Automatic Filter Selection Using Image Quality Assessment

Andrea Barretto de Souza

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

May 2003

Your file Votre référence
ISBN: 0-612-83899-4
Our file Notre référence
ISBN: 0-612-83899-4

ABSTRACT

Automatic Filter Selection Using Image Quality Assessment

Andrea Barretto de Souza

We present a method for automatically selecting the best filter to treat poorly printed documents using image quality assessment. In order to estimate the quality of the image, we introduce five quality measures: stroke thickness factor, broken character factor, touching character factor, small speckle factor, and white speckle factor. Based on the information provided by the quality measures, a set of rules uses a two-stage decision process to choose the best filter among 4 morphological filters to be applied to an image. Other preprocessing tasks implemented are: skew correction, connected components analysis, and detection of reference lines. Our database contains 736 document images that were divided in three sets: training, validation and testing. Most images have one or more of the following degradations: broken characters, touching characters and salt-and-pepper noise. A training set of 370 images was used to develop the system. Experimental results on the test set of 183 images show a significant improvement in the recognition rate from 73.24% using no filter at all to 93.09% after applying a filter that was automatically selected. The recognition rate refers to the number of characters that were correctly recognized in the image using a commercial OCR. Three commercial OCR's were used to demonstrate the improvement obtained in the recognition rates in the training set.

Acknowledgements

I would like to thank my supervisor, Dr. Ching Y. Suen, for his guidance during the development of my research, as well as for providing an exceptional working environment at Cenparmi. I would also like to thank my co-supervisor, Dr. Mohamed Cheriet, for the valuable discussions and excellent suggestions made throughout this project. Without their help, it would not have been possible to develop and complete this work.

I am very grateful to my colleagues at Cenparmi, in particular Dr. Marisa Morita, Dr. Luis Eduardo Oliveira, Dr. Susan Wu and Karim Abou-Moustafa, who helped me in different stages of my research.

I would also like to express my gratitude to my parents, who helped me accomplish my dreams during all my life.

I would like to thank my husband, Stephen, for his love and support in every moment.

Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1. Problem Description

Optical Character Recognition (OCR) is the process of converting the text of a document image into a format that can be edited by a computer. There are many commercial OCR systems available, however, most of the time, poor quality printed documents yield low recognition rates when submitted to them. One way of increasing these recognition rates is to improve the quality of these documents by applying a filter that can minimize the image degradations. However, when the database contains document images with different types of degradation, such as broken characters, touching characters, and salt-and-pepper noise, it is difficult to find a single filter that can address all these problems at the same time.

The solution that we found for our database, which contains the types of degradation mentioned above, was to use different filters for different problems. If only one filter is applied to the whole database, some images will be improved while others will be even more degraded. By using different filters, it is possible to improve the quality of all images, assuming that the appropriate filter is applied. However, this solution leads to another challenge, which is how to select the most suitable filter based on the type of image degradation. The solution was to obtain information on the quality of the image prior to the selection of the appropriate filter.

This research is about assessing the quality of a document image in order to obtain information about the type of degradation that it may contain and use this information to automatically select the best filter possible among a specific group.

The motivation for developing this research was the need to automatically improve the images of a database that contains poorly printed documents. Our system estimates the quality of each image using quality factors. The result is used to select the most suitable filter, among four available, that will improve the image quality. This improvement is measured by comparing recognition rates before and after applying the filter.

1.2. Importance of Image Quality Assessment

The estimation of the quality of a document image can be useful in different ways. In the case of adaptive OCR algorithms, in which the parameters for recognizing a document image are set according to the input [12], image quality assessment may provide essential information for this adaptive classification [8].

Image quality assessment can also be used in estimating OCR accuracy, since the image quality is directly related to OCR errors [13]. Depending on the error rate that such a system would estimate, it may be more cost-effective not to submit the document to the OCR because this would imply in spending time to manually correct a large amount of errors. This would be especially useful in large-scale OCR environments, in which the whole process is automated [8].

In the specific situation of our problem, in which there are different types of degradation, the use of image quality assessment provides information about these types of degradation before the filter selection. It is this quality estimation that makes possible the automatic selection of an appropriate filter to enhance the quality of a given image. Without this information, we would have to use a single filter to perform image restoration in the whole database, which would not be appropriate for all images and, therefore, would not provide the same improvement in the recognition rate as when different filters are used.

1.3. System Overview

The input of the system consists of binary printed images of poor quality, which is due to different types of image degradation found in the database. There are four filters available in the system, and the objective is to automatically choose and apply one of them, according to the type of degradation of the image. This automatic choice is possible because quality measures are used to estimate the image quality before applying the filter. The output of the system is a new image with skew corrected and with the least amount of noise possible.

This whole process is part of the preprocessing stage in a recognition system, and its results can be used as an input for a classifier. However, since we do not have the actual recognition stage in our system, we will use the term preprocessing only to refer to those tasks that are performed prior to image quality assessment and the automatic filter selection. The objective is just to clarify the distinction between the different blocks of the system.

The system is divided into three major parts: preprocessing, image quality assessment, and automatic filter selection, as illustrated in Figure 1.

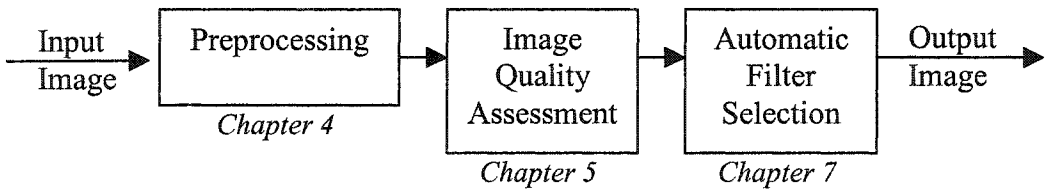


Figure 1: Block diagram of the system

Preprocessing includes skew correction, connected component analysis, and detection of reference lines. Skew correction is performed for all images, independently of the type of degradation that it may have. The detection of connected components and reference lines is performed because this information is used to estimate the image quality.

Image quality assessment involves the calculation of five quality measures that will be used to estimate the image quality and then decide on the type of degradation present in the document image.

Automatic filter selection refers to the process of automatically choosing and applying the most suitable filter for an image. This is done using a set of rules that were defined based on the analysis of the quality measures of the images from the training set.

1.4. Thesis Organization

The remainder of this thesis is divided into 9 chapters. Chapter 2 contains the state of the art regarding image quality assessment, noise reduction, and skew correction. In chapter 3, the database is described. Preprocessing tasks used in the system, such as skew correction, connected component analysis, and detection of reference lines, are explained in chapter 4. Chapter 5 describes the quality measures used for image quality assessment. The filters used in the system are presented in

chapter 6. Chapter 7 explains the sequence of tasks performed by the system, as well as the automatic filter selection process. Experimental results are in chapter 8. Chapter 9 contains the conclusion and perspectives for future work.

Chapter 2. State of the Art

Document image analysis is a sub-field of pattern recognition that aims to "recognize the text and graphics components in images and extract the intended information as a human would" [2]. Another definition, presented in [19], defines the objective of this field as "the task of deriving a high level representation of the contents of a document image". Document image analysis can also be defined as the determination of "the logical structure of a page image following a resolution into physical components" [20].

A document image analysis system uses a document image as input and tries to recognize its content. This process is divided into different stages, basically: image acquisition, preprocessing, and recognition. The focus of our work is in the preprocessing stage, which objective is to perform tasks that will result in a better image for recognition and/or obtain in formation that may help the recognition process. Many tasks can be performed at this point, such as connected component analysis, detection of reference lines, skew correction, noise filtering, contour detection, etc.

The objective of this chapter is to present related works especially in image quality assessment, which is the main focus of our research. A brief review of different methods of noise reduction is also presented.

2.1. Image Quality Assessment

Image quality assessment, as the name suggests, aims to estimate the quality of an image. This assessment can be done in different ways, and the result can be used for multiple purposes.

After reviewing the literature for related works on image quality assessment, we could only find one research that is similar to ours. The work developed by [7] presents a method to automatically improve the quality of document images by choosing an optimal enhancement method. The method is divided in two parts: first, five quality measures are used to estimate the quality of the image: small speckle factor, white speckle factor, touching character factor, broken character factor, and stroke thickness factor. As the names suggest, these quality measures focus on background speckles, broken characters, and touching characters. Based on this assessment, the second part of the method chooses a restoration algorithm using a linear classifier. The system uses only typewritten documents, and their database consists of 139 poor quality images. The skew angle of the document, if there is any,

is not considered. The effectiveness of their method is proven by the decrease in the OCR error rate after applying the selected filter to each image of the database.

The work developed by [13] also uses quality measures but with a different purpose. In this case, three quality features that focus on background speckles, touching characters and small white connected components are extracted. The objective is to determine if a poor quality document image can be successfully submitted to an OCR. Their database consists of nine versions of the same page of a book, each of them with decreased quality. Repeated photocopies were made in order to obtain the different versions of the page.

The objective of the research presented by [8] was the development of a classifier that can predict OCR accuracy using image quality assessment. The system uses three rules to classify a document as either "good quality" or "poor quality". In the first case, a high OCR accuracy is expected, while in the second case a low accuracy is expected. The rules are based on the image degradations that were observed in a small set of sample pages, which can cause OCR errors. The quality measures computed are small speckle factor, broken character factor, and a third measure that detects inverse video regions (white letters on a black background). The database used in this work consists of two sets, one with 460 pages of scientific and technical documents, and another with 200 pages from magazines.

2.2. Noise Reduction

The objective of a noise reduction algorithm, or filter, is to improve the quality of an image by removing as much noise as possible. There are various noise reduction algorithms that can be used for different types of problem, as well as different types of document. For example, regarding the types of problem, there are background speckles, broken characters, and touching characters. As for the type of document, some filters are intended for handwritten materials, while others are for machine-printed ones. The filters described below do not represent the totality of methods available in the literature. The objective is to show some of the different options available.

The work described by [21] deals with the restoration of images that have "undefined pixel values at known pixel locations". A set of algorithms intended for image continuation is presented. It is assumed that the defective pixels have already been detected. The objective is to restore these pixels starting from a reliable portion of the document.

In [22], an algorithm for restoring machine-printed characters is presented. The system deals a set of degraded binary images of a single unknown character and tries to obtain an output image that is as close as possible to the ideal image. The input images are superimposed, the intensities are added up at each point, and a threshold is used to obtain a new binary image.

An adaptive technique that restores touching characters and broken characters is presented by [23]. The objective is to "restore distorted character images by

adaptively generating an inverse distortion model". A distorted image is processed by an OCR system. Using this output and a distorted text image, an adaptive restoration filter is trained and then applied to the distorted image. The authors demonstrate that this feedback technique results in an improvement in the OCR accuracy.

A macrostructure analysis method that mends single handwritten broken characters is presented by [14]. The method is based on skeleton and boundary information. First, a set of masks is used to fill holes that may exist in strokes and to compensate single pixel width strokes. After that, the boundary and the skeleton of the character are obtained. The skeleton ends are then extended in order to preserve the tending direction of the strokes. A post-processing step smoothes the character using a set of masks.

The system developed by [10] deals with cleaning and enhancing handwritten form items that cross or touch the frames or any other preprinted text in the form. Mathematical morphology operations are used to clean the data. Another technique applied is edge smoothing, which uses 3x3 masks to either fill or remove pixel, depending on the situation. A method for mending broken characters that is based on [14] is used. However, some heuristic rules were added to decide when the algorithm for mending the strokes should be applied, and to avoid mending characters incorrectly.

Mathematical morphology operations can be used for noise removal and image enhancement [9]. There are two basic operators, dilation and erosion. The first one basically enlarges the image, while the second one erodes its boundaries. All the other mathematical morphology operations, e.g. closing and opening, are derived from the

erosion and dilation. In all cases, a structuring element is used, which is a mask or pattern that will define the effect of the operator on the image. Examples of structuring elements used are cross, square, horizontal line and vertical line.

Chapter 3. Database

The database used for the development and testing of the system consists of 736 binary document images. All images were taken from databases available at Cenparmi. In order to develop the system, 370 images were used. Throughout the remainder of this thesis, this set is called "training set". Please note that there is no actual learning by the system, as the term training may suggest, because we are not using a neural network. This set was used to define the rules that make the automatic selection of the filter that should be applied to an image. The term "training set" was chosen in order to differentiate this set form the others. The remaining images were divided into two groups: a validation test (183 images), which was used to verify the results obtained by the training set and make final adjustments to the set of rules, and a test set (183 images).

This database has three distinctive characteristics. First, all images contain only one printed text line in English. There are no graphics, tables or drawings in any document. Second, a large variety of font sizes, types and styles can be found among the images.

The third characteristic, and also the most important for our work, is that almost all images suffer from at least one of the following types of degradation:

- Broken characters
- Touching characters
- Salt-and-pepper noise

Images that present these types of degradation still represent a challenge to modern OCR's [18]. Images with broken characters usually have thinner strokes than the original font type presents, this is the reason why the characters fragment into smaller pieces. This type of problem requires a restoration algorithm that will basically add pixels to the image. On the other hand, noisy images that contain salt-and-pepper noise, and images that contain touching characters because of thickened strokes, require a filter that will remove the excess of pixels. So, these three types of degradation require conflicting solutions.

The images in the database were manually analyzed in order to determine the number of images that had each type of degradation, as well as the combinations of problems that could be found. The table below shows the results:

Table 1: Distribution of problems in the database

Type of Degradation	Number of Images	
Only Broken Characters	369	50.14%
Only Touching Characters	45	6.11%
Only Salt-and-Pepper Noise	17	2.31%
Touching Characters with Salt-and-Pepper Noise	229	31.11%
Broken-Characters with Touching Characters	22	2.99%
Broken Characters with Touching Characters and Salt-and-Pepper Noise	1	0.14%
No Degradations	53	7.20%
Total	736	100%

Analyzing these results, we can see that half of the database (50.14%) contains only broken characters. Grouping together the images that have either touching characters or salt-and-pepper noise or the combination of both, the total amount represents 39.54% of the images. Both groups together represent almost the entire database, 89.68% of the images. Based on this information, we can conclude that it is necessary to have at least two types of filter to improve the quality of the images of our database.

Examples of images from the database are presented below. They are grouped according to the type of problem found in each of them.

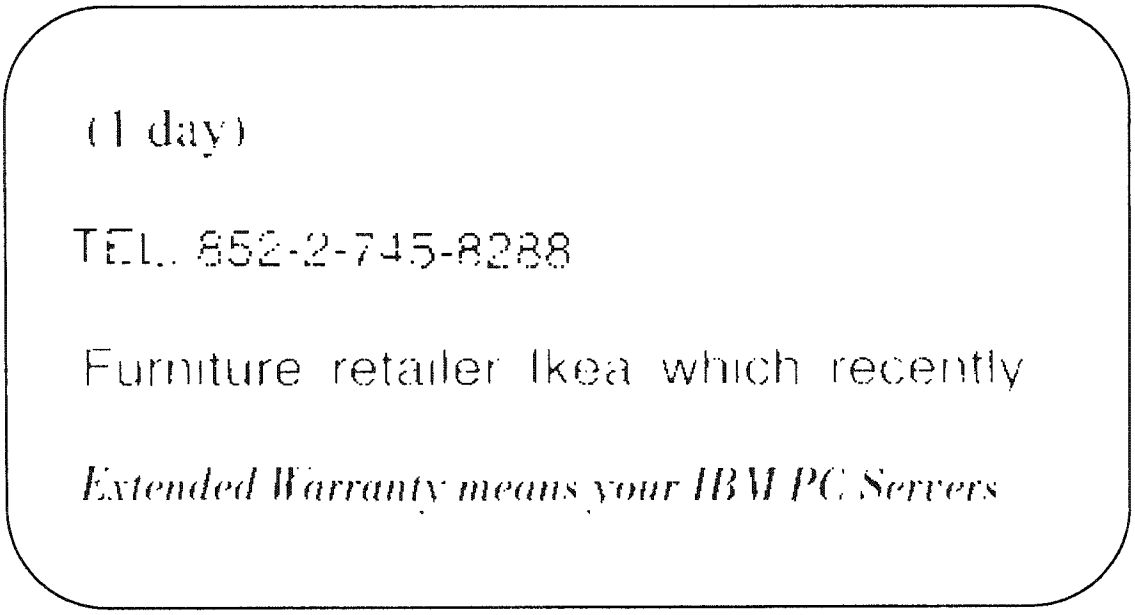


Figure 2: Sample images with only broken characters

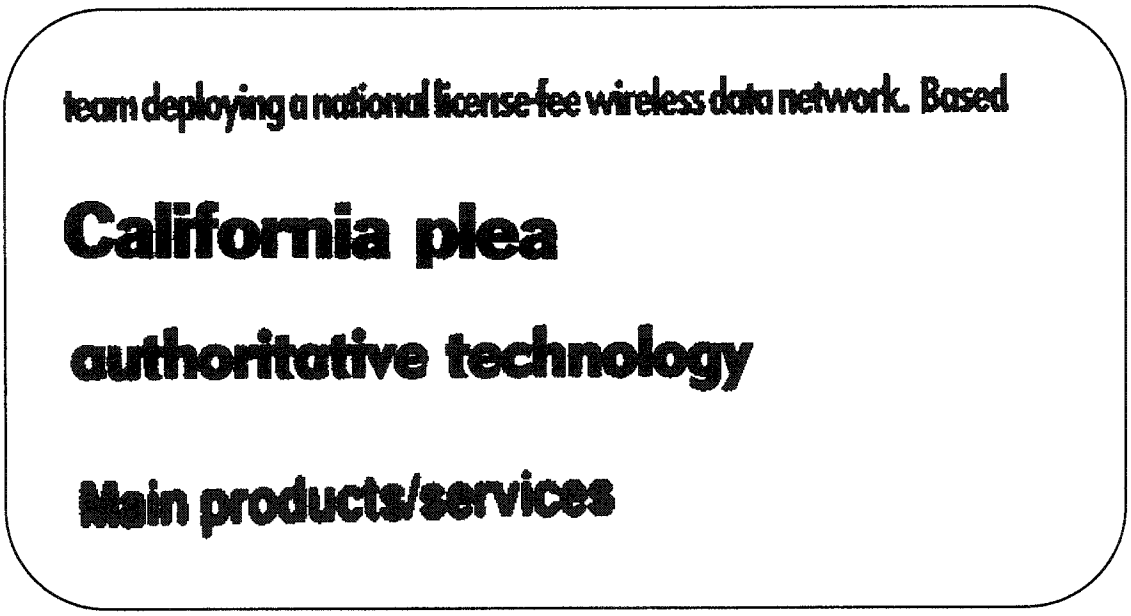


Figure 3: Sample images with only touching characters

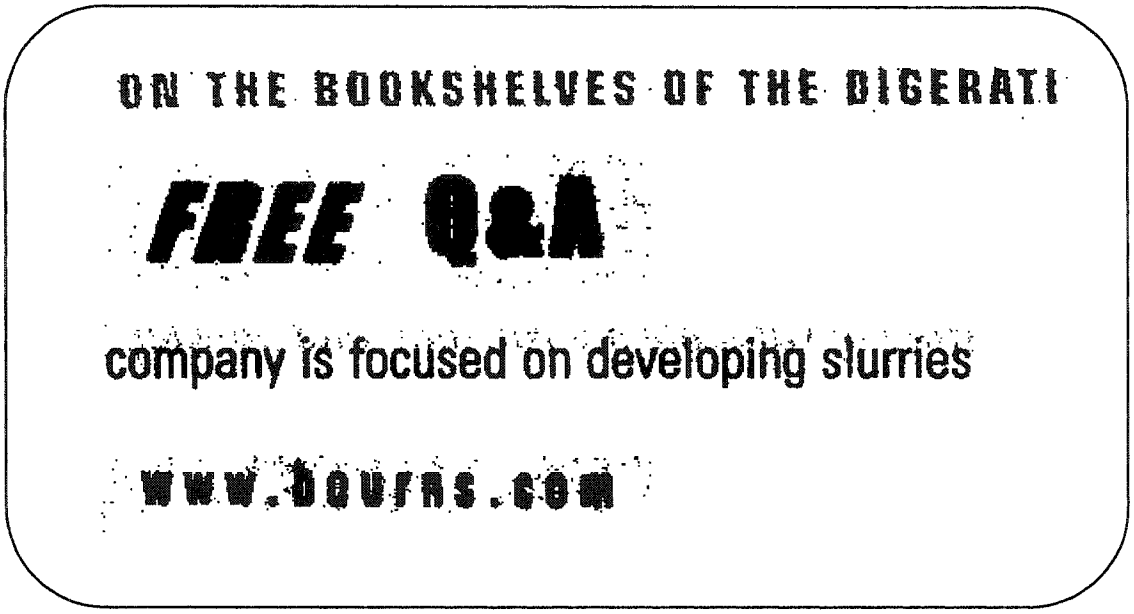


Figure 4: Sample images with only salt-and-pepper noise

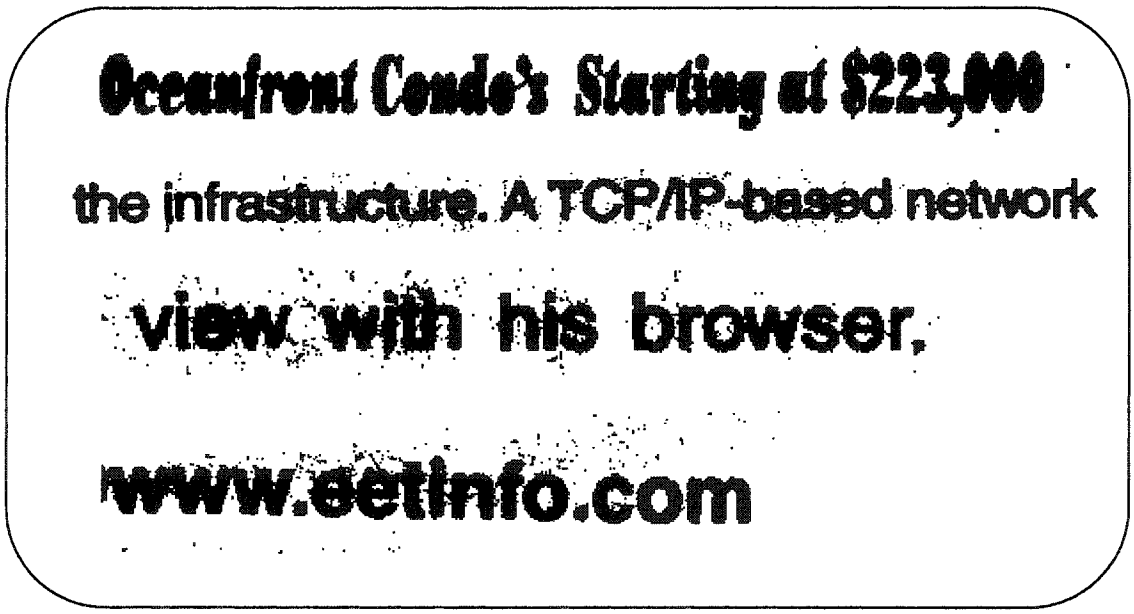


Figure 5: Sample images with touching characters and salt-and-pepper noise

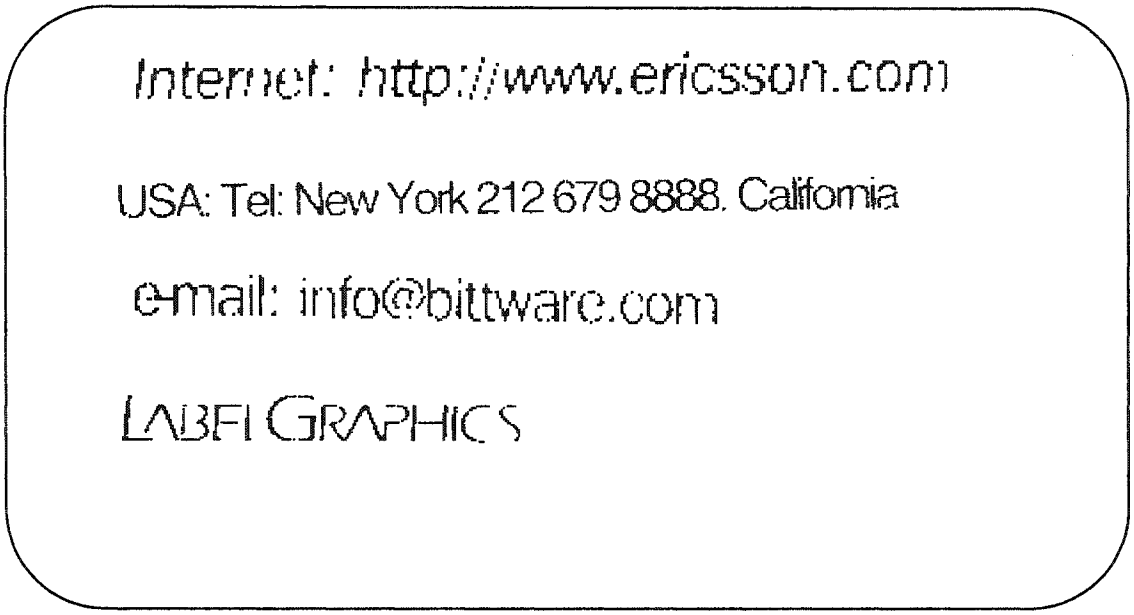


Figure 6: Sample images with touching and broken characters

Only one image was found that had all three types of degradation at the same time. This image is shown below.

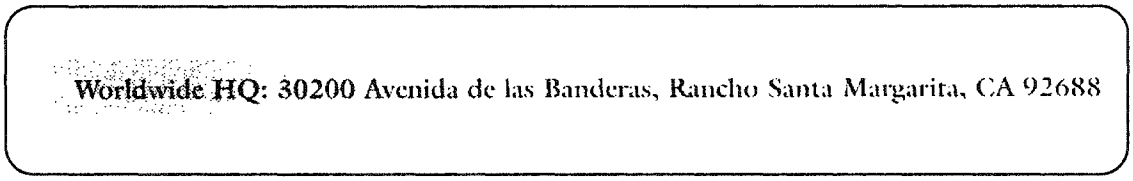


Figure 7: Image with touching and broken characters, and salt-and-pepper noise

In the case of the images that were manually classified as having no degradations, it does not necessarily mean that all of them are good quality images. In most cases, these images have either thinned strokes but not thin enough to fragment the characters, or thickened strokes but not thick enough to merge the characters. Examples of this group are presented below.

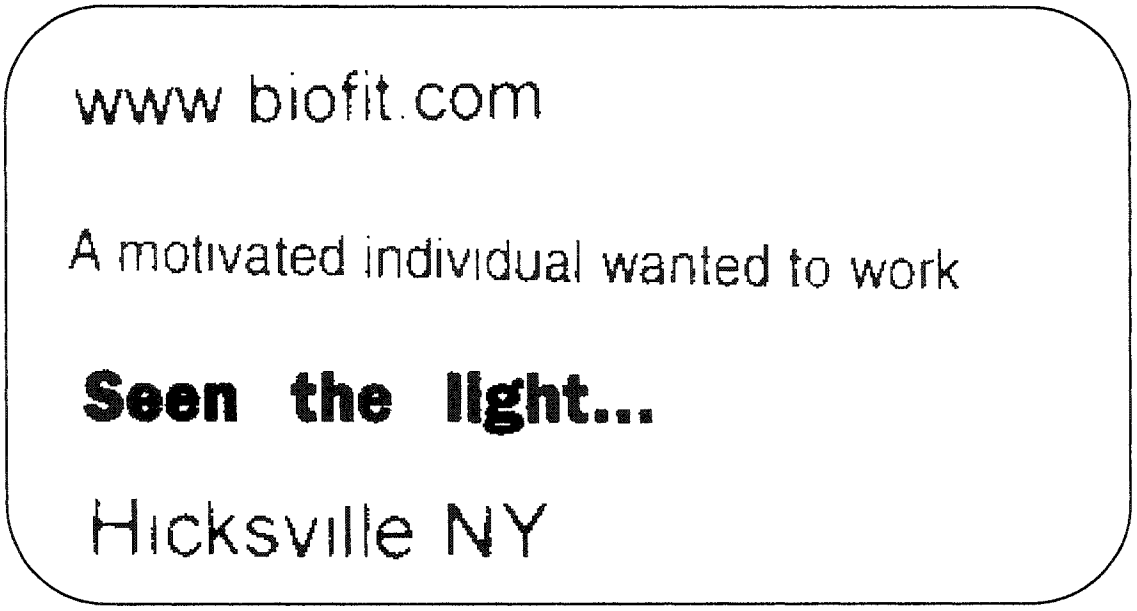


Figure 8: Sample images with no degradations

Chapter 4. Preprocessing

There are different definitions for document image analysis. It is concerned with "deriving a high level representation of the contents of a document image" [19]. Its objective is to "recognize the text and graphics components in images and extract the intended information as a human would" [2]. It can be "viewed as the interaction of generic components for the extraction of primitives, layout analysis and symbol recognition with an application specific knowledge base" [20].

A classical document image analysis system has four parts: image acquisition, preprocessing, feature extraction, and classification. Our research concentrates on some tasks that are performed at the preprocessing level. Preprocessing is the group of procedures that will enhance the original image in order to facilitate and/or provide information to the subsequent parts of the system.

Several procedures can be performed in the preprocessing module of a system, for example: skew correction, contour detection, connected component analysis, filtering,

detection of reference lines. In our system, there are three preprocessing tasks that are performed before the computation of the quality measures: connected component analysis, skew correction, and detection of reference lines. The following sections describe each of them.

4.1. Skew Correction

A document may or may not have a skew angle independently of its quality. Since a skew angle may have a negative impact on the recognition process, and the filters that are applied to the image in order to improve its quality do not address this problem, skew correction is performed in our system without considering the image quality assessment.

This task is divided into two parts. First it is necessary to detect the skew angle, in case there is any. The method described in [4] is used to perform this detection. The entropy associated with several histograms for different y projections of the image is computed. Entropy (E) is the measurement of the compactness associated with the density of points.

$$E = -\sum_{i} p_i \log (p_i) \qquad p_i = \frac{N_i}{N}$$

N_i is the total number of pixels with ordinate y_i in the histogram, and N is the total number of pixels in the image contour. The probability *pi* gives the occurrence of the ordinate y_i in the histogram. The histogram that gives the lowest value of entropy presents the skew angle.

The image contour is found by checking each black pixel as the center of a 3x3 window. If at least one of its neighbours is white, the pixel is considered part of the contour [2].

After obtaining the skew angle, if there is any, the skew correction is performed by rotating each pixel of the image using the Hook transformation [5]:

$$x' = x * cos(skew\ angle) + y * sin(skew\ angle)$$

$$y' = y * cos(skew\ angle) - x * sin(skew\ angle)$$

(x, y) → original coordinates

(x', y') → new coordinates after rotation

After finding the new coordinates, there is a test to verify if the new pixel has a fractional address. In this case, an interpolation is made in order to get integer values for x' and y'. Examples of images before and after the skew correction are shown below.

Figure 9: Sample images before and after skew correction

4.2. Connected Component Analysis

A binary image consists of black and white pixels. A black pixel can be connected to its neighbours in a 3x3 window through 8 different directions. Any group of two or more black pixels that are connected to one another is called a connected component. An important feature of a connected component is its bounding box, which is the smallest frame that can be drawn surrounding a group of connected pixels [1]. The bounding boxes provide important information about the connected components, e.g., their height and width.

There are different techniques to detect the connected components of a document image. The one used here is called region labeling or connected-component labeling [2]. A two-pass algorithm scans the image from left to right and from top to bottom assigning a different label to each group of black pixels. In the first pass, for each black pixel, the previous labeled connected pixels to the left and above are examined. If there are no labeled pixels, a new label is set to the pixel being analyzed. If there is a labeled pixel, the same label is assigned to the pixel being analyzed. If there is more than one labeled pixel but with different labels, only one label is set to the pixel being analyzed and these labels are put in an equivalence class to be merged later. The second pass of the algorithm assigns a single label to each equivalence class. Then, it reassigns the labeled pixels that were in the equivalence classes to their respective correct label. Usually four coordinates define each bounding box of connected components: the top left row and column, and the bottom right row and column.

The detection of the connected components represents an important step in our system because it will be used in the calculation of the quality measures. Examples of images from our database with their bounding boxes detected are presented below.

Figure 10: Sample images with connected components detected

4.3. Detection of Reference Lines

Four reference lines are detected for a text line, as shown in the image below.

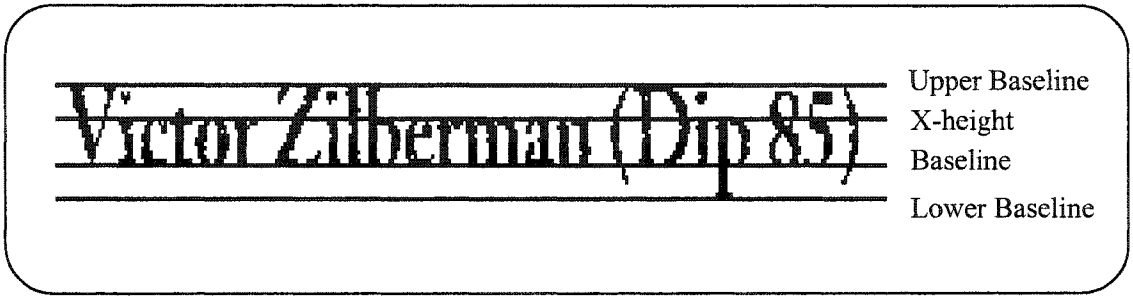


Figure 11: Image with reference lines detected

The baseline is detected using linear regression, based on [6]. A set of points is found using the middle point of the base of each connected component. The boxes that are too small to be a character, and are probably punctuation marks or noise, are discarded. A box is considered too small when its height is less than or equal to half of the distance between the baseline and the x-height.

Then, using the least squares method [6], a straight line that best fits the set of points is found. This is the first baseline. Any boxes that go below this first baseline are probably descenders and, therefore, are also discarded. Then a second baseline is found using the same method (least squares method). The first baseline is used only to discard the descenders, while the second one is more accurate. Using this method, the baseline can be correctly detected even if the image is skewed.

In order to find the x-height, the upper baseline, and the lower baseline, a horizontal histogram of black-white transitions [6] is used to analyze the core region

of a text line, which is delimited by the baseline and the x-height. This region presents a higher concentration of black-white transitions, which is shown in the histogram. A threshold defined dynamically detects the probable region of the x-height based on this information. To find the upper baseline, we start scanning the histogram from the baseline going towards the top until we find a row with no entries. The previous row is considered to be the upper baseline. Doing the inverse, going downwards in the histogram, the lower baseline is detected.

The detection of the reference lines is important for two reasons: for the calculation of the font height, and for one of the filters. Both tasks will be explained in the following chapters. Some examples of images with their reference lines detected are presented below.

Figure 12: Sample images with the reference lines detected

Chapter 5. Image Quality Assessment

5.1. Quality Measures

In order to estimate the quality of the image, five quality measures, or quality factors, that correspond to the specific degradations found in the database were defined. They are: stroke thickness factor, touching character factor, small speckle factor, broken character factor, and white speckle factor. These quality measures were chosen because they are related to the types of degradation that are found in the database, already described in chapter 3, which are broken characters, touching characters, and salt-and-pepper noise.

Almost all of these quality measures are computed using two types of information: the connected components of the image and the font height. The first one

is detected using the method described in section 4.2. The font height calculation is described below, followed by the quality measures.

5.1.1. Font Height

The font height refers to the height of characters that lie between the baseline and x-height, such as a, c, and e. Please note that the font height is image resolution independent, since it refers to the distance in pixels between these two reference lines.

The font height is calculated using the baseline and x-height as references. These reference lines are found using the method described in section 4.3. The font height is given by:

$$Font\ Height = Baseline - x\text{-}height$$

For example, in the image below the baseline corresponds to row 29 and the x-height row 12. The *font height* will be 17.

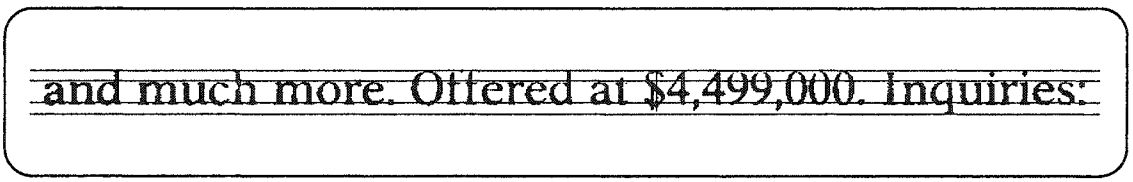


Figure 13: Image with reference lines detected

Even if the detection of the baseline and the x-height is not perfectly precise, the font height calculation is not really affected because most of the time this imprecision corresponds to only one or two rows.

5.1.2. Stroke Thickness Factor

Each character is formed by strokes that may have the same thickness or not. Different characters in the same document image may also have different stroke thicknesses. The type of font may cause these differences or some degradation like, for example, touching characters, which will connect strokes of different characters. However, even if the document has characters with different stroke thicknesses, usually there is a measurement that is the most common in the image.

The stroke thickness factor refers to this most frequent thickness found in the document. In order to calculate it, the image is scanned from top to bottom and from left to right, and searched for black pixels, either isolated or horizontally connected. A histogram is computed and its peak corresponds to the most frequent stroke thickness found in the image. Isolated pixels and small connected components that correspond to salt-and-pepper noise are not considered in this calculation. Applying a filter called edge smoothing, which will be presented in section 6.2.2, eliminates these noisy pixels.

There is a relation between the stroke thickness and the type of degradation found in the image. Images with broken characters usually have thin strokes (between 1 and 5 pixels wide), and images with touching characters usually have thicker strokes (4 or more pixels wide). The figures below illustrate these two situations by showing document images followed by the histograms that give the corresponding stroke thickness factor.

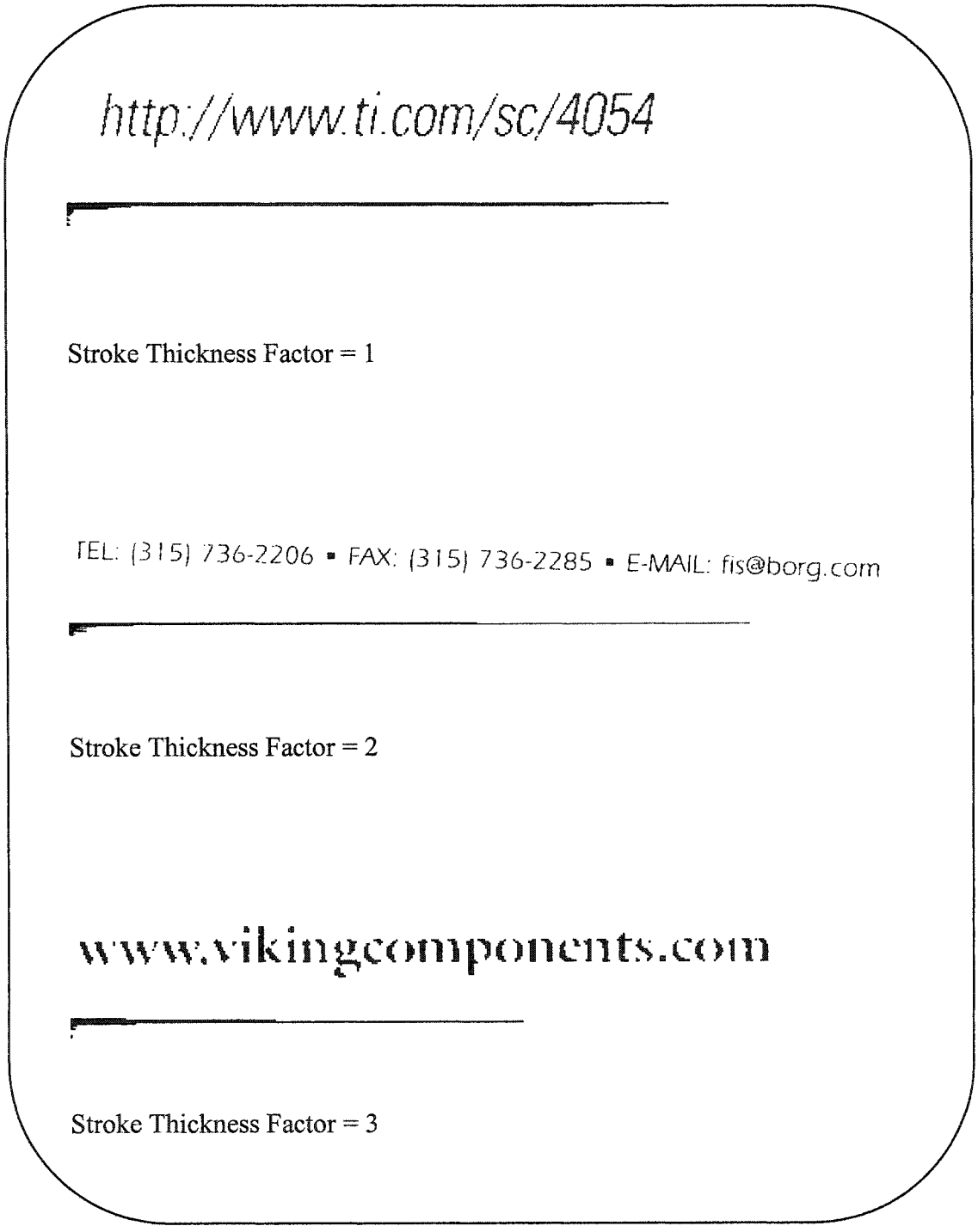


Figure 14: Images with broken characters and stroke thickness histograms

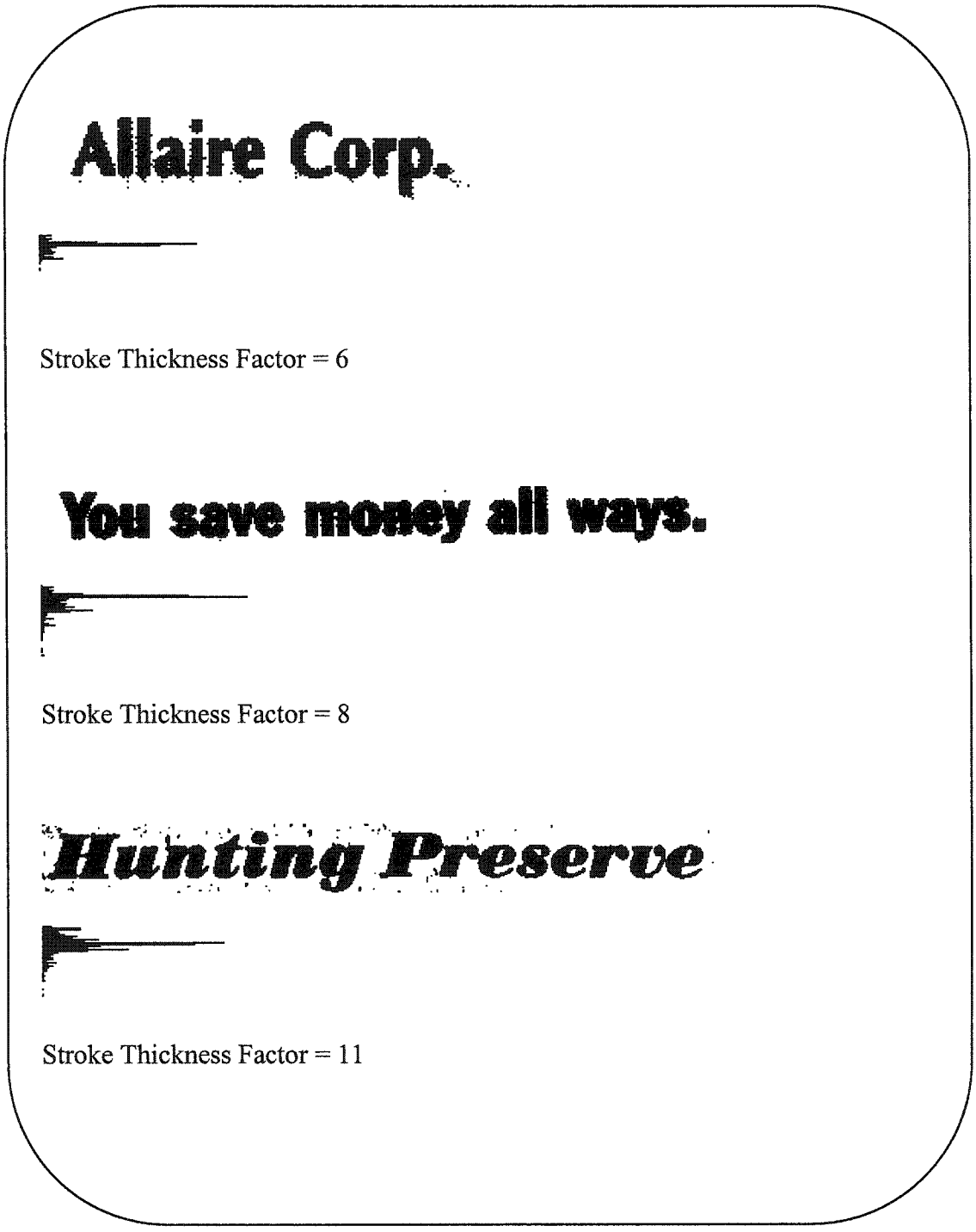


Figure 15: Images with touching characters and stroke thickness histograms

5.1.3. Touching Character Factor

In many cases where touching characters are found, the problem is related to widened strokes, which cause the characters to merge. Another cause is the presence of background speckles (salt-and-pepper noise) between the characters in an amount that may cause them to merge. These situations may happen when a page is photocopied multiple times, or when there is some problem with the photocopy machine and background speckles are inserted, or due to problems when printing the document. Independent of the reason, the presence of touching characters in a document image represents a challenge for an OCR.

The touching character factor verifies the number of touching characters in the image. This quality measure is calculated from the font height and the connected components.

A connected component that represents a character that does not touch its neighbours is roughly square, while a connected component containing touching characters is usually wider. A connected component is wide enough to be considered as a touching character when its height-to-width ratio is less than 0.75. Small speckles and broken characters are avoided by disregarding connected components with fewer pixels than *3 * (font height)* or shorter than *0.75 * (font height)*. Big globs of background speckle are avoided by disregarding connected components taller than *2 * (font height)* [7].

Note that this quality factor does not always detect all touching characters that may exist in a document, but it detects enough to identify if there are

touching characters in the image or not, especially when this factor is combined with the white speckle factor, which will be explained later in this chapter. Sometimes it identifies some characters as touching when, in fact, they are not. This happens especially to the letters "m" and "w".

The following figure shows examples of images with their touching characters detected. Bounding boxes around some groups of characters identify the ones that are touching.

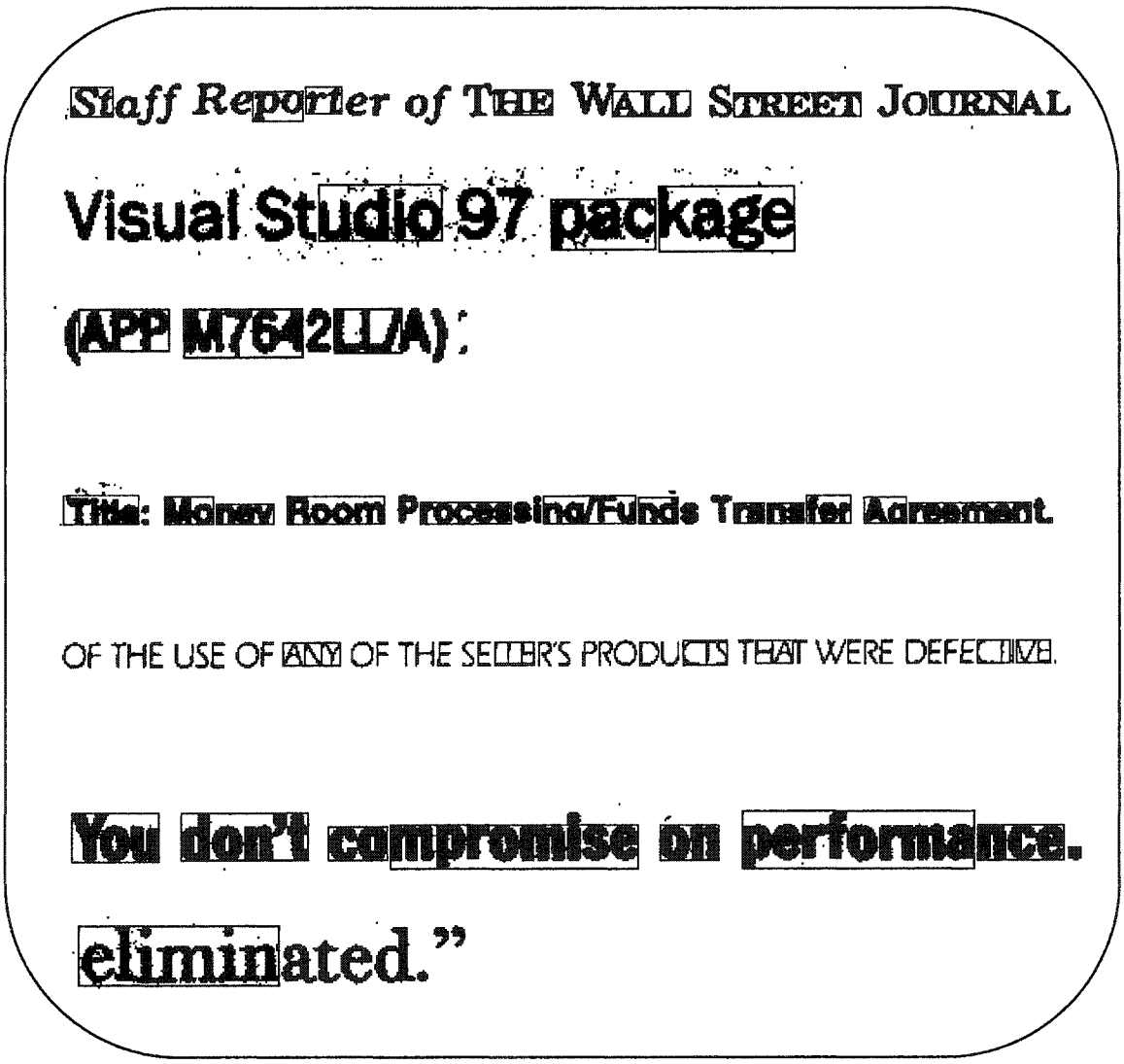


Figure 16. Sample images with touching characters detected

5.1.4. Small Speckle Factor

This quality measure checks the amount of small speckles in the image (salt-and-pepper noise). It is derived from the font height and the connected components.

As explained in the previous section, the origin of salt-and-pepper noise in a document image can be due to problems in the printing process, or in the photocopy machine, or even the result of multiple copies of a document.

Any connected component in which the amount of pixels is less than or equal to *0.5 * (font height)* is considered a small speckle.

Extracted from [7], the figure below shows the relation between a small speckle and a touching character in terms of size.

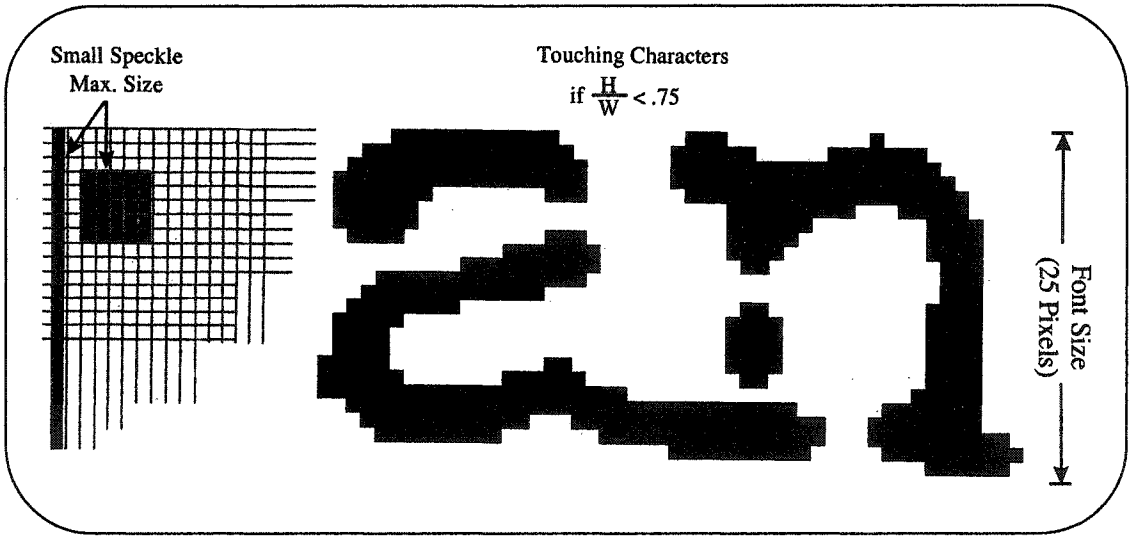


Figure 17: Touching character factor and small speckle factor

5.1.5. Broken Character Factor

This factor counts the number of connected components that are possible fragments of broken characters in the image. The calculation of this factor, which uses the connected components, is based on what is defined in [8] but with some modifications.

The bounding boxes are counted based on their height and width. In order to be considered as a broken character or a fragment of a broken character, a bounding box must have height and width smaller than 75% of the average height and width, respectively. This frequency distribution is plotted as a 3-D histogram (height, width and count of bounding boxes) divided in cells of one pixel by one pixel. An example of the histogram, extracted from [8], is presented below. The boxes that represent broken characters appear near the origin, this area is called *broken character zone.*

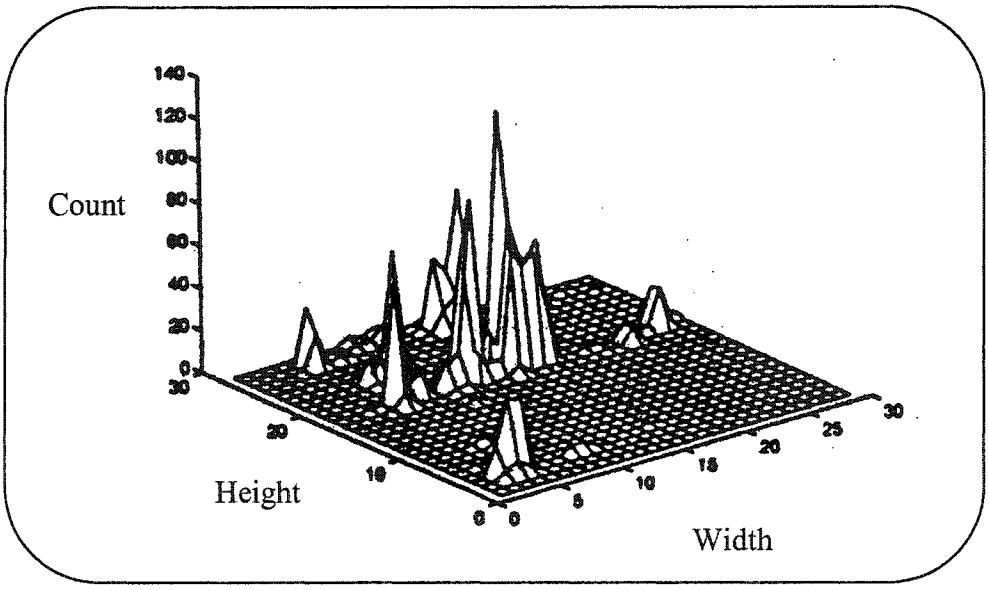


Figure 18: Frequency distribution of bounding boxes

After generating the histogram with all valid bounding boxes of the image, the calculation of the *broken character factor (BCF)* will be the sum of frequencies of all cells occupied divided by the number of cells of the histogram, as shown in the formula below:

$$BCF = \frac{\sum Frequency\ of\ Occupied\ Cells}{Number\ of\ Cells}$$

5.1.6. White Speckle Factor

Images with characters that have thick strokes caused by distortions usually have touching characters. Another consequence of this situation is the reduction or even the elimination of the white loops in letters such as "b" or "e", as illustrated in the figure below. The white speckle factor detects these small white loops in the image.

The calculation of this quality measure requires the detection of white connected components in the image. It uses the same algorithm for detecting black connected components but this time for white connected components smaller than or equal to 3x3 pixels, which are considered as white speckles. The calculation of the *white speckle factor (WSF)* is [8]:

$$WSF = \frac{\sum White\ CC \leq 3\ x3}{\sum White\ CC}$$

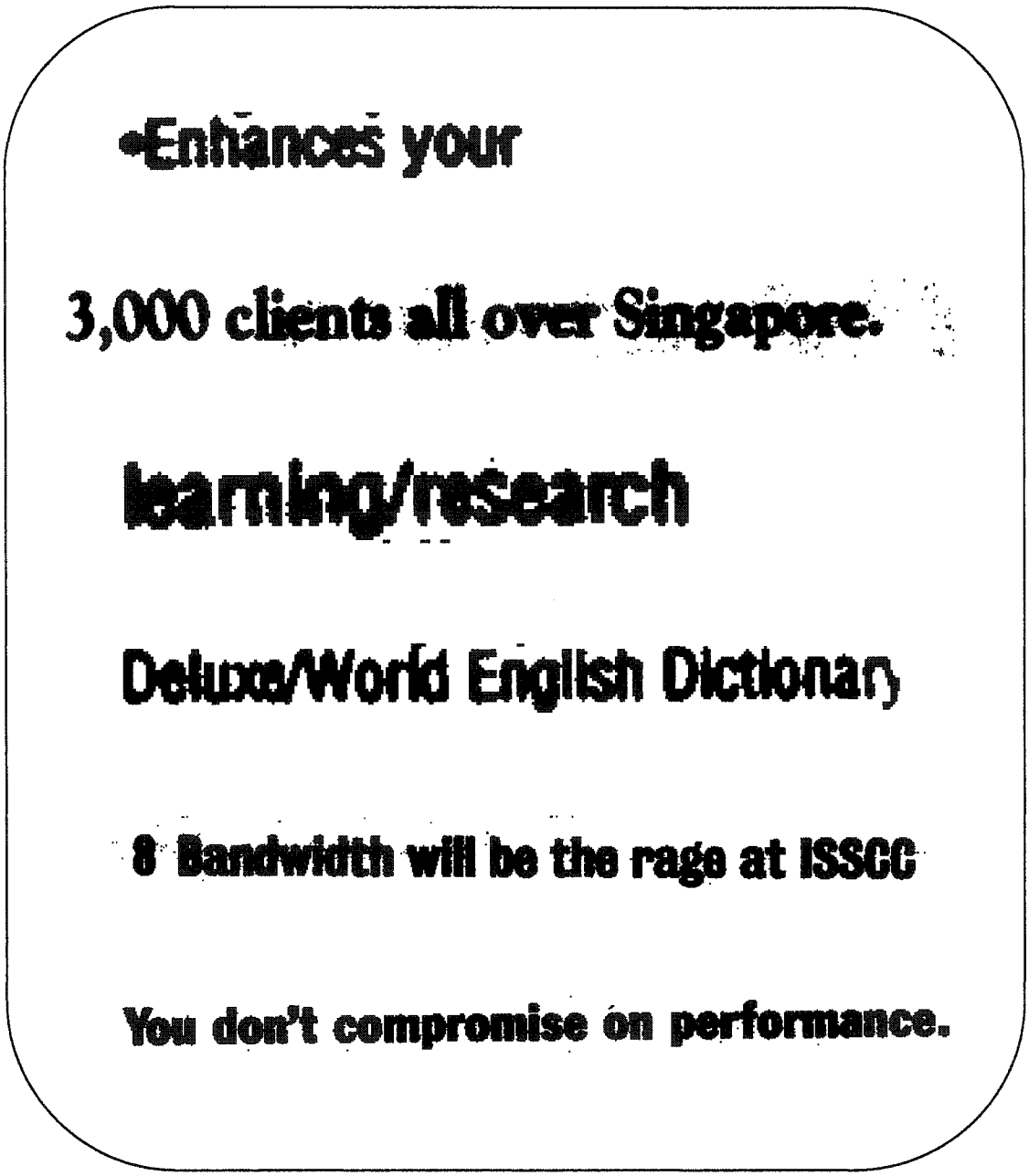


Figure 19: Sample images with reduced white loops

Chapter 6. Noise Reduction

Noise reduction or image restoration is a key task in improving the quality of the image, which will consequently provide an increase in the recognition rate. Although there are different techniques for noise reduction, we chose to work mainly with morphological filters because of the improvement obtained in the images of our database using morphological operations. It does not mean, however, that other types of filter would not work properly with our approach of estimating the image quality and then selecting a filter. Any filter can be used with the system, as long as it is appropriate for a considerable number of images in the database, and a correlation can be made between the values of the quality measures and the images that require that specific filter.

This chapter presents all restoration methods that were implemented, and also explains how the most suitable ones for our database were chosen.

6.1. Mathematical Morphology

Mathematical morphology was formalized in [9]. Its operations "tend to simplify image data by preserving their essential shape characteristics and eliminating irrelevant noise" [10].

There are two basic operations in mathematical morphology, erosion and dilation. Based on these operations, two others are defined, opening and closing. In all cases, the image is scanned from top to bottom and from left to right using a structuring element.

The following sections explain the structuring elements implemented, as well as how each operation works.

6.1.1. Structuring Elements

A structuring element is a set of point coordinates, usually with a specific shape. Four structuring elements were implemented: cross, square, horizontal line, and vertical line [9, 11], which are shown below.

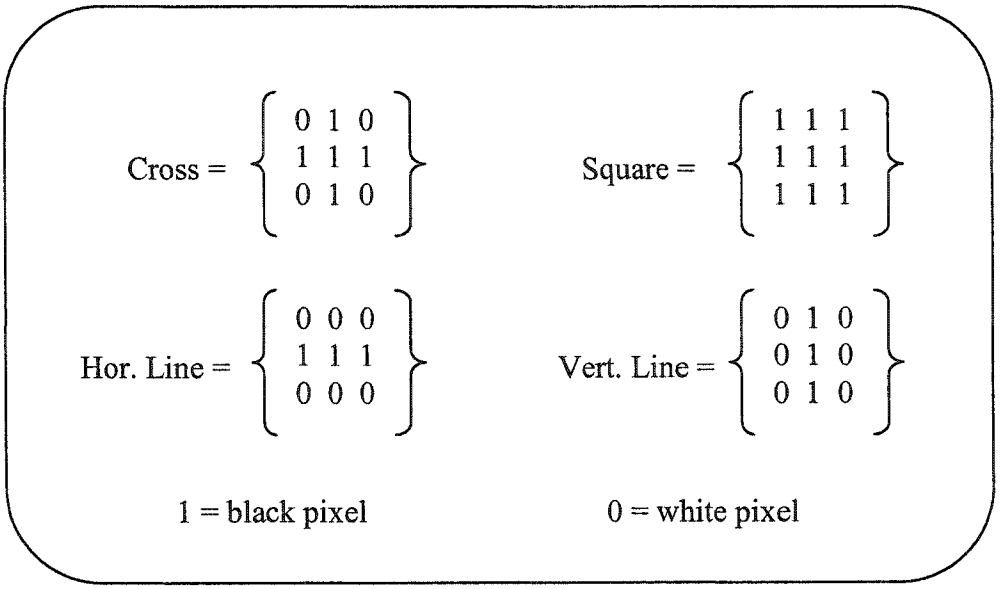


Figure 20: Structuring elements implemented

Any structuring element can be used with any operation. Consider the pixel being analyzed as "p". When the image is scanned, "p" is compared with the central pixel of the structuring element, and its neighbourhood is checked. In the output image, "p" will be either a black or a white pixel depending on the morphological operation used.

6.1.2. Morphological Operations

Erosion and dilation are the basic morphological operations. In the erosion, every time "p" and its neighbours match the structuring element, "p" will be a black pixel in the output image. Otherwise, it will be a white pixel. In the

dilation, if at least one pixel in the part of the image that is being analyzed (either "p" or any of its neighbours) matches the structuring element, "p" will be a black pixel in the output image. Otherwise, it will be a white pixel.

Based on these two operations, the operations opening and closing are defined. Opening means that an erosion operation is followed by a dilation operation, and these operations are repeated the same number of times, also called number of iterations. For example, if the erosion is performed twice, the dilation will also be performed twice. Closing is the opposite, a dilation is followed by an erosion, and also using the same number of iterations.

The four morphological operations erosion, dilation, opening and closing can be defined as [25]:

Consider A as the set that represents the original binary image:

$A = \{(x_j, y_j) \mid i(x_j, y_j) = 1\}$

Consider B as a structuring element.

$(A)_b$ is defined as the translation of A by the vector $b = (x_b, y_b)$:

$(A)_b = \{(x_j, y_j) + (x_b, y_b) \mid (x_j, y_j) \in A\}$

Erosion $A \ominus B = \bigcup_{b \in B} (A)_b$

Dilation $A \oplus B = \bigcap_{b \in B} (A)_{-b}$

Opening $A \circ B = (A \square B) \oplus B$

Closing $A \bullet B = (A \oplus B) \square B$

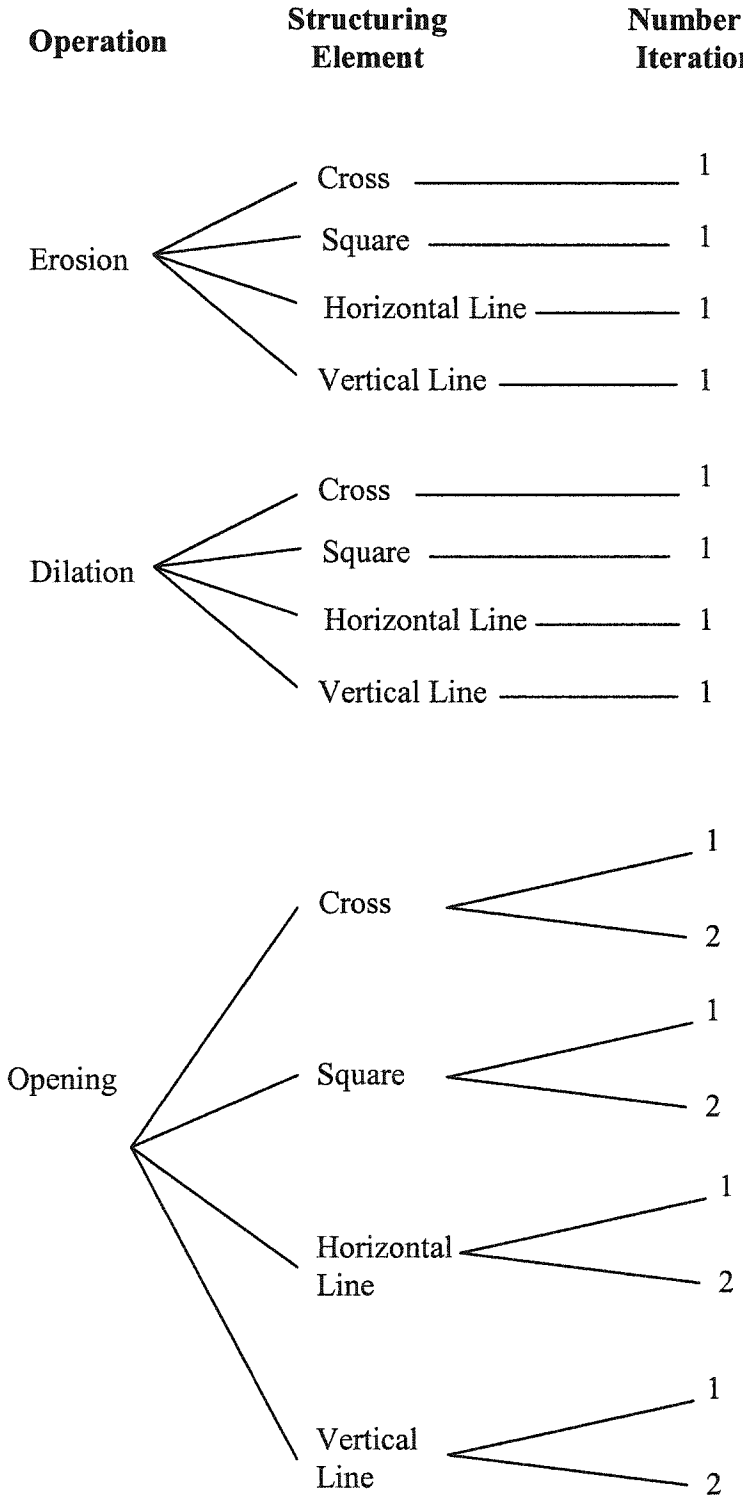
6.2. Filters

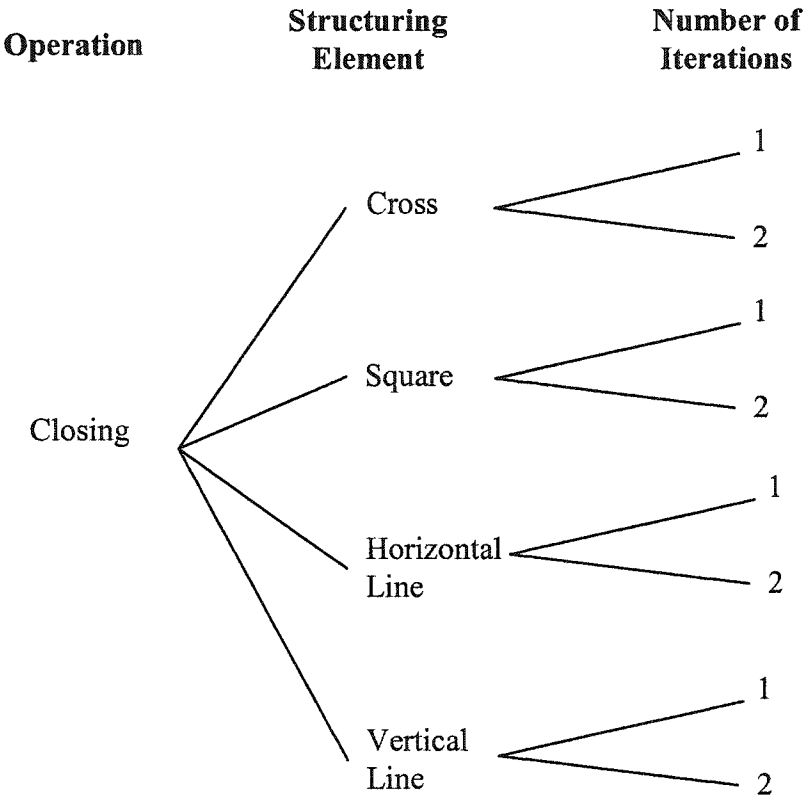
The total number of filters implemented was 25. Almost all of them (24) use mathematical morphology operations. The purpose of implementing so many filters was to choose the best among them for the database used. In the final version of the system, the number of filters was reduced to 4. This reduction was necessary not only to make an automatic selection of which filter to apply to an image, but also because many filters were not providing good results.

The filters implemented are described in the next subsection, followed by the explanation on the process of reducing this set to the most suitable ones.

6.2.1. Morphological Filters

Each morphological operation (erosion, dilation, opening, and closing) was implemented using each structuring element (cross, square, horizontal line, and vertical line). The erosion and dilation operations were implemented using one iteration, which means that they are performed only once. For the opening and closing operations, one and two iterations were tested. In the end this gives a total of 24 options of filters, 4 erosions, 4 dilations, 8 openings, and 8 closings, as summarized below.


Operation
Structuring Element
Number of Iterations
Erosion
Cross
1
Square
1
Horizontal Line
1
Vertical Line
1
Dilation
Cross
1
Square
1
Horizontal Line
1
Vertical Line
1
Opening
Cross
1
2
Square
1
2
Horizontal Line
1
2
Vertical Line
1
2



6.2.2. Smoothing

This last filter consists of a technique called edge smoothing, described in [10]. A 3x3 window is used to scan the image. When any of the patterns of figure 21a or 21b is found, the central pixel is filled. When any of the patterns of figures figure 21c or 21d is found, the central pixel is deleted. Figures 21b, and 21c are rotated 90°, 180°, and 270°.

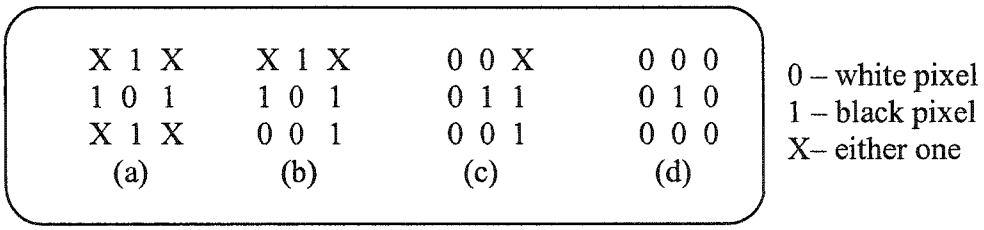


Figure 21: Patterns for edge smoothing

After this step, there is an additional test to see if there is any group of connected components smaller than a certain threshold, which is also eliminated.

6.2.3. Reduction of Filters

Although 25 filters were implemented, this number had to be reduced in order to have an automatic selection. The 25 filters were applied to the 370 images of the training set. A commercial OCR (identified in the remainder of this thesis as OCR 1) was used to select the best filter (or filters) for an image. The output that had more characters correctly recognized was considered as the best filter for the document. The option of not applying any filter at all was also tested. So for each image there were 26 outputs from OCR 1 (25 filters plus no filter) to be analyzed, giving a total of 9620 output images.

Based on the results of this OCR, which were manually analyzed, 9 filters were eliminated: all openings and closings with 2 iterations (8 filters), and edge smoothing. The 2-iteration openings and closings were not appropriate for most of the images because they either added too many pixels (closings) or removed too

many (openings). The number of images that had any of these filters as the best option was very small, in some cases even non-existent. The edge smoothing technique was not suitable because in many cases the "i" dots and/or punctuation marks were removed.

After eliminating these options, the number of filters was reduced to 16 plus the option of using no filter at all. The output images of the training set were submitted to two other commercial OCR's (OCR 2 and OCR 3). The objective was to demonstrate that the improvement obtained in the recognition rate by applying the filters was not subject to only one OCR. The results were:

Table 2: Comparison of OCR's

OCR	Recognition rate without filter	Recognition rate with best filter (manually chosen)
OCR 1	65.04%	89.84%
OCR 2	71.47%	86.65%
OCR 3	66.86%	94.44%

The best filter to each image was manually chosen based on the output that had more characters correctly recognized. OCR 3 was considered the best one because it obtained the highest recognition rate after applying the best filter for each image. Besides that, it allows the user to decide if options like "despeckle the image" should be used or not. This provides more control of the process at the same time that shows more clearly the improvement that can be achieved when a better quality version of the image is tested. This may explain why this OCR had the lowest performance without using any filter at all, since it was not possible to "turn off" some preprocessing tasks of the other two OCR's.

The number of filters was again reduced to the most efficient ones based on the recognition results provided by OCR 3. The option of using no filter at all was eliminated because, although there were many images that had this option as one that provided the best result possible, this only happened probably because all OCR's have some noise reduction algorithm already built in, and it was not possible to shut off completely this option in any case. Since almost all images have very poor quality, for this specific database it is better to always have some kind of filter applied.

The remaining 16 filters were manually reduced to 4 also by checking the options that had more characters correctly recognized, providing therefore the highest recognition rates. These 4 filters are the ones that are available to be automatically selected by the system to be applied to a document image.

The 4 filters selected as the most suitable ones for the training set were:

- Erosion with Cross
- Opening with Horizontal Line
- Dilation with Cross
- Closing with Vertical Line

The system will choose only one filter to apply to a given image. Because of our selection of filters to use, there is no need to apply more than one of these options to an image. The options "Opening with Horizontal Line" and "Closing with Vertical Line" already represent the combination of the morphological operations erosion and dilation. The combination of filters would actually decrease the quality of the image.

Chapter 7. The System

The previous chapters explained the different parts of the system, such as: preprocessing tasks, image quality assessment, and the filters available for automatic selection. Preprocessing tasks involve skew correction, detection of connected components, and detection of reference lines. Image quality assessment involves the computation of five quality measures, which are stroke thickness factor, touching character factor, small speckle factor, broken character factor, and white speckle factor. Four morphological filters are available in the system to improve the quality of the image. Only one is automatically chosen for a given image.

The following section describes the sequence in which all these tasks are performed prior to the automatic filter selection. Also described in this chapter is the automatic filter selection process, as well as how the quality measures interact.

7.1. Sequence of Tasks

The first step is the skew detection and correction of the document image. After this, a temporary image is created to contain the skew corrected original image. This temporary image is required because the edge smoothing technique (described in section 6.2.2) is applied to the image. This noise reduction is necessary to eliminate any small speckles that could interfere with the calculation of the font height, the stroke thickness factor, and the broken character factor. The original image needs to be preserved because it will be used later to calculate the other quality measures, and, eventually, apply the selected filter.

Using this temporary image, the reference lines are detected. Any black pixels above the upper baseline and below the lower baseline are removed. Then the font height, the stroke thickness factor, and the broken character factor are calculated. Next comes the calculation of the remaining quality measures: touching character factor, small speckle factor, and white speckle factor.

The five quality measures are used to select the best filter for an image, which has to be only one from the four options available. This selection is made using a set of rules that were defined based on the combination of the quality measures (examples of these rules are given in the following section). The selected filter is then applied to the skew corrected original image. The reference lines are detected in this new image to remove any remaining black pixels above the upper baseline or below the lower baseline. The output of the system is a newer image with the least amount of noise possible.

7.2. Interaction of Quality Measures

The values of the individual quality measures vary within a certain range for each group of morphological operations, dilation or closing and erosion or opening. Usually the analysis of each isolated quality measure cannot make a decision on the type of operation to use, since there is an overlap between the two groups, which means that for each quality measure, there is a range of values that is common to both types of morphological operations. Extreme values of a quality measure, which sometimes could be the minimum and sometimes the maximum, can indicate the choice to be made. However, for most images in the training set, the values to be analyzed were exactly in this overlapping range. This is why it is necessary to analyze the interaction of the quality measures in order to make the decision on which type of morphological operation to be applied to an image.

Once the type of operation is defined, the same quality measures are used to define the structuring element. Once again only in a few cases extreme values of a quality measure will determine the correct choice. In most cases the combination of these factors will lead to a decision. However, in this stage the choice between the types of structuring elements is not as clear as the choice of the type of operation. But it is possible to make a decision by analyzing the quality measures together.

The stroke thickness factor can identify the images that need a dilation/closing operation or an erosion/opening operation. Based on the training set, images with stroke thickness of more than 7 do not require a dilation/closing operation, and most images that need this operation, have this value between 1 and 5. Images that need an

erosion/opening operation usually have the stroke thickness value equal to or larger than 4. The graph presented in figure 22 below illustrates this quality measure. Analyzing this graph, we can see images with stroke thickness larger than 7 in the group of dilation/closing and also images with stroke thickness smaller than 4 in the group of erosion/opening. This happens because there are a few cases of images that require either a dilation/closing operation or an erosion/opening operation. Usually these images have better quality than the majority of images in the database.

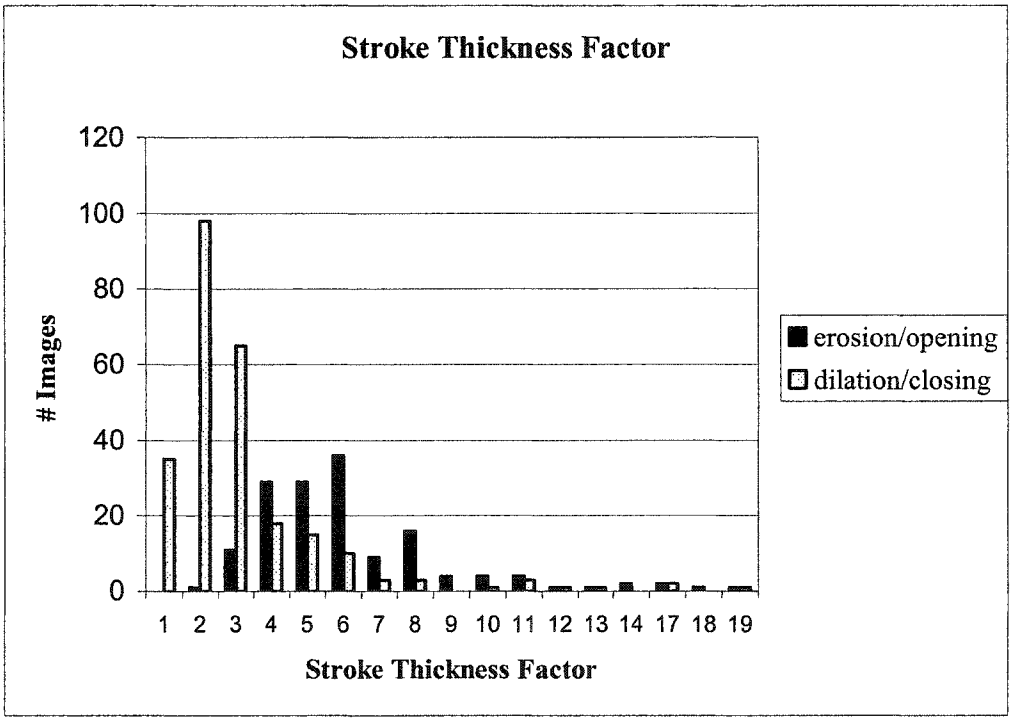


Figure 22: Distribution of values for Stroke Thickness Factor

The touching character factor does not detect all the touching characters precisely but it detects enough to decide if the image contains touching characters or not. The value for the touching character factor varies between 0 and 1 for most images that need a dilation/closing operation. In images that need an erosion/opening operation, this value is usually larger than 0. The graph below shows the distribution of values for this quality measure.

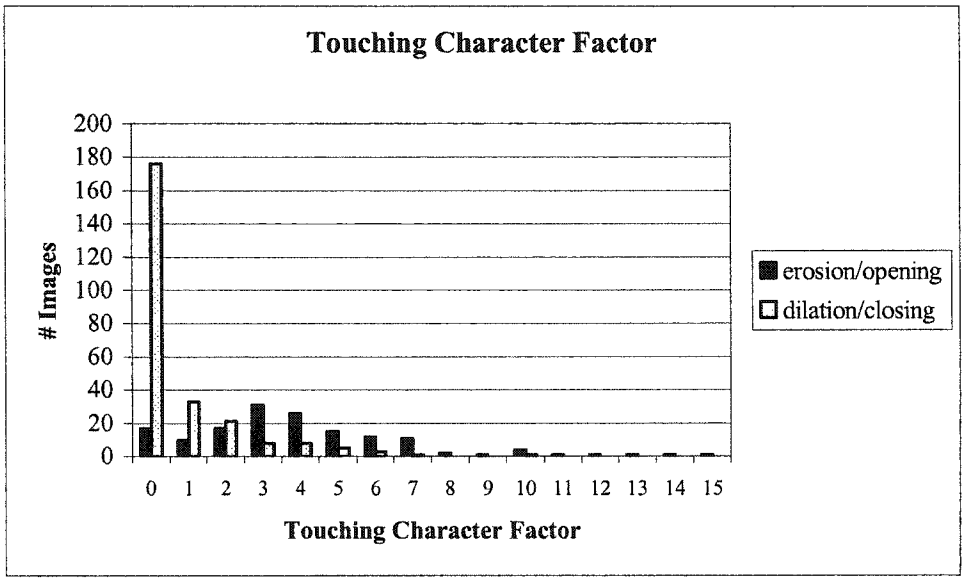


Figure 23: Distribution of values for Touching Character Factor

The white speckle factor indicates the presence of widened strokes, which can be often found in images with touching characters. So this quality measure usually complements the previous one, the touching character factor, and they are analyzed together. Most of the time, images with reduced white loops have their white speckle factor bigger than zero, which means that they need an erosion/opening operation.

The touching character factor is usually positive in these cases. Images without the problem of reduced white loops usually have this value equal to zero and need a dilation/closing operation. The touching character factor varies between 0 and 1 in most of these cases. The graph below illustrates the white speckle factor.

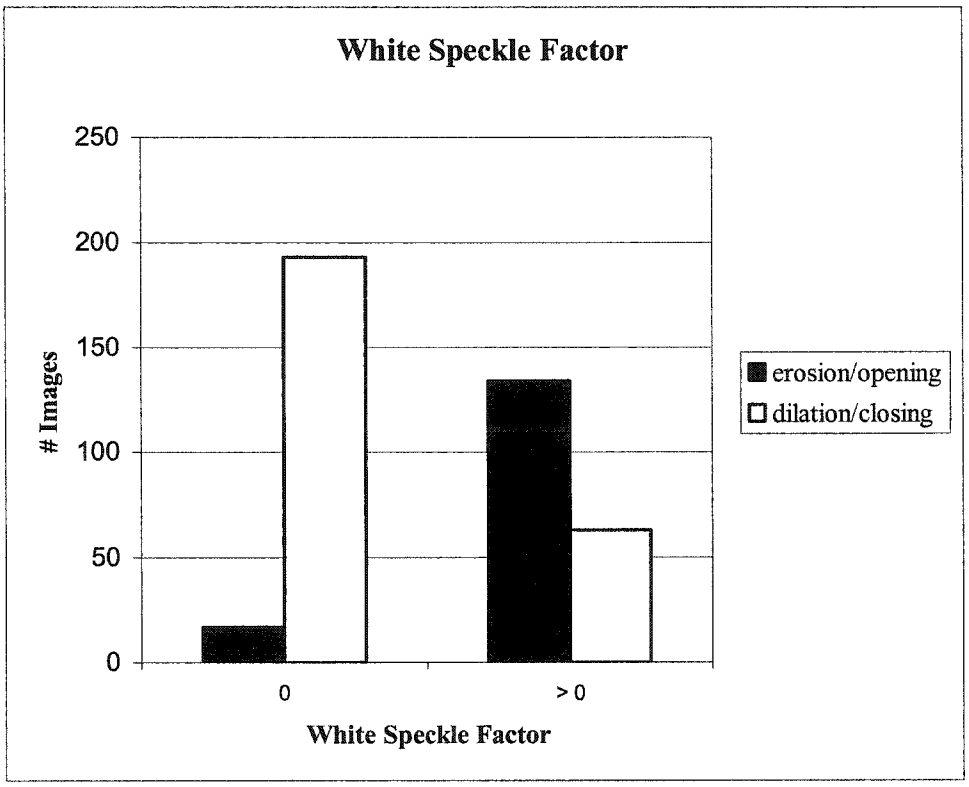


Figure 24: Distribution of values for White Speckle Factor

When the broken character factor is equal to or less than 0.01, the image usually needs an erosion/opening operation. Otherwise, it needs a dilation/closing operation, as shown in the graph in figure 25.

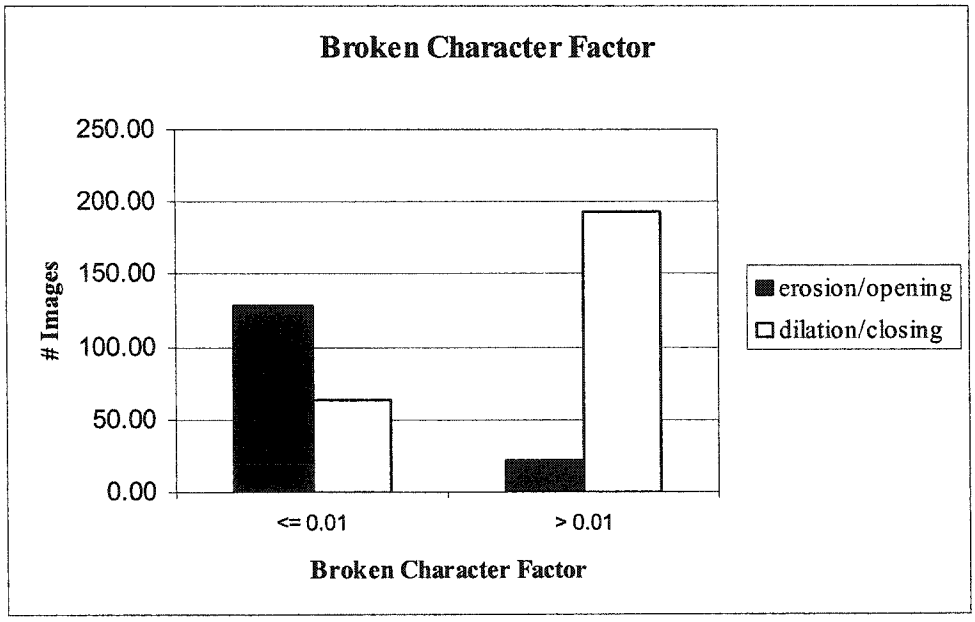


Figure 25: Distribution of values for Broken Character Factor

The small speckle factor detects correctly the salt-and-pepper noise but sometimes also detects small fragments of broken characters as noise. This happens especially when the image has thin strokes. The values vary a lot for both groups of morphological operations, so this quality factor is usually analyzed with the others. A large value for the small speckle factor with a large value for stroke thickness or a large value for the touching character factor usually means that the image has a lot of salt-and-pepper noise with touching characters. However, a large value for the small speckle factor with a large value for broken character factor usually means that the image has many small fragments of broken characters.

7.3. Automatic Filter Selection

The four morphological filters available in the system are: Erosion with Cross, Opening with Horizontal Line, Dilation with Cross, Closing with Vertical Line.

After computing the values for the five quality measures, the decision process for selecting the best filter for an image is divided in two stages. The first one is related to the type of operation that will be performed. There are two choices, either erosion/opening or dilation/closing. Erosion and opening are grouped together at this point because both refer to removing pixels from the image. Similarly, dilation and closing are grouped together because both refer to adding pixels to the image. Besides that, the values of the quality measures make a clear distinction between these two groups. The first group will be appropriate for images with touching characters and/or salt-and-pepper noise, while the second group will be used for images with broken characters.

The second stage is related to the structuring element that will be used. For the erosion/opening group, the choice is between cross and horizontal line. For the dilation/closing group, the options are cross and vertical line. For this second stage of decision, there is no specific characteristic in the image that makes obvious the choice of one structuring element over the other in each group, but it is possible to make a choice when the quality measures are analyzed together.

There are no specific values of each quality measure that lead to the selection of a particular filter for an image. It is rather the combination of these values that leads to a decision. A set of rules was defined based on the values observed in the training set. A

score is computed for each option of filter and in the end the highest one is selected as the best filter for the image. Examples of these rules are presented in section 7.5. The table below shows the rate in which the two stages of the decision process were correctly made.

Table 3: Automatic choice of filter

Set	First Stage	Second Stage
Training	97.57% (361/370)	90.58% (327/361)
Validation	97.27% (178/183)	88.76% (158/178)
Test	96.17% (176/183)	85.80% (151/176)

The calculation of the rates for the second stage was based on the results of the first one. For example, in the training set the right choice was made in 361 out of 370 cases for the first stage. So the statistics for the second stage are based on the total of 361 images, since it is not possible any more to achieve a correct choice in all 370 cases for the second stage if some of these choices were wrong in the first one.

Please note that both stages of decision are necessary in order to choose the filter to be applied. The first stage only selects the type of operation that will be used, which is not enough if the structuring element is not selected. The first stage of decision, however, plays a critical role in improving the recognition rate, while the second stage indeed plays a complementary role. Hence even in one of the worst cases that the first stage of decision is always right and the second is always wrong, there will be an increase in the recognition rate from 66.86% to 71.65% in the training set. By

choosing the correct option in the second stage, the recognition rate can increase up to 93.10% in the training set.

The reason for the importance of the choice of the right type of operation is that an error in the first stage of decision will, most of the time, have as a consequence the decrease of the recognition rate when compared to the option of not using any filter at all. However, once the correct option is chosen in this first stage, in most cases any of the options for the structuring element (second stage of decision) will provide some improvement in the quality of the image and in the recognition rate, but there is one that will be better than the other when the recognition rates of both options are compared.

7.4. Errors in the Automatic Filter Selection

As explained in the previous section, an error in the first stage of decision is the critical one. The errors that occurred in our tests were mainly because of values in the quality measures that led to the wrong decision. This happened, for example, in cases where existing touching characters were not detected, therefore the quality measures pointed to dilation when actually erosion had to be chosen. The opposite also happened when characters such as "m" and "w" were detected as touching characters when they were actually single characters. Based on this information, erosion was chosen instead of a dilation operation.

7.5. Decision Rules

The set of rules that decide which filter to apply to a given image was defined based on the analysis of the quality measures. There are 23 rules in the first stage of decision. In the second stage of decision, there are 19 rules for the choice of the structuring element for the dilation/closing group and 15 rules for the erosion/opening group. In both stages of decision only a certain amount of these rules are executed for an image, since there are different paths available depending on the values of the quality measures. Basically all quality measures are checked, individually and in combination with one another. A score is computed for the options available in each stage of decision. The rules make the score increase or decrease for each option. In the end, the option with the highest score is the one chosen.

A few examples of the rules used in both stages of decision are presented below. Before we review how the quality measures are calculated. For a detailed description of the quality measures, please refer to chapter 5.

- Quality Measures
  - Stroke Thickness Factor = most frequent horizontal stroke thickness in the image
  - Touching Character Factor = $\frac{\text{CC Height}}{\text{CC Width}} < 0.75$
  - Small Speckle Factor = CC Pixels ≤ 0.5 * Font Height

- Broken Character Factor = $\dfrac{\sum \text{Frequency of Occupied Cells in BCZ}}{\text{Number of Cells in BCZ}}$

- White Speckle Factor = $\dfrac{\sum \text{White CC} \leq 3\text{x}3}{\sum \text{White CC}}$

Where:

Font Height = baseline – x-height

CC = Connected Component

BCZ = Broken Character Zone

➢ Examples of Rules of the first stage of decision

- If *Stroke Thickness Factor* < 3 ==> Dilation / Closing
- If *Stroke Thickness Factor* > 7 ==> Erosion / Opening
- If *Broken Character Factor* > 0 ==> Increase score of Dilation / Closing
- If *Broken Character Factor* > 4 ==> Dilation / Closing
- If *Broken Character Factor* = 0 AND

  Small Speckle Factor ≥ 40 ==> Erosion/Opening
- If *Touching Character Factor* > 0 AND

  White Speckle Factor > 0 ==> Increase score of Erosion / Opening
- If *Touching Character Factor* = 0 => Increase score of Dilation / Closing
- If *Touching Character Factor* > 0 AND

  Small Speckle Factor > 0 => Increase score of Erosion / Opening
- If *Touching Character Factor* > 6 => Erosion / Opening

- Examples of Rules of the second stage of decision when the first stage was Dilation / Closing

  - If *Stroke Thickness Factor* > 5 ==> Cross
  - If *Stroke Thickness Factor* > 4 OR

    Stroke Thickness Factor < 2 => Cross
  - If ((*Stroke Thickness Factor* >= 2 AND *Stroke Thickness Factor* <= 4)

    AND (*Broken Character Factor* > 3.6 OR *White Speckle Factor* > 0.6))

    ==> Vertical Line
  - If *Broken Character Factor* > 7 ==> Cross
  - If *White Speckle Factor* = 0 AND *Broken Character Factor* > 0.54 ==> Cross
  - If *White Speckle Factor* > 0 AND *Stroke Thickness Factor* > 4 ==> Cross

- Examples of Rules of the second stage of decision when the first stage was Erosion / Opening

  - If *Stroke Thickness Factor* > 8 ==> Cross
  - If *Touching Character Factor* >= 3 AND *Touching Character Factor* <= 7

    AND *Stroke Thickness Factor* < 4 => Horizontal Line
  - If *Touching Character Factor* >= 3 AND *Touching Character Factor* <= 7

    AND *White Speckle Factor* > 0.76 => Cross
  - If *Small Speckle Factor* > 340 ==> Horizontal Line
  - If *Broken Character Factor* > 0.09 ==> Horizontal Line

Figure 26 below shows an example of an image before and after applying the filter. The first (a) is the original image, and the second (b) is the image after the filter. The values of the quality measures for the original image are:

Stroke Thickness Factor = 9

Touching Character Factor = 5

Small Speckle Factor = 66

Broken Character Factor = 0

White Speckle Factor = 0.47

In the first stage of decision, the score computed was -47 for dilation/closing and 11 for erosion/opening. So the choice is erosion/opening in this stage. After that, the score is 6 for the structuring element cross and –150 for the structuring element horizontal line. The choice of filter for this image is Erosion with Cross.

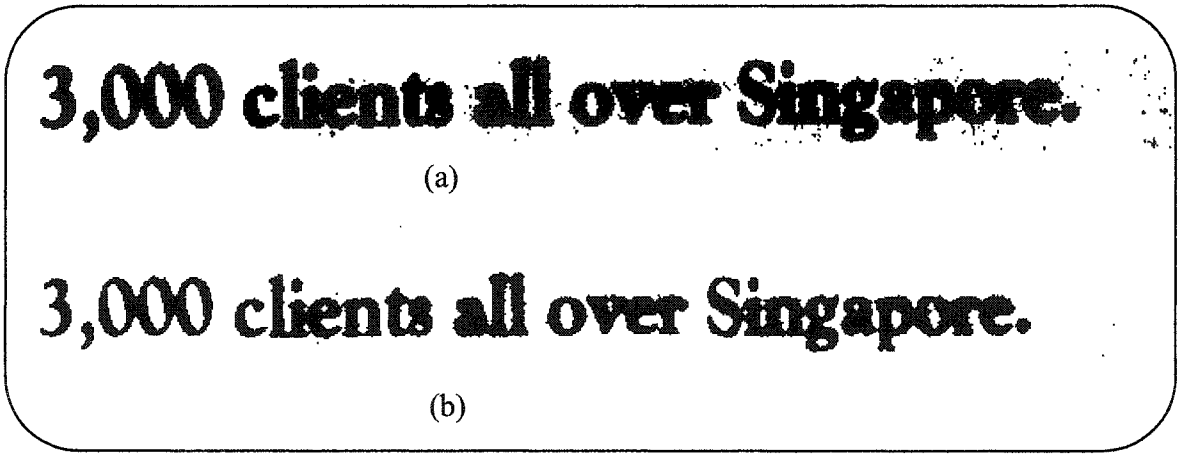


Figure 26: (a) Original image (b) Image after filter (selection and use)

Chapter 8. Experimental Results

Although the objective of the system is to automatically select the most suitable filter for an image based on its quality, the best way to verify its performance is by analyzing the improvement obtained in the recognition rates once a filter is applied to an image. This is why our experimental results will be expressed in terms of the recognition rates achieved before and after a filter is applied to each image. We do not make any comparison with experimental results obtained in other works because we could not find any other that is similar to ours, using the same database or at least poor quality printed documents. The closest one, as mentioned in section 2.2, uses only typewritten images.

The four filters selected as the most suitable ones for the database were applied to the 370 images of the training set, and the output was submitted to OCR 3. The best filter among the 4 was manually chosen for each image based on the output with the highest number of characters correctly recognized. In some cases, more than one filter could

provide the same best result. The original images were also submitted to OCR 3 in order to compare the recognition rates before and after applying the best filter.

The original images of the training set were processed by the system to have the best filter for each of them automatically chosen and applied. Only one among the four filters available is chosen and applied to a given image. The results could be verified because the best filter for each image was manually chosen previously.

The same steps were repeated with the validation set in order to make final adjustments to the set of rules. After reaching the final version of the system, these steps were repeated with a test set of 183 images to compare the results, which are presented in the table below:

Table 4: Experimental results using OCR 3

Set	Recognition Rate without Filter	Recognition Rate with Best Filter (manually chosen)	Recognition Rate with Best Filter (automatically chosen)
Training	66.86%	93.10%	90.37%
Validation	71.73%	95.45%	93.97%
Test	73.24%	95.13%	93.09%

Table 4 shows the recognition rates achieved for each set in three different situations. First, without using any filter (recognition rate without filter), which means that the original images were submitted to OCR 3 without any improvement at all. The second situation shows the recognition rates when the best filter for each image was manually selected. The system was not used in this case. The last column shows the recognition rates when the best filter for each image was automatically chosen, which means that these are the results obtained by the system that was implemented. The recognition rates

are higher when the filters were manually chosen, as expected, because in this case there are no errors in choosing the right filter. Comparing these rates with the ones achieved by the automatic filter selection, we can see that the values are close.

Analyzing the increase in the recognition rates before and after applying the filters proves the effectiveness of the method chosen to select the best filter possible to an image among a specific set. The test set achieved better recognition rates when compared to the training set most probably because its images have a slightly better quality.

The advantage of using image quality assessment to choose the best filter for a specific image becomes even clearer when we compare the results achieved by this method with the recognition rates that are obtained by applying only one filter to the whole database. Without estimating the quality of the image before attempting to apply a cleaning algorithm, it would not be possible to address different types of degradation with different filters. Table 5 shows the recognition results when each filter is applied to the whole database, without choosing the best option. The last column shows the option of choosing the best among the four filters.

Table 5: Recognition rates for each filter using OCR 3

Set	Closing with Vertical Line	Opening with Horizontal Line	Dilation with Cross	Erosion with Cross	Recognition Rate with Best Filter (automatically chosen)
Training	64.94%	40.02%	69.71%	33.60%	90.37%
Validation	73.72%	40.87%	71.50%	35.22%	93.97%
Test	69.64%	42.01%	74.74%	31.51%	93.09%

The recognition rates of each filter independently are so low, even when compared to the option of using no filter at all, because the use of an inappropriate filter will, most of the time, further deteriorate a poor quality image. Therefore, the OCR cannot recognize any character at all in many of these images. When an opening, and especially an erosion operation, is applied to an image with broken characters, the resulting image has even more fragments of broken characters, and sometimes nothing at all. The opposite happens when a closing, and especially a dilation operation, is applied to an image with touching characters and/or salt-and-pepper noise. In this case, the output image has even more touching characters and/or noise, which again will result in very low recognition rates. Figures 27 and 28 below illustrate these two situations.

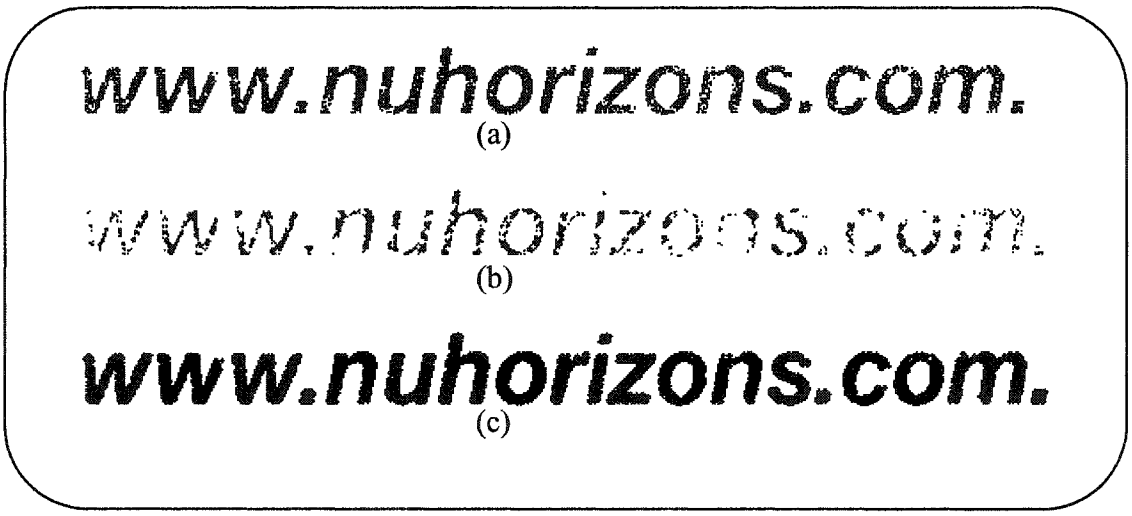


Figure 27: (a) Original image, (b) Image after wrong filter, (c) Image after right filter

In Figure 27, the best filter for the image, according to the recognition results, is Dilation with Cross. However, the system made a wrong choice in the first stage of the decision, which was Erosion. After that, the choice of the second stage of decision was the structuring element cross. The original image is the first one (a). The output of the

wrong choice of filter is the second image (b), which has more broken characters than the (a). The third image (c) shows the result that is achieved when the correct filter is applied.

Figure 28 shows the opposite situation. The best filter, based on the recognition results, is Erosion with Cross. Again the first stage of the decision made the wrong choice, which was Dilation. Cross was the choice of the second stage of decision. When we compare the original image (a) with the output of the wrong filter (b), we see thickened strokes that cause some characters to merge, as well as more evident salt-and-pepper noise. The third image (c) shows the result that is achieved when the correct filter is applied. Both examples are really errors that happened in the experiments.

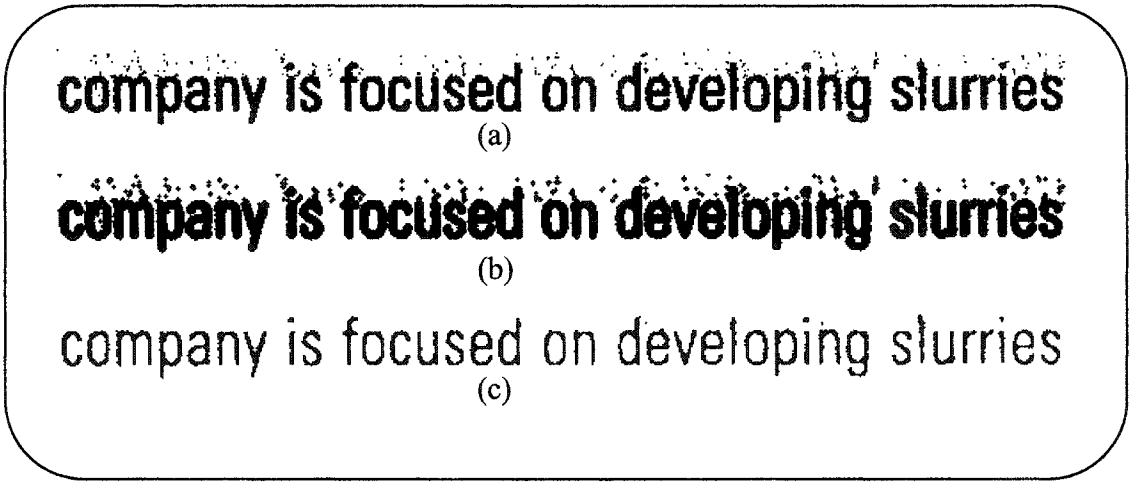


Figure 28: (a) Original image, (b) Image after wrong filter, (c) Image after right filter

8.1. Comparison of OCR's

In section 6.2.3 we compared the recognition rates obtained for the training set using three different commercial OCR's. At that point, the process of reducing the number of filters that were implemented was being explained, so those results are based on 16 filters and there was no automatic filter selection. Because OCR 3 was considered the best among these three, it was the only one used for the development of the set of rules that makes the automatic filter selection. All the experimental results presented in this chapter are also based on this OCR.

In order to demonstrate that the improvement in the recognition rates achieved with our system is not dependent on the results of only one OCR, we compare again the results obtained by these three commercial OCR's for the training set. However, only the four filters available in the system for the automatic filter selection are considered. The following table presents the recognition results when no filter is used, as well as the results after the manual and the automatic filter selection.

Table 6: Comparison of OCR's

OCR	Recognition Rate without Filter	Recognition Rate with Best Filter (manually chosen)	Recognition Rate with Best Filter (automatically chosen)
OCR 1	65.04%	85.87%	80.85%
OCR 2	71.47%	83.54%	80.75%
OCR 3	66.86%	93.10%	90.37%

The improvement in the recognition results is much better with OCR 3, as expected, because the system was developed using the results of this OCR. However, there is also a considerable improvement in the results obtained with the other two OCR's with the automatic filter selection.

8.2. Comparison of Number of Connected Components

Another analysis that can be made before and after applying the filters is regarding the number of connected components in the document images. Although we do not have access to the type of features extracted by the commercial OCR's in the recognition process, it is possible that this information is one of them, since connected component analysis is a widely used feature.

In most images of the database, the number of connected components in the image is usually smaller after applying a filter, as shown in the table below. The results are presented according to image degradations to facilitate the analysis.

Table 7: Comparison of Number of Connected Components

Type of Degradation	Number of Images		# Connected Components before applying a Filter	# Connected Components after Automatic Filter Selection
Only Broken Characters	369	50.14%	23262	15705
Only Touching Characters	45	6.11%	1062	1069
Only Salt-and-Pepper Noise	17	2.31%	1447	456
Touching Characters with Salt-and-Pepper Noise	229	31.11%	23264	5589
Broken-Characters with Touching Characters	22	2.99%	1093	553
Broken Characters with Touching Characters and Salt-and-Pepper Noise	1	0.14%	315	290
No Degradations	53	7.20%	1272	1198
Total	736	100%	51715	24860

This reduction in the number of connected components happens to the images with broken characters because some fragments are grouped together after the filter. Images with salt-and-pepper noise also have less bounding boxes after the filter because the noise is removed from the image. There is a small increase in the number of connected components in the images with only touching characters because some merged characters are separated after the filter.

8.3. Images Before and After the Filters

Examples of images before and after the automatic filter selection process are presented in the figures below. The original image is followed by the output obtained after the selected filter is applied. Figures 24, 25 and 26 show examples of the filters Dilation with Cross and Closing with Vertical Line. Figures 27 and 28 show examples of the filters Erosion with Cross and Opening with Horizontal Line.

Figure 29: Sample images before and after applying a filter

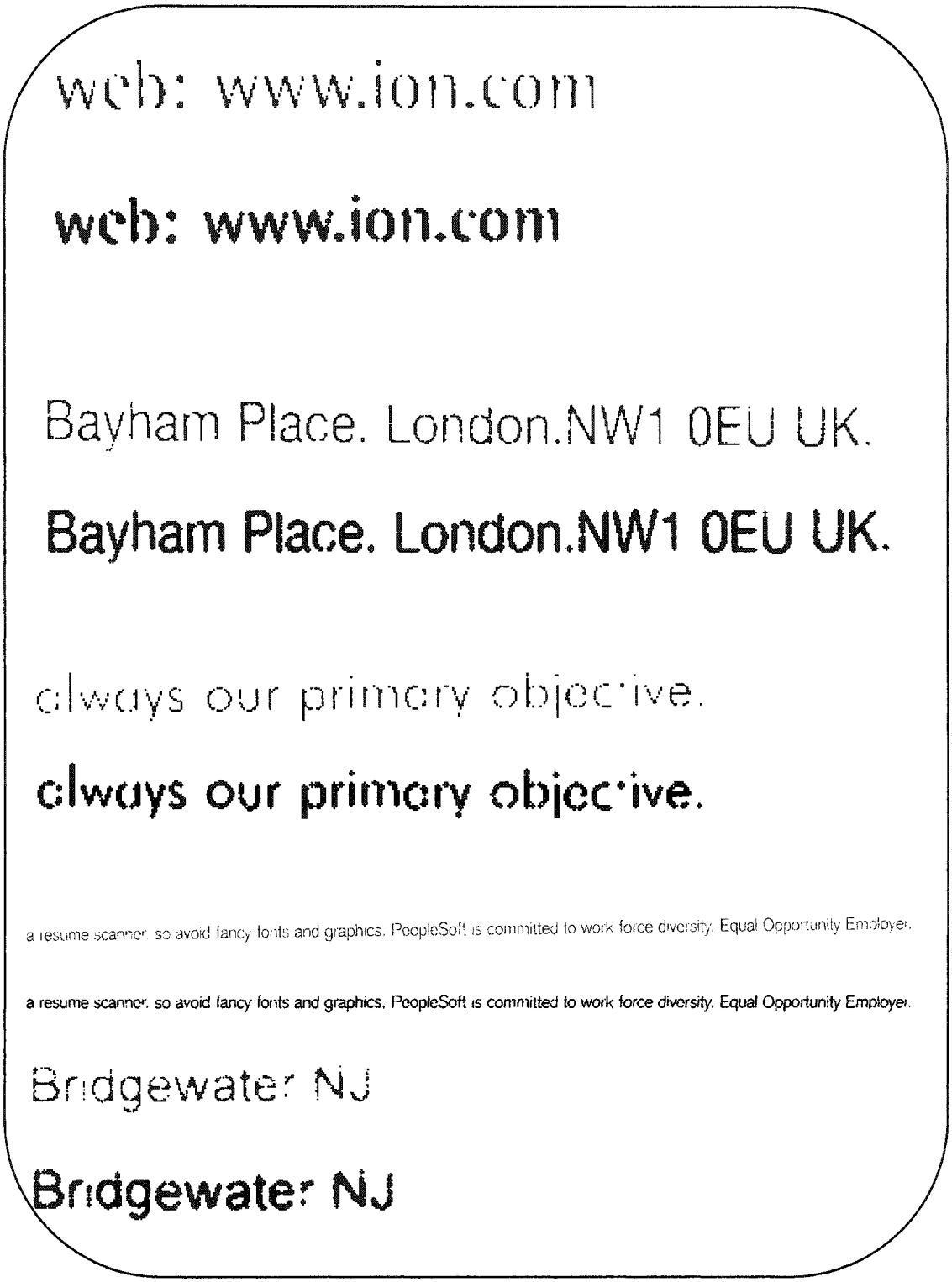


Figure 30: Sample images before and after applying a filter

Figure 31: Sample images before and after applying a filter

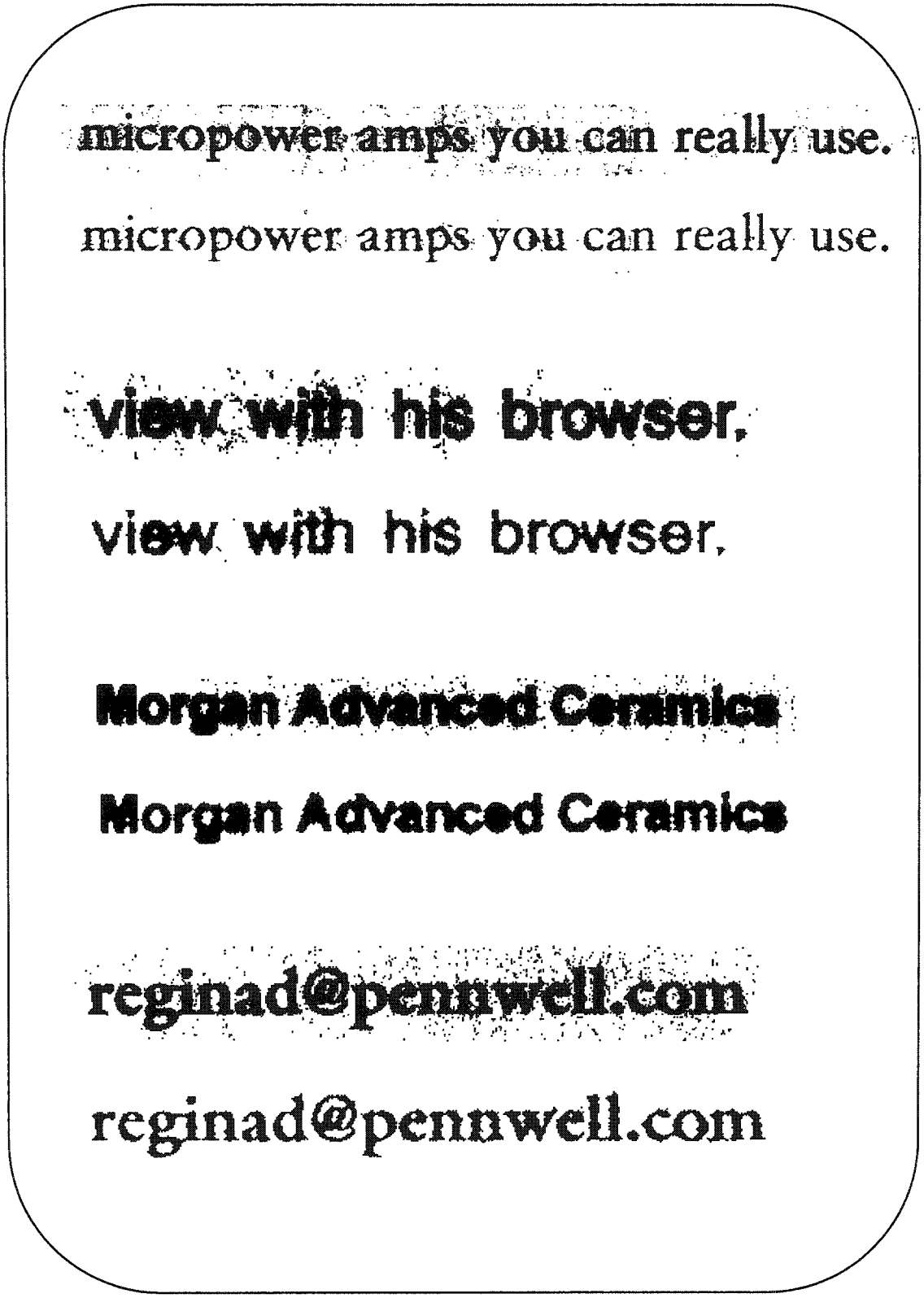


Figure 32: Sample images before and after applying a filter

Figure 33: Sample images before and after applying a filter

Chapter 9. Conclusion

We have presented a method for automatic filter selection using image quality assessment. Five quality measures are calculated in order to estimate the quality of the image, which are stroke thickness factor, touching character factor, small speckle factor, broken character factor and white speckle factor. The definition of the quality measures are based on the types of degradation observed in the database, such as broken characters, touching characters, and salt-and-pepper noise.

The results obtained with the quality measures are used to automatically select the most suitable filter for an image. There are four filters available in the system, and all of them use mathematical morphology operations. The filters are: Dilation with Cross, Erosion with Cross, Closing with Vertical Line, and Opening with Horizontal Line.

The automatic filter selection is a two-stage process that uses a set of rules to decide which filter should be applied to an image. The first stage of decision defines the type of operation that will be used. There are two possibilities, either erosion (or opening), or

dilation (or closing). The second stage of decision defines the structuring element, which can be cross, vertical line or horizontal line.

We consider the method successful given the significant increase in the recognition rate of an OCR engine in the test set from 73.24% using no filter at all to 93.09% after the automatic selection of a filter.

Although we chose to use only morphological filters, other types of filter that do not apply mathematical morphology can also be included. The choice of filters to be implemented should be based on two conditions. First, the filter should be suitable for a considerable amount of documents in the database. In this case, suitable means that the filter should improve the quality of the image, which can be measured by comparing recognition rates before and after applying the filter. Second, there must be a correlation between the values of the quality factors and the images that require a specific filter, otherwise it is not possible to automatically select the filter.

Other images can also be used with the system, although the set of rules used for the automatic filter selection may have to be adjusted in order to obtain similar results.

The system developed can be used as the preprocessing module of a recognition system.

9.1. Contributions

In our research for similar works, we did not find any research that applies image quality assessment to poor quality printed document images with the objective of automatically selecting a specific filter. As presented in chapter two, the work by Cannon et al [7] is similar to ours because it also deals with different types of image degradations in the same database, such as touching characters, broken characters and salt-and-pepper noise. The solution found by them was also to get information about the quality of the image prior to applying any restoration algorithm. However, an important characteristic of their database is that all images are typewritten, so the fonts have fixed widths, which is an advantage, according to the authors.

Still comparing the two works, our database does not only contain exclusively printed documents but also a wide variety of font sizes, types and styles. This represents a greater challenge because our documents have proportional spacing and most printed characters do not have a fixed width, which makes the problem more complicated. There are other differences between the two works. First, the types of filters implemented, we use only morphological filters and [7] uses other types of filters. Second, our database contains 736 images, while there are only 134 in the referred work. Third, the method used to select the best filter for each image. We use a set of rules in a two-stage decision process, while [7] uses a linear classifier. Fourth, the quality measures used in both works, although similar, are calculated in different ways. Regarding the methodology used in both works, we tried three different OCR

engines to demonstrate the effectiveness or our method, compared to only that was used in the referred research.

A paper describing this research, the system and the experimental results was accepted for publication and oral presentation at the International Conference on Document Analysis and Recognition (ICDAR) 2003 that will take place on 3-6 August in Edinburgh, Scotland. The acceptance of our work in this conference indicates its contribution, since this is a well-known and respected event in the Document Analysis and Recognition field.

9.2. Future Work

Despite the good results achieved by the system, it is not perfect, and therefore there is always room for improvement and implementation of new ideas, which will be discussed in this section.

The system was developed and tested using 736 images. As a consequence, the set of rules defined for the automatic filter selection process is too dependent on the database used, especially in the second stage of decision. It would be interesting to use a bigger database, both for training and testing. This would probably make the rules for selecting the filter even more generic and robust, and the system should perform well with other images other than those in the database.

In this current version, the system assumes that any input image requires some type of restoration. It was developed in this way because almost all images in the database really have some type of degradation. However, good quality images could also be part of the database. This means that the option of not applying any filter at all to an image should be available in the system, besides the option of applying a filter. This also makes the system more generic.

The two-stage decision process for filter selection could be reduced to a single step. At first, this could be done in two ways. If the same filters were used, more quality measures would have to be computed in order to provide enough information to make the decision in only one stage. In this case, it would be necessary to investigate which other quality measures could be useful. Another alternative is to make available in the system other types of filter that do not use mathematical morphology.

Independently of trying to change the automatic filter selection process, new filters could be tried, especially if new images are added to the database.

References

[1] J. Ha, R. M. Haralick, I.T. Phillips "Recursive X-Y Cut using Bounding Boxes of Connected Components," *Proceedings of the Third International Conference on Document Analysis and Recognition*, Montreal, Canada, 1995, pp. 1119-1122.

[2] L. O'Gorman, R. Kasturi, *Document Image Analysis*, IEEE Computer Society Press, Los Alamitos, CA, 1995.

[3] D. Drivas, A. Amin, "Page Segmentation and Classification Utilizing Bottom-Up Approach," *Proceedings of the Third International Conference on Document Analysis and Recognition*, Montreal, Canada, 1995, pp. 610-614.

[4] M. Coté, E. Lecolinet, M. Cheriet, C. Suen, "Automatic Reading of Cursive Scripts Using a Reading Model and Perceptual Concepts – the PERCEPTO System," *International Journal on Document Analysis and Recognition*, Volume 1, Number 1, 1998, pp. 3-17.

[5] A. E. Yacoubi, "Modélisation Markovienne de l'Écriture Manuscrite Application à la Reconnaissance des Adresses Postales," PhD Thesis, Université de Rennes 1, France, September 1996.

[6] M. Schußler, and H. Niemann, "A HMM-based System for Recognition of Handwritten Address Words," *Proceedings of the Sixth International Workshop on Frontiers in Handwriting Recognition*, Tajeon, Korea, 1998, pp. 505-514.

[7] M. Cannon, J. Hochberg, and P. Kelly, "Quality Assessment and Restoration of Typewritten Document Images," *International Journal on Document Analysis and Recognition*, Volume 2, Number 2, 1999, pp. 45-52.

[8] L. Blando, J. Kanai, and T. Nartker "Prediction of OCR Accuracy Using Simple Image Features," *Proceedings of the Third International Conference on Document Analysis and Recognition*, Montreal, Canada, 1995, pp. 319-322.

[9] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press Inc., London, 1982.

[10] X. Ye, M. Cheriet, and C. Y. Suen, "A Generic Method of Cleaning and Enhancing handwritten Data from Business Forms," *International Journal on Document Analysis and Recognition*, Volume 4, Number 2, 2001, pp. 84-96.

[11] J. Facon, *Morfologia Matemática: teoria e exemplos*, Editora Universitária Champagnat da PUC, 1996.

[12] Y. Xu, G. Nagy, "Prototype Extraction and Adaptive OCR," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 21, Number 12, 1999, pp. 1280-1296.

[13] M. Cannon, P. Kelly, S. Iyengar, N. Brener, "An automated system for numerically rating document image quality," *Proceedings of the Symposium on Document Image Understanding Technology*, Annapolis, Maryland, 1997, pp.161-167.

[14] J. Wang, H. Yan, "Mending broken handwriting with a macrostructure analysis method to improve recognition," *Pattern Recognition Letters*, 20, 1999, pp. 855-864.

[15] H. Baird, "Document Image Defect Models," from *Structured Document Image Analysis*, H. S. Baird, H. Bunke, and K. Yamamoto, eds., 1992, pp. 546-556.

[16] H. Baird, "Document Image Quality: Making Fine Discriminations," *Proceedings of the Fifth International Conference on Document Analysis and Recognition,* Bangalore, India, 1999, pp. 459-462.

[17] M. Cannon, J. Hochberg. P. Kelly, "QUARC: A Remarkably Effective Method for Increasing the OCR Accuracy of Degraded Typewritten Documents," *Proceedings of the Symposium on Document Image Understanding Technology*, Annapolis, Maryland, 1999, pp. 154-158.

[18] A. Coates, H. Baird, R. Fateman, "Pessimal Print: A Reverse Turing Test," *Proceedings of the Sixth International Conference on Document Analysis and Recognition,* Seattle, WA, 2001, pp. 1154-1158.

[19] S. Srihari, G. Zack, "Document Image Analysis," *Proceedings of the Eighth International Conference on Pattern Recognition*, Paris, France, 1986, pp. 434-436.

[20] R. Casey, G. Nagy, "Document Analysis – A Broader View," *Proceedings of the First International Conference on Document Analysis and Recognition*, Saint-Malo, France, 1991, pp.839-849.

[21] N. Billawala, P. Hart, M. Peairs, "Image Continuation," *Proceedings of the Second International Conference on Document Analysis and Recognition*, Tsukuba Science City, Japan, 1993, pp. 53-57.

[22] J. Hobby, H. Baird, "Degraded Character Image Restoration," *Proceedings of the Fifth Annual Symposium on Document Analysis and Image Retrieval*, Las Vegas, Nevada, 1996.

[23] P. Stubberud, J. Kanai, V. Kalluri, "Adaptive Image Restoration of Text Images That Contain Touching or Broken Characters," *Proceedings of the Third International Conference on Document Analysis and Recognition*, Montreal, Canada, 1995, pp. 778-781.

[24] R. Cattoni, T. Coianiz, S. Messelodi, C. Modema, "Geometric Layout Analysis Techniques for Document Image Understanding: a Review," Technical Report 9703-09, ITC-IRST, Trento, Italy, 1998.

[25] T. Ha, H. Bunke, "Image Processing Methods for Document Image analysis," from *Handbook of Character Recognition and Document Image Analysis*, H. Bunke, P. S. P. Wang, eds., 1997, pp. 1-47.