

EDGE DETECTION IN DOCUMENT ANALYSIS

Mihai Zaharescu¹
Ionuț Cosmin Petrescu²

ABSTRACT

The significant improvements in edge detection, vectorization, contour specification, and skeleton extraction that were made in the past years should be included in the document analysis pipeline. We discuss the general OCR system, introduce an edge detection specialized for characters, present a geometrical based preprocessing pipeline and open the door for further research in the communication of the OCR with the preprocessing stage.

Keywords: OCR, Edge Detection, Vectorization, Binarization

1. INTRODUCTION

In few cases the attention was brought to preprocessing a scanned image document before sending it to the Optical Character System (OCR). The usual preprocessing steps are pixel value alterations meant to remove shadows, enhance contrast, binarize the image and remove the skew.

Examples that use geometrical representations exist, but they are incorporated inside the OCR itself [12][13], converting the letters to skeletal representations and using those for comparison. Converting the entire image to geometry, using theoretical description of how lines are represented [12], along with feedback from the OCR can improve OCR results.

2. OPTICAL CHARACTER RECOGNITION – A COMPLETE SYSTEM

A usual system [8] starts from scanning an image (possibly stitching multiple smaller images together in order to scan a larger document, like a newspaper page [1]).

This image then goes a simple to medium complexity preprocessing step, based on changing pixel values in order to remove shadows or light spots, usually using local adaptive contrast enhancement [2].

Then follows a well studied process: binarization; this transforms the grayscale/color image into a binary image (black and white). These algorithms can be global (if luminosity is very uniform and the document has clear writing) or local and adaptive (sometimes a preprocessing isn't used, so this procedure needs to take its place and extract letters from

¹ Engineer , Department of Computer Science and Engineering, Faculty of Automatic Control and Computers Science, University “Politehnica” of Bucharest, Splaiul Independenței 313, Bucharest, 060042, Romania, mihai.zaharescu@cti.pub.ro

² Engineer , Department of Computer Science and Engineering, Faculty of Automatic Control and Computers Science, University “Politehnica” of Bucharest, Splaiul Independenței 313, Bucharest, 060042, Romania, ionut.petrescu1612@cti.pub.ro

non-uniform background) [3]. This includes removing backside writing, speckles of ink, paper coloration and it can be a challenging phase when dealing with old documents.

The global algorithms try to split the histogram in two balanced parts, or to create clusters having components with small variance but them themselves being as different as possible.

Local algorithms choose a local window around a pixel and apply the binarization considering only the pixels inside the window. These steps require human intervention in order to set parameters for the amount of information on page or the window size or the specific document type (depending on the quality of the document) though approaches have been made in order to eliminate human intervention with successful results [4].

Some additional steps follow, depending on what the user wants to do. A must do phase is analyzing and removing skew and it usually uses pixel based clustering and uses the clusters for an estimation of rotation [5]. More complex algorithms can use Hough transform methods, using the center bottom pixel of letters, after extracting the text elements; [6] or even correct the bending of the paper [7].

For medium sized projects, like digitization of a newspaper article, a layout analysis step splits the scanned image segments using morphological or computational geometry methods and after reading the text characteristics of each region decides what each block is (letter, word, paragraph, title, subtitle, image, white space, page number) [9]. Its purpose is to send ordered pieces of text to the OCR, being able to handle “strange” multicolumn layouts with inserted graphics.

For large scale projects, like digitization of entire books for libraries, all these fragments need to be put in a logical order. The Hierarchy analysis engine does this, adding extra information about each object (book title, author, preface, chapter, table of contents).

The actual conversion module (what is popularly known as OCR) receives binary images, consisting of text only, as input and gives digital text as output, either by matrix matching or feature extraction.

The final stage includes estimating font characteristics [20] and textual error correction using dictionaries.

3. THE “DOCUMENT IMAGE ANALYSIS” PROBLEM REVISITED

This paper focuses on the preprocessing but also touches a bit the ORC system stating that the OCR should work together with the image processing, giving it back information whether it should change the binarization level, add or remove information.

It also presents a simpler approach of transforming the initial image into geometry, into a vectorial image, which can be post-processed using computational geometry methods. Such methods have been used before for other document analysis purposes [10][11], but we propose to use them at the stage of letter pre-processing.

Today's computer vision methods rely more on edge processing than on binary image processing, so this kind of algorithms are better studied and more mature than binarization methods.

Binarization for feature extraction isn't that much useful; one example that demonstrates this is an image containing two important objects next to each other. Binarization can either put both objects in the same cluster or eliminate one of the important objects. None of the results is useful.

Segmentation and edge detection can resolve this issue. Alongside with edge detection, corner detection can aid deskewing and layout-analyze algorithms.

4. EDGE DETECTION

The domain of edge detection has been much studied, reaching the point of simulating human vision in areas like: texture extraction (selecting the outline of an object even if it has strong texture), optical illusions (extracting the contour from objects that only “seem” to have a contour), noise and illumination ignoring [14].

I. First Derivative Methods

The simplest idea is to take every two pixels and see the difference in color between them, firstly in horizontal direction then in vertical direction. The difference between adjacent pixels is the first derivative of the discrete image. The amplitude of the edge is equal to the summation of the amplitudes in both directions. The direction (normal to the edge) is the arctangent from the division of the two derivatives. Because the arctangent needs infinite values to represent horizontal and vertical lines the simplest edge detection operator, the Roberts cross, uses the presented idea, but rotates the derivative by 45 degrees, calculating the derivatives following the first and second diagonal of the image.

+1	0	0	+1
0	-1	-1	0

Fig 1. Roberts Cross convolution kernel

The amplitude or strength of the edge is:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

And the direction:

$$\theta = \arctan(G_y / G_x) + 3\pi / 4$$

In order to lessen the effect of single pixel noise, a filter can be used in conjunction with the derivative. The Prewitt operator does an arithmetic median of the neighboring pixels and calculates the derivative:

-1	0	+1	+1	+1	+1
-1	0	+1	0	0	0
-1	0	+1	-1	-1	-1

Fig 2. Prewitt convolution kernel

The amplitude and direction are calculated the same, but without the $3\pi/4$ compensation for the direction.

The Sobel filter uses a better Gaussian-like smoothing:

-1	0	+1	-1	-2	-1
-2	0	+2	0	0	0
-1	0	+1	+1	+2	+1

Fig 3. Sobel convolution kernel

II. Second Derivative Methods

The second derivative utilizes the zero crossing as reference for the positioning of the edge.

Because the gradient slowly rises to an abrupt slant (increases) then slowly comes back to 0 (decreases), the derivative of the derivative shows where the peak of the first derivative is located, or where the center of the edge is.

The downside is that even very faint artifacts influence the final result, so, even filters like Marr-Hildreth, which use the Laplacian of Gaussian in order to filter the noise, are prone to false positives, but are much better localized.

III. Canny

Up until Canny's research, the definition of an edge followed a more intuitive than rigorous path. Canny, in his work, made a rigorous formulation of what an edge is and developed a multistage algorithm for finding the optimal edge, taking his definition as reference.

The algorithm starts by blurring the image with a Gaussian kernel of adjustable size, which decreases noisy pixels.

Then, any of the presented edge detection algorithms can find the derivative intensities and directions along horizontal or vertical directions.

Using the definition, a single edge must give a single response, which means the edges must be one pixel thin. In order to filter the extra responses from one edge a non-maximal suppression stage compares each pixel with the neighbors, taking in consideration the calculated directions.

Only directions to neighboring pixels are significant, so the calculated orientations are truncated to the horizontal vertical and diagonal directions. This stage compares the edge pixels with the adjacent pixels of similar orientation to decide upon which should be included in the edge. If the pixel has the strongest amplitude, then it is a possible edge, otherwise it is an echo of an edge and must be suppressed. The result is an image containing 1 pixel thick edges.

However, a single threshold that generates a good result may possibly not exist. That is why the following procedure selects only confident edges and fills the possible holes using edges found using a smaller threshold and the directional information.

The canny edge detector is considered optimal for detecting edges using gradient analyze.

Optimizations have been made in order to simulate optical illusions and eliminate textures [15].

5. CANNY IMPROVEMENT

This paper is based upon a modified Canny algorithm.

In order for the edges to retain their positions and sharp angles, we must avoid the preprocessing blurring step, needed for the canny algorithm. But we also want to avoid false positives.

Canny uses a blurring preprocessing step that removes noise. This blurring also reduces corners and makes the edges positions less accurate. Instead, we decided to use a windowing algorithm:

For every pixel, we use a window, containing neighboring pixels up to a distance. The size of the window is proportional to the strength of the noise and size of the letters. Inside this window we calculate forces from the current pixel to each of the neighbors. The force's amplitude is the color difference between the pixels and the direction is the angle between their positions. The resulting force, characteristic to the center pixel of the window, is the vectorial sum of all the forces.

The window can be variable in size. The noisier the image, the greater the window should be. Noise will finally fade, because it is random, compared to the real signal.

Having a set of lines and orientations is not enough because it could be possible for a thick line to induce echo responses or for a thin line to be interrupted by noise; these problems can

be addressed in this geometrical definition (amplitude and orientation) much better than in the pixel value description.

A similar to Canny's non-maximal suppression procedure eliminates the echo lines, but an extra step, that considers the window's size, simplifies the edges having length up to the window's size to straight lines and having only the principal 4 directions. This is useful for simplifying the generated edge shape.

The inspection of the neighboring pixels as far as the window's size using the directional information filling possible edge gaps.

6. RESULTS

As a test, we used a speckled scanned image from an old book.

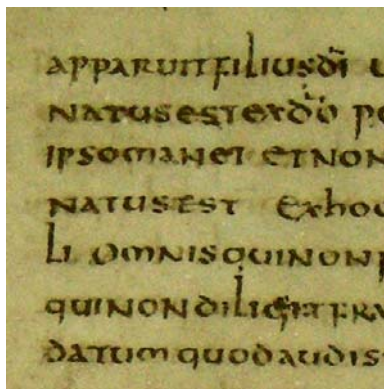


Fig 1. Test Image

Standard binarization techniques classify pixels in two buckets: black for foreground and white for background, either by using locally adaptive, global methods or complex algorithms [16]. Letter discontinuities can be noticed and can't be overcome by this method because noise levels are quite high in these kinds of images.

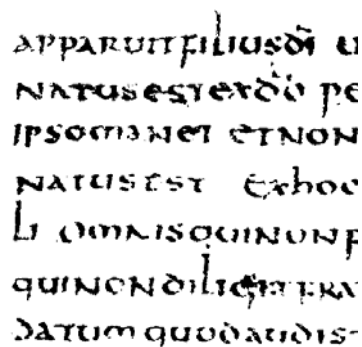


Fig 2. Global binarization

Edge detection techniques try to overcome the visible problem of letter discontinuity by intuitively closing an open edge. Canny edge detections perform rather well.

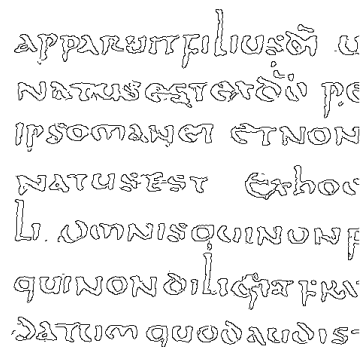


Fig 3. Canny edge detection

The proposed algorithm tries to keep the letter corners sharper by using the windowed approach, but also to eliminate strong noise.

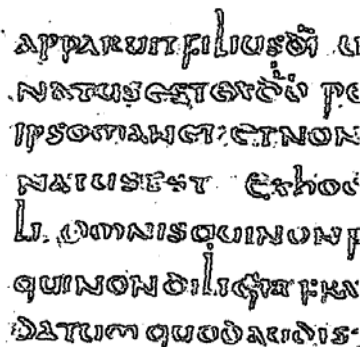


Fig 4. Edge detection - rough output

Thanks to the fact that we have a vector image, post-processing steps can improve the image a lot, for example simplifying inconstant edges and deleting small objects.

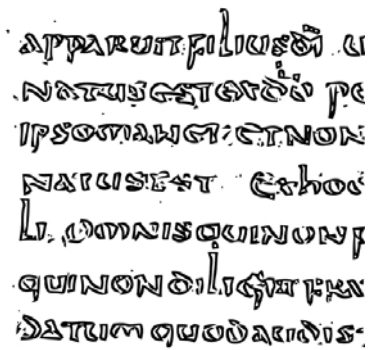


Fig 5. Edge detection - simplified

An improved hysteresis uses more edge intensity levels. This stage should be incorporated in the digitization system and communicate with the OCR. If the OCR doesn't recognize an object, it sends back a signal to add another lower intensity level.

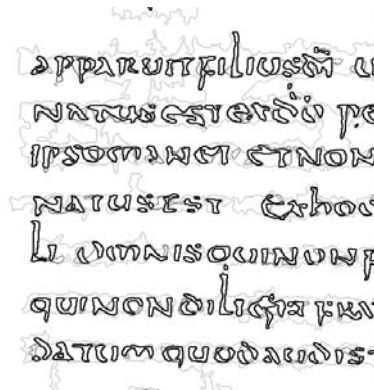


Fig 5. Edge detection – hysteresis

In this example 3 strengths were used: black for sure line, dark gray for possible completion and white gray for unlikely completion. It can be seen that the first letters are incomplete at the first intensity level, by using the second they become complete and clear.

A voting based algorithm can bring major improvements because it combines general color information with color variation information. The input algorithms are normal binarization, which selects objects as being foreground or background, relaying on pixel colors and usually ignoring transitions; edge detection algorithms which rely upon color transitions and usually ignore pixel color values; and vectorization, which connects components, smoothes edges but keeps corners. A simple technique which uses all these steps is presented, which takes the binary and a vector image, selects only the elements from inside the borders of the edge detection as font colorization and the edges which overlay the binarized edges as font margins. A closing morphological operation is used to eliminate discontinuities. The result is better than any of the individual methods alone. The white

inside is surely part of the text, and it can be completed with the black contour, in case a letter is not matched.

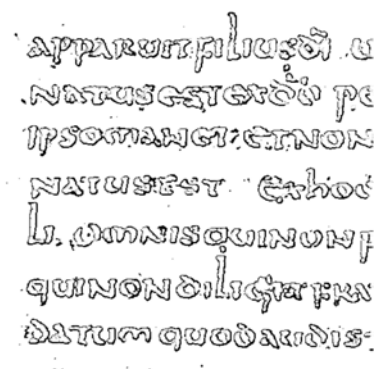


Fig 6. Edge detection – voting

7. FUTURE WORK

In this paper we used a simple median of the colors algorithm in order to convert the image to gray-scale for the following steps, but using color to gray-scale conversion algorithms that enhance the difference between colors, making the text more evident can enhance the results [19][17][3].

The parameters for window size can be set automatically. In one of the articles [18], the program sets its window size, for a locally adaptive binarization, automatically. It does this by using statistics: it starts with a small window around each pixel and enlarges it. When the standard deviation reaches a maximum, it considers that the optimal window for the respective pixel and applies the Otsu binarization there. Edge detection can improve the statistical heuristic. An edge image is created using a high threshold, in order to find the sure edges. In the growing process, if a window touches one of these edges, it is certain that the respective window contains foreground and background elements, improving binarization.

8. CONCLUSIONS

Retro-conversion systems can gain a lot from passing from pixel-based operations to geometrical representation. The preprocessing steps presented here show that better results can be achieved even with simple operations. All the systems incorporated inside the digitization program, including the preprocessing step, should be able to communicate to one another in order to offer clues to whether parts of their job should be refined.

9. REFERENCES

[1] Usual Scenarios and Suitable Approaches Used in Automatic Merge of Scanned Images, C A Boiangiu, A C Spataru, A I Dvornic, I Bucur

- [2] Modern Preprocessing Techniques for Automatic Content Conversion Systems, C A Boiangiu, A I Dvornic
- [3] Methods of Bitonal Image Conversion for Modern and Classic Documents, C A Boiangiu, A I Dvornic
- [4] Local Thresholding Algorithm Based on Variable Window Size Statistics, C A Boiangiu, A Olteanu, A Stefanescu, D Rosner, N Tapus, M Andreica
- [5] Text Line Processing for High-Confidence Skew Detection in Image Documents, D Rosner, C A Boiangiu, A Stefanescu, N Tapus, A Olteanu
- [6] Skew Detection Using the Radon Transform, B Raducanu, C A Boiangiu, A Olteanu, A Ștefănescu, F Pop, I Bucur
- [7] Confidence Measure for Skew Detection in Photographed Documents, C A Boiangiu, D Rosner, A Olteanu, A. V Stefanescu, A. D. B Moldoveanu
- [8] eUPB: Towards an Integrated e-Service Platform in Large Scale Distributed Environments, F Pop, M Andreica, C Dobre, R Rughinis, A Moldoveanu, C Boiangiu, M Ionescu
- [9] Document Layout Analyze Using Hierarchical Processing, C A Boiangiu, D C Cananau, I Bucur
- [10] 3D Mesh Simplification Techniques for Image-Page Clusters Detection, C A Boiangiu, B Raducanu
- [11] The Beta-Star Shape Algorithm for Document Clipping, C A Boiangiu, I Bucur, A I Dvornic
- [12] A protocol for performance evaluation of line detection algorithms, Liu Wenyin, Dov Dori
- [13] An Implementation of OCR System Based on Skeleton Matching, Ning LI
- [14] Edge and line oriented contour detection: State of the art Giuseppe Papari, Nicolai Petkov
- [15] Edge and line oriented contour detection: State of the art Giuseppe Papari, Nicolai Petkov
- [16] Binarization for Digitization Projects Using Hybrid Foreground-Reconstruction, C A Boiangiu, A I Dvornic, D C Cananau
- [17] Bitonal Image Creation for Automatic Content Conversion, C A Boiangiu, A I Dvornic
- [18] Local Thresholding Image Binarization using Variable-Window Standard Deviation Response, C A Boiangiu, A Olteanu, A. V Stefanescu D Rosner, A. I Egner
- [19] Color2Gray: Saliency-Preserving Color Removal, Amy A. Gooch, Sven C. Olsen, Jack Tumblin, Bruce Gooch
- [20] Normalized Text Font Resemblance Method Aimed at Document Image Page Clustering, C A Boiangiu, A C Spataru, A I Dvornic, D C Cananau
- [21] Geometric Methods for Optical Character Recognition, George N. Sazaklis