Radon transform-based invariant image recognition

by

Osama K. Al-Shaykh

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE

Department: Electrical Engineering and Computer Engineering
Major: Electrical Engineering

Iowa State University
Ames, Iowa
1992

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

Many of the real life applications of image processing aim to describe and classify objects in images. Mail sorting, text reading, chromosome analysis, tumor detection, parts identification on assembly lines, non-destructive evaluation, motion control for robots, fingerprint matching, and target detection and identification are examples of such applications [1].

Any pattern recognition process can be divided into three major stages (Figure 1.1) which are:

1. Data presentation and feature extraction.
2. Learning, or Training, involving the determination of optimum decision procedures, which are needed in the identification and classification stage.
3. Identification and classification.

While extracting features from objects, we make little use of the position, size, or orientation of the object in the image. Rather, the information is contained in the shape of the object, which might be defined as properties of an object that are invariant under rigid-body motion [3]. It is rare that we want to recognize an object from a transformed version of it. An example of this is recognizing "6" from "9".

Invariant recognition *(recognition of visual patterns irrespective to their position,*

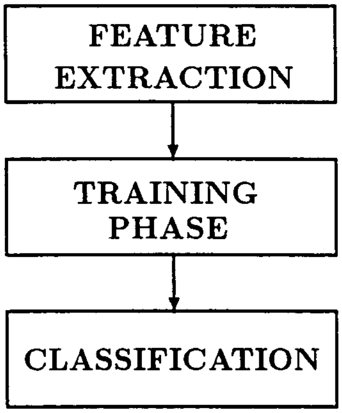


Figure 1.1: Pattern recognition process.

orientation, scale, and deformation) is one of the main concerns of researchers in the fields of pattern recognition and neural networks. Researchers have tackled this problem since the beginnings of the research on pattern recognition. Pitts and McCulloch [4] discussed in 1947 the perception of auditory and visual forms. They described a neural mechanism which exhibits invariant recognition of forms. They proposed averaging over a group of transformations to achieve invariance with respect to this group of transformations.

Hu introduced the method of moments to achieve invariant recognition. He used the ordinary, geometric, moments and the methods of algebraic invariants [5]. Afterwards, other types of moments were introduced, such as Legendre moments, Zernike moments, rotational moments and complex moments [6, 7, 8].

Fourier descriptors, defined as the discrete Fourier transform of the complex combination of the coordinates of the boundary of the object, have been used for invariant recognition. They were used in character recognition, aircraft identification and for non-destructive evaluation [1, 9, 10].

Stochastic models have been used to represent and classify boundaries. The coefficients of the autoregressive model of an image are invariant to translation, scale, and starting point [1].

Many researchers have found the application of artificial neural networks (ANN) to invariant pattern recognition particularly attractive because of the possible relationship between ANN and biological neural nets [11]. There are three techniques that render neural networks invariant to transformations [11]:

1. Invariant feature space.
2. Invariance by training.
3. Invariance by structure.

Invariant feature space includes using Fourier descriptors, method of moments, or any other invariant features of the object. These features are used as the input to the neural network.

Invariance by training is achieved by presenting transformed versions of the input to the neural network in the training phase. The training patterns should be chosen in a way to represent the expected transformations of the patterns. A major drawback of this approach is the need for large networks and memory matrices to store these patterns. Another drawback is the poor understanding of the way the neural network chooses its decision surfaces and the difficulty understanding the generalization that the neural network is forming.

Imposing invariance into the structure of the neural network is done by creating connections between the neurons which force transformed versions of the same input to have the same output [11]. An example of this is Fukushima's model, the neucognitron, which was developed from 1980-1988 as a follow up to the cognitron neural network developed by Fukushima himself during the period 1969-1981. The neucognitron consists of 50732 processing elements divided into four layers, each layer consists of two sublayers. The input to the network is a 19 × 19 image. The first sublayer of the network extracts local features from the image such as lines and then the second sublayer re-expresses the responses of the first sublayer by combining nearby responses and giving insensitive response to small changes and distortions. Then, the local features are combined and passed to the second layer. This kind of process is run in all layers. The second sublayer of the fourth layer consists of ten output nodes tha t correspond to the ten possible classes. The neucognitron is invariant to scale and translation. However, it is a very complicated network and is considered to be one of the largest developed to date [12, 13].

Higher order, "$\sigma - \pi$", neural networks, Figure 1.2, fall under the same category of the neucognitron. Maxwell utilized the concept of averaging proposed by Pitts and McCulloch along with higher order neurons to achieve invariance to translation, scale, and orientation [14]. Higher order neurons are neurons that have a preprocessing stage in which functional polynomials are applied to the input of the neuron. Afterwards, it applies a nonlinearity to the sum of the preprocessing stage. Third order neural networks proved to be invariant to translation, scale, and orientation [8, 14, 15]. However, the combinatoric explosion of higher order terms as the size of the image increases has made the use of this network impractical.

Widrow proposed a model that consists of ADALINE networks, which are invariant to translation and 90° rotation [16]. Fukumi, Omatu, Takeda, and Kosaka

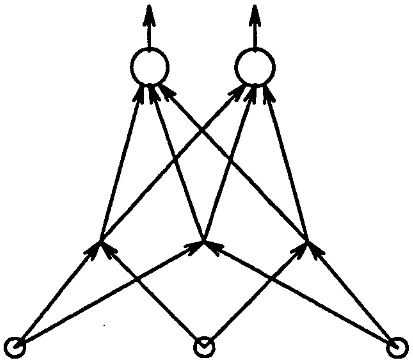


Figure 1.2: Second order neural network.

modified Widrow's network and proposed preprocessing retinal patterns with a rotationally invariant fixed neural network followed by a trainable neural network [17].

Chan has proposed two networks based on the back-propagation neural networks. The first one is translation invariant and capable of multiple object recognition. It consists of two back-propagation neural networks [18]. The second network reduces the inhibitory effect of the back-propagation in the training phase. It is capable of multiple object recognition only [19].

Projection space presentation and manipulation were used for image filtering using one dimensional filters. It was also used for image analysis; algorithms have been developed to approximate Hough transform, compute the convex hull and other features of objects, and compute statistical features such as centroids and principal components of binary objects [20]. U and Flachs utilized the projection to extract global structural features for pattern recognition purposes [21].

Sanz, Hinkle, and Jain provided an efficient engine, Parallel Pipeline Projection Engine P^3E, for changing representation from image space to Radon space and back again [20]. In order to speed up the process, methods have been proposed for computing fast Radon transform and fast inverse Radon transform [28].

In this thesis, an approach that falls under the invariant feature space category is presented. This approach is based on utilizing the projections (Radon transform, shadowgram, or gray sum) of the image to achieve invariant features. The singular values of the projection image, constructed by row-stacking each projection, are used as the invariant features.

One of the most important applications of Radon transform is computerized tomography for which Hounsfield and Cormack have shared the Nobel prize in medicine in 1978. Klug was also awarded the Nobel prize in chemistry in 1982 for a series of papers on tomographic methods in electron microscopy. the Radon transform has other applications in the fields of radio astronomy, economics, and nuclear medicine.

Outline

This thesis is organized as follows: Chapter two overviews the definitions and properties of Radon transform and singular value decomposition. It also discusses the methods of moments and feed forward neural networks. Chapter three develops the projection-based invariant recognition system. Chapter four illustrates the effectiveness of the proposed method. Finally, Chapter five presents conclusions. Appendix one presents the proofs of the equations of Chapter four.

CHAPTER 2. OVERVIEW

Introduction

In this chapter, a mathematical overview of the transforms and the methods that are used in this work is presented. First, the Radon transform is discussed. Afterwards, singular value decomposition is presented along with some of its applications in image analysis. The back-propagation neural network is also presented. Finally, the method of moments is presented.

The Radon transform

In 1917, Radon presented a paper on the determination of functions from their integrals along certain manifolds. He derived an explicit formula for the reconstruction of a function on the plane given its integral over all lines [24].

The first applications of the Radon transform appeared in radio astronomy by Bracewell, and subsequently in electron microscopy by Klug and Vanstein [23]. The construction of tomographic images from projections appeared simultaneously with computers due to the extensive computations needed for tomographic reconstruction. In 1970, Hounsfield introduced the first computerized tomographic scanner, CT scanner, for medical applications. He and Cormack, who developed the mathematical and computational aspects of the CT scanner, were jointly awarded the Nobel prize in medicine in 1978. In 1982, Klug was awarded the Nobel prize in chemistry for a series of papers on tomographic methods in electron microscopy [25]. Radon transform has other applications including nuclear medicine and economics [26].

Radon transform has been applied to image processing in areas other than tomographic reconstruction. It was used in image segmentation [1], structural extraction by projections. [21], determining the orientation of an object, recognition of Arabic characters [27], and one dimensional processing, filtering and restoration of images [20].

The parallel pipeline projection engine, P^3E, introduced by Sanz, Hinkle, and Jain is an efficient engine for transforming the image space to Radon space and vice versa [20]. Other methods and VLSI architectures have been proposed for fast Radon transform and Fast inverse Radon transform [28].

Definition

The Radon transform, ray sum, shadowgram, or projections of a function, denoted as $g(s,\theta)$ is defined as its line integral along a line inclined at an angle θ from the y axis at a distance s from the origin (Figure 2.1) [1]. That is

$$\begin{aligned} g(s,\theta) = \mathbf{R}(f) &= \int\int f(x,y)\delta(x\cos\theta + y\sin\theta - s)dxdy && (2.1) \\ &= \int f(s\cos\theta - u\sin\theta, s\sin\theta + u\cos\theta)du, && (2.2) \\ & \quad -\infty < s < \infty, 0 \leq \theta < \pi \end{aligned}$$

The definition of the Radon transform can be extended to n-dimensional functions [23]. If points in $\Re^n$ are designated by $\mathbf{x} = (x_1, x_2, \cdots, x_n)$, functions de-

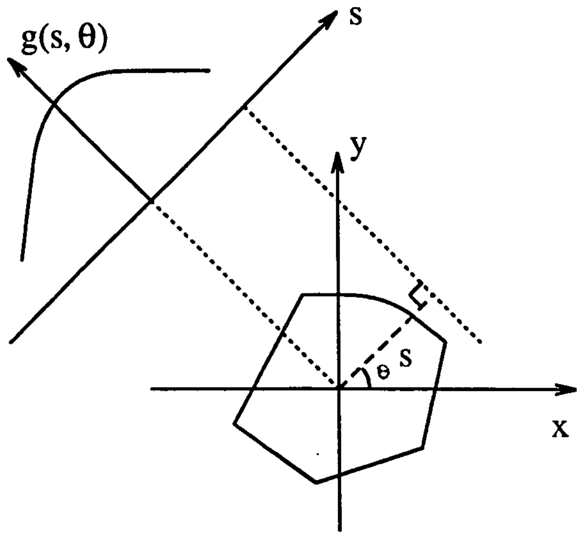


Figure 2.1: The Radon transform of an object

fined on $\Re^n$ by $f(\mathbf{x}) = f(x_1, x_2, \cdots, x_n)$, the volume element is designated by $d\mathbf{x} = dx_1 dx_2 \cdots dx_n$, and ζ is a unit vector that defines the orientation of a hyperplane with equation

$$s = \zeta \cdot \mathbf{x} = \zeta_1 x_1 + \zeta_2 x_2 + \cdots + \zeta_n x_n \tag{2.3}$$

The Radon transform of f may be written as

$$g(s, \zeta) = \mathbf{R}(f) = \int f(\mathbf{x}) \delta(\zeta \cdot \mathbf{x} - s) d\mathbf{x} \tag{2.4}$$

For two dimensional space

$$\zeta = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \tag{2.5}$$

If $f(\mathbf{x})$ is defined on $\Re^n$ then its Radon transform, $g(s, \zeta)$, would be defined on $\Re \times S^{n-1}$, where S^{n-1} is the unit hypersphere in $\Re^n$.

The inverse Radon transform, image reconstruction from its projections, can be found using the back-projection theorem which states

$$f(x,y) = \int_o^{\pi} \hat{g}(x\cos\theta + y\sin\theta, \theta)d\theta \tag{2.6}$$

where

$$\hat{g}(s,\theta) = \frac{1}{2\pi^2}\int_{-\infty}^{\infty} \left[\frac{dg(t,\theta)}{dt}\right] \frac{1}{s-t}dt \tag{2.7}$$

The derivation of the back-projection theorem relies on the Fourier-slice theorem which states that the one-dimensional Fourier transform of the projection equals the polar Fourier transform of the image along the angle of that projection. That is

$$G(\zeta,\theta) = F_P(\zeta,\theta) = F(\zeta\cos\theta, \zeta\sin\theta) \tag{2.8}$$

Properties of the Radon transform

The Radon transform has several useful properties. These properties can be utilized to achieve invariant features of the image in the Radon transform. These properties were used to process images using one dimensional systems.

Let $g(s,\theta)$ be the projections of an image, two dimensional function, $f(x,y)$ or $f_P(r,\phi)$ in the polar coordinates, then the following properties hold [1]:

- Linearity

$$\mathbf{R}\{a_1 f_1(x,y) + a_2 f_2(x,y)\} \quad = \quad a_1 g_1(s,\theta) + a_2 g_2(s,\theta) \qquad (2.9)$$

- Space limitedness

  If

$$f(x,y) = 0 \quad , \qquad |x|,|y| > \frac{D}{2}$$

  then

$$g(s,\theta) = 0 \quad , \qquad |s| > \frac{D}{\sqrt{2}}$$

- Symmetry

$$g(s,\theta) \quad = \quad g(-s,\theta \pm \pi) \qquad (2.10)$$

- Periodicity

$$g(s,\theta) \quad = \quad g(s,\theta + 2k\pi) \qquad (2.11)$$

$$k = \text{ integer}$$

- Shift

$$\mathbf{R}\{f(x-x_0, y-y_0)\} \quad = \quad g(s - x_0 \cos\theta - y_0 \sin\theta) \qquad (2.12)$$

- Rotation

$$\mathbf{R}\{f_p(r,\phi+\phi_0)\} \quad = \quad g(s,\theta+\phi_0) \qquad (2.13)$$

- Scaling

$$\mathbf{R}\{f(\alpha x, \alpha y)\} \;=\; \frac{1}{|\alpha|} g(\alpha s, \theta) \tag{2.14}$$

- Mass conservation

$$M = \int\int f(x,y)dxdy = \int g(s,\theta)ds \tag{2.15}$$

- Convolution

  The Radon transform of the convolution is the convolution of the Radon transforms [23].

$$\mathbf{R}\{f_1 * f_2\} \;=\; g_1 * g_2 \tag{2.16}$$

Singular value decomposition

Definition

Let A be a linear operator between separable Hilbert spaces X, Y

$$A \;:\; X \to Y \tag{2.17}$$

The triple $\{u_n, v_n, \sigma_n\}_{n\geq 0}$ is called a singular value decomposition (SVD) of the operator A if

$\{\mathbf{u_n}\}_{n\geq 1}$ is a complete orthonormal system in X,

$\{\mathbf{v_n}\}_{n\geq 1}$ is an orthonormal system in Y,

and $\{\sigma_n\}$ is a set of non-negative and real numbers,

then

$$Au_n = \sigma_n v_n \quad \text{and} \quad A^* v_n = \sigma_n u_n$$

where A^* is the adjoint of A

The singular values , σ_n, are usually ordered such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$ [29].

The singular functions $\{u_n\}$ are sometimes called the generalized eigenfunctions, since it can be shown that

$$(A^* A)u_n = {\sigma_n}^2 u_n \tag{2.18}$$

A can be constructed from the triple by

$$A = \sum \sigma_n v_n {u^*}_n \tag{2.19}$$

Back-propagation neural networks

The term neural network, or more likely neuronal networks, originally referred to a network of interconnected neurons [30]. Neural networks have been studied to achieve human-like performance in the fields of speech and image recognition.

Artificial neural networks (connectionist models, parallel distributed processing models, or neuromorphic systems) are structured based on our understanding of the biological nervous systems. They are basically constructed from simple processing units (neurons, nodes, or adalines). These processing units are connected with each other either fully or partially. Neural networks can be classified based on the process done by the neuron, e.g., linear and nonlinear classifiers [31]. Neural networks are also classified based on the way the neurons are connected, e.g., feed forward and bidirectional associative memories. In this section, the back-propagation neural network, which is a non-linear feed forward neural network, will be discussed.

The Back-propagation neural network (the multi-layer perceptron) is a feed forward neural network, i.e., there is no feedback connections between its layers and between the neurons of the layers themselves. It consists of non-linear neurons (Figure 2.2). The neurons perform a weighted sum of their inputs and pass the sum through a non-linearity (Figure 2.3). That is

$$y_i = f\left(\sum_{j=0}^{N} w_{ji}x_j - \theta_i\right) \tag{2.20}$$

where

y_i is the output of node i

w_{ji} is the connection strength between neuron i and neuron j

x_j is the output of neuron j which is from the previous layer in feed forward networks

θ_j is an internal offset in the neuron j

$f(.)$ is the nonlinearity

The training of the back-propagation is a generalization of the least mean square algorithm where the error is propagated from the output nodes to the input nodes. The following describes the algorithm [32]:

1. The weights and the offsets are initialized using small random values.

2. Desired inputs, $\acute{\mathbf{x}}$, and desired outputs, $\mathbf{d}$, are presented to the network.

3. The actual outputs of the neural network are calculated by calculating the output of the nodes and going from the input to the output layer. The most common non-linearity is

$$f(\alpha) = \frac{1}{1+e^{-\alpha}} \tag{2.21}$$

4. The weights are adapted by back propagating the error from the output to the input layer. That is

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x'_i \tag{2.22}$$

where the δ_j is the error propagated from node j. If node j is an output node, then

$$\delta_j = y_j(1-y_j)(d_j - y_j) \tag{2.23}$$

However, if node j is not an output node, then

$$\delta_j = x'_j(1-x'_j)\sum_k \delta_k w_{jk} \tag{2.24}$$

where the sum is over the nodes of the following layer.

5. This process is done over all training patterns.

Kolmogorov proved that any continuous function $f : [0,1]^n \rightarrow \Re^m, f(\mathbf{x}) = \mathbf{y}$, can be implemented by a three layer network [12]. However, there is no proof for the convergence of the back propagation algorithm. Several modification on the back-propagation algorithm have been proposed to speed-up the convergence and to prevent convergence to a local minima in the weight space.

Methods of moments

Methods of moments have been used in several forms to extract invariant features from images. Hu first introduced invariant recognition using moments [33]. Based on geometric moments and algebraic invariants, he derived a set of invariant features. Afterwards, orthogonal moments have been suggested, e.g., Zernike, pseudo Zernike, and Legendre moments. Other types of moments have been used too, e.g., rotational and complex moments [7].

In this section geometric moments, which are the most common type of moments used, and Zernike moments, which outperform other types of moments, will be discussed.

Geometric moments

Geometric moments are defined as the projections of the image, $f(x, y)$, onto the polynomial $x^p y^q$. That is

$$m_{pq} = \int\int x^p y^q f(x,y) dx dy \tag{2.25}$$

where m_{pq} is the geometric moment of order $(p+q)$ of $f(x,y)$. The basis set $\{x^p y^q\}$ is complete but not orthogonal [7].

The central moments are defined as

$$\mu_{pq} = \int\int (x - \bar{x})^p (y - \bar{y})^q f(x,y) dx dy \tag{2.26}$$

where

$$\bar{x} = m_{10}/m_{00} \quad , \quad \bar{y} = m_{01}/m_{00}$$

The central moments are invariant to translation. They are used to derive the invariant features.

If an image , $f(x, y)$, is scaled by α then the normalized moments [1], defined as

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{\gamma}} \quad , \quad \gamma = (p + q + 2)/2 \tag{2.27}$$

are invariant to scaling.

Hu used the algebraic invariants to derive these features that are invariant to translation, scaling, and rotation [33, 1, 34]:

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} && (2.28)\\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_1 1^2 && (2.29)\\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + \eta 03)^2 && (2.30)\\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 && (2.31)\\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad +(\eta_{03} - 3\eta_{21})(\eta_{03} + \eta_{21})[(\eta_{03} + \eta_{21})^2 - 3(\eta_{12} + \eta_{30})^2] && (2.32)\\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad +4\eta_{11}(\eta 30 + \eta_{12})(\eta_{03} + \eta_{21}) && (2.33)\\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad +(\eta_{30} - 3\eta 21)(\eta 21 + \eta 03)[(\eta_{03} + \eta_{21})^2 - 3(\eta_{30} + \eta_{12}^2] && (2.34)
\end{aligned}$$

Zernike moments

Zernike moments, A_{nl}, of an image , $f(r, \phi)$, are defined as its projections on a class of polynomials, called Zernike polynomials. These polynomials are separable in the polar coordinates and are orthogonal over the unit circle [1, 8, 7, 35]. That is

$$A_{nl} = \frac{n+1}{\pi} \int_{2\pi} \int_{0}^{\infty} \left[V_{nl}(r,\phi)\right]^{*} f(r,\phi) r dr d\theta \tag{2.35}$$

V_{nl} are the Zernike polynomials and are defined as

$$V_{nl}(r,\phi) = R_{nl}(r) e^{jl\phi} \tag{2.36}$$

where

$$R_{nl}(r) = \sum_{s=0}^{(n-|l|)/2} (-1)^{s} \frac{(n-s)!}{s!\left(\frac{n+|l|}{2}-s\right)!\left(\frac{n-|l|}{2}-s\right)!} r^{n-2s} \tag{2.37}$$

The image can be reconstructed from the Zernike moments by

$$f(r,\phi) = \sum_{n=0}^{\infty} \sum_{\substack{l \\ n-|l|=even \\ |l|\leq n}}^{\infty} A_{nl} V_{nl}(r,\phi) \tag{2.38}$$

It can be approximated by

$$f(r,\phi) = \sum_{n=0}^{\acute{n}} \sum_{\substack{l \\ n-|l|=even \\ |l|\leq n}}^{\infty} A_{nl} V_{nl}(r,\phi)$$

In pattern recognition applications, $\acute{n}$ usually equals 9, 10, or 12 [8]. This results in using 28, 34, or 40 features respectively [35].

The performance of the Zernike moments has been shown to outperform other type of moments in terms of sensitivity to image noise, information redundancy, and capability for image representation [7].

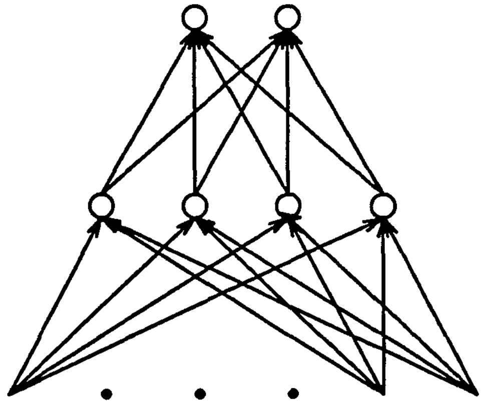


Figure 2.2: A two layer feed forward neural network

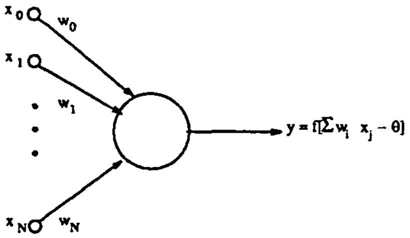


Figure 2.3: Neuron or computational element which does a weighted sum of its inputs and then performs a nonlinearity on the sum

CHAPTER 3. PROJECTION-BASED INVARIANT FEATURES

Introduction

The Radon transform representation of an image can be utilized to process it using one-dimensional signal processing techniques [20]. Moreover, features extracted from the Radon representation of the image have been used for pattern recognition purposes [37, 38, 39]. The Hough transform, which is a special case of the Radon transform, was used in image analysis for edge detection and feature extraction purposes [41].

Any image can be approximately reconstructed from its projections. The angular resolution at which projections are taken depends on the maximum reconstruction error acceptable. The linearity, shift, rotation and scaling properties of the Radon transform can be utilized to achieve an invariant feature vector. Here, an algorithm that utilizes these properties to achieve invariance to translation, rotation and scaling is presented. The singular values of a matrix constructed by row-stacking of projections are used to construct the invariant feature vector. This feature vector will be used as input to a classifier which is here the back-propagation neural network followed by a maximum-output-selector. A performance function is introduced to evaluate the performance of the recognition system. This performance function can also be used to indicate how closely the pattern matches the decision template.

Projection based invariance

The invariance with respect to translation, rotation and scaling will be discussed separately. Afterwards, all procedures will be incorporated in a complete system that can achieve invariance to translation, rotation and scaling.

A complete proof of the equations presented is in Appendix I.

Invariance to translation

If an object is translated then, according to the shift property of the Radon transform, each projection will be translated by a distance which is a function of the translation distance, (x_0, y_0), and the projection angle, θ. That is

$$s_\theta = x_0 \cos\theta + y_0 \sin\theta \tag{3.1}$$

where s_θ is the projection translation caused by a translation of (x_0, y_0) at a projection angle θ.

Centering each projection around its center of mass will eliminate the translation effect. This can be justified by the fact that centering each projection around its center of mass will result in centering the object around its center of mass. This is true because

$$\hat{s} = \frac{m^1}{m^0} \tag{3.2}$$

where

$$m^0(\theta) = \int\int f(x,y)dxdy = M \tag{3.3}$$

$$m^1(\theta) = M_x \cos\theta + M_y \sin\theta \tag{3.4}$$

where $\hat{s}$ is the center of mass of the projection taken at angle θ. The quantities m^0 and m^1 are the average and first order moment of the projection. M, M_x and M_y are the average and the moments in the x and y directions of the object, respectively.

Invariance to rotation

Invariance to rotation of an object by an arbitrary angle is achieved by extracting features that are independent of the ordering of the projections. That is, if each projection is placed in a row of a matrix (Figure 3.1) the features must be independent of the way the rows of the matrix are organized. There are a few algebraic parameters which are invariant to the order of the rows of a matrix but not all are distinct for each matrix, e.g., the trace. However, the singular values of the projection-matrix are invariant to the order of rows. This means that singular values are invariant to rotation of integer multiples of the angular resolution of projections.

It is shown in Appendix one that the image can be represented as a sum of multiplications of the singular values with functions of the radial distance from the origin and of the angle taken from the abscissa. That is

$$f(r,\phi) = \sum_k \frac{\sigma_k}{2\pi^2} \int_0^{\pi} u_k(\theta) d\theta \left\{ \int_s \frac{\left[\partial w_k(s)/\partial s\right]}{r\cos(\theta-\phi)-s} ds \right\} \tag{3.5}$$

where

σ_k	Singular values of the Radon transform
$w_k(s)$	Right singular vector spanning the domain of the projections
$u_k(\theta)$	Left singular vector spanning the angle , θ, of the projections
k	runs over the rank of the projection-matrix.

The Radon transform is taken over $[0^{\circ}, 180^{\circ})$. The symmetry property of the Radon transform equates the projection at angle θ with the abscissa reversed projection at angle $\theta + 180^{\circ}$. As a result, if the image is rotated then at least one row of the matrix will be reversed. Hence, the singular values will not be the same. However, taking projections over whole 360° range would give invariance at increments of $180^{\circ}/N$, where N is the number of projections taken over the image. This can also be done by utilizing the symmetry property, i.e., by reversing the projections alternately. This will result in a 360° coverage.

The rotation angle can be estimated by calculating the translation in the left singular vectors spanning the angle. This is because

$$\begin{aligned} \mathbf{R}f(r, \theta + \phi) &= g(s, \theta + \phi) \\ &= \sum_k \sigma_k u_k(\theta + \phi) w_k(s) \end{aligned} \tag{3.6}$$

where ϕ is the angle of rotation.

Invariance to scaling

If an object, $f(x, y)$, is scaled by a scaling factor α, the scaled object, $f_{\alpha}(x, y)$, will be

$$f_{\alpha}(x, y) = f(\alpha x, \alpha y) \tag{3.7}$$

When the object is scaled, the left singular vector spanning the angle of the projection will not be affected. However, the right singular vector spanning the domain of the projections as in 2.3 will change. That is

$$\mathbf{R}\left\{f_{\alpha}(x, y)\right\} = g_{\alpha}(s, \theta)$$

$$
\begin{aligned}
&= \frac{g(\alpha s, \theta)}{|\alpha|} \\
&= \sum_k \frac{\sigma_k}{\alpha^{3/2}} u_k(\theta) \left[\sqrt{\alpha} w_k(\alpha s)\right] \qquad (3.8)
\end{aligned}
$$

where the last step is shown in Appendix one.

The right singular vector would be $\sqrt{a}w_k(\alpha s)$ and the singular values are $\sigma_k/\alpha^{(3/2)}$. Consequently, when the object is scaled by a scaling factor of α, the l_2 norm of the singular-value vector, i.e., Frobenius norm of the projection matrix, would be scaled by $\alpha^{-3/2}$.

Constructing invariant feature vector

The following material summarizes the projection-based algorithm (Figure 3.1) to achieve invariance to translation, rotation and scaling:

1. Projections are taken over 360°. In most practical systems, only 180° coverage is available. However, the coverage can be extended to 360° by alternately reversing the projections.

2. Each projection is centered around its center of mass. This will center the image and introduce translation invariance in the extracted features.

3. A projection image matrix, is constructed, where the row vectors are the centered projections.

4. SVD is applied to the projection matrix. The order of the singular values is invariant to rotation with the angular resolution.

5. The singular value vector is normalized to achieve invariance to scaling.

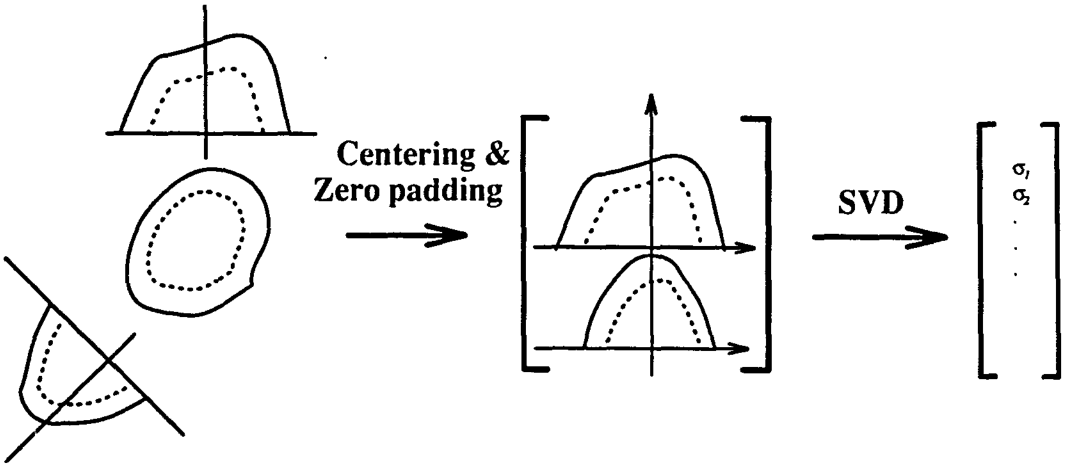


Figure 3.1: Steps to achieve invariant feature vector

Recognition system

After the feature vector is constructed, it is applied to the input of the recognition block or the classifier. The classifier used here is a back-propagation neural network followed by a maximum-output-selector.

The adaptive nature of neural networks as well as their ability to generalize from training data justifies their use for classification in our system. They are expected to overcome the rotation resolution invariance problem. This will be illustrated by the results obtained for worst case resolution, i.e., when the angle of rotation taken is in the middle of the angular sampling period.

The performance of the recognition system involves studying the recognition rate, the similarity of features extracted from patterns of the same class, and the differences in features extracted from patterns of different classes. To incorporate these parameters, we consider a function that sums all the normalized differences between the output nodes and the output of the node corresponding to the correct class as a performance measure. That is

$$P(n) = \frac{\sum_{\substack{k=1 \\ k \neq n}}^{N} Y(n) - Y(k)}{N - 1} \tag{3.9}$$

where

$0 \leq Y(i) \leq 1$ is the output of node i of the neural network

n is the node corresponding to the correct output class

N is the number of output nodes of the neural network

This performance measure produces values between -1 and 1. It gives its best output, 1.0, when the output of the node corresponding to the correct class equals 1.0 and the output of the rest of the nodes equal 0. It gives its worse output when all the output nodes have values of 1.0 and the output of the node corresponding to the correct class equals 0.

This function is used to give an indication of how closely a pattern matches the template the system classified. This function is analogous to the membership function defined in fuzzy set theory [36]. Along with some defined rules, it can be used to give a syntactic description of the output of the neural network. When this function is used as a membership function, its range will be $[0, 1]$. This is because $Y(n) \geq Y(k)$ for all $k \neq n$. Table 3.1 illustrates the effectiveness of this function when used for a neural network with three output nodes.

Table 3.1: Performance measure and membership values[a]

Output of			Performance	Membership
node 1	node 2	node 3	measure	value
1.000	0.000	0.000	1.000	1.000
1.000	1.000	0.000	0.500	0.500
1.000	1.000	1.000	0.000	0.000
0.000	1.000	0.000	-0.500	1.000
0.000	1.000	1.000	-1.000	0.500

[a] This is assuming the pattern tested belongs to Class 1, i.e., the output pattern should be $(1, 0, 0)$

CHAPTER 4. RESULTS

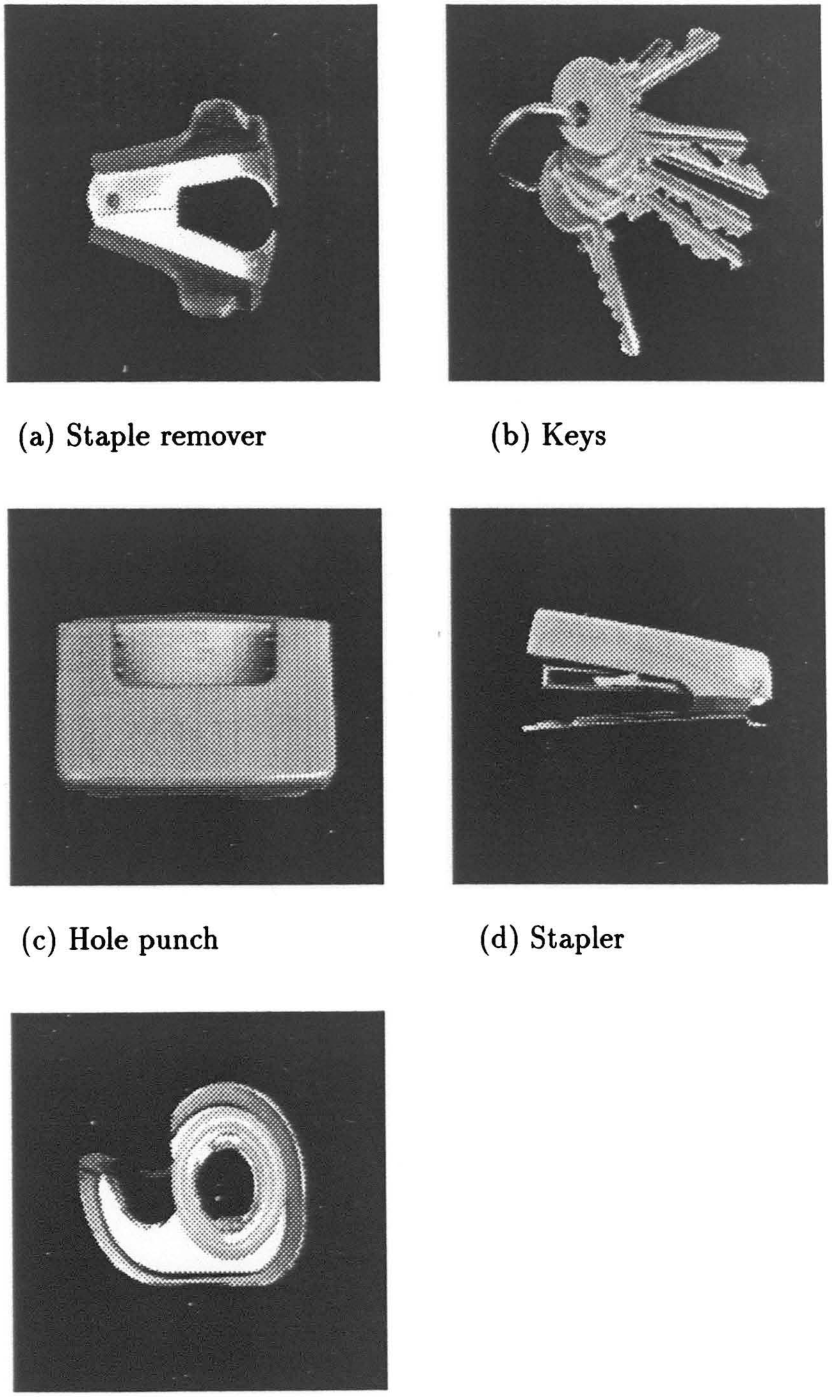
Introduction

In this chapter, the proposed method is examined and its merits are illustrated. The method is then compared to the method of moments. The example chosen for testing (Figure 4.1) consists of staple remover, stapler, keys, hole punch and tape dispenser. The objects in this example are chosen because they contain similarities and differences among them. The staple remover and the stapler have the same general shape, i.e., the "V" shape. The hole punch and the tape dispenser have almost the same region of support. However, the keys have a different shape.

The images were taken using an 8-bit black and white camera. They were displayed using Happi software, which was developed by the image processing group at Iowa State University. Happi was used along with other programs to transform the images. It was also used to create noisy versions of the images.

The neural network used for classification has the same number of layers, hidden nodes and output nodes in all experiments conducted. It consists of three layers where the number of nodes in the input layer equals the length of the feature vector. The hidden layer consists of 14 nodes. The output layer consists of 5 nodes corresponding to the five output classes.

The singular values are calculated using the subroutine provided by Press et al.



(a) Staple remover

(b) Keys

(c) Hole punch

(d) Stapler

(e) Tape dispenser

Figure 4.1: Objects used in testing the method

Table 4.1: Feature vector length experiment

Parameter	Value
No. of projections	Variable
No. of Train. patterns	5 (Original)
No. of Test. patterns	5×10 = 50 each
S/N	∞

[40].

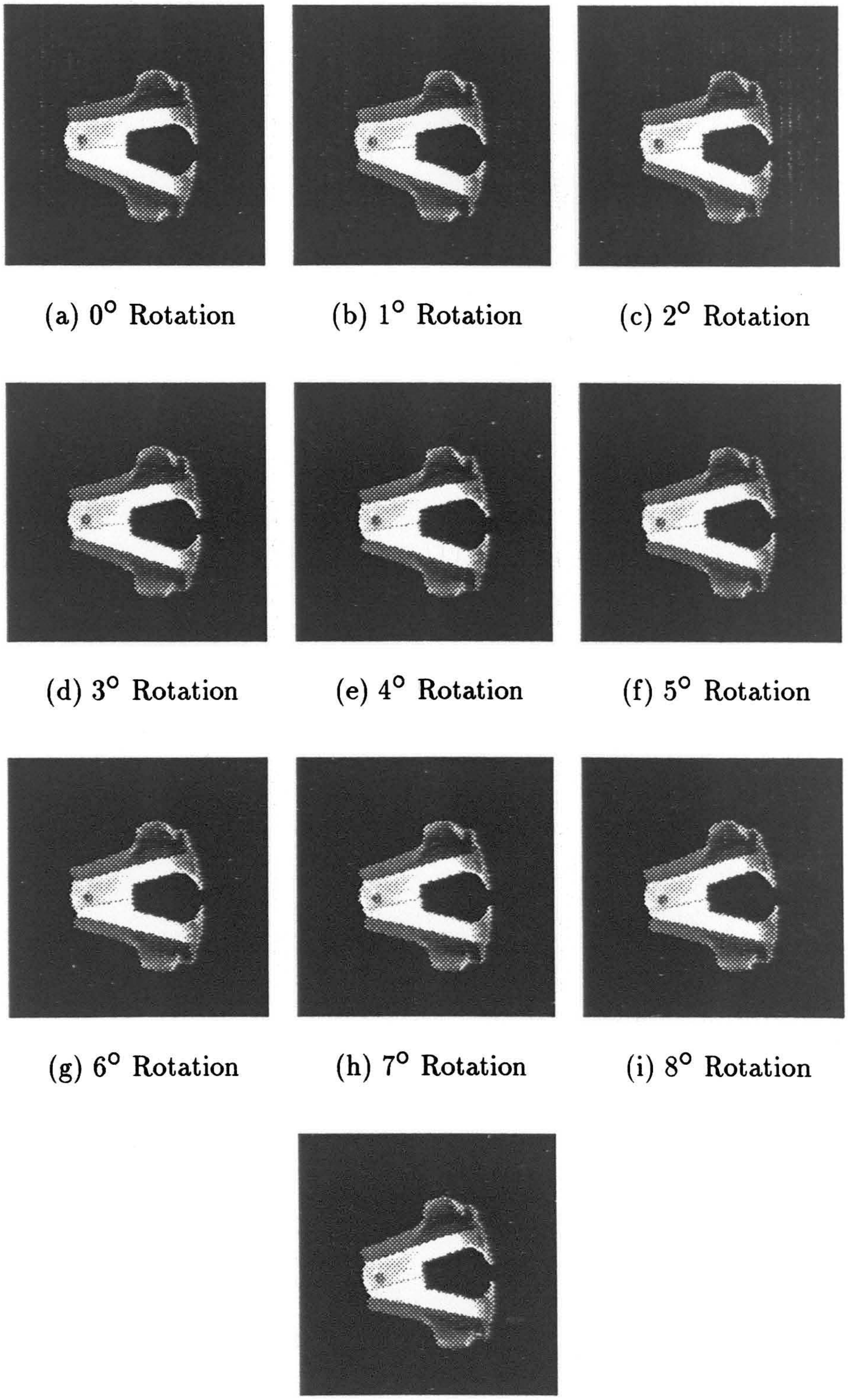
This chapter is organized as follows: The second section discusses the effect of the number of projections used to construct the projection matrix on recognition. The third section illustrates the robustness of the extracted features when the object is rotated or scaled. The fourth section studies the effect of white noise added to the object on recognition. The fifth section discusses combined transformations. Finally the method is compared with the method of moments.

Effect of feature vector length

This experiment (Table 4.1) is conducted to determine the variation of the recognition rate with feature vector length. Changing the length of the feature vector results from changing the angular resolution at which the projections are taken. Changing the angular resolution will affect mostly the recognition of images rotated with angles that are not integer multiples of the angular resolution.

To study this effect, ten images (Figure 4.2) rotated in the image plane by integer multiples of 10% of the angular resolution are used for testing. The neural network is trained using the feature vectors of the objects in Figure 4.1.

Figure 4.3 and Figure 4.4 show the recognition percentage and the performance



(a) 0° Rotation

(b) 1° Rotation

(c) 2° Rotation

(d) 3° Rotation

(e) 4° Rotation

(f) 5° Rotation

(g) 6° Rotation

(h) 7° Rotation

(i) 8° Rotation

(j) 9° Rotation

Figure 4.2: Staple remover rotated with 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 degrees respectively

Table 4.2: Feature vector robustness experiment

Parameter	Value
No. of projections	18
No. of Train. patterns	N/A
No. of Test. patterns	$5 \times 10 = 50$
S/N	∞

measure versus the vector length respectively. As evident, the recognition percentage reached 100% when using 6 or more projections to construct the projection-matrix, i.e., an angular resolution of 30° or less. This results in reducing the size of the projection-matrix P and the matrix PP^T which is used to calculate the singular values, e.g., to find the singular values of the projection-matrix constructed using 6 projections, a 6×6 matrix is used. This means a fast feature extraction system and a small neural network.

Feature vector robustness

The rotation invariance property is illustrated by this experiment (Table 4.1). The length of the feature vector used in this experiment is 18, i.e., angular resolution of 10°. Ten representations of each object of Figure 4.1 are generated by rotating each object by integer multiples of 1°. Table 4.3 is a list of the magnitudes of the first six elements of the 18-length feature vector. The respective variance, σ, to mean, μ, percentage, $\sigma/\mu\%$, which indicates the percentage of the spread of the values from their corresponding mean for all objects is shown in Table 4.4. The average of $\sigma/\mu\%$ percentages is 3.1%.

The rotation angle can be estimated from the shift in the left singular vectors

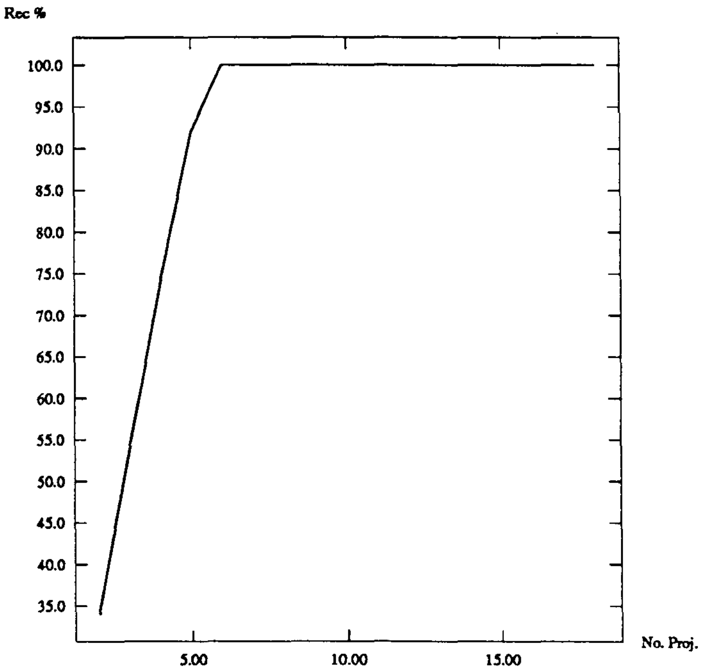


Figure 4.3: Recognition percentage versus number of projections. It can be seen that the recognition was perfect when the number of projections used was above six

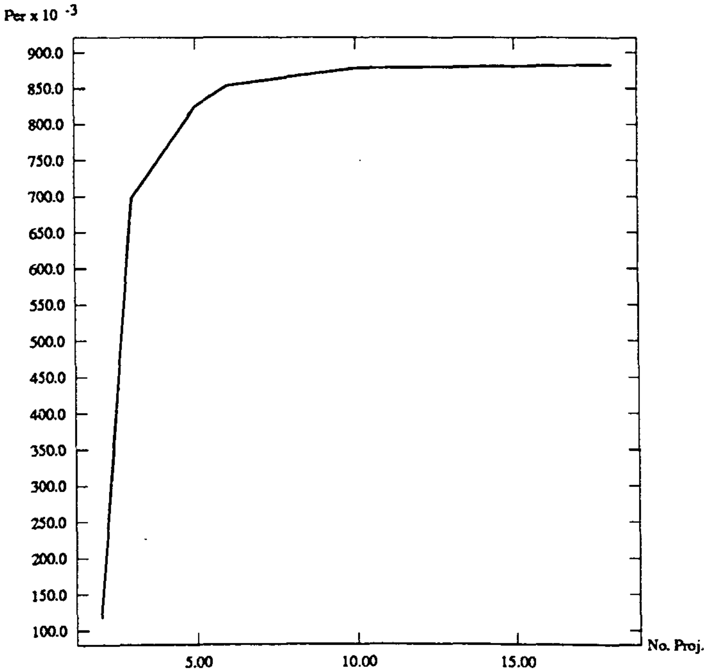


Figure 4.4: Performance measure versus number of projections. The performance measure stabilized around 0.87 when the number of projections used was above six

Table 4.3: The first six features of rotated staple remover. As seen, the features are almost constant against rotation

Angle of rotation	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6
0	0.949880	0.253046	0.116033	0.089392	0.062695	0.047311
1	0.949954	0.253400	0.115608	0.087676	0.063863	0.048238
2	0.949549	0.254868	0.116049	0.088979	0.064080	0.048806
3	0.950039	0.253376	0.115791	0.089452	0.064196	0.049605
4	0.950269	0.252327	0.114461	0.091734	0.064976	0.051829
5	0.950302	0.253028	0.117651	0.090796	0.061883	0.049290
6	0.950132	0.254024	0.117760	0.090544	0.061863	0.049779
7	0.950280	0.252846	0.119163	0.087849	0.060932	0.050372
8	0.950314	0.252748	0.118294	0.088158	0.061528	0.047344
9	0.950264	0.252315	0.118293	0.085925	0.062260	0.048379

spanning the angle. To illustrate this fact, The left singular vector corresponding to the largest singular value of 0°, 10° and 20° rotated images of the staple remover are shown in Figure 4.5. The translation between each vector of them is one unit, i.e., the angle of rotation is 10°.

The scaling invariance property is illustrated the same way (Table 4.5 and Table 4.6). Scaled images (Figure 4.6) with scaling factors of 0.2, 0.4, 0.6, 0.8, 1.0, and 1.2 are used in this experiment. The average of the $\sigma/\mu\%$ percentages is 6.48%.

All the patterns used in this experiment were classified correctly by the neural network used in the previous experiment.

The scaling factor can be estimated using Equation (3.8). Table 4.7 shows the estimated scaling factor. As evident, the estimation is almost the same as the original.

Table 4.4: Variance to mean ratio (percentage) of all the features of the rotated objects. The ratio is small and by the average it is 3.1%. This table further illustrates the robustness of the feature vector against rotation

Feature number	Keys	Staple remover	Hole punch	Stapler	Tape dispenser
1	0.016	0.025	0.017	0.079	0.020
2	0.328	0.292	0.031	0.383	0.631
3	0.566	1.229	0.692	1.516	0.713
4	1.557	1.834	1.204	5.107	0.300
5	3.814	2.049	1.580	5.680	2.506
6	2.999	2.691	5.286	6.313	2.429
7	1.392	4.340	7.168	12.72	3.934
8	3.091	5.914	2.261	6.657	2.462
9	3.040	10.920	6.322	5.730	2.732
10	3.783	8.483	4.480	5.705	2.181
11	4.658	7.078	4.461	4.280	4.377
12	4.769	7.213	5.708	6.597	2.717
13	4.833	6.365	5.961	6.935	4.805
14	5.572	6.157	7.513	5.906	2.911
15	3.426	9.868	6.478	6.771	3.886
16	5.083	8.869	6.633	3.636	4.325
17	7.484	8.690	7.602	5.863	5.573
18	9.800	6.164	7.458	5.452	7.813
Average	2.059	3.715	2.903	4.989	1.741

Table 4.5: The first six features of scaled stapler. As seen, the features are constant and robust against scaling

Scaling factor	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6
0.2	0.902867	0.363903	0.145755	0.125911	0.062013	0.055039
0.4	0.904646	0.358199	0.152912	0.117272	0.074648	0.053142
0.6	0.904751	0.357453	0.151978	0.120282	0.076433	0.051824
0.8	0.904539	0.355983	0.155194	0.125554	0.075964	0.049724
1.0	0.903940	0.355767	0.155259	0.126030	0.077862	0.050962
1.2	0.904409	0.357483	0.155013	0.122711	0.076896	0.050097

Table 4.6: Variance to mean percentage of all the features of the scaled objects. The ratio is small and by the average it is 6.5%. This table further illustrates the robustness of the feature vector against scaling

Feature number	Keys	Staple remover	Hole punch	Stapler	Tape dispenser
1	0.068386	0.032415	0.004781	0.071463	0.056927
2	1.862529	0.931086	0.455430	0.759601	1.870536
3	0.833467	1.028653	0.541184	2.187228	2.571677
4	2.332124	2.138348	3.531025	2.665441	0.923069
5	1.209885	2.517066	2.941441	7.346515	3.177121
6	1.989057	3.809874	2.228511	3.542740	5.660154
7	2.076525	2.244302	1.771226	3.704285	7.833734
8	2.155904	8.808520	10.633659	3.874142	8.478910
9	2.524689	16.365229	12.709072	2.227137	7.728633
10	3.606891	11.691761	11.681385	7.543673	7.570441
11	4.320522	10.603070	12.876406	8.542035	7.945292
12	4.421000	15.615104	4.191956	9.326546	4.276928
13	3.139513	8.984010	12.331057	18.669847	3.594523
14	4.524668	4.823858	5.722368	13.996540	1.712528
15	3.981825	4.570429	16.803171	12.159415	2.830502
16	2.445495	3.401840	18.787466	12.033607	5.861289
17	7.184366	15.953438	12.128682	6.145161	9.295919
18	6.109355	25.314453	17.585051	7.935613	17.050816
19	3.109910	8.261209	7.749619	5.653704	7.645201

Table 4.7: The estimated scaling factor[a] factor using Equation (3.8)

Original scaling factor	Keys	Staple remover	Hole punch	Stapler	Tape dispenser
0.2000	0.1995	0.2020	0.2000	0.2021	0.2006
0.4000	0.4000	0.4003	0.4004	0.4005	0.4005
0.6000	0.5997	0.5997	0.6000	0.6004	0.5998
0.8000	0.8003	0.8007	0.8008	0.8013	0.8007
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1.2000	1.1940	1.1972	1.1983	1.1980	1.1978

[a] The scaling factor is determined relative to the objects of Figure 4.1

Table 4.8: Effect of white noise experiment

Parameter	Value
No. of projections	18
No. of Train. patterns	5 (Original)
No. of Test. patterns	5 x 4 = 20 each
S/N	Variable

Effect of white noise

This experiment (Table 4.5) is conducted to study the effect of white noise on recognition. The length of the feature vector in the experiment is 18. In the training phase, only the original non-noisy objects are used. Four transformed objects (Figure 4.7) of each object are used for testing; the original, translated by $(-10, 10)$, rotated by 45^o, and scaled by a scaling factor of 0.5. White noise with zero mean was added to each testing image (Figure 4.8) Recognition percentage and performance measure for each S/N were calculated. Figure 4.9 and Figure 4.10 show the results obtained.

Table 4.9: Combination of transformations experiment

Parameter	Value
No. of projections	18
No. of Train. patterns	5 (Original)
No. of Test. patterns	$5 \times 10 = 50$
S/N	Variable

Figure 4.11 shows the performance of each transformation done to the objects versus noise added.

Combination of transformations

This experiment (Table 4.9) is conducted to study the effect of combining transformations. The length of the feature vector is 18. Five transformed objects (Figure 4.12) of each object are used for testing; scaled by 1.2 and rotated by 45°, scaled by 0.2 and rotated by 45°, scaled by 0.2, rotated by 45°, and translated by (-50, 50), scaled by 0.2, rotated by 45°, and translated by (-50, 0), and scaled by 0.2, rotated by 45°, and translated by (0, 50). White noise with zero mean was added to each testing image. Performance measure for each S/N was calculated. Figure 4.13 shows the result obtained.

This result shows that the performance declines fast when the object is shrunk and translated. This results from the poor alignment of noisy-pulse-shaped projections. Better performance is expected if the projections were low-pass-filtered before constructing the projection matrix.

Comparison with the method of moments

This experiment is conducted to compare the performance of the Radon-based invariant image recognition method with the method of Zernike moments. Moment-based invariant image recognition compares well with the results of other popular invariant feature extraction [8]. The performance of the Zernike moments has been shown to be superior to the performance of other types of moments in terms of sensitivity to image noise and information content [7].

The same experiment conducted to study the effect of white noise on recognition was conducted here. The order of the Zernike moments used to conduct this experiment was 11 and the classifier used was a three layer network with 40 input nodes, 53 hidden nodes, and 5 output nodes. Figure 4.14 and Figure 4.15 show that the recognition percentage of the Radon-based invariant features is better than that of the Zernike moments. This is further illustrated by the performance measure where the performance of the proposed method is much better than that of the Zernike moments even when the recognition percentage is close.

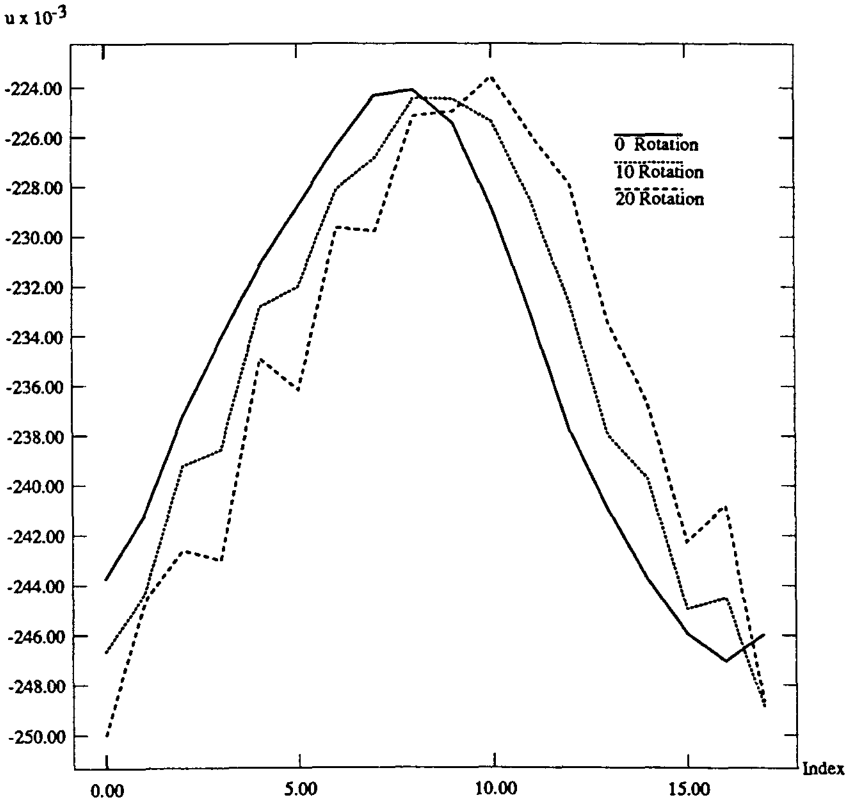
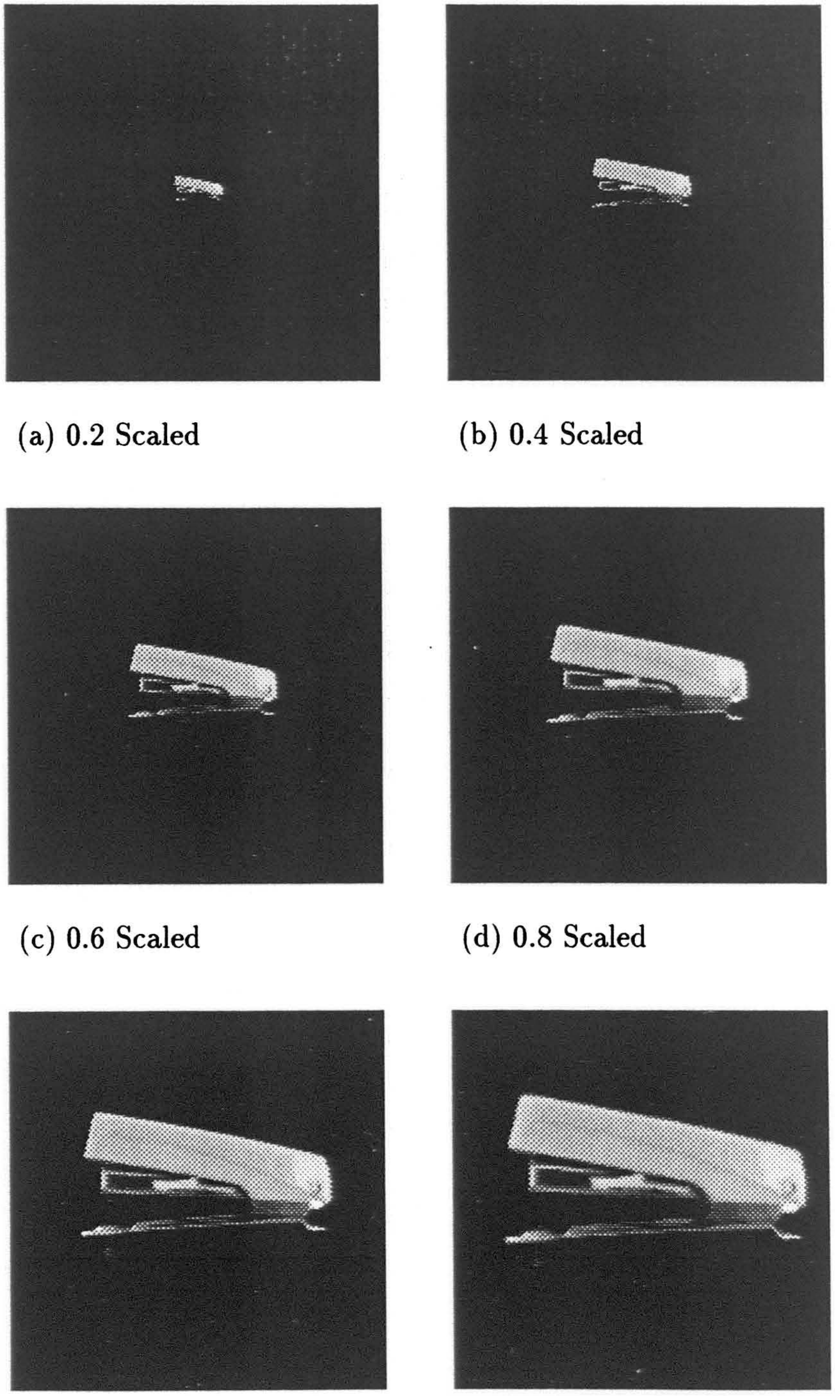


Figure 4.5: The shift of the first left singular vector due to the rotation of the staple remover by an angle of 0°, 10°, and 20° respectively. The shift in the left singular vector equals to one unit which corresponds to rotation of 10°



(a) 0.2 Scaled

(b) 0.4 Scaled

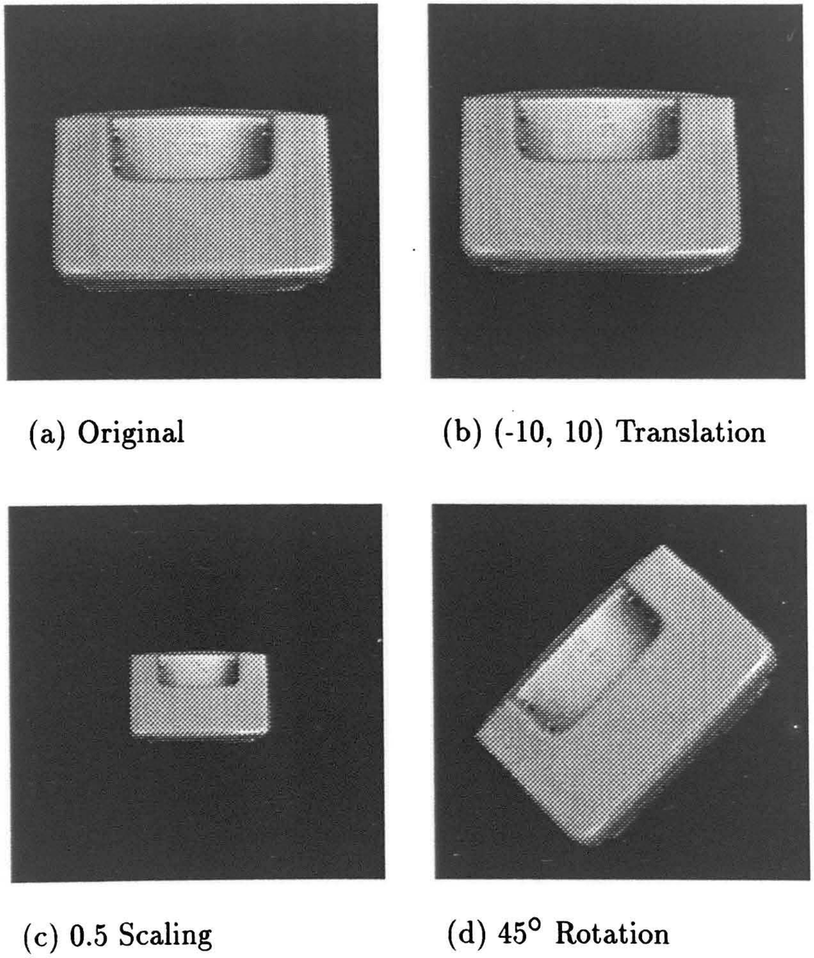
(c) 0.6 Scaled

(d) 0.8 Scaled

(e) 1.0 Scaled

(f) 1.2 Scaled

Figure 4.6: Scaled images of the stapler



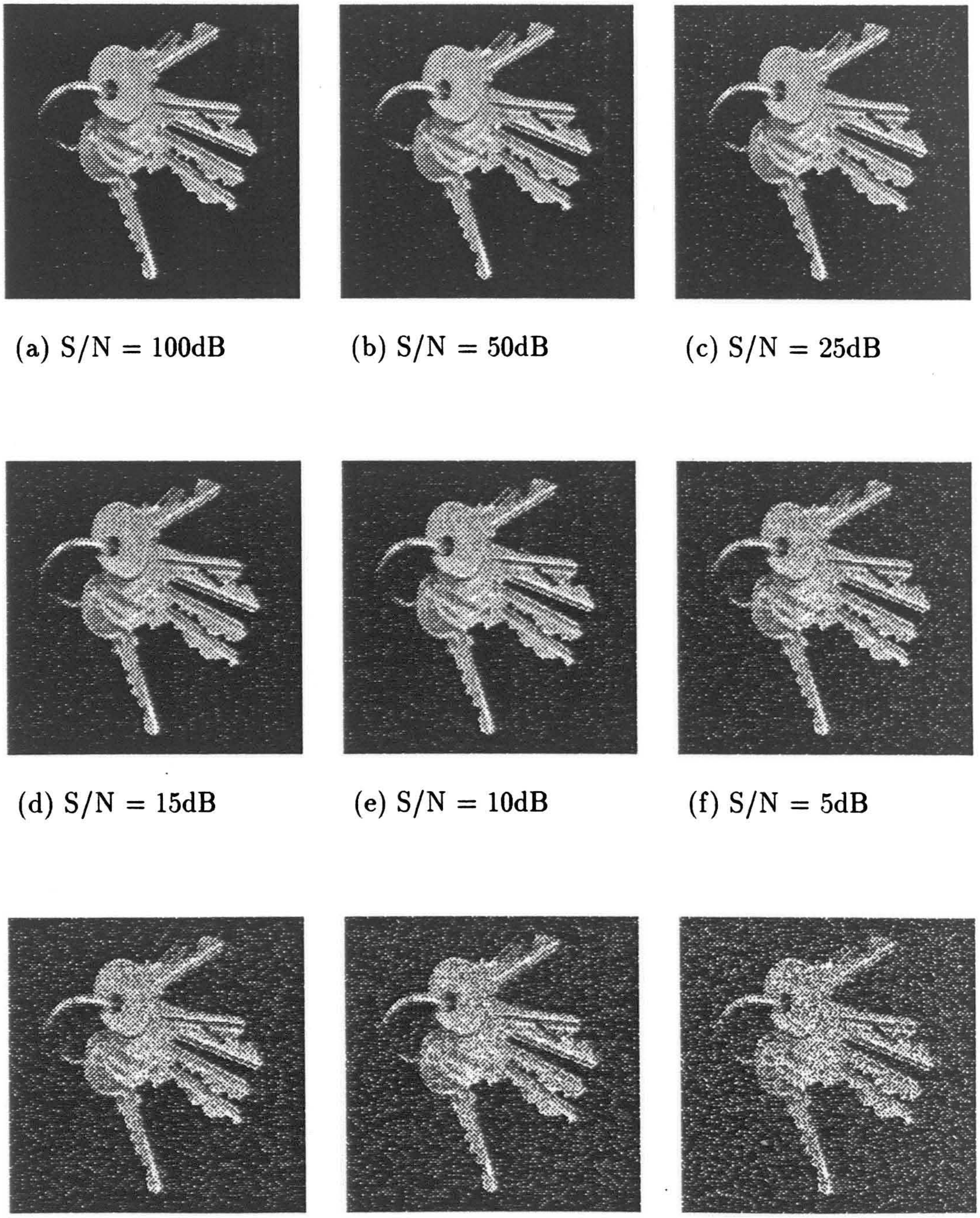
(a) Original

(b) (-10, 10) Translation

(c) 0.5 Scaling

(d) 45° Rotation

Figure 4.7: Transformation used to study the white noise effect



(a) S/N = 100dB

(b) S/N = 50dB

(c) S/N = 25dB

(d) S/N = 15dB

(e) S/N = 10dB

(f) S/N = 5dB

(g) S/N = 3dB

(h) S/N = 2dB

(i) S/N = 1dB

Figure 4.8: A qualitative illustration for the noise added to the image of the keys when studying the effect of white noise on recognition

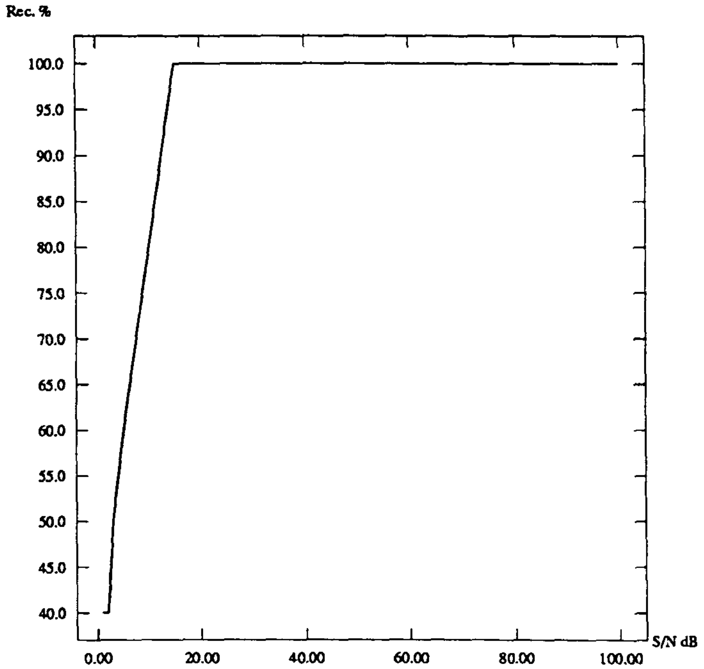


Figure 4.9: Recognition percentage versus S/N

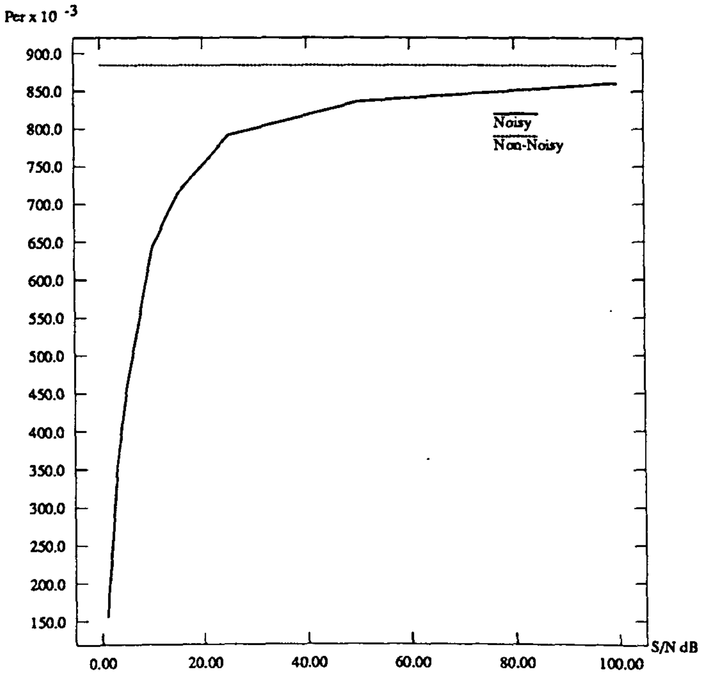


Figure 4.10: Performance measure versus S/N

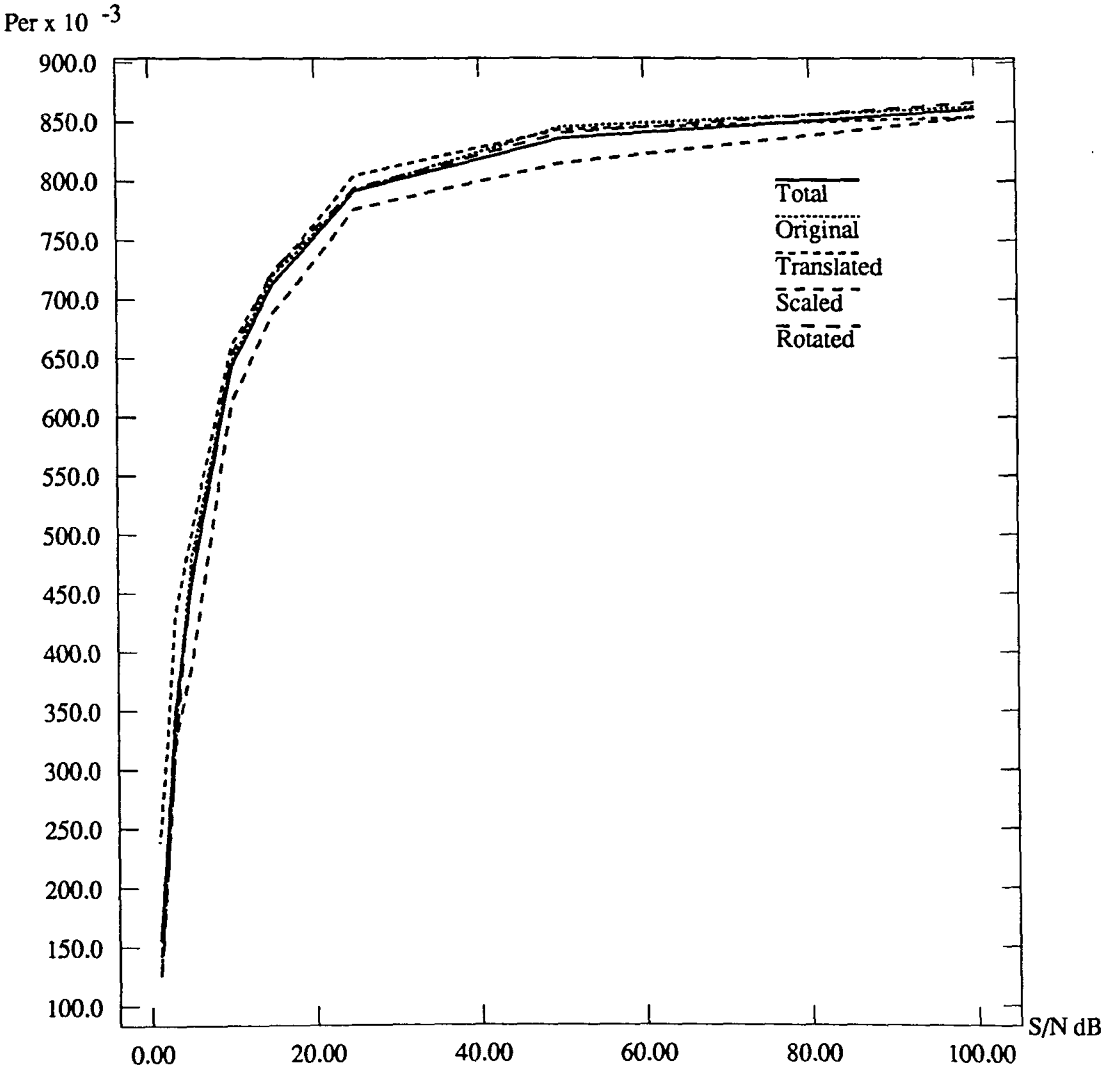
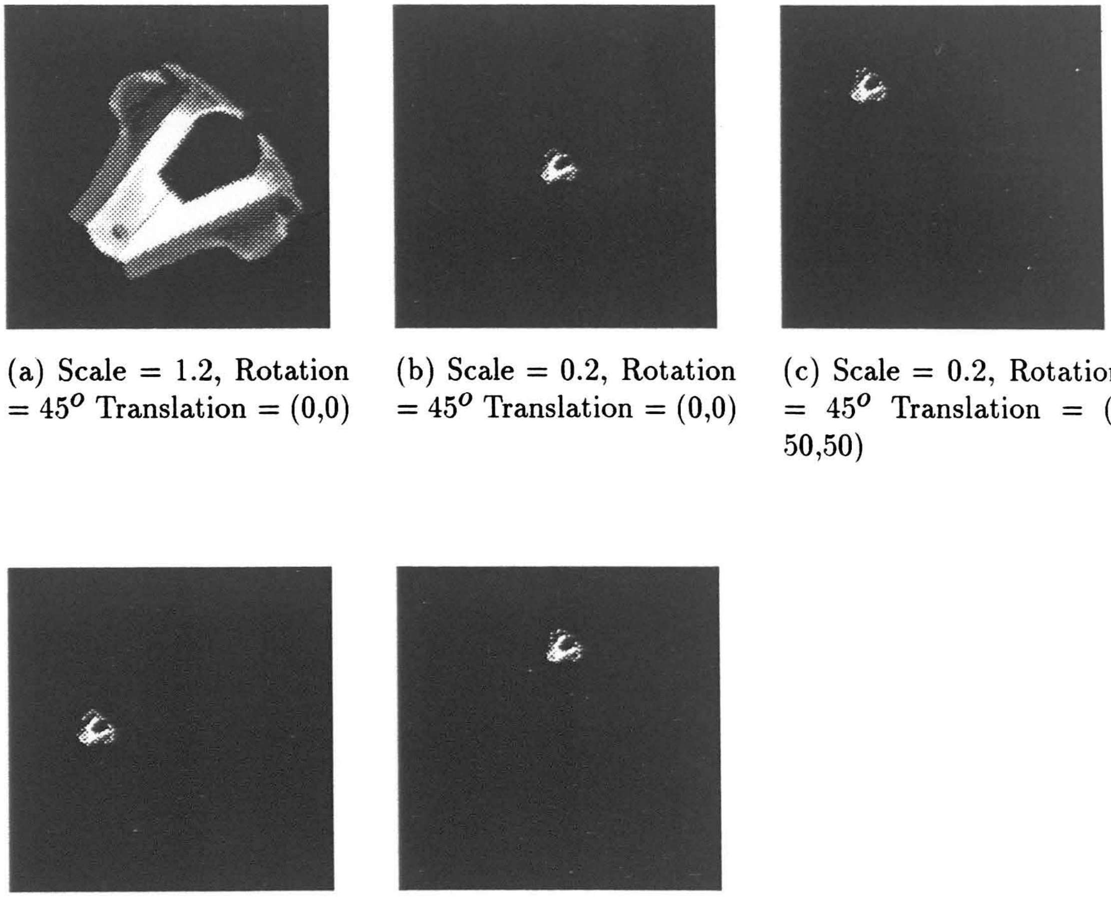


Figure 4.11: Performance measure of each transformed set versus S/N. The curves overlap. This further illustrates the robustness of the feature vector against the transformations and noise



(a) Scale = 1.2, Rotation = 45^o Translation = (0,0)

(b) Scale = 0.2, Rotation = 45^o Translation = (0,0)

(c) Scale = 0.2, Rotation = 45^o Translation = (-50,50)

(d) Scale = 0.2, Rotation = 45^o Translation = (-50,0)

(e) Scale = 0.2, Rotation = 45^o Translation = (0,50)

Figure 4.12: The Combined transformation done to the staple remover

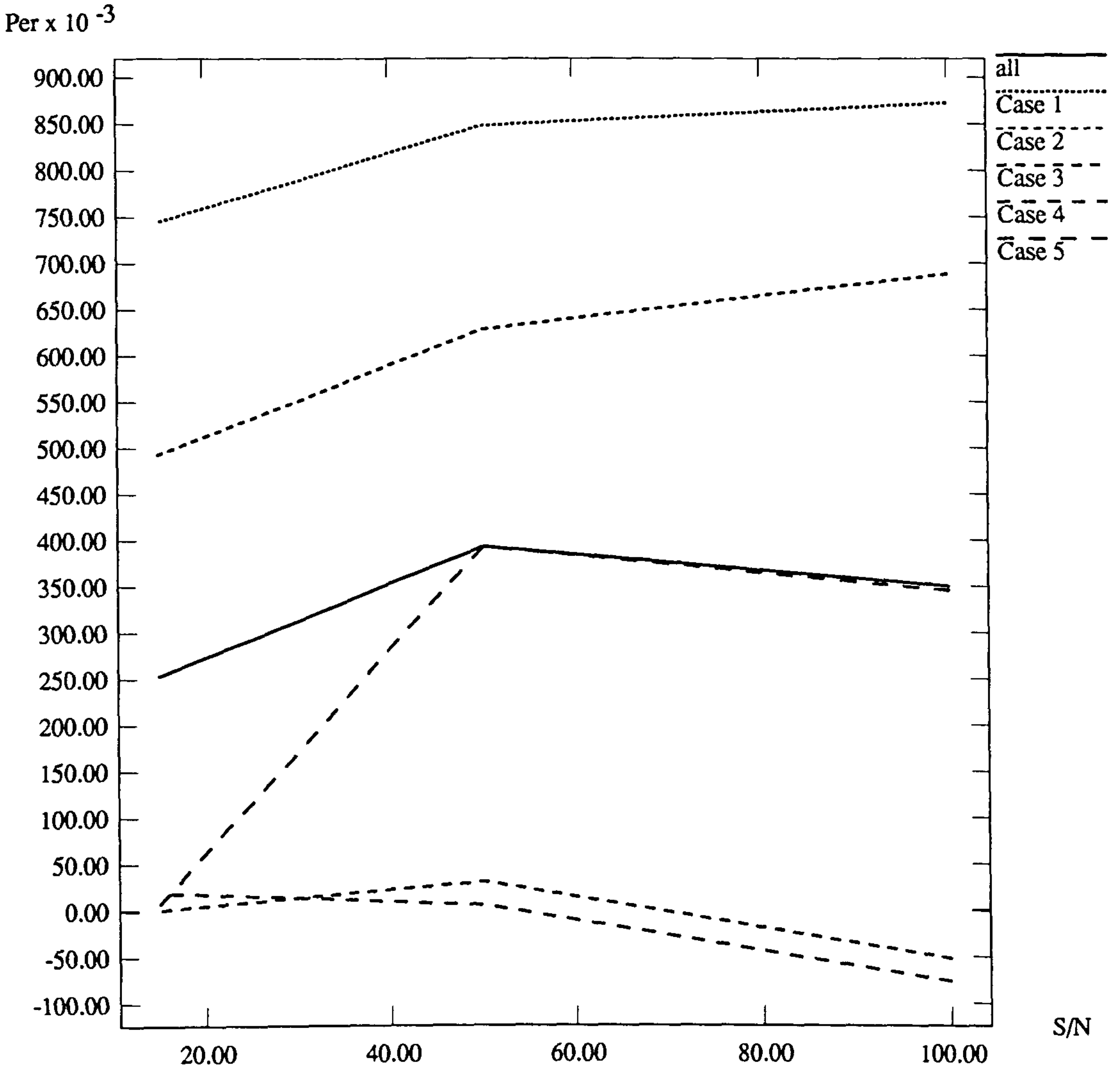


Figure 4.13: Performance measure versus S/N for combined transformations

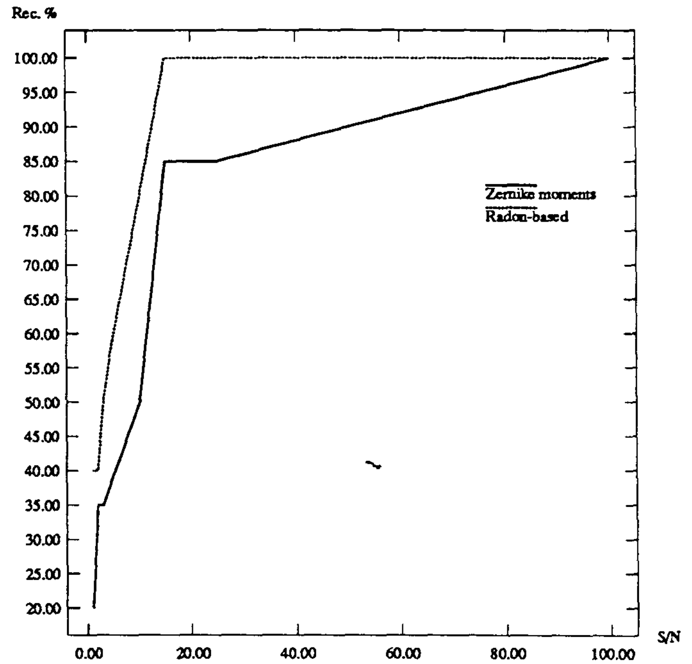


Figure 4.14: Recognition percentage versus S/N for Zernike moments and Radon-based features. As seen, the performance of the projection-based method is better than that of the method of the Zernike moments

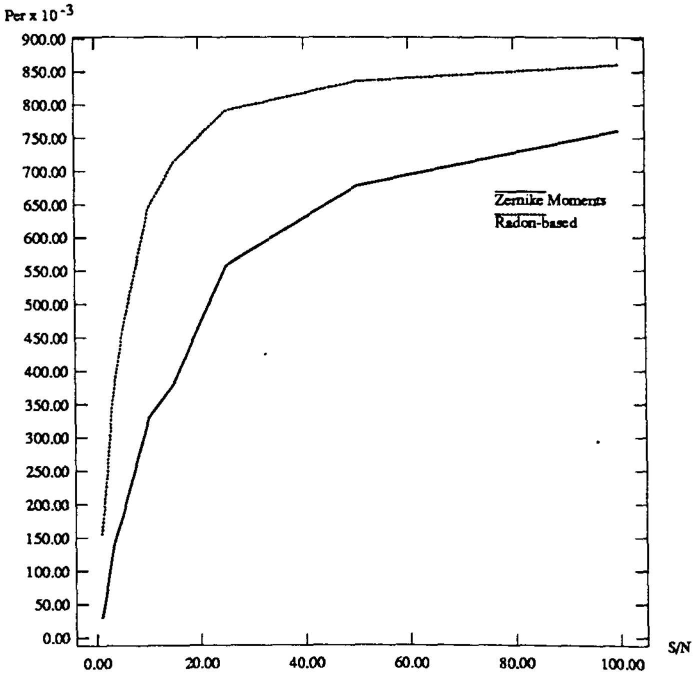


Figure 4.15: Performance measure versus S/N for Zernike moments and Radon-based features. As seen, the performance of the projection-based method is better than that of the method of the Zernike moments

CHAPTER 5. CONCLUSIONS

A discrete Radon transform method for invariant image analysis using artificial neural networks was developed. This method is based on utilizing the projections of the image to achieve invariant features. The singular values of the projection image, constructed by row-stacking each projection, are used as the invariant features. The feature vector is used as input to a back-propagation neural network followed by a maximum-output-selector to classify the image.

A performance function was introduced to evaluate the performance of the recognition system. This performance function gives an output of 1 when there is no ambiguity in classification, an output of -1 when there is no ambiguity in misclassification, and an output of 0 when it is most ambiguous. This function can be used to give an indication of how closely a pattern matches the template the system classified as.

This method is examined and its merits are illustrated. It is also compared with a leading feature extraction method, Zernike moments. The proposed method outperformed the method of Zernike moments in terms of recognition percentage, performance measure, sensitivity to image noise, and number of features used.

The effect of noise was more evident when the objects were shrunk and translated. The projections were not centered correctly. This resulted in poor recognition rates. This affected all methods that center the image to introduce translation invariance in the features. Low-pass filtering the projections is proposed to solve this problem. To overcome the problem of deformed projections, the projections are processed before building the matrix and training the system.

The discrete cosine transform (DCT) is a very close approximation to the Karhunen-Loéve expansion of a first order stationary Markov sequence [1]. The Karhunen-Loéve expansion is the same as the SVD transformation for a square matrix except that the Karhunen-Loéve coefficients (the eigenvalues of the matrix) are the square of the singular values. Using the DCT instead of the SVD will result in a faster recognition system.

The transformation parameters can be calculated. The translation and the scaling factor can be calculated without any significant computation. However, the rotation angle requires calculating any left singular vectors. These parameters along with the membership value, which is calculated using the performance function proposed, can be used to confirm the recognition of an object. They can also be used in building an intelligent vision system that can recognize two dimensional objects invariantly and perform some actions accordingly.

This method can be extended to three dimensional object recognition where the system is trained using different views of the objects. The object is then classified as the template that has the closest view to that of the tested object.

BIBLIOGRAPHY

[1] A.K. Jain, *Fundamentals of digital image processing.* Prentice Hall, Englewood Cliffs, NJ,1989.

[2] J.T. Tou and R.C. Gonzalez, *Pattern recognition principles.* Addison-Wesley, Reading, MA, 1974.

✓[3] E. Wong and J.A. Steppe, *Invariant recognition of geometric shapes.* Methodologies of pattern recognition, Edited by S. Watanabe, Academic press, New York, NY, pp. 535-546, 1969.

✓[4] W. Pitts and W.S. McCulloch, *How we perceive universals: The perception of auditory and visual forms.* Bulletin of Mathematical Biophysics, vol. 9, pp. 127-147, 1947.

✓[5] M.K. Hu, *Visual pattern recognition by moment invariants.* IRE Transactions on Information Theory, vol. IT-8, pp. 179-187, February 1962.

✓ [6] R.Y. Wong and E.L. Hall, *Scene matching with invariant moments.* Computer Graphics and Image Processing, vol. 8, pp. 16-24, 1978.

[7] C.H. Teh and R.T. Chin, *On image analysis by the methods of moments.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 4, pp. 496-512, July 1988.

✓[8] S.J. Perantonis and P.J.G. Lisboa, *Translation, rotation, and scale invariant pattern recognition by higher-order neural networks and moment classifiers.* IEEE Transactions on Neural Networks, vol. 3, no. 2, pp. 241-251, March 1992.

[9] E. Persoon and K.S. Fu, *Shape Discrimination using Fourier descriptors.* IEEE Transactions on Systems, Man, Cybernatics, vol. SMC-7, no. 3, pp. 170-179, March 1977.

[10] S.S. Udpa and W. Lord, *A Fourier descriptor classification scheme for differential probe signals.* Materials evaluation, vol. 42, pp. 1136-1141, August 1984.

✓[11] E. Barnard and D. Casasent, *Invariance and neural nets.* IEEE Transactions on Neural Networks, vol. 2, no. 5, pp. 498-508, September 1991.

[12] R. Hecht-Nielsen, *Neurocomputing.* Adisson-Wesley Publishing Company, Reading, MA, 1990.

[13] K. Fukushima and N. Wake, *Handwritten alphanumeric character recognition by the neucognitron.* IEEE Transactions on Neural Networks, vol. 2, no. 3, pp. 355-365, May 1991.

✓[14] T. Maxwell and CL. Giles, *Transformation Invariance Using High Order Correlation in Neural Net Architectures.* Proc. IEEE International Conference on Systems, Man and Cybernetics, Atlanta, Georgia, pp. 627-632, October 1986.

✓ [15] S. Kollias, A. Stafylopatis and A. Tirakis, *Performance of Higher Order Neural Networks in Invariant Recognition.* Neural Networks: Advances and Applications, E. Gelenbe (Editor), Elsevier Science Publisher B.V (North-Holland), pp. 79-108, 1991.

✓[16] B. Widrow, R.G. Winter, and R.A. Baxter, *Layered neural nets for pattern recognition.* IEEE Transactions on Acoustics, Speech, Signal Processing, vol. 36, no. 7, pp. 1109-1118, 1988.

[17] M. Fukumi, S. Omatu, F. Takeda, and T. Kosaka, *Rotation-invariant neural pattern recognition system with application to coin recognition.* IEEE Transactions on Neural Networks, vol. 3, no. 2, pp. 272-279, March 1992.

✓[18] L.W. Chan, *Neural networks for collective translational invariant object recognition.* To appear in the International Journal of Pattern Recognition and Artificial Intelligence.

✓ [19] L.W. Chan and F. Fallside, *Inhibitory networks for multiple object recognition.* Proceeding of the International Neural Network Conference, Paris, France, vol. 1, pp. 97-90, 1990,

✓ [20] J.L.C. Sanz, E.B. Hinkle, and A.K. Jain, *Radon and projection transform-based computer vision.* Springer-Verlag, Berlin, 1988.

✓[21] Y.H. U and G.M. Flachs, *Structural extraction by projections.* Region V IEEE Conference digest on electrical engineers for this decade, Austin, Texas, pp. 15-19, April 1976.

[22] A.C. Kak and M. Slaney, *Principles of computerized tomographic imaging.* IEEE Press, New York, 1988.

✓[23] S.R. Deans, *The Radon transform and some of its applications.* John Wiley & Sons, New York, 1983.

[24] J. Radon, *Translation of Radon's 1917 paper.* Translated by R. Lohner, School of Mathematics, Georgia Institute of Technology, The Radon transform and some of its applications by Stanley R. Deans, John Wiley & Sons, New York, 1983.

[25] I.M. Gelfand and S.G. Gindikin, *Mathematical problems of tomography.* Translations of mathematical monographs, American mathematical society, vol. 81, Moscow, 1990.

[26] G.M. Henkin and A.A. Shananin, *Bernstein theorems and the Radon transform. Application to the theory of production functions.* Translations of mathematical monographs, American mathematical society, vol. 81, Moscow, 1990.

[27] H. Al-Yousofi and S.S. Udpa, *Recognition of Arabic characters.* IEEE Transactions on Pattern Analysis Machine Intelligence, vol. 14, no. 8, pp. 853-857, July 1992.

[28] B. Kelley and V.K. Madisetti, *The fast discrete Radon transform.* Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, IEEE Press, Piscataway, 1992.

✓ [29] P. Maass, *Singular value decomposition for Radon transforms.* ,Mathematical methods in tomography, Editors G.T. Herman, A.K. Louis, and F. Netterer, Lecture notes in mathematics 1497, Springer-Verlag, pp. 6-14, 1991.

✓ [30] C. Lau, *Neural networks theoretical foundations and analysis.* IEEE Neural Networks Council, New York, 1992.

[31] B. Widrow and M.A. Lehr, *30 years of adaptive neural networks: perceptron, madaline, and back-propagation.* IEEE Proceedings , vol. 78, no. 9, pp. 1415-1442, September 1990.

[32] R.P. Lipman, *An introduction to computing with neural nets.* IEEE ASSP Magazine, pp. 4-22, April 1987.

✓[33] M.K. Hu, *Visual pattern recognition by moment invariants.* IRE Transactions on Information Theory, vol. IT-8, pp. 179-187, February 1962.

[34] R.Y. Wong and E.L. Hall, *Scene matching with invariant moments.* Computer Graphics and Image Processing, vol. 8, pp. 16-24, 1978.

[35] A. Khotanzad and Y. H. Hong, *Invariant image recognition by Zernike moments.* Transactions on Pattern Recognition and Machine Intelligence, vol. 12, no. 5, pp. 489-497, May 1990.

[36] L.A. Zadeh, *Fuzzy sets.* Information and Control, vol. 8, pp. 338-353, 1965.

[37] O.R. Mitchell and S.M. Lutton, *Segmentation and classification of targets in FLIR imagery.* SPIE, vol. 155, Image Understanding Systems & Industrial Applications, pp. 83-90, 1978.

[38] A.P. Reeves and A. Rostampour, *Shape analysis of segmented objects using moments.* Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing, pp. 171-174, Aug. 25-28, 1981.

✓ [39] P.R. Madhvapathy, *Pattern recognition using simple measures of projections.* Master of science thesis, Colorado State University, Fall 1986.

[40] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical recipes in C, the art of scientific computing.* Cambridge University Press, 1988.

[41] W.K. Pratt, *Digital image processing.* Wiley-Interscience, New York, 1991.

ACKNOWLEDGMENTS

Praise be to God, Most Gracious, Most Merciful.

I would like to thank my major professor, Dr. J.F. Doherty, for his continuous guidance, advice, and friendship. I also thank Dr. J. Basart, Dr. S.S. Udpa, Dr. H.S. Hung, and Dr. P. Sherman for serving on my committee.

I would also like to thank Dr. I. Zabalawi, Dr. H. Hamdan, Dr. M.K. Abdialaziz, and Dr. M. Maqousi of University of Jordan for their help and advice.

I wish to thank USIA, AMIDEAST, and the Fulbright Scholarship for their financial support, help and advice. I am also thankful to the University of Jordan for their financial support.

Finally, I wish to express my gratitude to my family, Khalid, Laila, Obayda, Moa'ath, and Eyas for their love, patience and continuous support.

APPENDIX DERIVATIONS

Rotation invariance

To derive the relation between singular values of the Radon transform of the rotated image and the image itself. The following results from the construction of the projection matrix

$$g(s,\theta) = \sum \sigma_k u_k(\theta) w_k(s) \tag{A.1}$$

$$\int_s g(s,\theta) w_k(s) ds = \sigma_k u_k(\theta) \tag{A.2}$$

$$\int_\theta g(s,\theta) u_k(\theta) d\theta = \sigma_k w_k(s) \tag{A.3}$$

where

σ_k	Singular values of the Radon transform
$w_k(s)$	Right singular vector spanning the s-axis of the projections
$u_k(\theta)$	Left singular vector spanning the angle , θ, of the projections
k	runs over the rank of the projection-matrix.

The Inverse Radon transform theory [1] states that

$$f(r,\phi) = \frac{1}{2\pi^2} \int_0^\pi \int_s \frac{[(\partial g/\partial s)(s,\theta)]}{r\cos(\theta-\phi)-s} ds d\theta \tag{A.4}$$

Substituting Equation (A.2) into Equation (A.4) results in

$$
\begin{aligned}
f(r,\phi) &= \frac{1}{2\pi^2}\int_0^{\pi}\int_s \frac{\left[(\partial/\partial s)\left\{\sum_k \sigma_k u_k(\theta) w_k(s)\right\}\right]}{r\cos(\theta-\phi)-s} ds d\theta && \text{(A.5)}\\
f(r,\phi) &= \sum_k \frac{\sigma_k}{2\pi^2}\int_0^{\pi} u_k(\theta) d\theta \left\{\int_s \frac{\left[\partial w_k(s)/\partial s\right]}{r\cos(\theta-\phi)-s} ds\right\} && \text{(A.6)}
\end{aligned}
$$

And this proves 3.5.

Scale invariance

To derive the relation between the singular values of the Radon transform of the scaled image and the image itself.

Utilizing the scaling property of the Radon transform and the singular value expansion of the Radon transform of the scaled image.

$$
\begin{aligned}
g_S(s,\theta) &= \frac{g(\alpha s,\theta)}{|\alpha|} && \text{(A.7)}\\
&= \sum_k \frac{\sigma_k}{|\alpha|} u_k(\theta) w_k(\alpha s) && \text{(A.8)}\\
&= \sum_k \frac{\sigma_k}{\alpha^{3/2}} u_k(\theta)\left[\sqrt{\alpha} w_k(\alpha s)\right] && \text{(A.9)}
\end{aligned}
$$

This result is the SVD of the Radon transform of the scaled image. It can be shown that

$$
\begin{aligned}
\int \frac{g(\alpha s,\theta)}{\alpha}\sqrt{\alpha} w_k(\alpha s) ds &= \frac{\sigma_k}{\alpha^{3/2}} u_k(\theta) && \text{(A.10)}\\
\int \frac{g(\alpha s,\theta)}{\alpha} u_k(\theta) d\theta &= \frac{\sigma_k}{\alpha^{3/2}}\sqrt{\alpha} w_k(\alpha s) && \text{(A.11)}
\end{aligned}
$$

This result proves Equation (3.8).