition of critical equipment for this project.

References

[1] D. Koh, K. Rodda, Digitizing Braille Music: A Case Study, Proc. Archiving Conference, pg. 21-25. (2018).

[2] T. Shreekanth, V. Udayashankara, "A Review on Software Algorithms for Optical Recognition of Embossed Braille Characters," International Journal of Computer Applications, 81, 3, pg. 25-35. (2013).

[3] M. Hanumanthappa, V. V. Murthy, Optical Braille Recognition and Its Correspondence in the Conversion of Braille Script to Text – A Literature Review, Proc. International Conference of Computational Systems and Information Systems for Sustainable Solutions, pg. 297-301. (2016).

[4] S. Isayed, R. Tahboub, A Review of Optical Braille Recognition, The 2nd World Symposium on Web Applications and Networking, 1, pg. 1-6. (2015).

[5] B. Nugroho, I. Ardiyanto and H. A. Nugroho, Review of Optical Braille Recognition Using Camera for Image Acquisition, The 2nd International Conference on Biomedical Engineering, pg. 106-110. (2018).

[6] K. Smelyakov, A. Chupryna, D. Yeremenko, A. Sakhon and V. Polezhai, Braille Character Recognition Based on Neural Networks, IEEE 2nd International Conference on Data Stream Mining & Processing, pg. 509-513. (2018).

[7] J. Li, X. Yan, Optical Braille Character Recognition with Support-Vector Machine Classifier, International Conference on Computer Application and System Modeling, pg. v12-219 – v12-222. (2010).

[8] R. Li, H. Liu, X. Wang, J. Xu and Y. Qian, Optical Braille Recognition Based on Semantic Segmentation Network with Auxiliary Learning Strategy, IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pg. 2362-2368. (2020).

[9] T. Kausar, S. Manzoor, A. Kausar, Y. Lu, M. Wasif and M. A. Ashraf, "Deep Learning Strategy for Braille Character Recognition," IEEE Access, 9, pg. 169357-169371. (2021).

Author Biography

Lei He received his PhD in Electrical Engineering from the University of Cincinnati in 2003. Since then, he has worked as a faculty member at Georgia Southern University (Savannah Campus), GA, for six years and was awarded tenure in 2009 as an Associate Professor. Before joining the Library of Congress in 2011, Dr. He worked at the National Library of Medicine, NIH, for two years. He is a digital imaging scientist of the Collections Digitization Division within the Library Collections and Services Group in the Library of Congress.

---

[8] https://blogs.loc.gov/nls-music-notes/2019/12/braille-digitization-project-gets-a-big-boost/