Morphological filter for text extraction from textured background

Oleg G. Okun

Machine Vision and Intelligent Systems Group, Infotech Oulu and Department of Electrical Engineering, P.O.Box 4500, FIN-90014 University of Oulu, Finland

ABSTRACT

A new method for text extraction from binary images with a textured background is proposed. Text extraction in such a case is very important for successful character recognition, because many character recognition methods expect text printed on a uniform (and typically white) background and their performance significantly degrades if this condition is not satisfied. The methods that have been already proposed to solve this problem, attempt to extract primitives or elements composing the textured background in order to separate text from them. From experiments with commercial character recognition software we observed that such an approach easily leads to the significant growth of errors in character recognition because of degradations in extracted characters, introduced during text extraction. On the other hand, it is hardly possible to reconstruct (more or less precisely) the degraded characters without knowing their class labels and this information is not yet available at this stage. In contrast, we explore another approach similar to symbolic compression of text, which is implemented as a morphological filter using the top-hat transform. This approach detects characters having similar shapes from an original image and it thus avoids character degradations. As a result, the accuracy of character recognition can be improved.

Keywords: Mathematical morphology, top-hat transform, text extraction, textured background, character recognition

1. INTRODUCTION

Optical character recognition (OCR) made significant advances during past dozen years. For example, multfont and multisize OCR engines are nowadays not surprising anymore. There are also a lot of publications on this topic and several commercial products were released. In our opinion, a lot of efforts are still necessary in order to produce software capable of reading text as humans do.

One difficult problem not considered by many researchers and developers of commercial software concerns text printed on a textured background. Such kind of background usually decorates important text information intended to attract reader's attention. An example is given in Fig. 1, where text is overlaid on the texture background consisting of “bricks”. The image in Fig. 1 is grayscale in order to better show texture elements composing the background. Results of binarization of this image with different global thresholds are presented in Figs. 2-4. As one can see, it was impossible to eliminate the textured background without significant degradations of characters. As a result, OCR readers will fail to recognize such text.

In this paper, we propose a new approach to deal with text extraction from the textured background. It employs a morphological filter with the top-hat transform as a basis. This filter is used to find characters belonging to the same class rather than to remove background elements as in other approaches. An operation implemented by the filter is similar to one done in symbolic compression used to encode text data. Thanks to this operation, shape distortions are generally avoided during text extraction and therefore OCR readers can recognize extracted text with the higher accuracy.

The paper has the following structure. Section 2 analyzes previous work done in the field. Section 3 gives a short introduction to symbolic compression. Section 4 defines the morphological filter emulating symbolic compression. Section 5 introduces our method of text extraction. Section 6 describes several experiments with our method and Section 7 concludes the paper.

oleg@ee.oulu.fi; phone: +358 8 553 2898; fax: +358 8 553 2612; http://www.ee.oulu.fi/research/imag/document/
Figure 1. Example of text overlaid on a textured background.

Figure 2. Binarization of the image in Fig. 1 with a global threshold 76.

Figure 3. Binarization of the image in Fig. 1 with a global threshold 178.

Figure 4. Binarization of the image in Fig. 1 with a global threshold 229.
2. PREVIOUS WORK

Text extraction from the textured background is a particular case of a more general problem of text extraction from a complex (for example, noisy, non-uniformly illuminated or irregular) background. Several attempts were made to solve the problem.\textsuperscript{2,7}

The paper\textsuperscript{2} rather aims at a localization of areas containing text printed on the textured background than in text extraction. To do so, it relies on the stationary first order Hidden Markov Models.

Character extraction from the periodic textured background was studied in Ref. 3 by applying various combinations of binary morphological operations with 1D structuring elements. The authors investigated the case of a small ratio of the width of text characters to the width of background symbols. They view the task as background elimination with simultaneous minimization of shape distortions of the characters by reconstruction. Although they managed to eliminate the periodic background, but it seems to be questionable whether extracted characters are well reconstructed (for OCR) or not. To evaluate results, their visual (that is, subjective) inspection was only used. As noted in Ref. 4, defects in the periodic background cause this method to fail to completely reduce the background, while numerous morphological operations make it quite time-consuming. These facts significantly limit its usefulness.

To remedy these problems, optimal Boolean filters are employed in Ref. 4. These filters treat the textured background as noise so that they can serve either noise reduction or background elimination. To design such filters, one needs to have an “uncorrupted” image (without textured or noisy background) in addition to the input “corrupted” image. The optimal Boolean filters are determined during training when the “uncorrupted” and “corrupted” images are compared. If \( n = mn \) is the number of pixels in the filter window, then there are \( 2^n \) different Boolean functions (provided that pixel values can be either 0 or 1), each representing a particular pixel configuration. For each configuration, the cost function defined as the difference between the number of cases when a given configuration was a part of noise and the number of cases when it was a part of characters, is computed. If the cost function is negative, this configuration belongs to characters and should not be filtered out. If the cost function is positive, it indicates to noise. As a result of training, all configurations are divided into two sets whose sizes are not necessarily equal. Once have been trained on a particular background, the filters can be used for text extraction from such kind of background without retraining. This method leads to faster processing and it produces better (at least visually in opinion of the authors) results than the method in Ref. 3. Also defects in the periodicity of the background do not affect its performance. However, it requires “uncorrupted” images for training for each background type. It is unclear how it will perform when the width of texture elements will be approximately equal to the width of characters. Finally, its speed can be only fast if \( m \) is small, say 2 or 3, because in this case, look-up tables can be used.

All in Ref. 5 analyzed the case when the width of characters is larger than the width of background elements. He argued that the morphological opening with a 3x3 structuring element will be enough to extract text in this case. He also observed that background elimination has a side effect: the OCR accuracy decreases for text printed on the uniform (usually white) background. This led him to conclude that the background elimination should be only done for text on the textured background, while it should not affect regions containing text on the uniform background. It means that document regions should be first separated into two classes. To this end, an original image was divided into small non-overlapping tiles or blocks and neural network trained with the backpropagation learning algorithm was chosen to classify tiles. This method is the only one which used OCR to verify the “quality” of extracted characters, because humans can read text that OCR readers may be unable to read. Thus, it provides more objective evaluation of text extraction results than other methods, but it is feasible only if the width of characters exceeds the width of background elements.

Text extraction from irregular backgrounds is considered in Ref. 6. Examples of such backgrounds can be found in bank checks. In this case, emphasis is not on the detection of background elements but on the detection of characters that are modeled as consisting of strokes of a predefined width. A double-edge feature determined by using morphological operations identifies a stroke. Although this method can deal with more complex backgrounds than others, it is mainly tied to the specific images of bank checks and it is unclear what will happen if background elements are character-like, e.g., \( \text{"@"} \).

Edge features are also employed in text detection from headlines of Japanese newspapers.\textsuperscript{7} The background of these headlines is composed of geometric patterns characterized by more edges and higher gradient magnitudes than the characters have. Since it relies on the gradient magnitudes of edges, this method is only intended to process grayscale images and it is not applicable for binary ones.
Based on the analysis of the methods in this section, the following conclusions can be drawn. Almost all methods “understand” text extraction from a textured background as a removal of background elements from an image. This in turn unavoidably leads to character degradations introduced in this process, because characters are spatially overlapped with background elements. The degradations in turn cause a high error rate in OCR. In fact, almost all methods, except for one in Ref. 5, did not use OCR to verify text extraction results. It is rather strange, because extracted characters should be eventually fed to recognition. However, in Ref. 5 the background is simple and the usefulness of the method is therefore limited.

In this paper, we propose an alternative approach. It tries to find characters embedded in the background instead of extracting and removing the background elements. In addition, it does not distort the characters so that their reconstruction is unnecessary. In fact, it is closely related to the technique called symbolic compression.

3. SYMBOLIC COMPRESSION

Symbolic compression is intended to encode binary text images and to perform OCR. This technique is an integrated part of the JPEG2 standard. Symbolic compression can be viewed as searching for similar characters and replacing them with one character being a prototype of a whole class. That is, the redundancy caused by repeatedly appearing characters in the image can be reduced and compression is achieved.

If most characters are isolated in the image, connected component analysis is the first step in symbolic compression. For connected scripts, where a whole word forms one connected component, segmentation into characters is first necessary. The first component is considered to be a prototype for the first class and it is compared to all other components. When a match between the prototype and observed component is found, the latter is assigned to the class, while the former is refined to incorporate new data. If no match exists, the observed component forms the prototype for the next class, etc. In the end of this process, a set of prototypes is generated. Characters denoting the same object (say ‘a’) and having the same font size and style are typically clustered together, though it is still possible that some of such characters are assigned to different classes, depending on a matching algorithm. Examples of such algorithms are simple XOR, weighted XOR, Boolean AND-NOT. Errors are also unavoidable as demonstrated in Section 4, but they can be corrected.

As reported in Ref. 10, symbolic compression is typically twice as efficient as JPEG1 compression, which in turn is nearly twice as good as CCITT-G4.

OCR using symbolic compression may be slower that most isolated character recognition methods (when characters are recognized one-by-one), since it involves extensive bitmap comparisons. The matching method is of significant importance, too. However, symbolic compression is adaptive to new fonts and degradations and it does not require learning. It is what made this technique attractive for our task, because we cannot determine the font style and size of the characters embedded in the textured background.

We found that symbolic compression can be effectively emulated with a morphological shape band-pass filter, because morphological operations can be fast when implemented in special purpose hardware.

4. MORPHOLOGICAL SHAPE BAND-PASS FILTER

This filter is characterized by two shape patterns, where one of them is larger than the other. It is applied to a binary image and passes only those image objects which are larger than the smaller shape pattern but smaller than the larger shape pattern. That is, the objects of a predefined shape and size are only allowed to pass. When applied to a text image, it only detects characters of a specific font style and size. Because of its band-pass characteristic, it can capture important shape details shared by many characters belonging to the same class. At the same moment, it is tolerant to small shape distortions unavoidable in real world images.

The shape band-pass filter is defined as follows. Let $A$, $B_1$, and $B_2$ denote an input binary image and two shape patterns ($B_1 \subseteq B_2$), respectively. The shape band-pass filter is defined as

\[ A \triangle (A \circ B_2) \circ B_1, \]

where ‘ $\triangle$ ’ stands for set difference and ‘ $\circ$ ’ means the binary opening.
\((A \backslash A \circ B_2)\) is the top-hat transform and it passes the objects in \(A\) which are not larger than \(B_2\). The top-hat transform followed by the opening with \(B_1\) gives the objects larger than \(B_1\) but not larger than \(B_2\). As stated in Ref. 15, the defined filter is translation-invariant and idempotent.

The tolerance to distortions is controlled by choosing appropriate \(B_1\) and \(B_2\). To detect a particular character, \(B_2\) should be larger than this character, whereas \(B_1\) should be smaller than it. It means that \(B_1\) should be included in all representatives of the character in \(A\) and \(B_2\) should include all of them.

An application of the shape band-pass filter is shown in Fig. 5, where all the characters o’s and a’s in Fig. 5(a) are detected in Figs. 5(b) and 5(c), respectively. In the first case, the ‘o’ in the word ‘To’ was used as \(B_2\), whereas in the second case, the ‘a’ in the word ‘Target’ served the same purpose. The shape pattern \(B_1\) was created by eroding \(B_2\) with a 3x3 structuring element. However, Fig. 5 contains rather ideal characters.

In reality, characters belonging to the same class and having the same font size and style may often split into several different clusters. For example, see Fig. 6, where there are 5 appearances of the ligature ‘ar’. A template to be searched was the ‘ar’ in Fig. 6(b). Results are given in Fig. 6(c) (template itself is not shown). Here, 3 appearances are correctly detected in the words ‘Maryland’, ‘Park’ (upper-right corner), ‘Maryland’; one appearance was missed in the word ‘Park’ (bottom of the image); 2 appearances were false positives (words ‘Bertrand’ and ‘Reinhard’ are printed in bold); 3 appearances were erroneous when ‘an’ and ‘am’ in the words ‘Hamburg’, ‘Japan’, and ‘Maryland’ were confused with ‘ar’.

5. OUR METHOD

By keeping in mind the results in Fig. 6, we propose a method intended to extract text in binary document images. We assume that such images can be obtained by binarizing color or grayscale images. We also assume that background elements will become disconnected as a result of the binarization. Though the background is not necessarily extracted entirely from the text, it is quite obvious that the text and its background should not form a single connected component. A typical example of the image before and after global binarization is given in Figs. 1 and 3. Another assumption is that the number of different text symbols that can appear is not very large, like it happens in many European languages.

Our method consists of the following steps:

1. Detect connected components in the input image.
2. Separate them into character candidates and background candidates.
3. For each character candidate, do:
   (a) Find other patterns in the image, which are similar to it and assign them to a cluster.
   (b) Use all found patterns to produce a cluster prototype.
   (c) Generate the shape patterns \(B_1\) and \(B_2\) from the prototype.
   (d) Apply the shape band-pass filter to the input image.
   (e) Add a result of the previous operation to an output image.
4. Perform OCR on the output image (optional).

Because of the assumption that text and its background in the binarized image consist of many connected components, characters are already partly separated from the textured background. In many cases, simple geometrical features (area, sizes of a bounding box, density of black pixels, etc.) derived from connected component analysis are sufficient to perform rough separation into character-like and background-like components. When doing it, we rely on the fact that the elements composing the textured background are either much thinner or much smaller than the characters as is in many practical situations; otherwise even human would have problems to read text.

For each character candidate (template), we search for other patterns similar to it by using a conditional erosion-like operation. The similarity is computed for each pixel and it is defined as

\[
s = \frac{N_{\text{template+pattern}}}{N_{\text{template}}},
\]
Cross-Correlation Used To Locate A Known Target in an Image

(a) Binary image.

(b) Extraction results of 'o'.

(c) Extraction results of 'a'.

Figure 5. Application of the morphological filter to a synthetic image.
Figure 6. Application of the morphological filter to a real image.
where \( N_{\text{1,template}} \) is the total number of 1-pixels in the template and \( N_{\text{1,template-pattern}} \) is the number of times when 1-pixels were found in the same position in the template and in the pattern under consideration.

It means that \( s \) can vary from 0 (complete dissimilarity) to 1 (complete similarity). If \( s \) is larger than or equal to a predefined threshold, a given pixel is set to 1, otherwise to 0. After this operation, 1-pixels often form blobs. In each blob, we detect one pixel with maximal \( s \) and others pixels are ignored. The dilation with the template as a structuring element is then applied to the remaining pixels and the resulting image is ANDed with the original binary image. As a result, only the characters similar to the template remain in the image.

Bitmaps of all characters detected in such a manner are used in order to produce a prototype for the whole cluster of similar patterns. A 1-pixel in a given position retains its value in the prototype only if values of more than 50% of pixels in the same position in all patterns are equal to 1. This operation leaves the most common parts of all characters and filters out unimportant details. The background elements usually tend to overlap characters belonging to the same cluster in different places so that many background pixels are eliminated during prototype generation.

The prototype is then chosen to be the shape pattern \( B_2 \), whereas the shape pattern \( B_1 \) is obtained by eroding the prototype with the same element. Finally, the shape band-pass filter is applied to the original binary image in order to locate all instances of the current template. Here, the ordinary erosion is replaced with the conditional erosion as described above to tolerate shape degradations.

The found instances are added to an output image which accumulates results produced with each character candidate. When all character candidates are picked, OCR is applied to the output image.

6. EXPERIMENTS

To verify our approach, we decided to apply OCR to the extracted characters. Commercial software used was a component of PageKeeper Standard 3.0 which belongs to a family of document management software created by Caere Corporation. A recognition engine is OmniPage Pro 9.0 made by Caere, too.

Here are recommendations concerning recognition capabilities of the OCR engine and found in the documentation provided with PageKeeper:

- Use high-quality images to obtain the best OCR accuracy. With lesser-quality pages, the accuracy may be poorer.
- Characters distorted by marks will be unrecognizable.
- Characters should be separated from each other.
- Text in page images should not be underlined. It is difficult to recognize underlined text, because the underline changes the shape of descenders on the letters q, g, y, p, and j.

As one can see, these recommendations, though reasonable for text printed on a plain white background, are not acceptable for text printed on a complex textured background. The OCR results when using the recognition engine of PageKeeper for the image in Fig. 5(a) (but its resolution was increased to 300 dpi, because the engine was unable to find text at a lower resolution) are:

Cross-Correlation Use! To Locate A Known Target in an Image.

Two rejections in the words ‘Correlation’ and ‘Target’ were because of junctions between characters.

Now we will present results for the same image but with the artificially generated textured background when the shape band-pass filter as described in\(^{15}\) was applied to extract text (see Figs. 7-8). A background element was ‘@’. Its bitmap was used as \( B_2 \). Fig. 8 displays results of the top-hat transform, because the background element was too thin to be eroded and the opening with \( B_1 \) was not done. No attempts to reconstruct characters were made, because in our opinion, it would be difficult to obtain the desirable image quality with any reconstruction technique at this stage.

The OCR generated output was
Cross-Correlation Used To Locate A Known Target in an Image.

\[ \text{Cross-Correlation Used To Locate A Known Target in an Image.} \]

Figure 7. Image in Fig. 5(a) with background elements of 1 pt in size.

Figure 8. Text extraction results for the image in Fig. 7 with the morphological filter.

Cross-Correlation Tested To Identify A Known Target in an Image.

The OCR accuracy significantly fell compared to the case of the white background. It indicates that a method aiming at background reduction from the image may not be so effective as it seems without feeding results to OCR.

The result obtained with our approach is given in Fig. 9. The similarity threshold for the conditional erosion was 0.7. The OCR output was

Cross-Correlation Used To Locate A Known Target in an Image.

Despite of many rejections, it can be considered to be better than the previous one, because in many cases the reason was very thin junctions between adjacent characters. These junctions could be easily broken if the OCR engine were more sophisticated. Results with other test images also indicated that our method gave the better accuracy than the method using the shape band-pass filter.

7. CONCLUSION

In this paper, we proposed a novel approach to text extraction from the textured background. It is based on the morphological shape band-pass filter using two shape patterns (structuring elements), one of them being larger than another. This filter implements the top-hat transform with the larger structuring element followed by the opening with the smaller structuring element in order to detect all patterns in a binary image, which are larger than the smaller element but smaller than the larger element. In contrast to many other methods, our method does not
Cross-Correlation Used
To Locate A Known
Target in an Image

Figure 9. Text extraction results for the image in Fig. 7 by using our approach.

attempt to eliminate background elements from the image so that the characters are not degraded and therefore
the OCR accuracy is not negatively affected. We processed several binary images and got promising results, but
large-scale testing is still necessary in order to obtain more reliable results and to figure out possible bottlenecks of
our approach.

Unfortunately, not every document image may be successfully binarized. For example, images of bank checks
need to be grayscale, because this will facilitate their analysis. In this case, the grayscale or fuzzy morphology for
text extraction would be a better choice. For example, it is well known that the grayscale or fuzzy dilation shrinks
dark objects and enlarges bright ones, while the grayscale or fuzzy opening removes small bright objects without
enlarging dark objects. This fact was exploited in Ref. 16 for the background removal of bank and traveler's checks
by using the fuzzy morphology. Our future research will concentrate on images with grayscale and color backgrounds.

ACKNOWLEDGMENTS
This work was partly supported by the Academy of Finland.

REFERENCES
2. J.-L. Chen, “A simplified approach to the HMM based texture analysis and its application to document seg-
3. S. Liang, M. Almadi, and M. Shridhar, “A morphological approach to text string extraction from regular
periodic overlapping text/background images,” Computer Vision, Graphics, and Image Processing: Graphical
International Conference on Pattern Recognition, Vienna, Austria, pp. 758–762, 1996.
6. X. Ye, M. Cheriet, and C. Y. Suen, “Model-based character extraction from complex backgrounds,” in Proceed-
ings of the 8th International Conference on Document Analysis and Recognition, Bangalore, India, pp. 511–514,
1999.
7. H. Otsuwa and T. Nakagawa, “A character image enhancement method from characters with various background
images,” in Proceedings of the 2nd International Conference on Document Analysis and Recognition, Tsukuba,
8. T. M. Breuel, “Modeling the sample distribution for clustering OCR,” in Document Recognition and Retrieval
9. J. D. Hobby and T. K. Ho, “Enhancing degraded document images via bitmap clustering and averaging,” in Proceed-
ings of the 4th International Conference on Document Analysis and Recognition, Ulm, Germany,