TEXT EXTRACTION FROM GREY SCALE PAGE IMAGES BY SIMPLE EDGE DETECTORS

Matti Pietikäinen and Oleg Okun

Machine Vision and Intelligent Systems Group, Infotech Oulu and Dept. of Electrical Engineering
P.O.Box 4500, FIN-90014 University of Oulu, FINLAND
e-mail: {mkp, oleg}@ee.oulu.fi

ABSTRACT

Text extraction is one of key tasks in document image analysis. Many methods proposed for this purpose require a binary image as an input. However, binarisation can sometimes lead to a loss of significant information contained in the grey scale or colour images, and it is also unsuitable if the original image is noisy or the background is complex. In this paper, we propose a simple method based on edge detectors, such as the Sobel operator, for document image filtering and text extraction. Promising results are obtained for degraded document images and for images with complex coloured and textured backgrounds.

1. INTRODUCTION

Page segmentation is a key area of research in document image analysis. Many researchers have tackled the page segmentation problem (see survey papers [1, 3, 5, 11]). In general, the existing methods can be roughly classified according to the type of the image (binary, grey scale, or colour) to be processed.

A subtask of page segmentation is text extraction [2, 8, 10, 12], which aims at localisation of text regions in the image, because text is usually the main source of information and accurate text detection can greatly facilitate optical character recognition.

Some text extraction methods only concentrate on binary images, because they are easier to analyse than other types of images. However, every image cannot be binarised. Among these are complex advertisements in magazines. They are mostly coloured and may often contain text embedded in pictures or text in a non-uniform complex background (see Figs. 1 and 2). For these reasons, page segmentation and text extraction methods based on image binarisation are likely to fail. Segmentation of the original colour images does not seem to be always a good alternative, either, because colour image processing typically requires sophisticated techniques for colour analysis and takes a lot of time.

We prefer to use colour-to-grey scale conversion prior to page segmentation or text extraction. This is a good compromise, because it does not require binarisation, the number of pixels is the same as it is for the binary image, and important intensity information is still preserved (Figs. 1 and 2 show grey scale images obtained after colour-to-grey scale conversion of the original colour images). However, text extraction from grey scale images with a complex background is still a challenge. Many existing approaches assume that the background is uniform and they usually show poor performance when this assumption is not satisfied.

Another, often ignored issue is text extraction when various degradations, for example, such as shown in Fig. 3, are present in the image. This degradation consists of a thick dark strip in the middle of the image and thinner dim strips near to the image borders, and it comes from copying-
In page images, degradations are often generated by the copying-scanning process of the original paper document. One very common degradation is caused by binding of thick books or magazines (see Fig. 4). This figure shows a typical position of such a document, labelled as ‘Book’, on a glass surface ‘Glass’ of a reading device. The thickness of the document leads to its binding when pressed, which, in turn, leads to gaps between ‘Glass’ and ‘Book’ within the intervals marked I and II. Partly due to this and partly due to the characteristics of the optical components of the reading devices, dark strips are often introduced in the image in the areas near to the image borders (intervals I) and between two adjacent pages (interval II).

One might notice that the problem would become simpler (at least, we can avoid nasty strips in the middle of the image) if we scan one page at a time. Of course, this would be a partial solution but at the expense of the time spent on scanning. It is, however, impossible to avoid such kind of degradations when one has only a paper copy of a book or magazine instead of the original. In this case, two pages are often copied at once.

A straightforward solution could be filtering before page segmentation. This might be true if we only had to deal with the degradations near to the image borders, because those places are quite far from the page regions containing important data. But degradations in the areas between two pages may be hard to accurately eliminate because of their spreading onto page regions close to these areas (see Fig. 5). Adaptive binarisation, for example, such as described in [13], may solve the problem, but possibly at the cost of character degradations. Non-uniform illumination compensation techniques, such as homomorphic filtering applied in [9] to license plate images, are not suitable, either, because such techniques often assume that the intensity values slowly vary across the image. But this is not true for the image in Fig. 3, where the background intensity is relatively rapidly changing in the critical regions. Of course, it is still possible to filter out noise there but at the expense of character degradations introduced.

Another approach, which we will follow, attempts to reduce noise in the critical areas during text extraction. To do this, we need to choose discriminative features of characters that will allow us to separate text from noisy background.
without actual noise removal.

By looking at characters (of the European script, at least), it is possible to conclude that directional edges are good candidates to be such features. Fig. 5 also shows that character edges in the critical areas are still quite clearly seen, despite of heavy noise. We assume that these edges can be detected, whereas noisy strips can be reduced because they do not contain strong edges. The next section describes our approach.

3. OUR METHOD

There are various operators for edge detection. We selected the Sobel operator because of its simplicity and wide use.

The input and output images are grey scale, but the latter one is divided into small $n \times n$ pixels blocks classified as non-informative (clean or noisy background, interior parts of large pictures in some cases) or as informative (possibly text, though we do not exclude that parts of pictures and graphics can be detected as well). Non-informative blocks have all pixel values set to zero, while informative blocks are copied as they are from the original image.

The value of $n$ depends on the resolution of the original image. We assumed that the maximum $n$ cannot be greater than $(\text{res}/25)^2$, where res - image resolution in dots or pixels per inch (dpi), res/25 - image resolution in pixels per mm, 2 - coefficient corresponding to 2 mm. It implies that under this constraint a block cannot occupy an area larger than $2 \times 2$ mm$^2$ on a paper. A small block size is important, because too large blocks would often capture data belonging to different classes and this would make the segmentation too inaccurate.

The proposed algorithm consists of the following steps:

**Step 1.** If the original is a colour image, convert it to grey scale, otherwise go at once to the next step.

**Step 2.** Set a threshold $t_1$ for suppressing weak edges or let it be automatically computed during the next step. Set a threshold $t_2$ for block classification into informative or non-informative. Choose an appropriate $n$.

**Step 3.** Apply the Sobel operator with threshold $t_1$ to the grey scale image $I$. As a result $E$ (edge) and $G$ (gradient magnitude) images are generated.

**Step 4.** Divide $I$ into $n \times n$ pixels blocks. Create a new image $B$, where each pixel represents the corresponding block of $I$.

**Step 5.** For each pixel of $B$ (and therefore for each block of $I$), compute a feature $R$ by using $E$ and $G$ obtained in Step 3. If $R \geq t_2$, a given block is informative, otherwise it is non-informative.

**Step 6.** Create a thresholded image $F$ from $B$, where $t_2$ is the threshold (optional).

**Step 7.** Perform morphological postprocessing of $F$ (optional).

Now let us consider these steps in detail. As mentioned above, a colour image when fed to the input should be first converted to a grey scale one. Independently of a colour space, we first obtain an image in the NTSC colour space. We chose the NTSC space because it separates luminance and chrominomination information. The Y-component of NTSC sufficiently well represents grey scale data. It is defined with the RGB coordinates R, G, and B (don’t confuse with $G$ and $B$ in the algorithm description) as

$$ Y = 0.299R + 0.587G + 0.114B. \quad (1) $$

The Sobel operator is then applied to detect edges, followed by thresholding and non-maximum suppression. A pixel is considered to belong to an edge if 1) its gradient magnitude is larger than the magnitudes of its 4-connected neighbours and 2) its gradient magnitude is larger than $t_1$ so that weak edges are cut off. The value of $t_1$ can be set by a user or automatically computed as

$$ t_1 = \sqrt{\frac{1}{(h - 1)(w - 1)} \sum_{i=1}^{h-1} \sum_{j=1}^{w-1} (G_x^2(i,j) + G_y^2(i,j))}, \quad (2) $$

where $h$ and $w$ are the height and width of the image and $G_x(i,j)$ and $G_y(i,j)$ are x- and y-components of the gradient magnitude $G(i,j) = \sqrt{G_x^2(i,j) + G_y^2(i,j)}$ for a pixel at (i,j).

After edge detection we have two images: a binary edge image $E$ (1 - edge, 0 - non-edge) and a grey scale gradient magnitude image $G$. Using both images, we classify blocks of the original image into informative or non-informative. The image $B$ represents a block partitioning of the original
image, where the pixel (i,j) in B corresponds to the block of pixels between coordinates \(i_{\text{start}}\) and \(i_{\text{end}}\) and \(j_{\text{start}}\) and \(j_{\text{end}}\), where \(i_{\text{start}} = (i - 1)n + 1\), \(i_{\text{end}} = in\), \(j_{\text{start}} = (j - 1)n + 1\), \(j_{\text{end}} = jn\) (we assume that \(i,j\) start from 1). If the sizes of \(I\) are \(hw\) pixels, then those of B are \((h/n)x(w/n)\) pixels.

For each pixel \((i,j)\) of \(B\) (and therefore each block of \(I\)), a special feature \(R\) is computed with one of the following equations:

\[
R = \sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{H}(G(i,j) - t_1) \mathcal{H}(E(i,j) - 1),
\]

(3)

\[
R = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} G(i,j) \mathcal{H}(G(i,j) - t_1) \mathcal{H}(E(i,j) - 1)}{n^2},
\]

(4)

\[
R = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} G(i,j) \mathcal{H}(G(i,j) - t_1) \mathcal{H}(E(i,j) - 1)}{\sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{H}(G(i,j) - t_1) \mathcal{H}(E(i,j) - 1)},
\]

(5)

where \(\mathcal{H}\) is the step function:

\[
\mathcal{H}(x - a) = \begin{cases} 
1 & \text{if } x \geq a, \\
0 & \text{if } x < a.
\end{cases}
\]

\(R\) defined by Eq. 3 is the number of edge pixels per block. \(R\) in Eq. 4 is the average gradient magnitude per pixel and \(R\) in Eq. 5 is the average gradient magnitude per edge pixel. Such definitions of \(R\) reflect the fact that the number of edge pixels and their gradient magnitudes are higher in many cases for informative than for non-informative blocks.

Using \(R\), every block of \(I\) is labelled either as informative or as non-informative based on the threshold \(t_2\) set by a user and determined from the histogram of \(R\) (see [6] for details).

After initial segmentation, segmentation results are smoothed by producing \(F\) from \(B\) according to the following threshold transformation:

\[
F(i,j) = \mathcal{H}(B(i,j) - t_2) \mathcal{H}(E(i,j) - 1).
\]

(7)

Morphological opening with a 1x3 structuring element followed by closing with the same element are then applied to \(F\). These operations remove narrow bridges between adjacent page regions and small spurious regions, and fill in narrow holes in regions. The resulting image shows which blocks in the output image are informative (for them the corresponding value of \(F\) is 1).

### 4. EXPERIMENTS

In order to test the method, we copied and scanned 20 images from a book. We also used 20 images of advertisements from the database of the University of Oulu [7]. Examples of images are shown in Figs. 1-3. Image sizes varied from 1,600x1,200 to 2,000x3,000 pixels.

#### 4.1. Degraded document page images

For scanned images, the following parameters were set: brightness – 123, contrast – 133 (both were the same for all images), and sharpening (it varied from image to image) – ‘none’, ‘light’, ‘normal’, ‘heavy’, and ‘extra-heavy’. In all experiments the value of \(n\) was 12 pixels and the value of \(t_1\) was automatically determined according to Eq. 2. The value of \(t_2\) was manually chosen from an admissible interval as described in [6].

The image of Fig. 3 after edge detection is shown in Fig. 6. It can be clearly seen that almost all noise was successfully reduced, because the edges of the dark noisy strips are weak or they are absent.

![Fig. 6: Edge image for the image in Fig. 3](image_url)

The text extraction results (after morphological postprocessing) are shown in Fig. 7. The non-informative blocks are painted black, while the informative ones are copied from the original image as they are. \(R\) defined by Eq. 5 was used as the discriminative feature and \(t_2\) was set to 0.065. The dark strip in the middle of the image was successfully reduced while not affecting characters. Even small portions of text such as the page numbers were correctly identified and they did not disappear after morphological postprocessing. Of course, large parts of the plot were also detected, but this seems to be unavoidable because of the simple feature employed.

An example showing what actually happened in the middle area of the image is given in Fig. 8, where a zoomed in fragment of the whole image is displayed. As before, eliminated noisy blocks have black colour. Though Fig. 8 displays only a small area, similar results are common for the whole image.

Good results were obtained with the other test images taken from a book. The detected text regions often had holes, but these corresponded to the background or they only included a small part of the text data. In any case, this is not a flaw, because the holes can be easily filled or just ignored when circumscribing a whole region.
4.2. Images with complex backgrounds

Now we will describe results obtained for the images shown in Figs. 1 and 2. The text extraction results are shown in Figs. 9-12. Figs. 9 and 11 were obtained with \( R \) computed by Eq. 3, while \( R \) defined in Eq. 5 was chosen for the images in Figs. 10 and 12. Morphological operations were not applied to these images, because they often deleted informative blocks detected on the borders of large-sized characters.

Because the image sizes were large, we divided them into narrow strips whose widths were equal to that of the whole image and detected edges for each strip. As a result, \( t_1 \) varied from strip to strip when computed automatically.

As one can see, the text was extracted quite accurately despite the non-uniform complex background. The results were typically better for \( R \) computed by Eqs. 4 or 5 than they were for \( R \) with Eq. 3 in sense that the false positive rate was smaller in the former case. This is not very surprising because Eqs. 4 and 5 employ both the number of edges and their gradient magnitudes, whereas Eq. 3 only relies on the edge density per block.

In addition to the text, parts of non-text regions were also detected by contributing to the false positive rate. This is clearly seen in Figs. 11 and 12. Not only significant portions of non-text were classified as text, but also fragments of large-sized characters are sometimes missing.

Missing fragments of large-sized characters can be partly caused by the small values of \( n \). Nevertheless, this did not negatively affect correct detection of such characters in many cases (though interior parts were usually not classified as text because they did not contain any edges). That is, for large-sized characters, only blocks on their borders were typically detected. It seems that in general the blocks found in such areas contained less edge pixels than the blocks found in the areas of small-sized text. This may be one of the main reasons explaining why the large-sized characters were not always completely detected. It was still possible to improve the results by lowering the value of \( t_2 \), but at the cost of more false positives. Another reason contributing to missing border blocks for the large-sized text was a poor contrast between the characters and surrounding background, which sometimes resulted from the colour-to-grey scale conversion when several different colours were mapped into the same grey level.
4.3. Threshold selection and comparison

Although we did not yet extensively investigate the influence of different threshold settings on the text detection quality, the results were quite good for many thresholds. Here, ‘good’ means that text regions were reasonably accurately detected, and noise, if present, was reduced to an acceptable level.

As to $t_1$, it was always automatically computed according to Eq. 2 in our experiments because it produced adequate edge images. Unfortunately, we are unable to say so about $t_2$, where the range of admissible values varied, depending on the definition of $\mathcal{R}$. We can only say that this range is generally narrower when $\mathcal{R}$ defined by Eq. 3 is used than in the case of $\mathcal{R}$ defined by Eq. 5. The range is difficult to automatically determine because the histogram of $\mathcal{R}$ is almost flat, except for $\mathcal{R} = 0$. We are currently using heuristics to bound it, but more intelligent techniques need to be developed.

An example demonstrating a rather good stability of text extraction when $t_2$ varies within the admissible range is shown in Figs. 13 and 14. The admissible range was from 0.004 to 0.044. As one can see, the results do not much differ from those in Fig. 10.

To compare our method, we selected a method based on the discrete cosine transform coefficients proposed in [14].

It is intended to localise text in video and still images encoded in JPEG. The comparison results when this method was applied to the images we used for testing, are discussed in [6]. Based on the comparison, we can say that the method of Zhong et al. performed worse for large-sized characters, while it was nearly as good in the extraction of small-sized text (say, 10-12 pt) as our method. We think that this was due to the fact that the ‘text periodicity’ claimed as a principal feature for text detection in [14], requires a lot of characters and text lines in order to be reliably estimated. The comparison results indicated that the edge information plays a more important role in accurate text extraction than the periodicity.

5. DISCUSSION

Text extraction from grey scale or colour images remains a challenging task because of the complex background or degradations introduced during scanning or copying a paper document. We proposed a simple solution to this problem based on edge detection.
For colour images, we currently apply colour-to-grey scale conversion before edge detection. This is a computationally cheap approach, and it is also well known that most of the edge information in colour images is included in the luminance component of the colour image. Further investigation on how to use colour to aid the analysis is still needed, however.

To be robust, the text extraction should be more or less insensitive to page skew, typically manifested as a tilt of text lines. We did not conduct extensive experiments with skewed images, but it seems that our approach performs well in this case, too (see Figs. 15 and 16).

Our future research will concern refining the text extraction results in order to reduce the number of falsely detected non-text blocks. One possibility for automatically adjusting thresholds might be to learn them with the help of a special tool GROTTO [4]. The tool is designed to generate a description, which one should obtain as a result of successful segmentation of a given page image. We also plan to investigate the use of colour to aid the analysis.

Acknowledgments

The financial support of the Academy of Finland is gratefully acknowledged. We would like to thank Mr. Topi Mäenpää for his comments, Mr. Ville Brummer for help in programming, and Ms. Kati Vainio for program testing.

References


Fig. 15: Skewed image with a skew angle of 5 degrees

Fig. 16: Text extraction results for the image in Fig. 15


