

A Robust Braille Recognition System

A. Antonacopoulos and D. Bridson

Pattern Recognition and Image Analysis (PRImA) group, Department of Computer Science
University of Liverpool, Liverpool, L69 3BX, United Kingdom
<http://www.csc.liv.ac.uk/~prima>

Abstract. Braille is the most effective means of written communication between visually-impaired and sighted people. This paper describes a new system that recognizes Braille characters in scanned Braille document pages. Unlike most other approaches, an inexpensive flatbed scanner is used and the system requires minimal interaction with the user. A unique feature of this system is the use of context at different levels (from the pre-processing of the image through to the post-processing of the recognition results) to enhance robustness and, consequently, recognition results. Braille dots composing characters are identified on both single and double-sided documents of average quality with over 99% accuracy, while Braille characters are also correctly recognised in over 99% of documents of average quality (in both single and double-sided documents).

1 Introduction

Information in written form plays an undeniably important role in our daily lives. From education and leisure, to casual note taking and information exchange, recording and using information encoded in symbolic form is essential. Visually impaired (blind and partially sighted) people face a distinct disadvantage in this respect. Addressing this need, the most widely adopted writing convention among visually impaired people is Braille. Since its inception in 1829, significant developments have taken place in the production of Braille and Braille media as well as in the transcription of printed material into Braille.

However, although the production of Braille documents is relatively easy now, the problem of converting Braille documents into a computer-readable form still exists. This is a significant problem for two main reasons. First, there is a wealth of books and documents that only exist in Braille that, as with other rare/old documents, are deteriorating and must be preserved (digitized). Second, there is an everyday need for duplicating (the equivalent of photocopying) Braille documents and for translating Braille documents for use by non-Braille users. The latter application is quite important, as it forms the basis for written communication between visually impaired and sighted people (e.g., a blind student submitting an assignment in Braille).

The automated recognition of Braille documents is not straightforward due to the special characteristics of the documents themselves (see below) and the constraints of the application domain. More specifically, in addition to the natural expectations for

high efficiency demanded from a document conversion application, a Braille recognition system must also be easy to use by visually impaired people (it should not require complicated setup etc.) and cost-effective (should only use commercially available standard equipment).

In order to appreciate the idiosyncrasies of the problem and to set the scene for the subsequent description of the system, a brief description of Braille and of automatic Braille recognition systems is given below. For a comprehensive account on the subject, the reader is referred to another publication [1] by the first author.

Braille is a particular system of representing information in tactile form. As such, Braille documents are formed by groups of protruding “dots” representing characters and various symbols (including music). Each Braille character is an arrangement of six points in two columns of three (the Braille *cell*) as can be seen in Fig. 1 (there is also an 8-point representation but, for brevity, it is not covered here). Each point can be either raised (a protrusion, or Braille *dot*) or flat. The height and diameter of protrusions as well as the distances between dots in the same and adjacent Braille characters are standardized—e.g., by the UK’s Royal National Institute for the Blind (RNIB) and by the Library of Congress in the US.

The meaning of each Braille character depends on the type of Braille encoding used. In Grade 1 Braille, for instance, there is a one-to-one correspondence between printed characters and their Braille representation (with the exception of some punctuation marks). In Grade 2 Braille (also called *contracted*, or *literary* Braille) there are conventions for representing whole strings of printed characters by a single Braille one. A particular Braille character may correspond to more than one string of printed text and this association is only resolved by examining the context of the character (e.g., whether the character is on its own, or in the beginning, middle or end of a word). There are, of course, different conventions for Braille in different countries (e.g., English, American, French, etc. Braille) and domains (e.g., music, mathematics, computer code etc.)

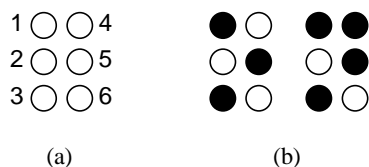


Fig. 1. (a) The Braille cell. (b) The word “ON” in Grade 1 English Braille (filled dots represent protrusions).

The nature of Braille has direct implications on the physical characteristics of documents. The thickness of the page material (most commonly card) and the added thickness introduced by the protrusions result in very bulky documents, in comparison to printed documents containing the same information (a dictionary can occupy a whole bookcase). One attempt to make Braille documents more concise is the introduction of contracted Braille, as explained above. Another attempt, at the physical level, is the introduction of double-sided Braille documents. These documents, also referred to as *inter-point* Braille have dots (protrusions) on both sides of the page. The implication

is that on a given page of inter-point Braille there exist protrusions (the dots of the recto) and depressions (the dots of the verso). This fact impacts on the difficulty of the recognition problem as will become evident in the discussions in the remainder of this paper. Most bound Braille documents (e.g. books) encountered will be double-sided, while most personal documents produced with a Braille typewriter or printer will be single-sided.

An obvious, perhaps, but significant characteristic of Braille documents is the absence of any information visible in a colour contrasting the background. The only information recorded on a Braille page is in terms of the protrusions created by embossing the card (under uniform illumination a Braille document page appears blank). The fact that Braille documents are not intended to convey any visual information also has repercussions on the quality of card used to produce them. It is not uncommon for the card to be of low (visual) quality, with visible grain and imperfections (dark and light regions). This fact can affect the recognition of Braille documents by visual means (the objective of any realistic automated conversion system).

There have been a number of attempts to recognize Braille documents using relatively complex (in terms of a visually impaired person) setups of a camera and oblique lighting (in order to reveal shadows from the protrusions) [2][3][4]. In addition to the non-standard setup and equipment (potentially expensive and difficult to use), the images obtained frequently suffer from the problems of camera-based image acquisition (e.g., aberrations, irregular lightness, relatively low resolution etc.).

In contrast, the use of a commercially available flatbed scanner facilitates image acquisition and it is a cost-effective solution. The only crucial requirement is that the illumination of the Braille document is non-uniform, i.e. there is only one light source in the scanner and at a slight distance from the image sensor (CCD). While scanner output quality has increased dramatically in the last decade and the price of top-quality scanners has dropped equally dramatically, manufacturers have not eliminated the “problem” of non-uniform (sideways) illumination. In fact, there is an abundance of very low priced scanners that “suffer” from non-uniform illumination. It is very fortunate that precisely this kind of scanner fulfils both the cost effectiveness and suitability criteria for optical Braille recognition.

One of the first approaches to use a flatbed scanner to appear in the literature is that of Ritchings et al. [6]. It is applied to both single and double-sided Braille documents, scanned at 100dpi at 16 greylevels (for economic reasons at the time). It performs few image-based operations and it is relatively flexible to skew as it identifies Braille characters based on character-region search. Results reported for double-sided Braille documents were around 96.5% correct (recognition of recto Braille characters).

The approach of Mennens et al. [5], also using a flatbed scanner, appeared in parallel in the literature. Higher quality images are obtained (200dpi at 256 greylevels) and dots are located using image-based operations (performing correlation with a particular mask). To identify Braille characters, a grid is placed on the image where Braille dots are expected to be. Results reported are good (99.75% correct Braille character recognition) on documents without major defects but the approach fails when distortions are present (due to the fixed grid).

The new approach described in this paper is a radical re-development of the approach of Ritchings et al. [6]. There are several improvements on both approaches

mentioned above, while the cost effectiveness and usability of the system (using standard equipment) remains at least as high. A major advantage of the new system is the robustness it achieves at each stage of the process, even with poor quality documents. Image-based operations are drastically reduced to a minimum (only an initial thresholding and labeling) and a flexible grid is used to recover from potential errors. Most importantly, it is the first system to perform contextual post-processing at the word level using knowledge of likely specific Braille production errors.

An overview of the new system is given in the next section, where the stages involved are described in more detail, each in a separate subsection. The paper concludes with presentation of results and discussion.

2 The System

The new system is applied to both single-sided and inter-point (double-sided) Braille documents. In the case of double-sided documents, the Braille characters on both sides are recognized from the image of only one side of the page. The system comprises the following stages.

First, an image of a Braille document page is obtained using a flatbed scanner. The image is thresholded so that only three classes of regions exist: dark, light and background. Having labeled each of the different types of regions, an initial identification of Braille dots is performed. A flexible grid of possible dot locations is then constructed and any dots that were not previously detected are recovered. Braille characters are subsequently recognized and, if the type of Braille is known (e.g., Grade 1) they are translated into the equivalent printed text. Finally, using this interpretation, a suitable dictionary and awareness of common Braille errors, post-processing is performed to correct wrongly recognized Braille characters.

Each stage is described below in more detail.

2.1 Scanned Braille Characteristics

In most low-cost scanners, the document page is illuminated from an offset angle. The direct implication for Braille documents is that the illumination of a protrusion in that page will not be even. The face of the protrusion, which is angled towards the light source, will be more brightly lit and the face of the protrusion angled away from the light source will be considerably less brightly lit. It is this property that can be exploited to enable the recognition of Braille documents.

The scanned Braille page appears with a mid-gray background, and for each protrusion and depression a highlight and shadow pair is present along the scanning direction (depressions are only present in double-sided documents). The order in which the shadow and highlight appear for each dot depends upon the model of scanner involved. Some models represent protrusions as shadow areas over highlight areas while other scanners produce the reverse. The scanner used with this system produces the former pattern and the possibility to reconfigure the system to work with other scan-

ners is provided. An example of a typical scanned double-sided Braille document can be seen in Fig. 2.

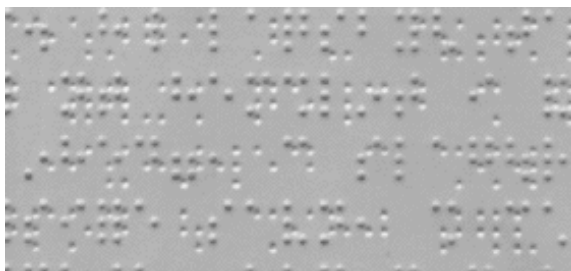


Fig. 2. An example of a scanned double-sided Braille document.

Experiments [5] have shown, and has been verified by the authors, that a scanning resolution of between 80dpi and 200dpi is the most appropriate for Braille documents. The system described here has been tested with images in the full range, however, the majority of the examples presented are 100dpi images.

The colour depth chosen for the images input to the system is 8-bit (greyscale). In actual fact the original scanned documents are in 24-bit colour. This can be useful to segment (in a potential pre-processing step) some paper defects, annotations or stamps introduced by archivists/librarians, for instance, if present.

2.2 Pre-processing

Since there are only three classes of useful information (shadows, light areas and background), a preprocessing step to reduce the greylevels in the image (to 3) is necessary. To cope with significant (in many cases) variations in lightness across the whole image, a local adaptive thresholding method was introduced.

The method works by dividing the image into 32×32 pixel regions (the window size is experimentally derived) and assesses whether each region contains whole dots, highlight(s) only, shadow(s) only, or just background. This assessment is based on a comparison of sets of ranges of greylevels observed in the region against equivalent ranges that are expected when a particular feature (dot, highlight or shadow) is present or not. In each of those four different cases, a different threshold (or a fixed value in the case of background regions) is applied to the pixels of the region.

The resulting image will have only black regions (corresponding to the shadows), white regions (corresponding to the highlights) and mid-grey (the majority, corresponding to the background). An example of a region of the image after this stage is shown in Fig. 4(b).

2.3 Initial Braille Dot Location

Braille dots manifest themselves in the image as white/black region pairs. The region order depends on whether the dot is a protrusion (recto) or a depression (protrusion on verso). Given a particular scanner model, we can define for the purpose of clarity in this paper that a protrusion will be represented in the image as a black region over a white one (and a depression will be represented with the two regions in reverse order).

The black and the white regions are not usually connected; they are separated by background pixels. However, it is not uncommon for regions of the same type to be connected. This situation happens only in double-sided documents, in which case the white region, for instance, of a protrusion is merged with the white region of a depression. This is one of the major problems encountered when attempting to identify dot-forming pairs (see below).

In the interest of efficiency, the image regions are labeled next and the subsequent processes in the system are carried on the connected component representation rather than on the image. There now exist two lists of connected components (bounding box co-ordinates), one for the black regions and one for the white regions.

Braille dots can be located by identifying vertically adjacent pairs of white/black regions. In double-sided documents, dots on both sides are identified at the same time. It should be pointed out that in the latter case it is not sufficient to locate a protrusion, say, by simply identifying a white region that lies below a black one. Due to the white regions and the black regions of protrusions and depressions being frequently merged, the dot locating method must be careful not to discard any black and white regions that have been used to describe a protrusion but that can also be used to describe a depression. An example of protrusions and depressions with merged back/white regions can be seen in Fig. 3.

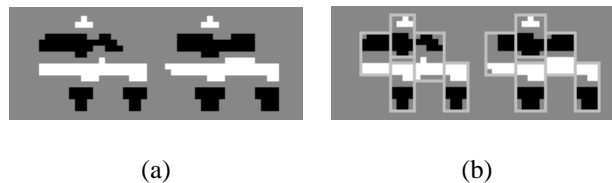


Fig. 3. (a) Example of merged components from depressions and protrusions, (b) with the corresponding dots (depressions/protrusions) outlined.

The algorithm to identify protrusions and depressions proceeds as follows. Each of the black connected components is analyzed in turn. If a white region exists above it within the expected limits (the expected size of the Braille dot in the given resolution), a depression has been found. Similarly, if a white region exists below it within those same limits, a protrusion has been found. Components that have been used for the location of a dot are marked as used and should not be considered again, unless their width is greater than the expected width of a Braille dot (at the given resolution). In that case, the wide component is split into two and only the part contributing to the dot will be marked as used, leaving the remainder to be used in the creation of the neighbouring dot (from the other side of the document).

The algorithm is cautious at this stage and only up to two dots on opposite sides of the document (a depression and a protrusion) will be identified, even in complex situations where combinations of more than two dots exist (e.g., the case in Fig. 3). Additional dots will be recovered with high confidence once topological information is also known (see below).

Upon completion of the algorithm two lists have been created, one storing the locations of protrusions and the other the locations of depressions (as the coordinates of a single point representing the top-left extremity of the upper—white or black—component). An example of located dots (protrusions in this case) is shown in Fig. 4(c).

2.4 Grid Formation

By definition, the placement of dots within a Braille character is regular. The space between adjacent characters in the same character line and between adjacent character lines is also regular (albeit small distortions and scanning skew may be present). Therefore, one could construct a grid whose intersections determine the possible position of dots on the page image. For double-sided documents, there will be two grids for each image, one for the protrusions (recto dots) and one for the depressions (verso dots). This is not a new idea, it has been used before [5] but in a fixed way that does not account for possible slight variations in character positioning in different lines.

The system described here constructs a relatively flexible grid by allowing variations in the positions of characters between different lines (i.e., the Braille characters need not be aligned in the vertical direction). The positions of all possible protrusions and depressions are calculated based on the dots identified so far. The grid-forming method is tolerant of wrongly recognised dots (in non-valid positions), of lines not containing any Braille characters and of characters with no dots in one of the two columns. The following process takes place for each of the lists of protrusions and depressions (separately).

First, rows of dots are identified by grouping together dots (points) that have the same (within a small tolerance) y-coordinates. This simple process is successful in identifying rows of dots even in the presence of small amounts of skew. For larger amounts of skew, it is straightforward (and fast) to perform a Hough transform on the dots to identify the precise orientation of the rows and “rotate” them so that they are horizontal—this correction is performed by adjusting the coordinates of dots in each of the two lists (protrusions and depressions).

Having identified the rows of Braille dots, a frequency histogram of the vertical distances between adjacent rows is calculated. The histogram should have two main peaks, one indicating the intra-character vertical distance between dots, and the other the inter-line (vertical) distance, i.e., the vertical distance between the bottom row of a Braille character line and the top row of the next.

Given the two vertical distances, consecutive pairs of rows of dots are examined. If their distance is judged to be the intra-character one they are labeled as rows of a specific character line. If, on the other hand, the vertical distance is judged to be the inter-line one the first one is labeled as the bottom row of one character line, and the

second one as the top row of the next line. It should be mentioned that the labeling process keeps track of the position of each row within a character line and labels each row of dots accordingly as the first, second or third row of a specific line.

Naturally, not all rows of dots will be labeled at this stage, as some Braille character lines happen not to have all three rows of dots present, therefore giving rise to larger vertical distances between their row(s) of dots and those of adjacent character lines. To identify the position of an unlabeled row within its character line (first, second or third row), the vertical distances between each row of the adjacent character lines and the unlabeled row are examined. If a distance is found to be a multiple of the inter-line one, the unlabeled row will receive the same label as the other row involved. For instance if the vertical distance between the second row of the character line above and the unlabeled row is a multiple of the inter-line distance, the unlabeled row will be identified as the second row within the current character line.

Having identified all existing rows of characters, a similar process takes place to identify and label the columns of dots within Braille characters. In this case the frequency histogram of horizontal distances is used to identify intra-character and inter-character column distances.

At the end of both the row and the column labeling processes, the Braille dot grid (a complete description of all possible dot locations in the image) is constructed. This construction process takes into account the “missing” dots in addition to those identified from the white and black components.

The resulting grid, therefore, is a list of all the possible valid dot occurrences, given the identified valid character lines. Since care has been taken in the process to allow for possible deformations, the grid is not a fixed global grid but it is flexible in taking into account local variations between lines of characters (i.e., independent column positions in each line). This flexibility is ensured by constructing a dot grid for each Braille character line and then joining the individual grids to form one for the whole page. An example visualization of the grid for a given portion of a character line (with gridline intersections indicating possible dot locations in that character line) is shown in Fig. 4(d).

2.5 Additional Braille Dot Recovery

The initial Braille dot location method (Sec. 2.3) successfully identifies most dots (on average about 98%). One of the reasons for the missed dots is that the initial location process is cautious in situations where black and white regions have been merged across more than two dots (recto and verso). At that stage it was necessary to be cautious, as there usually exist considerable amounts of noise regions that could be interpreted as parts of dots. An example of the situation where a series of dots from both sides of the document have merged regions is shown in Fig. 3(b) with the dots outlined.

Having the list of valid dot positions in the image (grid), the system can now verify the dots previously detected and, more importantly, attempt to recover dots that were previously missed.

To identify valid missed dots, each of the vacant locations in each of the two grids (separately for protrusions and depressions) is examined. Similarly to the initial dot location process, the system searches for a valid black/white or white/black component arrangement (depending on which of the two grids is used). In this case, however, the size/distance constraints are more relaxed (although care is taken not to use black or white components corresponding to noise—typically very small components comprising less than 3–4 pixels).

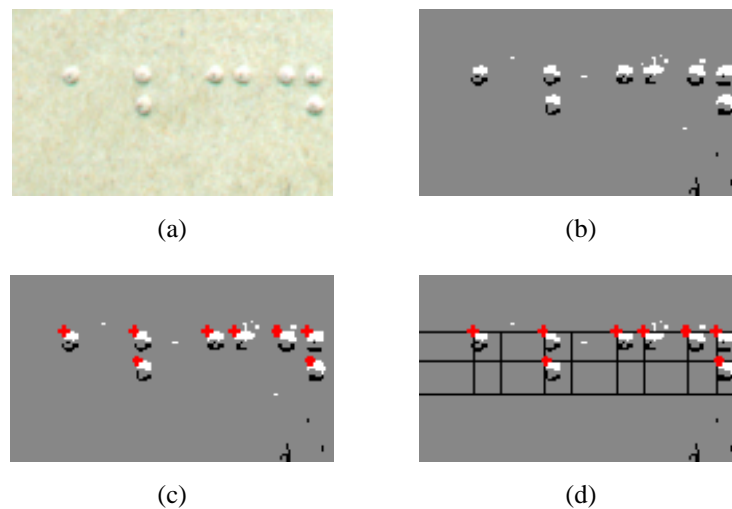


Fig. 4. (a) Part of a scanned Braille document. (b) Result of pre-processing. (c) Result of initial dot location. (d) Illustration of possible valid dot positions (grid).

2.6 Braille Character Recognition

The two identified grids of protrusions and depressions (Sec. 2.4) contain information not only on the valid positions of dots on the whole page but also on the groupings of columns (within lines) that can form Braille characters. Having identified all possible valid dots, the system at this stage segments the Braille characters in the image (both sides of the page) and recognizes the corresponding codes. A code in this case is a unique identifier of each Braille character. This digital representation of a Braille character takes the form of a 6-bit word (or 8 bits in 8-point cell Braille), each bit position corresponding to the dot position in the Braille character.

In each of the two grids, each grid point within a character is examined. When a dot is found at that point the bit corresponding to that position in the Braille character is switched on ('1'), otherwise it is left off ('0'). An illustration of the digital representation of the Braille characters found in the example of Fig. 4(a) can be seen in Fig. 5.

2.7 Braille Character Interpretation

At this stage the Braille character codes are translated into their textual equivalent. In contrast to traditional OCR, this is not straightforward as there is significant ambiguity depending on the type of Braille used. As mentioned in Sec. 1, the same Braille character will correspond to a single printed character in Grade 1 Braille, but may possibly correspond to a string of characters in Grade 2 Braille (assuming at least that the language of writing is known) or a non-character symbol in other types of Braille.













Character		Code	Grade 1																
1		4	A																
2		5																	
3		6																	
		<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td> </tr> </table>	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	1	
8	7	6	5	4	3	2	1												
0	0	0	0	0	0	0	1												
1		4	B																
2		5																	
3		6																	
		<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td> </tr> </table>	8	7	6	5	4	3	2	1	0	0	0	0	0	0	1	1	
8	7	6	5	4	3	2	1												
0	0	0	0	0	0	1	1												
1		4	C																
2		5																	
3		6																	
		<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td> </tr> </table>	8	7	6	5	4	3	2	1	0	0	0	0	1	0	0	1	
8	7	6	5	4	3	2	1												
0	0	0	0	1	0	0	1												
1		4	D																
2		5																	
3		6																	
		<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td> </tr> </table>	8	7	6	5	4	3	2	1	0	0	0	1	1	0	0	1	
8	7	6	5	4	3	2	1												
0	0	0	1	1	0	0	1												

Fig. 5. Illustration of Braille codes and Grade 1 English translation of the characters in Fig. 4.

Unfortunately, the type of Braille and originating language is not indicated within the Braille document. If this information is obtained from a different source (e.g., a librarian), there is still not a direct correspondence between Braille codes and printed characters. In Grade 2 Braille the corresponding string of characters will depend on the position of the Braille character within a line (relative to other Braille characters). Even in the simpler case of Grade 1 Braille, some Braille codes have dual meanings depending on context. For instance, a given code may represent a single letter unless it is preceded by a specific code, which signifies that the following codes should be interpreted as digits [1].

To demonstrate the functionality of this and further stages, the system currently supports Grade 1 English Braille. Further types of Braille can be supported by implementing suitable extensions. An illustration of the translation of Grade 1 English Braille codes to the corresponding printed characters is shown in the rightmost column in Fig. 5.

2.8 Character Validation and Correction

A unique characteristic of the system that further enhances its robustness is its provision for post-processing to recover from any recognition errors as well as from possible errors during the production of the Braille document (mistyping errors). Here, knowledge of Braille is used as well as of the language of the writing. This can be thought of as the equivalent of prioritizing alternative recognition results depending on a ranking of the most common spelling mistakes in conjunction with existing character features.

In contrast to printed character recognition (where there are distinct structural differences and similarities between characters), the nature of Braille characters does not provide with equivalently suitable clues. A given Braille character can be easily converted to another (similarly valid) one by the presence or absence of one or more dots at the pre-specified locations. It is not therefore possible to identify a suitable alternative to a “wrong” Braille character simply by examining its form (code).

On the other hand, examining only the translated text does not help in the correct ranking of the potential correct alternatives of wrong words. This is because, given a suspect Braille character, not all potential printed alternatives may be possible to be derived. Instead, only those printed characters that can result from a (limited) incremental addition of missed, or deletion of detected, dots of the given Braille character should be considered.

The system works at the word-level taking into account the structure of the underlying Braille characters (that give rise to each printed word in consideration). First, the Braille codes of the entire document are translated into the appropriate printed textual representation (here this is English, as explained in the previous section). The resulting text is then split into words. Each of these words is then looked up in a dictionary. Words that are found in the dictionary are left unchanged. Those that are not found in the dictionary are subjected to two stages of transformation in order to identify a suitable replacement.

In the first stage, the system searches (in the English dictionary) for words that are similar to the detected word but with one dot altered in the constituent Braille codes. To do this, n (where n is the number of dots in the word) copies of the word (as currently expressed in Braille codes) are created and in each one, a different dot is reversed from its current state (i.e., an existing dot is removed or an additional dot is introduced). These words are then evaluated. If a match is found in the dictionary for just one of the new words, the original incorrect word is replaced with that. If more than one match is found in the dictionary, they are presented as alternative options.

If no suitable matches are found in the first stage, the post-processing method proceeds to a second stage where, in a similar manner to the previous stage, $n (n - 1)$ copies of each incorrect word are created. Each alternative word is derived from the original by reversing the state of *two* dots in the underlying Braille codes. These words (the translation from the Braille codes) are then looked up in the dictionary and, depending on the number of correct matches, the original incorrect word is either directly replaced or (if more than one match is present) alternative options are listed.

5 Results and Concluding Remarks

The system has been tested with a wide variety of scanned Braille documents, both single and double sided, written in different languages and Braille types (including music), produced with different mechanical methods and scanned with different scanners. Due to the sheer variety of documents, it has not been possible to obtain textual ground truth. For our tests, ground truth was created manually by counting the number of Braille dots and characters in each document (both sides of a page if characters are present on the verso). As such, the general recognition results given here refer to the detection of dots and characters.

Due to the nature of Braille documents, it is expected for most documents to have some defects such as slight variations in background colour and a small number of dark spots (one type of paper imperfection). Such documents are referred to as being of *average* quality. Documents that are considered to be of progressively worse (referred to as being of *low* and *very low*) quality may have a larger number of dark spots, excessive paper grain and large variations in background colour. In addition, image quality depends on the extent of scanning artefacts, such as the effect of non-uniform illumination. These problems generally cause the lightness of the highlight part of Braille dots to be much less distinct from the page background and spurious regions to appear in the image that could be misidentified as the highlight or shadow parts of dots.

Overall, on single-sided documents with average defects 100% of the dots are correctly recognized. This rate drops slightly to 98.2% and 97.6% for low quality and very low quality documents, respectively. In terms of character recognition, on the same documents, the system achieves 99.9% for documents with average defects, 97.4% for low and 94.9% for very low quality documents.

On double-sided documents the overall dot detection performance of the system is at or slightly below 99% for both average and low quality (exhibiting excessive noise and detrimental illumination defects) documents. In terms of the character recognition performance, the system achieves 98.7% for average quality documents, 98% for low and 94% for very low quality documents.

Typical failure cases in the single-sided documents result from defects present in the image, such as worn (from prolonged reading) protrusions and paper defects (marks resembling valid dot components). In the case of double-sided documents, the additional problem of the merged dot components (white and/or black regions) is present as a potential cause of missed dots.

To put these results into perspective, the documents scanned by the authors using a standard scanner, all fall within the average difficulty category, where the system achieves near perfect results. The low and very low quality documents were obtained already scanned, some about a decade ago, using quite old scanner technology and as such, the authors are inclined to not consider them as relevant to the present and future performance of the system.

Moreover, the above results do not take into account (for reasons of clarity and to demonstrate the broad applicability of the system) the post-processing stage. While there are clear indications of performance improvement by including that stage, the dataset currently used does not lend itself to a representative and accurate quantifica-

tion of its performance (as explained in Sec. 2.8, the post-processing stage is not implemented for all the different types of Braille and languages presenting the dataset).

For comparison purposes, the dataset used by Mennens et al. [5] was obtained. This was limited to 7 documents and included cases in all three categories of quality grade. As Mennens et al. do not, however, specify for which images they obtain which specific results, the discussion here is limited to overall performance. The system presented here successfully recognized all documents with at least comparable results. Moreover, it is the authors' expert opinion that Mennens' et al. approach would fail in the very low quality documents, whereas the system described here achieves 94% success. In terms of the performance of the system by Ritchings et al. [6], the current system shows a significant improvement as a combination of the improved image quality as well as the enhanced robustness aspects introduced.

Overall, the system described in this paper has distinct advantages over previous approaches while it maintains a cost effective and visually-impaired-user-friendly realization. Robustness to cope with low quality scans and defective documents is built-in at different levels, from the initial thresholding through to the flexible Braille point grid construction. More significantly, though, a word-level post-processing stage is introduced that takes into account knowledge of both the language of the corresponding printed text and the unique characteristics of Braille character structure.

Further work is focused on eliminating some of the errors attributed to card (paper) defects and on expanding the translation capabilities to different types of Braille.

References

1. Antonacopoulos, A.: Automatic Reading of Braille Documents. In: H. Bunke, P.S.P. Wang (eds.): Handbook of Character Recognition and Document Image Analysis. World Scientific Publishing Company (1997) 703–728
2. Dubus, J.P., Benjelloun, M., Devlaminck, V., Wauquier, F., Altmayer, P.: Image Processing Techniques to Perform an Autonomous System to Translate Relief Braille into Black-Ink, Called: Lectobrilie, Proc. IEEE Engineering in Medicine and Biology Society 10th Annual Int. Conf., New Orleans (1988) 1584–1585
3. François, G., Calders, P.: The Reproduction of Braille Originals by Means of Optical Pattern Recognition. Proc. 5th Int. Workshop on Computer Braille Production, Heverlee (1985) 119–122
4. Hentzschel, T.W.: An Optical Braille Reading System, M.Sc. Dissertation, Faculty of Technology, University of Manchester (1992)
5. Mennens, J., van Tichelen, L., François, G., Engelen, J.J.: Optical Recognition of Braille Writing Using Standard Equipment. IEEE Trans. on Rehabilitation Engineering, vol. 2 no. 4 December (1994) 207–212
6. Ritchings, R.T., Antonacopoulos, A., Drakopoulos, D.: Analysis of Scanned Braille Documents, In: Dengel, A., Spitz, A.L. (eds.): Document Analysis Systems, World Scientific Publishing Company (1995) 413–421