CHAPTER 2

RESUMÉ OF RELATED LITERATURE AND PROFESSIONAL STUDIES

A. Related Literature

De-skewing

Image de-skewing is the process of straightening an image that has been scanned or photographed crookedly- that is an image that is slanting too far in one direction or one that is misaligned. Skew is an artifact that can occur in scanned images because of the camera being misaligned, imperfections in the scanning or surface, or simply because the paper was not placed completely flat when scanned. An example of a skewed image is shown in Figure 2.1. [Lao, 2009]

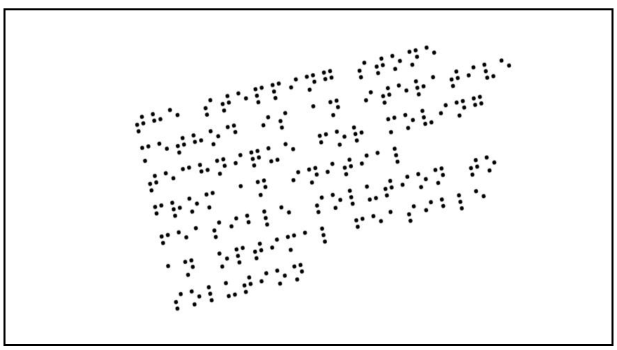

Figure 2.1
Skewed image tilted 13 degrees to the left

Double Sided 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 the 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 scanners is provided [Dengel et al., 2004] An example of a typical scanned double-sided Braille document can be seen in Figure 2.2.

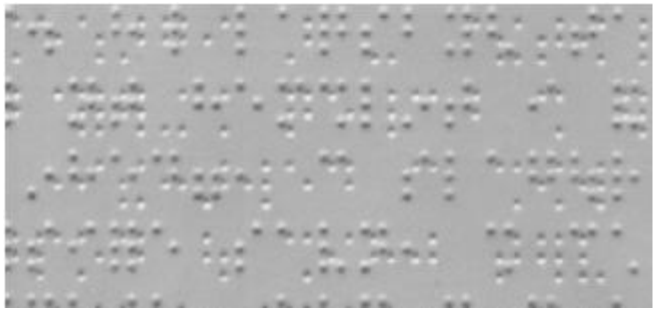

Figure 2.2
An example of scanned double-sided Braille document

Grade 2 Braille

Grade Two English Braille (America Edition) has 250 symbols for: letters, punctuation marks, composition signs, numerals, contractions, single-cell words, and short-form words [Charlie Web, 2005]. Grade 2 Braille was developed to reduce the size of books and make reading quicker. Other symbols are used to represent common letter combinations, for example in English 'OW', 'ER', and words such as 'AND' and 'FOR'. Combinations of two symbols are also used to represent some words, for example: 'THROUGH'. Some characters may change their meaning, depending on how they are spaced. [Sablé,2006]

Preprocessing

Since there are only three classes of useful information (shadows, light areas and background), a preprocessing step to reduce the gray levels in the image 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 32x32 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 gray levels 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 Figure 2.3.

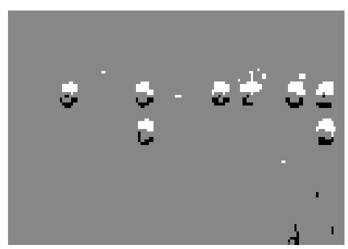

Figure 2.3
Example of a result of preprocessing

Thresholding

During the thresholding process, individual pixels in an image are marked as “object” pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as “background” pixels otherwise. This convention is known as threshold above. Variants include threshold below, which is opposite of threshold above; threshold inside, where a pixel is labeled "object" if its value is between two thresholds; and threshold outside, which is the opposite of threshold inside [Shapiro, et al., 2001]. Typically, an object pixel is given a value of “1” while a background pixel is given a value of “0.” Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's label. [www.wikipedia.com]

In an image (see Figure 2.4) that already underwent the process of thresholding, all image details are lost. The source image is scanned one pixel at a time and the pixels above the certain threshold are transformed into black pixels in their respective positions while the pixels below the threshold are transformed into white pixels, in effect the image is left with two pixel modes, black and white. A cluster of black pixels given a specific volume can represent a dot in a Braille cell.

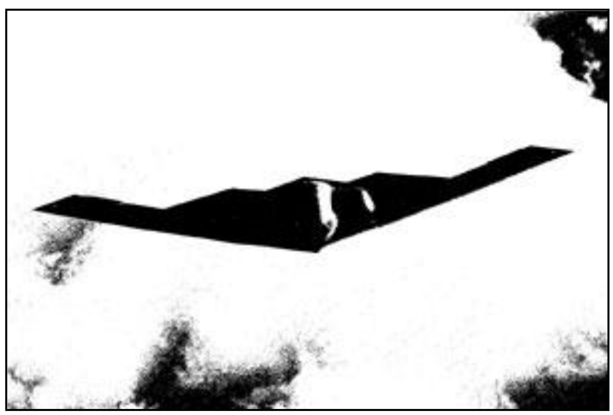

Figure 2.4
Example of thresholded image

B. Related Studies

Image pre-processing is an essential step during which errors that occurred while the images were taken are eliminated. Errors include noise, deformation, bad illumination or blurring. Image pre-processing can be used for image enhancement by reducing noise, sharpening images, or rotating a skewed page. The algorithms used differ from one system to another depending on the classification approach followed by researchers and developers. [Al-Salman et al., 2007]

In Optical Braille Recognition, the appropriate pre-processing algorithm must be chosen for the problems that occur in scanning a Braille document. The pre-processing algorithm in our study should focus on rotating skewed images.

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). Also, 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). [Antonacopoulos and Bridson, 2004]

A percentage of these Braille documents are in grade 2, which cannot be processed by the previous study.

One of the first approaches to use a flatbed scanner to appear in the literature is that of Ritchings et al. It is applied to both single and double-sided Braille documents, scanned at 100dpi at 16 grey levels (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. [Antonacopoulos and Bridson, 2004]

Attempts had been made to optically recognize embossed Braille using various methods. In 1998, Dubus and his team designed an algorithm called Lectobraille which translates relief Braille into an equivalent printed version on paper. Since then, research has built on knowledge of image processing techniques towards the goal of Braille to text translation. In 1993, Mennens and his team designed an optical recognition system which recognized Braille writing that was scanned using a commercially available scanner. The result was satisfactory with reasonably well formed Braille embossing. However, the system cannot handle deformation in the dot grid alignment. In 1999, Ng and his team approached the problem using boundary detection techniques to translate Braille into English or Chinese. The recognition rates were good, however no mention was made of grid deformed input, nor its efficiency. In 2001, Murray and Dais designed a handheld device which handles the scanning as well as the translation. Since the user is in control of the scanning orientation, and only a small segment is scanned at each instance, grid deformation is not a major concern, and a simpler algorithm was used to yield efficient, real-time translation of Braille characters. In 2003, Morgavi and Morando published a paper where they described the use of a hybrid system using neural network to solve the recognition problem. The paper also provides a means of measuring accuracy in Braille recognition, and the results show the system can handle a larger degree of image degradation compared with the algorithms that use more conventional and rigid image processing techniques. [Wong et al., 2004]

This reveals the evolution of Optical Braille Recognition through the years. The different pros and cons of these previous studies showed which problems our study might also face. This gives us a wide array of algorithms that can be potentially useful.

Horizontal illusory clues originate from the arrangements of characters into words and lines. Henceforth we shall indicate with horizontal the clues belonging to this dominant direction. The algorithm proposed in this paper to extract the horizontal illusory lines is summarised as follows. A pre-processing stage binarizes the input image, turning it into blobs representing either single characters or (portion of) words or lines, depending upon the font size and the resolution considered. These blobs are divided into elongated (major axis longer than thrice the minor axis) or compact. A pairwise saliency measure is computed for pairs of neighbouring blobs that represent how likely they are to be part of a text line. A network is then built using the blobs and their associations. The network then transverses to extract salient linear groups of blobs which constitute the illusory horizontal clues. Isolate elongate blobs are also considered as individual clues. In the following sections we shall describe these stages is more detail. [Pilu, 2001]

This study deals with de-skewing English alphabet letters, the concept can also be used in de-skewing an already skewed Braille document.

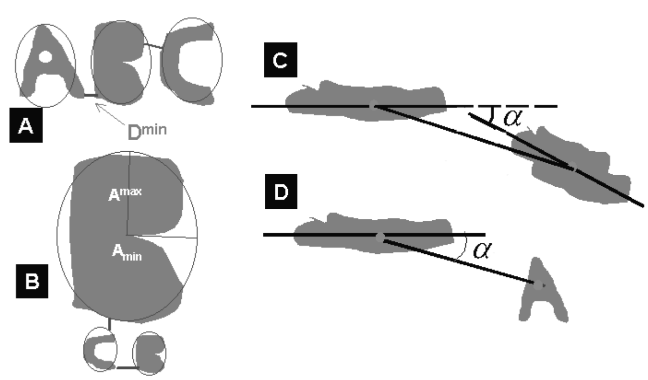

Figure 2.5
Illustration of the quantities used for the determination of the pairwise blob saliency

The reduction of noise (Figure 2.7) is valuable in correctly recognizing the Braille dots. Figure 2.6 shows a sample Braille image that underwent thresholding. From there, the image noises are more visible so it should undergo the process of noise reduction to be able to recognize the Braille dots correctly. Also, correcting the image skews is beneficial to preserve the data in the scanned image. However, the cropping process of the de-skewing phase sometimes fails due to some poor edge detection problems. The problem of recognizing the Braille characters was also solved by modifying the pattern generation algorithm. [Lao, 2009]

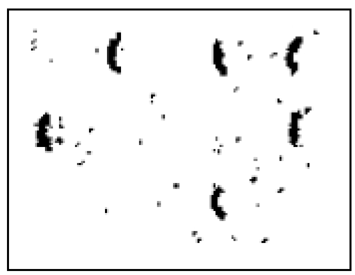

Figure 2.6
Scanned Braille image with noise

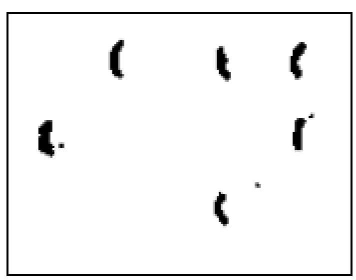

Figure 2.7
Noise reduction of Braille scanned image