Form processing with the Hough transform

De Chen
University of Nevada, Las Vegas
FORM PROCESSING WITH THE HOUGH TRANSFORM

by

De Chen

Master of Science in Physics
East China Normal University
1990

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Electrical and Computer Engineering
Department of Electrical and Computer Engineering
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
December 2003
INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.
To

my wife, Ying Yang

my son, Alex Young Chen
The Thesis prepared by

De Chen

Entitled

"Form Processing With the Hough Transform"

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

Examination Committee Chair

Dean of the Graduate College

Examination Committee Member

Examination Committee Member

Graduate College Faculty Representative
ABSTRACT

Form Processing with the Hough Transform

by

De Chen

Dr. Emma Regentova, Examination Committee Chair
Assistant Professor of Electrical and Computer Engineering
University of Nevada, Las Vegas

A form document processing system based on the Hough transform (HT) is developed. It performs form identification and form registration. For form identification, HT is applied off-line to master forms to calculate form features and build-up the feature database, and it is performed on-line for the input (scanned) forms to extract features to identify the form type based on feature matching. The derived features are rotation, translation and scale invariant. The proposed form description is compact, thereby allows for fast identification. The registration is feature/knowledge based. Two methods for control points detection are discussed; one implements template matching for finding frame corners. The second approach is based on detection of line crossings via the analysis of the parameter space of the HT. Detected control points are used to calculate parameters of geometrical transform and perform coordinates translation. Linear conformal and projective transforms are tested. The system is featured by fast and reliable type identification, and the moderate preprocessing time, which is attained by proper design of the Hough space.
# TABLE OF CONTENTS

ABSTRACT .................................................................................................................... iii

LIST OF FIGURES ....................................................................................................... vii

LIST OF TABLES .......................................................................................................... ix

ACKNOWLEDGMENTS ................................................................................................ x

CHAPTER 1 INTRODUCTION ......................................................................................... 1

CHAPTER 2 RESEARCH ESSENTIALS ........................................................................... 5

CHAPTER 3 FORM PREPROCESSING ........................................................................... 11

3.1 Hough transform for line detection ............................................................ 11

3.2 Text and form dropout ............................................................................... 15

3.2.1 Text dropout by thresholding the accumulator array ......................... 15

3.2.2 Inverse Hough transform .................................................................... 16

3.2.3 Adaptive thresholding ........................................................................ 18

CHAPTER 4 FORM TYPE IDENTIFICATION ............................................................. 21

4.1 Form matching by the global descriptor ..................................................... 21

4.2 Form matching by the physical descriptors ................................................ 26

CHAPTER 5 FORM CO-REGISTRATION ................................................................. 30

5.1 Image registration tasks ............................................................................. 30

5.2 Linear conformal transformation for form registration ......................... 33

5.3 Control points extraction ........................................................................... 35

5.3.1 Template based corner detection ..................................................... 35

5.3.2 Corners as line crossings ..................................................................... 37

5.3.2.1 Algorithm for extracting line crossings ......................................... 37

5.3.2.2 Corner points extraction .............................................................. 39

5.4 Form registration in examples ................................................................. 40

CHAPTER 6 EXPERIMENTS AND DISCUSSION ...................................................... 42

6.1 Effectiveness of the distance metric ........................................................... 42

6.2 Limitation of translation invariance ........................................................... 43

6.3 Feature and knowledge based registration techniques ......................... 44

6.4 Limitations of Hough transform for form analysis .................................. 45
6.5 Image registration with the projective transform ..................................... 50
6.6 Time Analysis ........................................................................................ 51
6.7 Possible improvements and future work ............................................... 52
   6.7.1 Secondary Hough transform ....................................................... 52
   6.7.2 Speeding up the Hough transform ............................................... 52

CHAPTER 7 CONCLUSION ......................................................................... 54

REFERENCES ......................................................................................... 57

VITA .......................................................................................................... 63
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Some sample forms of different resolution: a) Example of a form of size 441x543; b) Example of an input form of size 657x411; c) Example of a template form of size 614x391; d) Example of a form of size 615x709</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>Preprocessing of template and incoming forms</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Processing of input forms for form type identification and registration</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>a) Parametric line; b) image pixels in the parametric space</td>
<td>13</td>
</tr>
<tr>
<td>3.2</td>
<td>A sample form with printed and filled texts</td>
<td>27</td>
</tr>
<tr>
<td>3.3</td>
<td>a) Hough transform of the form in Figure 3.2.; (b) thresholded accumulator array of (a)</td>
<td>32</td>
</tr>
<tr>
<td>3.4</td>
<td>Result of inverse Hough transform for the form in Figure 3.2</td>
<td>34</td>
</tr>
<tr>
<td>3.5</td>
<td>Text dropout (a) by nonadaptive thresholding; (b) by adaptive thresholding</td>
<td>34</td>
</tr>
<tr>
<td>3.6</td>
<td>Text remaining after form dropout</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>(a) Master form; b) Lines remaining after text drop</td>
<td>39</td>
</tr>
<tr>
<td>4.2</td>
<td>Line counts for the form in Figure 4.1b</td>
<td>45</td>
</tr>
<tr>
<td>4.3</td>
<td>(a) Input form; (b) Lines remaining after text drop</td>
<td>46</td>
</tr>
<tr>
<td>4.4</td>
<td>Feature vector extracted for the form in Figure 4.3 a)</td>
<td>47</td>
</tr>
<tr>
<td>4.5</td>
<td>Example of two globally similar forms</td>
<td>48</td>
</tr>
<tr>
<td>4.6</td>
<td>Line counts of the forms in Figure 4.5</td>
<td>48</td>
</tr>
<tr>
<td>4.7</td>
<td>Two forms of the same line count</td>
<td>54</td>
</tr>
<tr>
<td>4.8</td>
<td>Examples: a) a template form; b) an input form</td>
<td>55</td>
</tr>
<tr>
<td>4.9</td>
<td>Two globally identical forms of different sizes</td>
<td>61</td>
</tr>
<tr>
<td>5.1</td>
<td>Patterns for detecting a) the upper left corner, b) the upper right corner, c) the lower left corner, d) the lower right corner</td>
<td>37</td>
</tr>
<tr>
<td>5.2</td>
<td>Line crossings detected for forms in Figure 4.8</td>
<td>39</td>
</tr>
<tr>
<td>5.3</td>
<td>Registration of the form in Figure 4.8 b)</td>
<td>45</td>
</tr>
<tr>
<td>5.4</td>
<td>Automated registration (a) Part of the unregistered filled-in form; (b) Part of the registered filled-in form; (c) Part of the blank template form</td>
<td>46</td>
</tr>
<tr>
<td>6.1</td>
<td>Translation of image with center of image moved out of a box</td>
<td>47</td>
</tr>
<tr>
<td>6.2</td>
<td>a) Example form; b) Processed version; c) Line counts (L)</td>
<td>48</td>
</tr>
<tr>
<td>6.3</td>
<td>a) A sample form; b) processed version; c) line count (L)</td>
<td>48</td>
</tr>
<tr>
<td>6.4</td>
<td>a) A sample form; b) processed version; c) line counts</td>
<td>54</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

First, I deeply thank Dr. Emma Regentova for encouraging me to pursue MS in EE, for her advising my thesis work and all the help she provided during my study years. I am grateful to Drs. Kazem Taghva, Shahram Latifi and Henry Selvaravij for serving as committee members. I am also indebted to Dr. Tom Nartker and Kazem Taghva for funding and suggestions on the project. I would like to thank all the faculty and the staff of the Department of Electrical and Computer Engineering for their support, patience and encouragement.
CHAPTER 1

INTRODUCTION

Nowadays, in spite of rapid advances of multimedia technology, scanning of documents is still performed, because of wide popularity of paper forms. Automated form processing is advantageous, because even a medium sized company can produce thousands of forms on a daily basis[1]. This number is doubled or even tripled for big governmental organizations. Survey forms, medical or employee data sheets, admission forms or other document, filled in manually or typed-in are scanned and stored for further processing.

Scanned forms are usually skewed, rotated and translated versions of the master forms, so preprocessing is always necessary before their use in the automated processing pipeline. Typically, form processing includes form modeling, archiving, matching, registration, editing, revising and data extraction.

Image based form analysis offers advantages of on-line processing. It is of great importance to develop an automatic form processing system that integrates form description, archiving, retrieval, or identification based on computer vision technology.

In this work, we introduce an automatic form processing system, in which the master forms are first preprocessed for obtaining form description and storing it in the form database. Master forms are mostly created by computer aided design tools and are considered of high quality and perfectly aligned. Input forms are scanned, hence prone to
geometrical distortions. To perform many of the above mentioned form processing operations, input forms are to be associated with their templates and registered, i.e., calibrated.

Distortion caused by the scanning process can make an input form irrelevant to its master form in the database. An accurate registration is needed to perform subsequent automated processing procedures using the same software as for the master forms.

Form is a special type of document. We adopt the image based approach wherein a form is a binary image (binarized or scanned in a binary mode). Also, form is a framed document containing a number of long horizontal and vertical lines. In this work, we use alternatively words “input”, “incoming” or “scanned” form for forms to be processed, and “master”, “template”, “reference” or “base” form for any form in the database.

The first task is to associate the input form with its template. This is performed by feature-based matching technique. The system retrieves forms from the database. Indexing scheme is always used for image retrieval to avoid direct matching. Among indexing approaches, key words and content-based retrieval are employed. For form retrieval, key words approach is impractical due to the difficulties in recognition of small characters of form numbers.

In context-based image retrieval, global content of the form image is acquired through feature vector extraction. The feature vector is to be compact to serve as an index. Line crossings, corners and lines are features commonly used for representing form layout. Among them, lines are the most commonly used features for describing physical layout, or for constructing high level logical descriptions [2,3,4,5]. Because of their low sensitivity to noise, lines are detected more reliably; hence they provide an accurate input
to logical description. There are extensive studies on obtaining form descriptive features, and a bold body of research carried out in this area [4,6,7,8,9,10].

Since majority of methods for form description/identification is based on line processing approach, the performance of line detection is crucial for attaining an accurate form representation. We use the Hough transform for line detection. We find it particularly suitable for form analysis, because it is known for detecting broken lines and also performs well even there is a moderate rotation. The extracted features comprise lines count for angle distribution, normalized line lengths and distances to the origin calculated for major form lines.

Form type identification is carried out by calculating distances between feature vectors of the input form and those stored in the database. Either zero or a minimum distance is allowed to indicate matching.

There is a variety of methods designed for image registration. One of the commonly used is the cross-correlation, which is the basic statistical approach to registration [11,12]. The optimal alignment of the input form with its template is obtained by calculating the transformation with the largest cross-correlation value. Other known registration methods are feature-based, area-based, knowledge-based, and Fourier-based [13,14]. In knowledge/feature based image registration techniques [15,16], several control points are set/selected from two images to be co-registered. The correspondence of the control points between images is established by similarity measurement. These techniques are more suitable for gray level or color images, rather than for binary forms. In Fourier based registration, all coefficients related to rotation, translation and scale can
be derived from the phase factors of the transformation. But Fourier approach requires large amount of memory.

In this work, we adopt feature-based registration to find parameters of the linear conformal transform and recover from rotation, scaling and translation. Two techniques are tested. First, we fulfill the template based approach, in which we use a sliding window containing a corner template to locate the coordinates of four corners. In the second technique, Hough transform is used to obtain coordinates of four corners of the form frame.

For performing registration with the linear conformal transform, coordinates of two pairs are sufficient. But to ensure the higher accuracy, we use coordinates of four points. We seek for form corners as a special type of line crossings. They are found based on the analysis of the accumulator array of the Hough transform, its processing and performing the inverse Hough transform. The coordinates are fed to the linear transform equations to find distortion parameters and translate coordinates of any given pixel of the input form.

The organization of the thesis is as follows. In the following section we give main definitions and the overview of the system. Chapter 3 presents the basics of the Hough transform and its application to form preprocessing. Chapter 4 describes the identification of form types. Form registration is demonstrated in Chapter 5. Chapter 6 presents experimental results and discusses possible improvements and the future work. Chapter 7 concludes the thesis.
CHAPTER 2

RESEARCH ESSENTIALS

The form images are obtained via document scanning; hence the images can be acquired in different modes. We use binarized versions of the input form images, assuming that either scanning is done in the binary mode, or the gray images are binarized prior to processing. It gives the privilege of processing of reduced datasets, and modifying the Hough transform for its faster implementation. Among distortions introduced by scanning, we examine only rotation, scaling and translation. All procedures developed in this work are viable for such additional geometrical and image distortions such as skew, noise, missing characters, and incomplete lines. But in all conducted experiments we use images of relatively good image quality and with no skew, as they are delivered by the laboratory scanner and its software. The generalization of the designed procedures and the extension of the registration methods to the skewed images are discussed briefly in the later chapters.

Our further studies are based on the following definitions and assumptions:

1. Forms are framed documents with a number of intersecting horizontal and vertical lines. Text added to the form can be either handwritten or typed-in.

2. Form images are bi-level.

3. Lines formed by filled-in text are shorter than that formed by frame lines.
4. Input forms can be scaled, slightly shifted and moderately rotated instances of the master forms.

Definition 1 is commonly used in the research on automated form processing. Assumption 3 is done for the reliable drop-out of the filled-in text and calculating form features inherent to empty forms only. The last assumption holds true for a great statistics of scanned forms. Additionally, we consider a moderate rotation of up to $5^\circ$, which is the maximum encountered under normal scanning conditions and to ensure the proper functioning of the form feature extraction and registration procedures. Some of the forms used in this work are presented in Figure 2.1 a) through d).

The procedures to be discussed here include form preprocessing, feature extraction, matching and registration. A generalized view of the system is presented in Figure 2.2. The Hough transform is applied once to the template forms off-line, and to any input form on-line. The parameter space of transform is processed (thresholded) for calculating form features and reliable detection of control points required for determining parameters of the geometrical transformation. Coordinates of these points for template forms along with the feature vectors are stored in the database (DB) of templates. For incoming forms, further steps comprise feature vector matching, detecting coordinates of control points, finding their correspondence to those recorded in the DB, and calculating the parameters of geometrical transform (see, Figure 2.3).

We use MATLAB environment for experimenting and displaying the results.
a) Example of a form of size 441x543
b) Example of an input form of size 657×411

c) Example of a template form of size 614×391
d) Example of a form of size 615x709

Figure 2.1 Some sample forms of different resolution.
Figure 2.2 Preprocessing of templates and incoming forms

Figure 2.3 Processing of input forms for form type identification and registration
CHAPTER 3

FORM PREPROCESSING

In this chapter, we discuss form analysis with the Hough transform. Form analysis concerns with the extraction of the form structure via detecting form lines in the raster image. That includes transforming raster images to the Hough space, thresholding for text/form dropout, and an inverse transform such that the major lines in the form images are preserved. A brief review of the applications of the Hough transform-based techniques in computer vision is given first, and then the description of line detection with Hough transform is provided.

3.1 Hough Transform for Line Detection

The Hough transform (HT), which was first introduced by Hough in 1962 for detecting the tracks of high-energy particles, is a technique widely used to detect shapes in images[17]. Hough transform is effective for detecting regular curves such as lines, circles, ellipses, and other shapes, because it specifies the desired features in a parametric form. But in the case when a simple analytic description of feature(s) is not possible, or there is no simple analytical form for shape description, such as for irregular shapes, a generalized Hough transform(GHT) is to be employed[18]. The main problem with GHT is its computational complexity and the memory requirements. The former is the major limitation of GHT for real-time, on-line applications. To overcome this
drawback, the randomized generalized Hough transform (RGHT) was introduced [18]. We use the Hough transform for line detection in the form images.

Hough transform maps a point \((x_i,y_i)\) in the image space into a sinusoid in the Hough space given by

\[
\rho = x \cos \theta + y \sin \theta \quad (3.1)
\]

where \(\rho\) is the norm from the origin to this point, and \(\theta\) is the orientation angle of \(\rho\) with respect to the X-axis. Pixels belonging to the collinear lines with coordinates \((x_i,y_i)\) in the image space form the butterfly curves which intersect at \((\rho,\theta)\) in the Hough space as it is shown in Figure 3.1. This intersection indicates a line in the image space with its position and orientation specified by \(\rho\) and \(\theta\).

The Hough transform is implemented by quantizing the parameter space into finite intervals, so called accumulator bins which form the accumulator array. Each image pixel at \((x,y)\) is transformed into a discretized \((\rho,\theta)\) curve, and the accumulator bins along this curve are incremented. Each pixel in the image will vote to a specific bin at \((\rho,\theta)\) in the accumulator array. See Figure 3.1 a) and b). Pixels belonging to the same line will vote to the same bin, resulting in a peak in the accumulator array. The latter indicates a straight line of a certain length.

In the implementation of the Hough transform, the coordinate system of the image space and its corresponding parameter space are set as follows. The center of the raster image is taken as an origin of the coordinate system. For a \(M \times N\) image, the Hough space \((\rho,\theta)\) is constructed with \(\theta\) spanning from \(-5^\circ\) to \(175^\circ\), and an increment of \(1^\circ\) in \(\theta\).
Figure 3.1 a) Parametric line; b) image pixels in the parametric space

is spanned from $-\rho_{\text{max}}$ to $\rho_{\text{max}}$, where $\rho_{\text{max}} = \sqrt{M^2+N^2}$. The minimum increment in $\rho$ is also taken to be 1, and each pixel in the raster image with coordinates $(x, y)$ is mapped into a parametric curve according to Equation (3.1). By the above construction, any $\rho$ value corresponding to the image pixel will be guaranteed to fall into a bin in the accumulator array, and form rotation in the range $\pm 5^\circ$ does not affect further processing.

By the above construction, the Hough space is a plane of $2\rho_{\text{max}} \times \theta_{\text{max}}$ bins each of size $\Delta \theta \cdot \Delta \rho = 1$. Since $\theta$ is taken from $-5^\circ$ to $175^\circ$, a bin at $(0^\circ, -\rho)$ represents a line at $180^\circ$ in the image space. Thus, we can represent lines distributed in $0^\circ - 360^\circ$ with the span of $180^\circ$ in the Hough space. This is due to the fact that each butterfly curve in the Hough space is centrally symmetric. Any point in the lower half of the Hough space corresponds to a negative $\rho$, which means that $\theta$ value of the related line is $\theta+180^\circ$. Making use of this symmetry, we halve the dimension of the Hough space, hence save the computational time and the memory requirements.

Figure 3.3 a gives the image of the parameter space obtained by the Hough transform of the form presented in Figure 3.2.
Figure 3.2 A Sample form with printed and filled text.

Figure 3.3 (a) Hough transform of the form in Figure 3.2.;
(b) thresholded accumulator array of (a)
3.2 Text and form dropout

In this section, we discuss line extraction, that is text or alternatively form dropout by processing the accumulator array. Since only the form layout is of interest in our analysis, we need to discard text data reliably, retaining only lines. Also, if we need alternatively to preserve the text, the procedure can be inverted. We demonstrate this by examples in the later parts of this chapter. The accumulator array can be used to extract lines, and based on that - other features for the form description. We are interested in extracting as many lines as possible.

In a filled form, text is enclosed in boxes formed by form lines. Thus, the form lines are always longer than those formed by text. This means that in the Hough space, a bin value corresponding to the relatively longer horizontal/vertical form line has a higher magnitude than that formed by text lines in the image space. Each pixel belonging to the same line in the raster image falls into the same accumulator bin of a size of $\Delta \rho \Delta \theta$ at a specific location defined by $(\rho, \theta)$.

3.2.1 Text dropout by thresholding the accumulator array

The value of the accumulator bin reflects the line length, i.e., the longer is the line the higher is the value of its corresponding bin in the accumulator array. Thus, a local maximum in the parameter space indicates a line in the raster image. Both form lines and text pixels are detectable by the Hough transformation. But, the difference is in that the number of votes for text lines is smaller than that for form lines. A proper threshold allows for removing lines formed by the text. The threshold is performed as follows.
\[ A'(i,j) = A(i,j), \quad \text{if } A(i,j) \geq T \]
\[ A'(i,j) = 0 \quad , \quad \text{otherwise.} \quad (3.2) \]

Here, \( A \) and \( A' \) are the local maxima of the accumulator bin and its thresholded value, respectively, and \( T \) is a threshold. \( T \) is set depending on the maximum encountered in the accumulator array, i.e., \( T = k \cdot A_{\text{max}} \). If \( k \) is greater than for example 0.7, then only major lines retain. In case of smaller \( k \), for example 0.3, short lines are also captured, including those formed by the filled-in text. An optimal \( k \) can be found experimentally. In our study it is selected to be about 0.7. Figure 3.3.b shows the thresholded accumulator array of one given in 3.3a. Each local maxima in this array represent a major form line.

### 3.2.2 Inverse Hough transform

In a thresholded accumulator array, all major lines are indicated by the local maxima in the Hough space. The information other than on the major lines is discarded by thresholding procedure. That means the text data are dropped in the parameter space. To restore the image space just for observing of the processing effect, we extract all local maxima and put them in the list of \( \theta \) and \( \rho \). This list is used as a look-up table for the major lines in the inverse Hough transform.

In the course of inverse Hough transform, we test a pixel \((x_i, y_i)\) of the image of size \( m \times n \), by calculating the corresponding \( \rho \) for each \( \theta \) value from the local maxima list. If the calculated \( \rho \) has a match with a specific \( \theta \), this pixel belonging to the form line. We restore the pixel value at \((x_i, y_i)\) as zero in a blank image of the same size, i.e. \( m \times n \). The pseudo code for this inverse Hough transform of form image \( I(x_i, y_i) \) of size \( m \times n \) is given below:
for i=1 to m
    for j=1 to n
        y(i, j) = 1  //generate a blank image of the same size as the original form image
    endfor
endfor

//Test each form pixel in the form image x(x_i, y_i)
for i=1 to m
    for j=1 to n
        if x(i, j) = 0  //it's a form pixel
            ρ = x cos θ + y sin θ
            for l=1 to k  //k is the length of the local maxima list
                if ρ = ρ_l
                    y(i, j) = 0  //restored one form pixel
                endif
            endfor
        endif
    endfor
endfor
print y
3.2.3 Adaptive thresholding

According to the above discussion, a bin value in the accumulator array is proportional to the length of the detected line. The forms to be processed are either flat or vertical rectangles. That means $A_{\text{max}}$ values for horizontal and vertical lines may differ. Vertical lines of a flat form can be dropped along with the text by thresholding procedure. (See, for example, Figure 3.5 a). This happens because of a single threshold applied. To avoid occasional drop of either horizontal or vertical lines, the thresholding is to be performed adaptively. Different thresholds, i.e. $kA_{\text{max}H}$ and $kA_{\text{max}V}$ are set for horizontal and vertical lines, respectively.

From the Hough space, one can observe that positions of bins corresponding to the vertical lines are located by the left side of the Hough space, while the bins corresponding to the horizontal lines reside at the middle. (See, Figure 3.3 b)).

Figure 3.4 Result of inverse Hough transform for the for in Figure 3.2.
Figure 3.5 Text dropout: (a) by nonadaptive thresholding; (b) by adaptive thresholding.

Figure 3.6. Text remaining after form dropout

The threshold $kA_{\text{max}V}$ is used for the bins residing around $\theta=0^\circ$, and the threshold $kA_{\text{max}H}$ is used for bins residing near $\theta=90^\circ$. The results of such adaptive thresholding for removing the text are demonstrated in Figure 3.5.
Form dropout and text dropout are complement operations. Form dropout is obtained by logic addition of the complement of the text-free form and the original one. Figure 3.6 exhibits the result of dropping the form given in Figure 3.2.
CHAPTER 4

FORM TYPE IDENTIFICATION

We propose a feature-based similarity detection by comparing features of the template and input forms. First, the global logical description is obtained. Then, physical features are calculated for matching with those forms, which exhibited similarity by the global descriptor. The global logical descriptor is the lines count for angle distribution that gives the information about orientation of form lines in the image space. The physical descriptors comprise vectors of normalized values of (a) line positions with respect to the origin and (b) line lengths. To evaluate the similarity, we defined a distance metric. The overall identification is rotation, scaling and translation invariant. The following sections give insight into the feature calculations and the identification procedures.

4.1 Form matching by the global descriptor

As required by on-line processing, the descriptor should be compact. First, we form a global feature vector which is scale and translation invariant by using the lines count in a scope of 360°

To evaluate matching, we compute the distance between the feature vectors of the input form and those in the database. Suppose the feature vector in the database is \( L_T \), and that of the input form is \( L_I \), then the distance between these two vectors is defined as:
where $N=360$, that is, the total number of angle values in the vector. If $D^L = 0$, it would be the match. If the query form is rotated with respect to its template, then $D^L$ will be greater than 0, and it is likely that the same difference would be produced by forms of the other types in the database. That is, this feature is not sufficient for type identification. Furthermore, matching with $D^L = 0$, does not mean definitely that matching is found, because if there is a line miss because of thresholding, different forms can produce the same global description. Additionally, form matching by this method is rotational variant.

We assumed that shapes are preserved by rotation, that is there is no skew, and parallel lines remain parallel. Recall, we have assumed that the maximum allowed rotation angle of $\pm 5^\circ$. Then, we solve the rotational variant problem by cyclic rotation of the query feature vector, that is, rotate the input form from $-5^\circ$ to $5^\circ$. The distance is calculated at each rotation step as

$$D^L_k = \min_{k=-5,-4,...,4} \sum_{i=1}^{N} |L_T(i) - L_d(i)|. \tag{4.2}$$

$D^L_k = 0$ indicates matching at the angle determined by k. This way we ensure the rotation invariant match, and at the same time determine the rotation angle of the input form. Figure 4.1 (b) shows text-free variant of the master form (a) form; The corresponding global logical descriptor is plotted in Figure 4.2.
Given in Figure 4.3.a and b there are occasionally rotated input form and its text-free (thresholded) variant, respectively. The lines count feature is exhibited in Figure 4.4.
For the forms shown in Figures 4.1a and 4.3a, \( D_k^L = 0 \).

The distance between two forms in Figure 4.5, is one. Indeed, there is a one additional horizontal line in the top of the left form. Their lines counts are presented in Figure 4.6. The distance \( D_k^L = 1 \). This confirms that matching using this feature vector is effective at the logical level.
Figure 4.5. Example of two globally similar forms

Figure 4.6 Line counts of the forms in Figure 4.5

For accurately identifying the form type, we need additional features. More features will allow to retrieve the database of the form types with a greater precision.
4.2 Form matching by physical descriptors

It is likely that there can be two or more different forms that produce the same logical descriptions, especially in a large database system. For example, two forms presented in Figure 4.7 produce the same lines count. Therefore, we consider incorporation of additional invariant features.

![Figure 4.7 Two forms of the same line count.](image)

We obtain additionally, line position ($p_i$) and line length ($l_i$) information and store it in a three column list. In Table 1, these features are calculated for two forms shown in Figure 4.8. In Figure 4.8, the input form is of size 411x657; and the template is of 391x614. Column $\theta$ contains orientation of lines with respect to x-axis; column $p$ gives the distance from the corresponding line to the image origin; and column $AC$ lists the thresholded values of the local maxima of the accumulator array, i.e. the information on the line length. The line positions and lengths given in Table 1 are scale and translation variant.
Table 1. Tabulated values of features for the forms in Figure 4.8.

<table>
<thead>
<tr>
<th>$\rho_T$</th>
<th>$\theta_T$</th>
<th>$AC_T$</th>
<th>$\rho_I$</th>
<th>$\theta_I$</th>
<th>$AC_I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>280</td>
<td>0</td>
<td>339</td>
<td>284</td>
<td>3</td>
<td>355</td>
</tr>
<tr>
<td>166</td>
<td>90</td>
<td>567</td>
<td>178</td>
<td>93</td>
<td>598</td>
</tr>
<tr>
<td>107</td>
<td>90</td>
<td>567</td>
<td>116</td>
<td>93</td>
<td>598</td>
</tr>
<tr>
<td>83</td>
<td>90</td>
<td>567</td>
<td>91</td>
<td>93</td>
<td>598</td>
</tr>
<tr>
<td>57</td>
<td>90</td>
<td>567</td>
<td>63</td>
<td>93</td>
<td>461</td>
</tr>
<tr>
<td>32</td>
<td>90</td>
<td>567</td>
<td>37</td>
<td>93</td>
<td>593</td>
</tr>
<tr>
<td>7</td>
<td>90</td>
<td>567</td>
<td>11</td>
<td>93</td>
<td>598</td>
</tr>
<tr>
<td>283</td>
<td>180</td>
<td>339</td>
<td>308</td>
<td>183</td>
<td>356</td>
</tr>
<tr>
<td>80</td>
<td>270</td>
<td>567</td>
<td>79</td>
<td>273</td>
<td>596</td>
</tr>
<tr>
<td>169</td>
<td>270</td>
<td>567</td>
<td>171</td>
<td>273</td>
<td>598</td>
</tr>
</tbody>
</table>

To make these features scale and translation invariant, we normalize $\rho$ by its mean value ($\mu$), and multiply by ten as:

$$\rho_n = 10^* \frac{\rho_i}{\mu}, \quad \text{where} \quad \mu = \frac{1}{N} \sum_{i=1}^{N} \rho_i . \quad (4.3)$$

The same normalization and adjustment procedure is carried out for the line length feature to obtain $l_n$. After normalization, we get a list of normalized values in Table 2.

Figure 4.8 Examples: (a) Template form; (b) Input form
Table 2. Tabulated values of vectors of normalized features for the forms in Figure 4.8.

<table>
<thead>
<tr>
<th>ρ_T</th>
<th>θ_T</th>
<th>AC_T</th>
<th>ρ_I</th>
<th>θ_I</th>
<th>AC_I</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.152</td>
<td>0</td>
<td>6.502</td>
<td>21.226</td>
<td>3</td>
<td>6.634</td>
</tr>
<tr>
<td>13.133</td>
<td>90</td>
<td>10.875</td>
<td>13.303</td>
<td>93</td>
<td>11.175</td>
</tr>
<tr>
<td>8.465</td>
<td>90</td>
<td>10.875</td>
<td>8.670</td>
<td>93</td>
<td>11.175</td>
</tr>
<tr>
<td>6.566</td>
<td>90</td>
<td>10.875</td>
<td>6.801</td>
<td>93</td>
<td>11.175</td>
</tr>
<tr>
<td>4.509</td>
<td>90</td>
<td>10.875</td>
<td>4.709</td>
<td>93</td>
<td>8.615</td>
</tr>
<tr>
<td>2.532</td>
<td>90</td>
<td>10.875</td>
<td>2.765</td>
<td>93</td>
<td>11.082</td>
</tr>
<tr>
<td>0.554</td>
<td>90</td>
<td>10.875</td>
<td>0.822</td>
<td>93</td>
<td>11.175</td>
</tr>
<tr>
<td>22.389</td>
<td>180</td>
<td>6.502</td>
<td>23.019</td>
<td>183</td>
<td>6.653</td>
</tr>
<tr>
<td>6.329</td>
<td>270</td>
<td>10.875</td>
<td>5.904</td>
<td>273</td>
<td>11.138</td>
</tr>
<tr>
<td>13.370</td>
<td>270</td>
<td>10.875</td>
<td>12.780</td>
<td>273</td>
<td>11.175</td>
</tr>
</tbody>
</table>

For invariant features, the distances are calculated as follows:

\[ D^{\rho} = \sum_{i=1}^{N} |\rho_T(i) - \rho_I(i)|^2 \]  \hspace{1cm} (4.4)

\[ D^{AC} = \sum_{i=1}^{N} |AC_T(i) - AC_I(i)|^2 \]  \hspace{1cm} (4.5)

where \( N \) is a total number of lines encountered in the form. And the integrated distance by these two features is defined as

\[ D_{phy} = |D^{\rho}| + |D^{AC}| \]  \hspace{1cm} (4.6)

If \( D_{phy} = 0 \), there is most likely the match. For other cases, a minimum distance indicates that a closest form type is found. It is clear that there should be a maximum allowed distance to prevent erroneous identification.

From the observation of Tables 1 and 2, one can see the form rotation by 3°. The integrated distance between the feature vectors of two different forms in Figure 4.8 is 0.

Scale invariance can be demonstrated by computations carried for two forms shown in Figure 4.9. The scale difference is made purposefully large. The resulting distances between the feature vectors of this two forms are \( D_{k^L} = 0 \) and \( D_{phy} = 0 \).
Figure 4.9. Two globally identical forms of different size
CHAPTER 5

FORM CO-REGISTRATION

In this chapter, we first review the image registration techniques. Then, we give the mathematical definition of the linear conformal transformation for form registration. At the later part of this chapter, we discuss two different approaches for corner detection, and use the detected corners as control points for form registration.

5.1 Image Registration tasks

Image registration is extensively used in computer vision and image processing. Image registration means mapping of coordinates of one image to coordinates of another, which is called a base, or a reference image. The applications of such a task can be found in analysis of medical images, remote sensing, astrophysics and many other areas where the following practical problems are to be solved.

1) Finding changes in images taken at different timelines or under different conditions [19].

2) Combining images taken separately from parts of the same scene [20].

3) Object recognition using model (template) based approaches [21].

4) Recovery of the 3-D characteristics of a scene [22].
In automatic registration, most commonly applied techniques fall mainly within two broad categories: feature based and area based image registration. There is other registration techniques such as knowledge- based [23] and Fourier- based [24] reported elsewhere.

In the feature- based registration, similar features in vicinity of some specific spots are extracted from the base and the input images, and one to one correspondence is expected to be established between the similar spots in both images. In registration terminology, these features are called control features and are referred to as control points. If the type of distortion is known, then with the extracted control points, parameters of geometrical transformation can be derived, which will finally align the input image with the base image.

The common features for registration are moments, curvatures and line segments, that is some vivid distinguishing characteristics of image objects. Also, it is desirable to define those that can be found easily. Various recognition techniques such as the Canny operator [25,26], the Laplacian of Gaussian (LoG) operator [27,28], thresholding [29] and other classification methods [30] have been proposed for edge and boundary features. Features are generally represented by chain code, moment invariants, Fourier descriptors, shape signatures, centroidal profiles, and shape matrices [31] [f23]. Centers of gravity of closed boundaries are usually taken as the control point. Other salient points [32] like corners, line crossings and coordinates of maximum curvature [27] have been employed as control points for precise registration.

In the area-based registration, a small window of a number of pixels of the input image is compared statistically with the window content of the same size of the
base image [32]. The similarity between the two given windows is measured to establish their correspondence. Once, the correspondence is established, the centers of the matched windows are taken as control points.

The important factor for the group of the area-based techniques is the similarity measure. One popular is the cross correlation. There are other metrics such as correlation coefficient and the sequential-similarity measures used for the problem.

In the knowledge-based registration, the information about images or objects is employed. In [33], the authors uses prior knowledge to select appropriate structures using Geographic Information System and then to extract their corresponding features from the sensor data. The knowledge is presented explicitly using semantic nets and rules.

Among frequency domain registration the FFT- based one is another known approach. Fourier transform is used to match images that are translated, rotated and scaled with respect to one another in [34]), for example. The Fourier domain technique for image registration was proposed by Kuglin and Hines [35]. They introduced the phase correlation technique by using some properties of the Fourier transform to align images. De Castro and Morandi developed a method to derive the rotation angle and the shift from the Fourier transform [36]. And the phase correlation technique was later extended by Reddy and Chatterji in 1996 to handle translation, rotation, and scaling. The rotation and scaling information is obtained from the Fourier scaling and rotational properties; and the translational movement is determined by the phase correlation technique. The scenario for the FFT-based technique for translation, rotation, and scale-invariant image registration is presented below [37]:
Let $f_2$ be the displaced version of image $f_1$ by an amount of $(x_0, y_0)$ i.e.

$$f_2(x, y) = f_1(x-x_0, y-y_0).$$  \hspace{1cm} (5.1)

Their corresponding Fourier transforms $F_1$ and $F_2$ will be related by

$$F_2(\xi, \eta) = e^{-j2\pi(x_0 \xi + y_0 \eta)} F_1(\xi, \eta)$$ \hspace{1cm} (5.2)

The cross-power spectrum $R$ of the two images $f_1$, $f_2$ with Fourier transforms $F_1$, and $F_2$ is given by

$$R = \frac{F_1(\xi, \eta) F_2^*(\xi, \eta)}{|F_1(\xi, \eta) F_2(\xi, \eta)|} = e^{j2\pi(x_0 \xi + y_0 \eta)}$$ \hspace{1cm} (5.3)

where $F_2^*(\xi, \eta)$ is the complex conjugate of $F$. The phase of the cross-power spectrum $R$ is equivalent to the phase difference between two images. By taking the inverse Fourier transform of $R$, the impulse function will be obtained, that is, it is approximately zero everywhere except at the displacement $(x_0, y_0)$. Thus, the latter is identified. The same way, other parameters such as rotation and scale can be obtained.

**5.2 Linear conformal transformation for form registration**

As it follows from the previous survey of image registration techniques, if there is no information about the type of distortion, then the transformation equations for mapping are to be obtained based on physical characteristics of image/objects. Generally, in the feature-based registration, if the distortion is known, then the features help with calculating the parameters of the transformation.

Particularly, for the distortions assumed for the form images, the linear conformal transformation is a simplest solution to register images. Given two sets of control points detected from the template and the input images and their correspondence,
the parameters of the linear conformal transformation can be derived from the following equations

\[ u = s \cdot x \cdot \cos(\alpha) + s \cdot y \cdot \sin(\alpha) + T_x \]  \hspace{1cm} (5.4)

\[ v = s \cdot y \cdot \sin(\alpha) + s \cdot x \cdot \cos(\alpha) + T_y \]  \hspace{1cm} (5.5)

where \( s \) is a scaling coefficient, \( \alpha - \) is a rotation parameter, \( T_x \) and \( T_y \) are shift parameters in X and Y directions, and \((s,v)\) and \((x',y')\) are coordinates of control points in the template and the input forms, respectively.

As control points, the intersections of form lines, particularly corner points of form frames can be used. The minimum number of control points needed for linear conformal transform is two. Then, four equations of four unknowns can be resolved to find \( s \cdot \cos(\alpha) \), \( s \cdot \sin(\alpha) \), \( t_x \) and \( t_y \). More points help with increasing the registration accuracy. We use four frame corners for calculations. The linear conformal transform equations in the matrix form can be written as:

\[
[u \ v] = [x \ y \ 1] \begin{bmatrix}
  s \cdot \cos(\theta) & s \cdot \sin(\theta) \\
  s \cdot \sin(\theta) & s \cdot \cos(\theta) \\
  t_x & t_y
\end{bmatrix}
\]  \hspace{1cm} (5.6)

For more than two control points, the matrix equation is as

\[
\begin{bmatrix}
  u_1 \\
  u_2 \\
  \vdots \\
  u_n \\
  v_1 \\
  v_2 \\
  \vdots \\
  v_n
\end{bmatrix} = \begin{bmatrix}
  x_1 & y_1 & 1 & 0 \\
  x_2 & y_2 & 1 & 0 \\
  \vdots & \vdots & \vdots & \vdots \\
  x_n & y_n & 1 & 0 \\
  y_1 - x_1 & 0 & 1 \\
  y_2 - x_2 & 0 & 1 \\
  \vdots & \vdots & \vdots & \vdots \\
  y_n - x_n & 0 & 1
\end{bmatrix} \begin{bmatrix}
  s \cdot \cos(\theta) \\
  s \cdot \sin(\theta) \\
  t_x \\
  t_y
\end{bmatrix}
\]  \hspace{1cm} (5.7)

In a generalized form, it can be represented as

\[ U = X \cdot R \]  \hspace{1cm} (5.8)
So,

\[ R = X^{-1} U \]  \hspace{1cm} (5.9)

where \( R \) is a matrix of transform parameters given by

\[
R = \begin{bmatrix}
    s^* \cos \theta \\
    s^* \sin \theta \\
t_x \\
t_y
\end{bmatrix}
\]

5.3 Control points extraction

Corners in a form are, by nature, the best control points for form registration. As the special points in a form, corners are the points which are easiest to be detected. In a form image, corners are the most widely spread points in a form, which is the best property for precise image registration. In this work, we use the four corners of a form as the control points for form registration.

5.3.1 Template based corner detection

We first used an approach that employs some knowledge of the input and template images to detect the corner points. This knowledge is mostly on the structure of form document, i.e., possible positions of corners. This knowledge is particularly available due to the identification. A sliding window of size 7x7 moves from the upper leftmost corner to the lower rightmost position in the form image. Each pixel is matched against its peer. The sliding window contains the patterns of the four corners as they are expected to appear in the scanned forms (see Figure 5.1).
These four patterns are derived for the ideal case, that is no “salt and pepper noise”, as well as the alignment of the form is perfect.

If a pixel being tested is a corner point, the sum of differences between the pixels in the window and those corresponding to the appropriate pixels in the image is expected to be a minimum.

With the specific placement of the corners, the procedure can be quite effective. The point being searched for registration is the central point in the window for which the match is recognized. The corner detection is carried out off-line for the template forms. The coordinates of the corner points are stored in the database along with the descriptive features needed by form identification procedure. And the procedure is applied on-line for the input forms. The coordinates of the corner points are fed to Equation (5-7) to calculate parameters of the linear conformal transform.
5.3.2 Corners as line crossings

The described above technique works well on noisless images. Corners can be defined as a special type of the line crossing, that is L-type intersection. To pursue as much functionality as possible in the parameter space, we endeavor to find a solution by processing the parameter space. Recall, HT was used as a preprocessing procedure. For form type identification it performs on a line base. Thus, the same framework can be used to detect major line intersections.

5.3.2.1 Algorithm for extracting line crossings

Described below, is the generalized algorithm for finding intersections of lines in the form image.

1) Perform Hough transform of the form image;

2) Preprocess the accumulator array of the Hough transform with a proper threshold, thus removing text part of the form.

3) Find local maxima in the accumulator array, and store them in the list which gives the angle and position distribution \((\rho, \theta)\) of the major lines in the form image;

4) For an image of size \(m \times n\) and the list of line distribution \((\rho, \theta)\) of size \(k\), each pixel on the line in the raster image is tested as follows:

   a. The \(\rho\) value is calculated for each \(\theta\) in the list according to Equation (3.1), that is

   \[
   \rho = x \cos \theta + y \sin \theta,
   \]

   where \((x,y)\) are coordinates of the line pixel, and \(\theta\) is the angle value in the list.
b. Each time when the calculated $p$ has a match for the corresponding $\theta$, the accumulator is incremented by one. Then if the accumulator $A$ is greater than 1, the line pixel $(x, y)$ is considered to be a line crossing. Below, is the pseudo code for step 4.

**Pseudo-code for Step 4**

```pseudo
for i := 1 to m
    for j := 1 to n
        if $x(i,j) = 0$ (to test if the pixel at $(i,j)$ is a line pixel)
            then for $l := 1$ to k
                $\rho = i \cos \theta(l) + j \sin \theta(l)$;
                if $\rho = \rho(l)$
                    $A = A + 1$;
                end if
            end for
        end if
        if $A > 1$
            $y(i,j) = 0$
        end if
    $A = 0$;
end for
```

Here, $x$ denotes the matrix of the form image, and $y$ denotes the matrix of the image of detected line crossings.

Shown in Figure 5.2 are the line crossings detected for the forms in Figure 4.8.
With these points detected in both the template and the input form, any four or more points can be used for solving equation (5.7) and finding the parameters of the linear transform. The last problem is to establish the correspondence between corners of the input form and those of the base form.

5.3.2.2 Corner points extraction

As a matter of fact, more than four control points are not necessarily resulting in more accurate registration since an error caused mainly by rounding accumulates. That leads to degrading the accuracy. Besides, too many control points complicate the task of establishing correspondence between corners from the two forms, and increasing the complexity of calculations. In our system, four corner points are used as the control points for the form registration. They are selected from a bucket of line crossings detected in the form images. The major reason for doing this is that corners are, in fact, scattered.

They have to be scattered by the basic requirement for the control point selection in image registration. Not less important is the fact, that the correspondence between the
control points detected in the input and the template images can be established based mainly on the heuristics.

Corners are easily extracted from the list of coordinates of crossing points. For the point of the smallest value of x, the smallest value of y tells about the upper left corner; while the largest y coordinate value indicates the lower left corner. And for a point of the largest x value, the smallest value of y coordinate manifests about the upper right corner. Accordingly, the largest y value indicates the lowest right corner.

5.4 Form registration in examples

We use the coordinates of the found corner points to calculate the scaling, rotation and translation parameters as it is discussed in Section 5.2. Once these parameters are obtained, the linear conformal transformation is performed for any pixel in the form image to be registered. The registration result for the form in Figure 4.8b is displayed in Figure 5.3.

Figure 5.3 . Registration for the form in Figure 4.8 b)
Figure 5.4. Automated registration. (a) Part of the unregistered filled-in form; (b) Part of the registered filled-in form.

Shown in Figure 5.4 is another example of form registration using the template base approach for corner detection. Both examples show the input form get aligned with the blank base form. The registered input form is ready now for text revising and editing which is the target application of our system.
6.1 Effectiveness of distance metric

For experiments, we have designed various form patterns for testing rotation, scaling and translation invariance. As described in Chapter 4, the rotation invariance is tested by the sample forms in Figure 4.3 with a zero distance \( D_k=0 \), defined in (4.2) between feature vectors of two forms found similar by the global matching procedure. With two sample forms in Figure 4.8, the translation invariance is verified by the zero distance \( D_{Phy}=0 \), defined by (4.6) between the feature vectors of the normalized line positions and the line lengths. The results are tabulated and presented in Tables 1 and 2.

Since the scaling in two forms in Figure 4.8 is not obvious, we, instead, used the designed form pattern shown in Figure 4.9 to test the scaling invariance. With the normalized line positions and line lengths, the distances between feature vectors of two forms are zero \( D_k=0 \), and \( D=0 \), that confirms the scaling invariance.

In the form identification process, the distance we used is the sum of squares of differences of the components of two vectors. It is simple, but reliable measure, used in the database retrieval. For example, we designed the distance to match two forms in Figure 4.5 the result is \( D^1=1 \). What we calculated is the square of sum of the number of lines in the scope of \( 0^\circ \) to \( 360^\circ \). The result \( D^1=1 \) tells that the two forms are different and that one form has one more line. If the forms have this difference being equal 2 lines,
then the distance according to the metric used in our study $D^k=4$, that is much larger. This clearly stresses the difference.

For two other features, i.e., the normalized line distance and the line length, the adjustment (scaling) of a factor of ten is used. For identical forms, the difference between each component of the vector is small; usually less than one, which after being squared, will become a smaller number. In this case, the square will generally make the distance for matched forms be less than one and a floor operation will finally make the distance almost close to zero ($D_{phy}=1$, $|D^0|=0$, $|D^{AC}|=1$. For two forms that do not match, the component differences can easily be greater than one, and when squared, the difference becomes large enough ($D>>1$) strengthening the mismatch between two forms.

6.2 Limitation of translation invariance

We have verified the translation invariance for our identification system. But in practice it is limited. Theoretically, this invariance holds only when the translation factor of the form image is not very large. Otherwise, the center of the image moves out the box which originally confines the center of the image. This is demonstrated in Figure 6.1.

![Figure 6.1](image)

Figure 6.1 Translation of image with center of image moved out of a box
Though, this is a violation of above declared the shift invariance property of the form identification system, in practice, most of the translation caused by the document displacement on the scanner board does not introduce such a big shift. This is true, particularly for A4 format documents, and forms with a moderate number of relatively large boxes. This limitation can be eliminated if we logically describe the forms with line distribution only in horizontal and vertical direction, instead of in the scope of 360°. The price of such an improvement would be the coarser description of the form layout. This is a tradeoff between precision and limitation.

6.3 Feature and knowledge based registration technique

We have applied two different approaches for control point detection, i.e., the template based and the Hough-based. In the template based approach, we judge if a pixel is a corner pixel by the sum of differences between the corresponding points in the 7×7 window of a template image of a corner and the pixels taken from the form image. The zero difference is the indication of the corner point, but a non-zero difference can still correspond to the corner point, because of imperfection of lines caused by scanning and binarization. For corner detection using templates in Figure 5.1, a tolerance value, which is 2 in our case, is to be set. Thus, if the difference is less than or equal the tolerance value, the corresponding pixel is judged as a corner point. If the tolerance value is too high, too many pixels will be recognized as corner pixels. On the other hand, if the tolerance is too low, corners are missed. Therefore the reliability of corner detection by use of this method is questionable.
In the HT-based approach, we extract line crossings as the features to be used for calculating the geometrical transform for further coordinate mapping. In general, forms can contain a large number of crossing points. To use all the detected line crossings as the control points needs first to establish one to one correspondence between points in the template and those in the incoming form. Apparently, the procedure is the time consuming and the error prone operation. To avoid this difficulty, we can limit ourselves to the four points, which are frames corners.

It was discussed in Chapter 5 that the correspondence between frame corners of the template and those in the incoming form is easy to establish using only heuristics and simple rules for checking. Thus, in our studies, we use a combination of the feature-based and the knowledge-based methods.

6.4 Limitations of the Hough transform for form analysis

By performing thresholding of the accumulator array in the preprocessing step, we preserve major lines. Short lines and lines created by the short text lines will be removed (see, Figure 3.4). This result obviously has an adverse effect, though not serious, if the global layout is concerned, or the number of the template forms of a same global layout in the database is limited.

Short lines most likely are discarded after thresholding. Analogously, long lines indicated by relatively long text might retain. In fact, this does not introduce much difficulties. For example, if the long lines are created by the printed text formed by the comment lines, they remain after thresholding in both template and incoming forms.
Thus, they are counted for the feature vectors for both. This is exhibited in Figure 6.2, 6.3 and 6.4.

The major problem though is the loss of short horizontal and vertical lines which are essential for differentiation of globally similar forms. But in practice, there cannot be many forms of exactly same global and physical content based on even few lines remaining after thresholding.

Another limitation of the Hough transform is its time complexity which is of $O(N^3)$, where $N$ is one dimension of image being processed.

Discussion of these two problems is provided in the Subsection 6.6.
Figure 6.2. a) Example form; b) Processed version; c) Line counts (L).
Figure 6.3. a) An example form; b) processed version; c) line counts (L).
Figure 6.4. a) Example form; b) processed version; c) line counts \
6.5. Image registration with the projective transform

It is worth to mention that the form can be recovered from such a distortion as skew using the projective transformation. Four corner points are sufficient to accomplishing registration for skewed, rotated, translated and scaled forms.

The following equation describe projective transform, where \( p_{ij} \) are unknown parameters.

\[
\begin{align*}
x' &= p_{11}x + p_{12}y + p_{13} - p_{31}xx' - p_{32}y x' \\
y' &= p_{21}x + p_{22}y + p_{23} - p_{31}xy' - p_{32}y y'
\end{align*}
\] (6.1)

We can rewrite these system for four pairs of coordinates to resolve unknown parameters as follows

\[
\begin{bmatrix}
x'_1 \\
y'_1 \\
x'_2 \\
y'_2 \\
x'_3 \\
y'_3 \\
x'_4 \\
y'_4
\end{bmatrix} =
\begin{bmatrix}
x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 \\
0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 \\
x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 \\
0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x_2 & -y'_2y_2 \\
x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -x'_3y_3 \\
0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3x_3 & -y'_3y_3 \\
x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -x'_4y_4 \\
0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4x_4 & -y'_4y_4
\end{bmatrix}
\begin{bmatrix}
p_{11} \\
p_{12} \\
p_{13} \\
p_{21} \\
p_{22} \\
p_{23} \\
p_{31} \\
p_{32}
\end{bmatrix}
\]
For that we have to know four corner points in an incoming forms \( \{(x_i', y_i'), i=1,2,3,4\} \) and its master variant \( \{(x_i, y_i), i=1,2,3,4\} \).

6.6 Time analysis

The overall time is an important characteristic of the processing system. On-line processing needs seedy form processing. In Table 3, the last column indicates the total time required for the identification in seconds. These estimates are performed for the documents given in Figure 2.1 which are of different resolutions, thus of different sizes. The Hough transform takes the maximum of the overall time, wherein the part of identification is miserable. The transform, local maxima calculation, thresholding and feature calculation times all are functions of the document size. The feature calculation time is also a function of the number of lines in the form. The time of identification by matching features is a function of a number of template forms in the database.

### Table 3. Processing time for the forms in Figure 2.1

<table>
<thead>
<tr>
<th>Figure 2.1</th>
<th>Dimension</th>
<th>Size</th>
<th>HT time(s)</th>
<th>Local max</th>
<th>Thresholding</th>
<th>Feature calculation</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>d)</td>
<td>615x709</td>
<td>536035</td>
<td>22.235</td>
<td>0.015</td>
<td>0.015</td>
<td>0.031</td>
<td>0</td>
</tr>
<tr>
<td>b)</td>
<td>635x406</td>
<td>257810</td>
<td>9.891</td>
<td>0.031</td>
<td>0.047</td>
<td>0.016</td>
<td>0</td>
</tr>
<tr>
<td>c)</td>
<td>614x391</td>
<td>240074</td>
<td>7.25</td>
<td>0.015</td>
<td>0.016</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>a)</td>
<td>441x543</td>
<td>239463</td>
<td>7.562</td>
<td>0.015</td>
<td>0.015</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

One can conclude that to minimize the processing time, the major effort is to be done to reduce the time of the transform.
6.7 Possible improvement and the future work

6.7.1 Secondary Hough transform

Here, we provide a possible approach to obtain information about short lines dropped after thresholding. As it was discussed, text is extracted by the form dropout. The loss of short lines can be observed in Figure 3.4. Short lines which is longer than those formed by the text can be seized by the secondary Hough transform of the image restored after the first hand processing, such as one in Figure 3.4.

Suppose, we perform HT of the image in Figure 3.6. Some of the short lines can be detected. After thresholding and detecting the local maxima, some detected short lines are recorded, say in the List 2. Recall, we obtain List 1 in the original preprocessing. Then, if List 2 is concatenated with List 1, the final list contains the information on all major lines along with those detected by the secondary HT. Secondary transform is more effective in regaining vertical short lines. This is because lines formed by text in vertical direction are weaker due to the larger inter-line space.

6.7.2 Time reduction for Hough transform

The accuracy of estimating parameters of geometrical transform, form features and respectively the form description depend significantly on image resolution and quantization of the parameter space. The smaller is the bin size, the higher is the processing accuracy (though, there is a limit of the effect of the bin size on the accuracy [38]), but the much longer is the processing time because of “one-to-many” property of the Hough transform. The Hough transform is computationally expensive. Speeding up of the HT is greatly desirable. In our future work, we consider carrying out the Hough
transform on images of a lower resolution. In that case, any “smart” resolution reduction
technique, such as JBIG [39], can be applied prior to the transform. However, the
registration accuracy might degrade, and this issue is to be studied with care. Another
way of speeding up is the parallel or/and hardware implementation [40,41,42,43].
CHAPTER 7

CONCLUSIONS

In this work, we have reported on the design and the implementation of the automated form processing system which takes the image processing approach. The Hough transform is instrumental for form preprocessing, feature extraction for form type identification, and detection of the control points for the form registration task.

Firstly, system preprocesses template forms for extracting feature vectors off-line. The obtained compact description of the forms is stored as an index the form type identification and registration procedures. Incoming forms are processed on-line by the same procedure as one applied for the templates. The system identifies the type of the form, that is, finds a matching template for co-registering images, that is aligning the input form with its template. Then, further editing/ revising directly with the input form is made possible because of coordinate matching.

Due to the ability of the Hough transform to detect lines even in imperfect environment (noise, skew, broken lines) it is selected as a major technique underlying most procedures in the designed system. The algorithm for the modified HT is developed. In the parameter space of the HT, thresholding is performed, such that when restored, image retains its major lines and the most of text is dropped.

For the form description, we have proposed the line angle distribution, called lines count, the normalized line lengths and the distances from the lines to the image origin as
the feature vectors. Also, the similarity measure, i.e., distance metric is proposed for the matching algorithm.

The identification is a two-step procedure. First, forms are matched by lines count feature. Cyclic rotation of the feature vector is performed to ensure rotation invariance. Thus, the form type is filtered out in a global manner, then another two features are invoked for characterizing the physical layout. Features are normalized such that scale and shift invariance is also achieved. Thus, in combination with the rotation invariant feature, the whole identification is made invariant to rotation, scaling and shifting.

In contrast to the template forms which are of perfect quality, input forms can be affected by the scanner and binarization noise and the geometrical distortions such as skew, rotation, shift and scaling. In our study we have shown first how to remedy from geometrical transforms, but also discussed that the Hough transform is able to reject salt and pepper noise and to capture lines, even if they are broken.

Since the distortions are known, we use the linear conformal transform to perform coordinate mapping. The feature-based approach is applied, wherein the coordinates of so-called control point are used to calculate parameters of the transform. As control points, four corners of the form frames are employed. They are selected due to the reliability of their detection. We have implemented two different versions of corner detection schemes. One is the template-based and the second use the parameter space of the Hough transform to find among all line crossings those corresponding to the frame corners. Four points are sufficient to provide accurate registration with the geometrical transform we use.
In a template-based form registration, a small window of a certain corner pattern is sliding over the form image to match one of four templates. This method though faces difficulties if the scanned document is contaminated by noise.

In the second technique, line crossings are extracted directly from the parameter space of Hough Transform. Due to the known locations of the corner points and the ease of estimating the correspondence between control points in the template form and those in the input form, the registration procedure becomes quite simple and reliable. The speed was greatly increased, as well as the accuracy of registration.

Hough transform is a classical method for pattern analysis. We effectively used the Hough transform to form document analysis. While it is slow in its generalized form, in our work, the modified version is implemented. That allows for speedy realization. The numerical results of our simulation have shown that the system is suitable for on-line processing and analysis.

Additionally, we provide discussion on how to register images if additionally, there is a skew introduced. The projective transform with the coordinates of four corner points is able to perform transformation for four different distortion types, that includes skew, rotation, translation and scaling. We have discussed also FFT-based registration technique in Chapter 5. This is done for the possible future research and the application of the Fourier-based technique to form processing. In the Fourier domain, text dropout can be performed by high pass filtering. In that case, short lines would probably remain, that is, the shortcoming of the Hough transform could be overcome.
REFERENCES


pp.793–797.


[13] Li-Te Cheng and John Robinson, “Fast Image Registration for Wearable Computing”, Multimedia Communications Laboratory Faculty of Engineering and Applied Science Memorial University of Newfoundland.


[18] Ping-Fu Fung, Wing-Sze Lee and Irwin King, Randomized Generalized Hough Transform for 2-D grayscale object detection, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong.


[24] Hongjie Xie, Nigel Hicks, G. Randy Keller, etc. “Automatic image registration based on a FFT algorithm and IDL/ENVI”, Technical report, Pan American Center for Earth and Environmental Studies(PACES), Department of Geological Sciences, Department of Computer, Sciences, University of Texas.

59
at El Paso.


[33] Stefan Growe, Ralf Tönjes, "A Knowledge Based Approach to Automatic
Image Registration”, International Conference on Image Processing (ICIP’97), scheduled for October 26–29, 1997 in Santa Barbara, CA, USA.

[34] Hongjie Xie, Nigel Hicks, G. Randy Keller, etc. “Automatic image registration based on a FFT algorithm and IDL/ENVI”, Technical report, Pan American Center for Earth and Environmental Studies (PACES), Department of Geological Sciences, Department of Computer, Sciences, University of Texas at El Paso.


VITA

Graduate College
University of Nevada, Las Vegas

De Chen Graduate Student

Home Address:
1600 E. University Ave. #126
Las Vegas, Nevada 89119

Degrees:
Bachelor of Science, Physics, 1983
East China Normal University

Master of Science, Physics, 1990
East China Normal University

Thesis Title: Form Processing with the Hough transform

Thesis Examination Committee:
Chairperson, Dr. Emma Regentova, Ph. D.
Committee Member, Dr. Shahram Latifi, Ph.D., P.E.
Committee Member, Dr. Henry Selvaraj, Ph.D.
Graduate College Representative, Dr. Kazem Taghva, Ph.D.