VIEW INVARIANT PLANAR-OBJECT RECOGNITION

A Thesis

Presented in Partial Fulfillment of the Requirements for
the Degree Master of Science in the
Graduate School of The Ohio State University

By

Panu Srestasathiern, B.Eng.

* * * * *

The Ohio State University

2008

Master's Examination Committee: Approved by

Asst. Prof. Alper Yilmaz, Adviser

Prof. Anton F. Schenk

Adviser
Graduate Program in
Geodetic Science &
Surveying
ABSTRACT

In many photogrammetry and computer vision applications, there ultimate goals is to recognize objects of interest. Various framework for object recognition problem have been developed. Among many framework, geometric invariances have been proven to be an efficient way to recognize object under geometric transformation i.e. affine transformation.

Motivated by geometric invariance framework, we propose a new method for recognizing the isolated planar object under the circumstance of geometric transformation. Namely, we assume that object shape is deformed by projective transformation. Firstly, we present a new object's shape representation based on an assumption that the boundary of the object's shape can be approximately represented by a set of piecewise conics. Secondly, a new projective invariant feature is derived based on the distribution of the projective relations between the conic pairs, which are estimated from the objects shape. We hypothecate that two objects of the same type, which are viewed from different viewpoints generate similar histograms, such that the distance between these two histogram is smaller than the histograms generated from other object types. The proposed method has shown promising performance our shape database in which object's shapes are deformed by projective transformation.
This page dedicated to my father, mother and sister
ACKNOWLEDGMENTS

First of all, I wish to express my heartfelt gratitude to my advisor, Dr. Alper Yilmaz for encouragement throughout this research and all of the time and energy he spent in meeting with me and discussing my research. His innovative teaching style and perspectives on vision science have greatly facilitated my achievements. Also, I would like to thank Professor Anton F Schenk for serving on Masters Examination Committee.

I would like to thank the faculty in the department of Civil and Environmental Engineering and Geodetic Science and department of earth science for making the Ohio state university one of the best place to study geodetic science. I would like to thank Sudhagar Nagarajan, Gabor Barsai, Kyungjin Park, Po-Lun (Ryan) Lai, and other members of photogrammetric computer vision laboratory, and all of my friends in the Ohio state university for making the time I have studied here one of the best time in my life.

I would like to acknowledge the financial support from the Thai Government. This support gave me the most valuable chance to pursue my graduate education. Last but not least, I would like to thank my parents for their for their unlimited support, encouragement, and belief in me throughout my studies at the Ohio State University.
VITA

February 1982 .................... Born - Bangkok, Thailand

May 2004 .......................... B.Eng. Electrical Engineering,
Chulalongkorn University,
Bangkok, Thailand

September 2006 - present ........ Graduate Student, Department of Civil
& Environmental Eng. and Geodetic
Science, The Ohio State University

PUBLICATIONS

Research Publications


FIELDS OF STUDY

Major Field: Geodetic Science and Surveying
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>iv</td>
</tr>
<tr>
<td>Vita</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>ix</td>
</tr>
<tr>
<td>List of Figures</td>
<td>x</td>
</tr>
<tr>
<td>Chapters:</td>
<td></td>
</tr>
<tr>
<td>1.   Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1  Overview</td>
<td>1</td>
</tr>
<tr>
<td>1.2  Related works</td>
<td>4</td>
</tr>
<tr>
<td>1.3  Motivation and contributions</td>
<td>10</td>
</tr>
<tr>
<td>1.4  Organization of thesis</td>
<td>11</td>
</tr>
<tr>
<td>1.5  Notation</td>
<td>12</td>
</tr>
<tr>
<td>2.   Projective Geometry: Basic Definitions</td>
<td>13</td>
</tr>
<tr>
<td>2.1  Projective Space and Homogeneous Coordinate</td>
<td>15</td>
</tr>
<tr>
<td>2.2  Hyperplane and Quadric</td>
<td>18</td>
</tr>
<tr>
<td>2.3  Projective Transformations and Homography</td>
<td>19</td>
</tr>
<tr>
<td>2.4  Duality Principle</td>
<td>21</td>
</tr>
</tbody>
</table>
3. Two-Dimensional Projective Geometry ........................................... 24
   3.1 Points and lines ................................................................. 25
      3.1.1 Intersection of lines and line joining points .................... 26
      3.1.2 Point and line at infinity ............................................ 27
   3.2 Conic section ................................................................. 30
      3.2.1 The representation of a conic section ............................ 31
      3.2.2 Dual conics ............................................................ 32
      3.2.3 Circular point .......................................................... 33
      3.2.4 The pole-polar relationship ........................................ 34
   3.3 2D Projective transformation ............................................. 36
      3.3.1 Representation of transformation .................................... 36
      3.3.2 Transformation of lines and conics ................................. 37
   3.4 Classes of transformations ................................................. 38
      3.4.1 Class 1: Euclidean transformations .................................. 39
      3.4.2 Class 2: Similarity transformations ................................ 39
      3.4.3 Class 3: Affine transformations ..................................... 40
      3.4.4 Class 4: Projective transformations ................................ 41
   3.5 Epipolar geometry and the fundamental matrix ......................... 41
      3.5.1 Epipolar geometry ..................................................... 42
      3.5.2 Fundamental matrix F ............................................... 44

4. Projective invariance ......................................................... 46
   4.1 Invariance of points and straight lines ................................ 47
      4.1.1 Cross ratio in one dimension ....................................... 48
      4.1.2 Cross ratio in n dimension ......................................... 50
      4.1.3 \(p^2\) invariance in one dimension ................................. 52
      4.1.4 \(p^2\) invariance in higher dimensions ............................. 57
   4.2 Invariances of smooth (non-algebraic) curve .......................... 58
   4.3 Joint conic invariants ..................................................... 63
      4.3.1 The pair of quadratic forms invariance ............................ 64
      4.3.2 The derivation of the pair of conics invariance ............... 66

5. View invariant object recognition ......................................... 74
   5.1 Shape representation ....................................................... 75
   5.2 Projective invariant feature .............................................. 81
   5.3 Shape matching ............................................................... 84

vii
6. **Experimental Results** ................................................................. 87
   6.1 Data set ............................................................................. 87
   6.2 Use of only non-isotropic or isotropic scaling ...................... 91
   6.3 Use of both scalings in the proposed method ....................... 94
   6.4 Experiment with biological vision ........................................ 103
      6.4.1 The first experiment on biological vision ...................... 103
      6.4.2 The second experiment on biological vision .................. 104

Bibliography .............................................................................. 108
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Characteristics of the four different geometries [16, pagep. 11]</td>
<td>14</td>
</tr>
<tr>
<td>6.1</td>
<td>Test images</td>
<td>88</td>
</tr>
<tr>
<td>6.3</td>
<td>Matching result. The bold face number indicate correct matching, while italic number indicate false matching.</td>
<td>95</td>
</tr>
<tr>
<td>6.4</td>
<td>Performance of the proposed recognition method</td>
<td>96</td>
</tr>
<tr>
<td>6.5</td>
<td>Cluster result</td>
<td>97</td>
</tr>
<tr>
<td>6.6</td>
<td>Clustering performance</td>
<td>101</td>
</tr>
<tr>
<td>6.7</td>
<td>Grouping result from the examinees who did not ask any equation during the experiment.</td>
<td>105</td>
</tr>
<tr>
<td>6.8</td>
<td>First incorrect image grouping</td>
<td>106</td>
</tr>
<tr>
<td>6.9</td>
<td>Second incorrect image grouping</td>
<td>107</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure | Page |
-------|------|
1.1 | The definition of object recognition can be separated into two category. In this example, the problem can be stated as recognizing my cat among other or distinguishing between cats and tigers. | 3 |
2.1 | Projection from $\mathbb{P}^3$ to $\mathbb{P}^2$. The point $M$ in $\mathbb{P}^3$ is projected to the projective (image) plane $\mathbb{P}^2$ through the optical center of a pin-hole camera. | 16 |
2.2 | Transformation of coordinates between Euclidean and projective spaces. | 17 |
2.3 | An example of projective transformation. The object space and image planes are related by projection matrices $P_1$ and $P_2$. If we take the image of the ground plane in object space, i.e. $z = 0$, the relation between two image planes $\Pi_1$ and $\Pi_2$ can be represented by planar homography which is represented by a $3 \times 3$ matrix. | 20 |
2.4 | The example of duality principle. The statement “the joining of two points define a line” is dual to the statement “the intersection of two lines define a point”. | 22 |
2.5 | The example of duality element. *Left:* the dual element of a point $x$ in $\mathbb{P}^2$ is a line $l$. *Right:* the dual element of a point $X$ in $\mathbb{P}^3$ is a plane $A$. The distance from origin to the dual element is the inverse of the distance from origin to the original element [27, page. 41]. | 23 |
3.1 | The mapping of points at infinity to the image plane. | 27 |
3.2 | The intersection of two parallel lines. Image of the points on two parallel lines intersect at the same vanishing point. | 29 |
3.3 The line at infinity. A pair of parallel lines will intersect at infinity. When we connect two vanishing points together, we will get a line called vanishing line on which all vanishing points lie. .......................... 29

3.4 The three types of non-degenerate conic. *Left* Ellipse; *Middle* Parabola; *Right* Hyperbola. .............................................................. 30

3.5 The duality between conic $C$ and $C^*$. A line $l$ tangent to the conic $C$ satisfy $l^T C^* l = 0$. In other words, the envelope of $C$ is the lines represented by the points of $C^*$. ................................. 32

3.6 The intersection of two conics. (a) Four points of intersection. (b) Two points of intersection. ................................................................. 33

3.7 The pole-polar relationship. (a) Pole lies outside the conic. (b) Pole lies on the conic. (c) Pole lies inside the conic. ................................. 35

3.8 Point correspondence problem. .................................................. 42

3.9 The illustration of terminology in epipolar geometry .................. 43

3.10 The derivation of the fundamental matrix $F$. ......................... 45

4.1 Cross ratio in one dimensional projective space. The right figure illustrate the perspective projection shown in the left image as a general projective transformation on between two lines. .......................... 49

4.2 Cross ratio in two dimensional projective space. The configuration of five points in projective plane is used to compute the cross ratio. Any nonsingular $3 \times 3$ matrix $H$ defines a planar projective transformation of the configuration. Although the configuration of five points is changed according to the projective transformation, the cross ratio is preserved. 52

4.3 Permutation problem in one dimensional projective space. After the points on a line are transformed, if we do not know the correct correspondence of points before and after transformation, the cross ratio value will differ from what it should be. ............................... 53

4.4 The invariants $J_1(\lambda)$ (solid line) and $J_2(\lambda)$ (dashed line). ................................. 55
4.5 The invariants $J_5(\lambda)$ (dash-dot line), $J_6(\lambda)$ (solid line) and $J_7(\lambda)$ (dash line). .......................... 56

4.6 The bounded invariants $J_8(\lambda)$. ........................................ 56

4.7 Preservation of bi-tangency and point of inflection. ....................... 61

4.8 Detection of other distinguished points. Two examples are shown here.  
Left: Ray casting approach. Right: Connection of two inflection points. 62

4.9 Images of two coplanar conics on plane $\Pi$. There images on view 1 and 2 are related by the homography $H_{3\times3}$ induced by the plane $\Pi$. The joint conic invariants derived from images of the two conic in either view 1 or 2 are independent from the homography. ................. 64

4.10 A point $p_1$ in the plane is the pole of polar $l_{polar}$ with respect to the conic $C_1$. Then, the pole $p_2$ of the line $l_{polar}$ with respect to the conic $C_2$ lies either inside or outside the conic $C_2$. (a) Pole $p_2$ lies outside the conic $C_2$. (b) Pole $p_2$ lies outside the conic $C_2$. .................. 67

4.11 The geometric configuration corresponding to the eigenvalue problem, i.e. the configuration when $p_1 = \lambda p_2$. Using the eigenvalue of the $C_1^{-1}C_2$ as the poles, the obtained polar is common for both conics. (a) The pole lines outside of both conics. (b) The pole lies outside one of the conics. .......................... 67

4.12 Self-polar triangle. Each side of it is the polar line of the opposite vertex. ..................................................... 68

5.1 Edge detection. Left: Silhouette image of a shape (French curve) before applying edge detection. Right: Detected edge after applying Canny edge detector to the silhouette image on the left. .................. 76

5.2 After the edge was extracted from the silhouette image, we sample $N$ points from the edge. ........................................... 77

5.3 Conic fitting. The conic $C_i$ comes from the fitting conic to the contour by using the $i^{th}$ point and its immediate neighborhood. .................. 77

xii
5.4 The example of two scaling methods. The isotropic scaling preserved the object shape. While non-isotropic scaling changes the object shape and make the sampled points dense at high curvature regions. .... 79

5.5 The result from fitting conics to the contour. In this example, 256 points was sampled from the contour, as shown in black dot. Each conic was fitted to the contour using 9 points and shown in red line. ... 80

5.6 An example of shape that can make an ambiguity in our approach. .. 81

5.7 Two conics $C_i$ and $C_j$ approximated from the contour boundary define an invariance. ....................................................... 82

5.8 Histogram of the negative invariant group, $\log |D^-|$, (Left) and the positive invariant group, $\log D^+$, (right) for the shape given in Figure 5.1, and both of them are log-scaled histogram. ............. 85

6.1 Matching result from using only isotropic or non-isotropic scaling. ... 93

6.2 Shape alignment ....................................................... 102

6.3 Some question arose during the first experiment. The images shown here are for the question "Are these two dogs considered as the same object?" ....................................................... 104

xiii
CHAPTER 1

INTRODUCTION

In science the important thing is to modify and change one’s ideas as science advances.

Herbert Spencer

1.1 Overview

Many biological vision systems work efficiently, such as butterfly can recognize flower shapes or a child can recognize a cat viewed from different viewpoints. In particular, changes in position, size, orientation or lighting of an object make a little or no difficulty in recognizing the object in human vision. However, the capabilities of computer systems for object recognition are far behind that of biological systems and are limited to only specific objects that are programmed or learned [81, page 1]. It, thus, makes the object recognition an ongoing research effort in the digital photogrammetry and computer vision communities. Moreover, object recognition has been applied in many application including medical diagnosis, homeland security, military. These reasons make object recognition a fascinating research area.
In general, the object recognition is referred to the research of how machines can learn to distinguish objects of interest from their background and assign labels to objects by using a priori information from object models [51]. However, the object recognition problem can be also separated into two categories. That is, object recognition may be referred to the identification of a specific object or labeling the object type. These definition have been used alternatively depending on the applications.

To give an example of the object recognition definition, let us use the problem of recognizing our cat from other cats. In this problem, we are looking for a cat of a specific texture, tail length, size, coat patterns or eye color such that the cat belongs to us in this case. On the other hand, object labeling is more general. For instance, labeling will classify cat examples as persian, turkish van or himalayan. Another labeling can be made for broad class clustering such as by distinguishing cats from tigers in the group of feline, as shown in figure 1.1.

In general, the problem of object recognition consists a number of subproblems:

- Data acquisition
- Data processing
- Feature extraction
- Feature matching

According to the above list of subproblems, the design of an object recognition system starts from the data acquisition which is usually designed based on the application. For example, data used for object recognition may be either gray-scale or color image. The next step is to process data to use for feature extraction.
Figure 1.1: The definition of object recognition can be separated into two category. In this example, the problem can be stated as recognizing my cat among other or distinguishing between cats and tigers.

The expectation from every object recognition system is to recognize objects independent of the variations in size, translation, rotation, viewpoint change or occlusion, make the feature extraction step surprisingly difficult task. Finally, to say that two objects are the same, there features must be matched.

Many research efforts have been expanded in the areas of feature extraction and feature matching. That is, the data acquisition and processings is assumed to be solved. Developing an object recognition system that is invariant to the aforementioned variations can be performed by two approaches:

1. Recognition by using common features and specially designed matching method invariant to shapes, or
2. Recognition by designing special features invariant to variation in the object appearance and using a general matching method

An example for the first approach is the SIFT descriptor [41]. This descriptor used points extracted from an object as its features. To match two features, a specially designed matching method was derived and proved to be invariant to affine transformation. For the second category of approaches, the method proposed in [19] derived a feature based on area moments invariant to affine transformation and matching between two features can be done by comparing between two values of two moments.

1.2 Related works

Since the object recognition problem was presented in photogrammetry and computer vision applications, many researchers have invented many frameworks to solve the problem. However, different frameworks have different drawbacks and advantages. A new framework is proposed to overcome the limitations or improve the performance of other frameworks. In this chapter, we introduce some frameworks in object recognition problem that are widely used.

Planar shape matching based on point correspondence has been used in the literature. In particular, after establishing point correspondence, a shape distance can be used to evaluate similarity [2]. In that study, the shape context has been successfully used to generate a 2-D histogram describing each edge point on object contour. Point correspondence is then generated by the minimization of a $\chi^2$ distance between two shape contexts [2]. Once point correspondence was establish, the similarity distance between two shapes is evaluated by the combination of error between two correspondence points i.e. low similarity distance yields the best match. Mori et al. [49]
improved the performance of shape context by using $L_2$ norm to compare shape contexts which are treated as feature vectors and it was shown that the result of matching is computationally faster than using $\chi^2$ distance. Since the shape context is highly influenced by the clutter, Thayananthan et al. [76] combined the shape context and a figural continuity constraint for improving the performance of the shape matching. Shape matching is then perform by employing the Viterbi's algorithm to minimize the cost function consisting of shape contexts, continuity costs and curvature costs.

Since some matching approaches involve finding point correspondences between two object contours, many researchers applied optimization techniques to find such correspondence. In other words, optimization techniques were employed to determine the identities of shapes. As a consequence, the shape recognition problem is converted to an optimization problem [78]. Dynamic programming has been applied find optimal correspondence or alignment between two shapes by using their intrinsic properties of the curve i.e. curvature [67, 40, 66, 59]. In [83], the correspondence problem was treated as a Quadratic Assignment Problem (QAP) and a specific Ant Colony Optimization algorithm was derived to solve QAP-based correspondence problem. Among many search technique, Genetic Algorithms have emerged as a promising solution for correspondence problem due to their abilities to find optimal solution in a large solution space and maintain of possible solution [79, 78, 73]. Yuen et al. [91] proposed an object recognition method based on the use of particle swarm optimization for recognizing broken boundary objects. Treated as a combinatorial optimization problem, the shape matching and recognition approach based on the combination of discrete particle swarm optimization and fuzzy theory was proposed in [12].

Spectral method is one of the most popular method in shape recognition. Since a shape can be represented by close contour, the coordinate of contour points can be represented by two 1D periodic signals and the frequency content of the coordinate signals is used as shape descriptors. Fourier transform is one of the popular method for obtaining the frequency content from a coordinate signal. It decomposes the coordinate signal into the infinite orthogonal basis functions i.e. cosine and sine functions. A set of Fourier coefficient is then used as a descriptor of a close curve and matching between two shapes [93, 50, 35, 92]. Invariance from starting point on shape boundary is one of the advantages of this method. Although the Fourier descriptors have been widely used in shape recognition problem, the use of phase components has been mostly ignored to generate the Fourier shape descriptors. The phase components of Fourier spectrum were first applied in shape matching in [18]. Fang and Lin [14] applied the cosine transform, an offspring of the Fourier transform, to decompose a set of Synthesized Feature Signals (SFS) from object boundary. After that affine invariant function for the shape recognition is then computed from SFS. Wavelet is also applied to derive invariant signature. For example, the dyadic wavelet was used for affine invariant in [33, 63]. In [34], Kunttu et al. combined the advantages of wavelet and Fourier transform to improve the performance of Fourier descriptor. Such method is called called multi-scale fourier descriptor. It is constructed by applying the Fourier transform to complex wavelet coefficient of the shape boundary.

A shape can be also represented by its skeleton. In the skeleton-based frameworks, the skeletons of two shape are extracted and features extracted from their skeletons are then used to perform shape matching [25]. The skeleton of an object can be extracted by using Blum's medial axis transform which is defined as the loci
of the centers of all maximal inscribed circles. In [28, 25], the topology of individual
nodes or edge in a skeleton such as the relationship with each neighbor, the length
of the edges, bifurcation angle, and the nodes connected neighbors is captured from
the skeleton and then used in matching process. Di Ruberto [64] proposed a shape
recognition method based on the use of attributed skeletal graph which is the medial
axis characteristic points, and matching two shapes can be done by graph matching
algorithm. The shock graph or shock three, which is a medial axis derived from the
locus of singularities (shocks) obtained during evolution of the Blum's grassfire, was
used by Sebastian et al. in [68] for shape matching by editing their shock graphs.

Motivated by partial differential equation describing physical phenomena, the
Helmholtz equation is employed to defined shape descriptor that is invariant to ro-
tation, scaling, translation in [32, 97]. The shape descriptor of a shape \( u \) defined as
a region in \( \mathbb{R}^2 \) are the eigenvalues \( \lambda \) of the elliptic equation \( \nabla^2 u - \lambda u = 0 \) with the
Dirichlet's condition and it was proved in [97] that the feature induced by eigenvalues
are invariant to affine transformation. In [21], shape descriptors based on the use of
Poisson equation is derived. In particular, the shape descriptor is the solution of the
Poisson equation subject to Dirichlet boundary conditions. Some of the shape prop-
terties can then be extracted from the solution of Poisson equation such as skeleton
or corner point which can be applied for shape matching. The Laplace equation was
exploited for establishing the one-to-one correspondence between two shape in [60].
The distance between corresponding points is then used to evaluate the dissimilarity
between two shape.

Invariant signature is frequently used to represent a shape. In particular, different
types of invariant signature are specific to a group of transformation i.e. projective
transformation and matching between two shapes is done by matching between two signature curves. In [86], Weiss introduced a framework on object recognition based on the used of differential invariance and this framework has been used thoroughly [29, 95, 23, 57]. Particularly, derivatives at each point on shape boundary is used to derive the shape invariant signature in this framework. An advantage of using differential invariants is that they can perform matching of partial occluded shape due to the local property of the signature [5]. However, the drawback of differential invariant is the requirement of high order derivative computation which is not numerically feasible with the presence of noise.

In order to diminish the effect of noise, the "semi-differential invariances" are introduced [48]. Furthermore, Weiss [87] proposed a differential invariant signature resisting the effect of noise. In order to improve the numerical performance of differential invariance, Calabi et al. [6] used the combination of first order derivative and curvature to derive a discrete approximation scheme and it was shown to be invariant to Euclidean and equi-affine transformation. However, Boutin [4] demonstrated that the numerical scheme proposed by Calabi et al. in [6] is incorrect, and corrected formulas are given. Scale-space theory has also been used to reduce the noise on shape boundary in [82, 85, 30]. In such framework, the curve is evolved to filter out the noise from curve while still preserves some entity such as dominant point.

The inverse operation of differentiation, integration, has been also used to discover a new invariant signature of a shape boundary. That is, the integral computation is used in this framework. Unlike differential invariances, the integral invariances are not sensitive to noise. Integral invariant can be also defined in terms of area computation. In [43], the integral invariant signature is computed from the area of overlap region
between contour and a circle with center at the point on contour and it is able to provide a natural multi-scale notion by giving the radius of circle as scale parameter. This integral invariance has been improved by Feng et al. [17] to be applicable for affine invariant. Similar to the definition of integration, the summation is also used. In [39], the parameters of affine transformation applied to a shape are eliminate to obtain invariant description by using just summation which insensitive to noise. A statistical approach called moment is also in the integral invariance framework. Chong et al. [9] employed Legendre polynomials to derive a new set of Legendre moments and this descriptor was proved to be invariant to translation and scale of planar shape. In [88], a new moment descriptor specific to a class of transformation was derived by using multiple integral of invariance entities such as distance or area in the case of Euclidean transformation. The significant limitations of moment method are that of the noise sensitive of high-order moments.

Algebraic curve (shape) models and particularly implicit polynomial have been proven to be a powerful technique for object representation and recognition in model based vision [10]. In [56], an object shape is modeled by using implicit polynomials. The invariant descriptor of such shape representation can be computed by employing some classical invariance theory of the implicit polynomial i.e. Gröbner bases, binary forms etc. [10]. Tarel et al. [75] proposed an affine invariant object recognition method by the use of fourth order implicit polynomials (quartics) to represent objects. After the quartics were decomposed into covariant conics acting as intrinsic coordinate system for the object, geometric invariances are then evaluated. In [61, 26], coplanar conics are used to defined the object boundary and invariances of coplanar conics are used to match two objects.
1.3 Motivation and contributions

There are a few object recognition methods that can recognize objects under projective transformation such as differential invariance. Most of the methods were designed to recognize objects under similarity or affine transformation. This reason motivated us to invent a new object recognition method that works when objects are deformed by projective transformation. Specifically, a novel object recognition for planar object is presented. Moreover, the aim of the proposed method is to recognize a specific planar object rather than the object class. The approach of this work is based on the use of geometric invariance to define a new shape descriptor.

In the proposed method, a set of conic is used to representation the object boundary. It is motivated by two advantages of conic. First, conic is projective invariance. In other words, the projection of conic is still conic. For the second advantage, conic is the fundamental image feature. That is, many man-made object and natural object have conical part i.e. pedal shape. In contrast to traditional shape representations method, we represent the object shape as a composition of conics, which are estimated from piecewise curve sections. Conics have previously been used for representing and matching of objects. The most related works to our approach are [61, 26]. Our proposed approach, however, differs from both of these methods by eliminating both the use of a limited number of conic sections, such as three conic sections defined in [61], and the requirement of establishing correspondences between conic sections across different views, which is required in [26]. Depending on the object size, our approach estimates piecewise conic sections. Pairs of these conic sections are used to compute a measure invariant under projective transformation. The distribution of
these measures is used to represent the object. The proposed representation provides a view invariant approach to match similar objects.

In conclusion, two main contributions of this thesis in the object recognition problem consists of a new shape representation and a new projective invariant features. Dissimilar from other approaches, our proposed shape representation uses a set of conic sections without using any prior information regarding the exact number of conics composed to be the object shape. A new projective invariant feature is then derived in terms of view invariant histograms which uses the invariance of conic pairs representing the object shape. As a result, matching between two shapes is performed by using histogram metrics such as city block or Euclidean distances.

1.4 Organization of thesis

The thesis consists of six chapters. The following chapter introduces some basic definitions and terms in the projective geometry. These basic definitions will be presented for the general case: n-dimensional projective geometry.

The third chapter deals with the 2-dimensional projective geometry which is useful for studying the geometry between two images of the same planar object when the images are taken from two different viewpoints. The fourth chapter presents the concept of projective invariance. This chapter lays out the details of the proposed object recognition, which is derived based on the projective invariance.

The proposed object recognition method is presented in chapter 5. The discussion about the proposed method begins with shape representation method then projective invariant feature extracted from our shape representation method. Finally, shape matching method is presented.
Chapter 6 contains some experimental results from the proposed object recognition method and discussion on those results. The conclusions from those experimental results and some ideas about future work on this proposed method are presented in chapter 7.

1.5 Notation

Throughout this work, the following notational convention will be used. A type-written upper case letter will refer to a matrix i.e. a conic matrix \( \mathbf{C} \). A bold face letter will denote a vector i.e. \( \mathbf{x} = [x \ y \ z]^T \) and the superscript \( T \) stands for the transposition of a matrix. The cross sign \( \times \) is used to express the cross product between two vectors, for example, the cross product of vectors \( \mathbf{a} \) and \( \mathbf{b} \) is written as \( \mathbf{a} \times \mathbf{b} \). The determinant of a (square) matrix \( \mathbf{C} \) is written as \( |\mathbf{C}| \).
CHAPTER 2

PROJECTIVE GEOMETRY: BASIC DEFINITIONS

Projective geometry is all geometry.

Arthur Cayley

The task of photogrammetry and computer vision is to infer information from three-dimensional world. In order to represent the three-dimensional world, we use the familiar Euclidean geometry which generally works so well. In the Euclidean geometry, the length of an object and angle between two lines are preserved when the Euclidean transformation is applied. Moreover, two parallel lines in the same plane never intersect.

However, when we sense information from three-dimensional world by using a camera (the pin-hole camera model is assumed) and then record data on a photograph, the length of a line, angle between two lines and parallelism are obviously not preserved. According to the geometric optic theory of the pin-hole camera, such effects come from the perspective transformation which is the nature of any lens system [15, page 7]. Due to these facts, the Euclidean geometry is not suitable to study the geometry of image. Thus, a space that accounts for all the important phenomena and easily to manipulate need to be used to study images. As a consequence, projective
geometry is a very efficient tool for studying the geometry of image under perspective transformation [15, p. 7].

In the global view, Euclidean geometry is a special case of projective geometry. There are also two geometric spaces between them: similarity geometry and affine geometry. In other words, Euclidean ⊂ similarity ⊂ affine ⊂ projective transformation. According to table 2.1, the imaging process of a camera can be efficiently modeled by using projective geometry because it consists of the largest class of transformations. In projective geometry, however, fewer geometric entities are preserved, such that we cannot avoid to confront the difficulty of the object description and recognition in the projective space.

<table>
<thead>
<tr>
<th>Transformations</th>
<th>Euclidean</th>
<th>similarity</th>
<th>affine</th>
<th>projective</th>
</tr>
</thead>
<tbody>
<tr>
<td>rotation, translation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>uniform scaling</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>nonuniform scaling, shear</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>perspective projection</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Invariants</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>angle, ratio of length</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>parallelism</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>incidence, cross ratio</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 2.1: Characteristics of the four different geometries [16, pagep. 11].

While projective geometry contains fewer invariants than other spaces, we can use duality principle to make points and lines equivalent. Moreover, with the existence of
homogeneous coordinate, the algebra of projective geometry can be linear and have simpler formulas [72, chapter 1]. Furthermore, the use of projective geometry can help us understand the behavior of the image of algebraic curve such as conic and cubic very well [31].

In this chapter, we will provide some basic definitions of projective geometry. We will begin with the definition of the projective space and the homogeneous coordinates. Secondly, two importance geometric entities, hyperplane and quadric, will be introduced. Next, we will talk about the projective transformation and homography which maps one projective space to another. Finally, one of the important features of projective geometry, duality principle, will be discussed.

2.1 Projective Space and Homogeneous Coordinate

There are many ways to define the projective space (for more detail please see [3, page. 41] and [7, page. 29]), but the definition that will be presented here will lead us to the idea of homogeneous coordinate and we will restrict the discussion to only the real projective space. According to [37, chapter 2], the projective space is defined as follows:

**Definition 1** (The real projective space of dimension $n$). Given a real $(n + 1)$-dimensional vector space $\mathbb{R}^{n+1}$, the projective space of dimension $n$, $\mathbb{P}^n$, induced by $\mathbb{R}^{n+1}$ is the set $\mathbb{R}^{n+1}/\mathbb{O}_{n+1}$ under equivalence relation $\sim$ defined such that for all $x, x' \in \mathbb{R}^{n+1}/\mathbb{O}_{n+1}$,

$$ x \sim x' \iff \exists \lambda \neq 0, x = \lambda x', $$

(2.1)

where $\lambda$ is a scale and $\mathbb{O}_{n+1}$ is $(n + 1)$-dimensional null vector.
A projective point is a vector \( x \in \mathbb{P}^n \) and all vectors \( x' \in \mathbb{R}^{n+1} \) which are equivalent with a vector \( x \) under the equivalence relation \( \sim \) are used to represent the projective point \( x \in \mathbb{P}^n \), so-called homogeneous coordinate representation [27, page. 20]. Thus, the proportional vectors \( x \) and \( x' \) (\( x = \lambda x' \)) will represent the same point in projective space which means the magnitudes of vectors \( x, x' \in \mathbb{R}^n \) do not matter, only their directions do, and regardless of the value of \( \lambda \), all projective properties of a point must be preserved [27, page. 20]. Furthermore, null vector has no meaning and is not defined in projective space according to definition 1.

![Diagram](image)

Figure 2.1: Projection from \( \mathbb{P}^3 \) to \( \mathbb{P}^2 \). The point \( M \) in \( \mathbb{P}^3 \) is projected to the projective (image) plane \( \mathbb{P}^2 \) through the optical center of a pin-hole camera.

To link an Euclidean space \( \mathbb{R}^n \) and a projective space \( \mathbb{P}^n \) together, let \( [x_1, \ldots, x_n]^T \) be the coordinate of a point in \( \mathbb{R}^n \). Then, we can build its projective coordinate by the correspondence

\[
\begin{pmatrix}
  x_1 \\
  \vdots \\
  x_n
\end{pmatrix} \quad \longrightarrow \quad \begin{pmatrix}
  x_1 \\
  \vdots \\
  x_n \\
  1
\end{pmatrix}.
\]  

(2.2)
For instance, let consider a point of Euclidean coordinate $[x, y]^\top$ in the (image) plane. Its correspondent homogeneous coordinate in $\mathbb{P}^2$ can be obtained by adding 1 at the end: $[x, y, 1]^\top$. Similarly, in the case of a point coordinate $[x, y, z]^\top$ in the three-dimensional Euclidean space, its projective coordinate in $\mathbb{P}^3$ is $[x, y, z, 1]^\top$. Thus, the object and image spaces will be considered as $\mathbb{P}^3$ and $\mathbb{P}^2$ respectively and $\mathbb{P}^2$ is also called the projective plane. Moreover, the mapping from $\mathbb{P}^3$ to $\mathbb{P}^2$ is used to model the imaging process of a camera (please see figure 2.1).

We can easily transform a point $[x_1, \ldots, x_n, x_{n+1}]^\top$ in the homogeneous coordinate back into Euclidean coordinate by dividing it by the last coordinate:

$$
\begin{pmatrix}
x_1 \\
\vdots \\
x_n \\
x_{n+1}
\end{pmatrix}
\rightarrow
\frac{1}{x_{n+1}}
\begin{pmatrix}
x_1 \\
\vdots \\
x_n
\end{pmatrix}.
$$

(2.3)

The conclusion of the transformation of coordinate from Euclidean to projective space forth and back is shown in figure 2.2.

![Diagram](image)

**Figure 2.2**: Transformation of coordinates between Euclidean and projective spaces.
According to equation 2.3, there are two cases for the last coordinate. That is \( x_{n+1} \neq 0 \) and \( x_{n+1} = 0 \). Firstly, all of the points in Euclidean space is the points in projective space with \( x_{n+1} \neq 0 \). Next, the points with coordinates \([x_1, \ldots, x_n, 0]^\top\) do not exist in n-dimensional Euclidean space and such points are known as points at infinity [16, page 6]. Thus, we can conclude that the Euclidean space contains less points than the projective space because the projective space contains point in the Euclidean space and the points at infinity [16, page 6].

Since the projective geometry treats points at infinity like any other point; in photogrammetry and computer vision applications, it is convenient to use the projective geometry to represent the three-dimensional world by extending it to three-dimensional projective space. Similar to 3D world, we also extend 2D Euclidean space, i.e. image space, to 2D projective space. Moreover, with the use of homogeneous coordinate, we can use a simple matrix multiplication to represent the projective transformation [24, page 3]. As a result, matrix algebra computations can be exploited to demonstrate many of the significant aspect of the projective transformation and projective geometry [1].

2.2 Hyperplane and Quadric

A hyperplane defined by a point \( h \in \mathbb{P}^n \) is a set of points \( x \in \mathbb{P}^n \) whose coordinate satisfy

\[
\sum_{1 \leq i \leq n+1} h_i x_i = h^\top x = 0. \tag{2.4}
\]

where the vector \( h \) contains the parameters of the hyperplane. For example, hyperplanes in \( \mathbb{P}^2 \) are lines; they are planes in \( \mathbb{P}^3 \). There is a remark from equation 2.4
that it is symmetric i.e. \( h^\top x = x^\top h = 0 \). This is to say, the form of the equation is not affected by the interchanging of the role of \( x \) and \( 1 \). This symmetric property is called \textit{duality} between a point and a hyperplane [16, page. 68] and we will discuss about duality principle later.

Another way to define a hyperplane \( h \) in \( \mathbb{P}^n \) is to use \( n \) linearly independent points \( x_1, \ldots, x_n \) in \( \mathbb{P}^n \) [27, page. 23]. For instance, three points \( x_1, x_2, x_3 \) in \( \mathbb{P}^3 \) define a plane similar to three points defining a plane in the Euclidean space. A point \( x \in \mathbb{P}^n \) is in the hyperplane \( h \) defined by \( n \) points \( x_1, \ldots, x_n \) if and only if

\[
| x_1 \ldots x_n x | = 0, \tag{2.5}
\]

where \(| \cdot |\) is the determinant of a \((n+1) \times (n+1)\) matrix. If we compute the determinant in equation 2.5 by expanding the last column, we will get the equation 2.4. Moreover, the equation 2.5 is very useful to check whether three points are collinear in \( \mathbb{P}^2 \) and whether four points are coplanar in \( \mathbb{P}^3 \).

A \textit{quadric} defined by a symmetric \((n+1) \times (n+1)\) matrix \( Q \) is a set of points \( x \in \mathbb{P}^n \) whose coordinates satisfy

\[
\sum_{1 \leq i, j \leq n+1} q_{ij} x_i x_j = x^\top Q x = 0. \tag{2.6}
\]

The quadrics are conic sections in \( \mathbb{P}^2 \), and they are quadric surfaces in \( \mathbb{P}^3 \). In the case of \( \mathbb{P}^1 \), they are two points in 1D projective space.

### 2.3 Projective Transformations and Homography

In general, a transformation is a mapping between two spaces. That is, a point \( x \) in a space is mapped to a point \( x' \) in another space by \( x' = T(x) \). In projective space, the \textit{Linear transformation of projective points} from \( \mathbb{P}^n \) to \( \mathbb{P}^n \) can be represented as a
Figure 2.3: An example of projective transformation. The object space and image planes are related by projection matrices $P_1$ and $P_2$. If we take the image of the ground plane in object space, i.e. $z = 0$, the relation between two image planes $\Pi_1$ and $\Pi_2$ can be represented by planar homography which is represented by a $3 \times 3$ matrix.

$(m + 1) \times (n + 1)$ matrix $P$ [27, page 22]. Let $x$ and $x'$ be a projective point in $\mathbb{P}^n$ and $\mathbb{P}^m$ respectively. Thus, the mapping from $\mathbb{P}^n$ to $\mathbb{P}^m$ is

$$x' = Px. \quad (2.7)$$

The matrix $P$ is called projection matrix. If $n = m$, we use a special notation $H$ for transformation matrix $P$, and it is called homography [16, page 18]. One of the fundamental properties of the projective transformations is given by the following statement.

**Theorem 1** (Conservation of Collinearity). Under projective transformation, images collinear points are collinear.

Let us use the imaging process as an example of projective transformation. Let $X$ be a point in object space, $\mathbb{P}^3$, and $x_1$ and $x_2$ be the image of the point $X$ on image planes $\Pi_1$ and $\Pi_2$ respectively, as shown in figure 2.3. The relation between the point $X$ and its images on image planes is given by
\[ x_1 = P_1X, \quad x_2 = P_2X, \]

where \( P_1 \) and \( P_2 \) are 3 \( \times \) 4 matrices, and we call a projection from \( \mathbb{P}^3 \) to \( \mathbb{P}^2 \) a \textit{perspective projection}. A remark is that the projection matrix \( P_1 \) and \( P_2 \) are different due to the different orientations of cameras and some camera parameters i.e. focal length. Additionally, if there are collinear points in the object space, such points will be map to be collinear points in image planes. With this result, we will come up with another theorem in projective transformation.

\textbf{Theorem 2} (Conservation of Line). \textit{Under projective transformations, images of lines are lines.}

If we assume that point \( X \) lies on the ground plane, i.e. \( z = 0 \), then the projection matrices \( P_1 \) and \( P_2 \) will become 3 \( \times \) 3 matrices. According to equation 2.3, we have

\[ x_2 = P_2P_1^{-1}x_1 = h x_1. \] (2.8)

This homography between two image plane is induced by the ground plane in the object space. Thus, it is also called \textit{planar homography}.

\section{2.4 Duality Principle}

Without losing of generality, let us use the two dimensional projective space as the show case for the duality principle. In two-dimensional projective space, a point and a line are dual elements i.e. if a line is through a point, then the point is on the line. According to duality principle, we can find the dual theorem of any two-dimensional projective geometry theorem by interchanging the role of points and lines in the original theorem. Adapted from [7, p. 20], the duality principle for two-dimensional projective geometry can be stated as following:
Figure 2.4: The example of duality principle. The statement "the joining of two points define a line" is dual to the statement "the intersection of two lines define a point".

**Theorem 3** (The Duality Principle of $\mathbb{P}^2$). *If $T$ is a theorem valid in two-dimensional projective geometry, and $T'$ is the statement obtain from $T$ by making the following changes:*

\[
\begin{align*}
\text{point} & \iff \text{line}, \\
\text{collinear} & \iff \text{concurrent}, \\
\text{join} & \iff \text{intersection},
\end{align*}
\]

*and whatever grammatical adjustment that are applied to $T$, then $T'$ is called Dual Theorem* and is valid in projective geometry.

For example, according to the statement "the joining of two points define a line", with the duality principle, we can states it dual statement as "the intersection of two lines define a point", see figure 2.4. In general, the duality principle is valid to n-dimensional projective geometry in which a point and a hyperplane are dual elements [16, page 106]. For example, a point and a plane in three-dimensional projective space are dual elements in $\mathbb{P}^3$, see figure 2.5. Another feature of the dual principle is that if the distance from the original element to the origin is $d$, then the distance from its dual element to the origin is $1/d$ [27, page 41].

22
Figure 2.5: The example of duality element. *Left:* the dual element of a point $\mathbf{x}$ in $\mathbb{P}^2$ is a line $l$. *Right:* the dual element of a point $\mathbf{X}$ in $\mathbb{P}^3$ is a plane $A$. The distance from origin to the dual element is the inverse of the distance from origin to the original element [27, page. 41].

The use of a homogeneous vector to represent both point and hyperplane suggests that a point $\mathbf{x}$ can be interpret as a hyperplane such that the homogeneous vector of the point $\mathbf{x}$ is the parameter vector of its dual hyperplane. For instance, for a point in $\mathbb{P}^2$, $[5 \; 3 \; 1]^T$, the equation of its dual line is $5x + 3y + 1 = 0$. 
CHAPTER 3

TWO-DIMENSIONAL PROJECTIVE GEOMETRY

Projective geometry has opened up for us with the greatest facility new territories in our science, and has rightly been called the royal road to our particular field of knowledge.

Felix Klein

The image plane can be represented in the two-dimensional projective space. This observation suggests to study two-dimensional projective space to understand the characteristics of image plane. The two-dimensional projective geometry, in contrast to \( n \)-dimensional projective geometry, is easier to understand due to the fact that we can visualize the geometry [24, page. 25].

We will start with the study of points, lines and conics which are the fundamental entities in 2D projective space in homogeneous notation. These entities are defined in the homogeneous coordinate for the ease of manipulation in projective space. Moreover, projective transformations which are a fundamental concept all over geometry will be detailed.
3.1 Points and lines

In the previous chapter, we introduced the definition of projective space. A point in the projective plane, $\mathbb{P}^2$, can be represented by $(x, y, 1)^\top$ or $(\lambda x, \lambda y, \lambda)^\top$ which are the same point under the equivalent relation. To represent a line in the projective plane, let us start with the standard line equation in Euclidean space

$$ax + by + c = 1^\top x = 0,$$

where $x = (x, y, z)^\top$ is the homogeneous coordinate of a point and $1 = (a, b, c)^\top$. Since the different value of $a, b$ and $c$ give a different line, a line may be represented by a vector $1 = (a, b, c)^\top$ [24, page. 26]. However, the vector $(a, b, c)^\top$ is not the unique representation of the line $1$ since

$$ax + by + z = \lambda ax + \lambda by + \lambda c = 0.$$

As a result, a line in 2D projective space can be represented by the vectors $(a, b, c)^\top$ and $(\lambda a, \lambda b, \lambda c)$ regardless of non-zero $\lambda$. Similar to the definition of point in projective space, we can represent a line $1$ by the vector $(a, b, c)^\top$ under equivalence relation.

To determine whether a point lie on a line or not, such point must satisfy the line equation, namely

**Theorem 4 (Incident relation).** The line $1$ contains the point $x$ if and only if $1^\top x = x^\top 1 = 0$.

The product $x^\top 1$ is just the inner or scalar product of two vectors. Furthermore, there are also two other importance theorem that are used to check the collinearity of three points, and concurrence of three lines. We have

25
Theorem 5 (Collinearity). Three points $x_1, x_2$ and $x_3$ are said to be collinear if and only if
\[
\begin{vmatrix}
  x_1 & y_1 & z_1 \\
  x_2 & y_2 & z_2 \\
  x_3 & y_3 & z_3 \\
\end{vmatrix} = 0.
\]

Theorem 6 (Concurrence). Three lines $l_1, l_2$ and $l_3$ are said to be concurrent if and only if
\[
\begin{vmatrix}
  a_1 & b_1 & c_1 \\
  a_2 & b_2 & c_2 \\
  a_3 & b_3 & c_3 \\
\end{vmatrix} = 0.
\]

3.1.1 Intersection of lines and line joining points

Given two lines $l$ and $l'$. We can easily find the intersection of these two line by using following theorem.

Theorem 7 (Intersection of two lines). Line $l$ intersects line $l'$ at point $x = l \times l'$.

The operator $\times$ is the cross or outer product. This theorem can be easily proved by using the vector identity $l \cdot (1 \times l') = l' \cdot (1 \times l') = 0$ and theorem 4. Similar to the basic fact that a line can be defined by two points, the line joining two points $x$ and $x'$ is expressed by the cross product of two point vectors. That is,

Theorem 8 (Line joining two points). The line joining two points $x$ and $x'$ is $l = x \times x'$

Since the cross product is a linear operator, it can be expressed as a matrix multiplication i.e. $u \times x = [u]_x x$ where $[u]_x$ is the skew-symmetric matrix which its left and right null space are the vector $u$ [16, page. 8]:

\[
[u]_x = \begin{bmatrix}
  0 & -u_3 & u_2 \\
  u_3 & 0 & -u_1 \\
  -u_2 & u_1 & 0 \\
\end{bmatrix}.
\]
Figure 3.1: The mapping of points at infinity to the image plane.

Thus, the geometric operation that joins two points to form a line or intersection of two lines to form a point becomes an algebraic operation.

3.1.2 Point and line at infinity

The most important feature of projective geometry that make it more general than Euclidean geometry is the inclusion the points at infinity [16, page. 6]. Let us start from a point in two-dimensional projective geometry \((x, y, z)^\top\). We can find its Euclidean coordinate by \((x/z, y/z)^\top\) where \(z \neq 0\). When the \(z\) value is approaching zero, its correspondent Euclidean coordinate is approaching to infinity and the points with \(z = 0\) are known as ideal points, vanishing points or points at infinity [16, page. 6]. Thus, we have

**Theorem 9** (Ideal point representation). *The homogeneous coordinate of ideal points in \(\mathbb{P}^2\) is \((x, y, 0)^\top\).*
To illustrate how points at infinity can be mapped to the image plane, let us use an example from reference [53] shown in figure 3.1. Without losing of generality, we assume that the image plane is perpendicular to the ground plane. Assuming that there is a point moving on the line perpendicular to the image plane, when the point moves infinity, its image on the image plane will converge to a vanishing point. Obviously, although the points on ground are evenly spaced on ground plane, there images are not evenly spaced.

Next, let's consider the intersection of two parallel lines. Clearly, in Euclidean space, two parallel never intersect but they intersect in projective space. Given two parallel lines \( ax + by + c = 0 \) and \( ax + by + c' = 0 \). Thus, there homogeneous vectors are \( \mathbf{l} = (a, b, c)^T \) and \( \mathbf{l}' = (a, b, c')^T \). According to theorem 7, there intersection is \( \mathbf{l} \times \mathbf{l}' = (b, -a, 0)^T \) which is the ideal point. Namely,

**Theorem 10** (Intersection of two parallel lines). *In projective space, two parallel lines intersect at a vanishing point.*

To visualize how two parallel lines intersect at vanishing in projective plane, let us again use an image plane as an example. According to figure 3.2, here we see the moving of two point to infinity on two parallel lines on the ground plane. One can see that the loci of the image points on image plane intersect at the vanishing point.

According to the fact that the joining of two points define a line, if we connect two vanishing points together, what we get is a line called *line at infinity* as shown in figure 3.3. We can compute the line at infinity by using theorem 8, then we have \( \mathbf{l}_\infty = (0, 0, c)^T \). Moreover, we can prove that all the vanishing points lie on the line at infinity because the dot product between ideal point \( (x, y, 0)^T \) and ideal line \( \mathbf{l}_\infty = (0, 0, c)^T \) is always zero. Namely,
Figure 3.2: The intersection of two parallel lines. Image of the points on two parallel lines intersect at the same vanishing point.

Figure 3.3: The line at infinity. A pair of parallel lines will intersect at infinity. When we connect two vanishing points together, we will get a line called vanishing line on which all vanishing points lie.
Figure 3.4: The three types of non-degenerate conic. Left Ellipse; Middle Parabola; Right Hyperbola.

**Theorem 11** (Collinearity of ideal points). In projective space, all of the points at infinity lie on the line at infinity \( l_\infty = (0, 0, c)^T \).

### 3.2 Conic section

Conic section is an algebraic plane curve which is represented by a second-order polynomial equation. It can also be geometrically generated by the intersections of a plane with a circular conical surface. The three type of conics shown in figure 3.4 are the circle, ellipse, hyperbola and parabola, so-called *non-degenerate conics*. They, however, are all equivalent under projective transformation because any one type can be transform to the other [24, page. 59]. In other words, conics are preserved under projective transformation.

In this section, we will discuss about the representation of conic section in projective space and also the projective properties of conic. That are dual conic (line conic), circular point and pole-polar relationship.
3.2.1 The representation of a conic section

General equation of a conic section in the Euclidean plane is given by:

\[ ax^2 + bxy + cy^2 + dx + ey + f = 0. \]

To represent the conic section in projective space, we homogenize the conic sections by repressing \( x \) and \( y \) with \( x/z \) and \( y/z \) respectively. We then have

\[ ax^2 + bxy + cy^2 + dxz + eyz + fz^2 = 0, \]

which can be written in the matrix form as

\[ x^\top C x = 0. \]

The matrix \( C \) is a \( 3 \times 3 \) symmetric matrix containing the conic coefficients and it is called a point conic. Thus, we have

\[ C = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}. \]

There are two possible cases for the matrix \( C \). For the case that conic \( C \) is full rank, the conic is one of the three main types: hyperbola, ellipse or parabola. In contrast, when the conic \( C \) is not full rank, it is called a degenerate conic including two lines (rank 2) and repeated line (rank 1).

Similar to the homogeneous representation of point and line, the conic \( \lambda C \) where \( \lambda \) is non-zero scalar is the same as conic \( C \). In other words, only the ratios of conic section parameters are important. This is because

\[ \lambda ax^2 + \lambda bxy + \lambda cy^2 + \lambda dx + \lambda ey + \lambda f = ax^2 + bxy + cy^2 + dx + ey + f = 0. \]

As a result the degree of freedom of a conic is given as five. The line \( l \) tangent to a conic \( C \) at a given point \( x \) is given in the homogeneous coordinate as:
Theorem 12 (Line tangent to a conic). The line that tangent to C at a point \( x \) is \( l = Cx \).

3.2.2 Dual conics

In the previous section, we discuss the point conic, which defines an equation of the locus of points. According to the duality principle, it is possible to define another type of conic called line (or dual) conic. Just the same as the point conic, we use a \( 3 \times 3 \) symmetric matrix to represent a dual conic denoted by \( C^* \) which is the adjoint of the point conic \( C \) \[24, \text{page 31}\]. That is, for full rank symmetric matrix \( C^* = C^{-1} \) (up to scale) \[24, \text{page 31}\]. A line \( l \) satisfies the line conic equation \( l^T C^* l = 0 \) if and only if the line \( l \) tangent to the conic \( C \) as shown in figure 3.5. Moreover, the envelope of the conic \( C \) is its dual conic \( C^* \) which has five degree of freedom.

![Diagram](image)

Figure 3.5: The duality between conic \( C \) and \( C^* \). A line \( l \) tangent to the conic \( C \) satisfy \( l^T C^* l = 0 \). In other words, the envelope of \( C \) is the lines represented by the points of \( C^* \).
3.2.3 Circular point

The concept of a circular point arises from the problem of the intersection of two conics [24, page 5]. According to fundamental theorem of algebra, we must have four solution from the set of two second-order polynomial equations. Thus, we should get four points from the intersection of two conics in Euclidean space as shown in figure 3.6a. However, geometrically, there are some situations that the intersection of two conics gives us only two points. That is the intersection of two circles as shown in figure 3.6b.

![Figure 3.6: The intersection of two conics. (a) Four points of intersection. (b) Two points of intersection.](image)

Let's consider the homogenized equation of a circle is of the form

$$(x - az)^2 + (y - bz)^2 = r^2z^2$$

where $(a, b)$ and $r$ are the center and radius of the circle. It is very easy to check that the points $I = (1, +i, 0)^T$ and $J = (1, -i, 0)^T$ satisfy above equation. This means that every circle contains these points and any two circles intersect at these points.
For these reasons, we call them *circular points at infinity*. Moreover, since these points are complex, they cannot be visualized. This fact defines the construction of projective geometry. That is, the combination of all points in Euclidean space, line at infinity, $l_\infty$, and these two circular points, $I$ and $J$, lying on the line at infinity is the projective space [16].

### 3.2.4 The pole-polar relationship

According to theorem 12, the line $l$ tangent to conic $C$ at a point $x$ is $l = Cx$. This relation is a special case of the *Pole-Polar* definition, which is restricted to the case of points lying on the conic $C$. If a line $l$ is defined by $l = Cx$, we call $l$ the *polar* of $x$ with respect to conic $C$ and the point $x$ is called the pole of $l$ with respect to conic $C$. In [24, page. 58], the generalized version of theorem 12 is written in terms of pole-polar as follow:

**Theorem 13.** The polar line $l = Cx$ of the point $x$ outside a conic $C$ intersects the conic in two points. The two lines tangent to $C$ at this points intersect at $x$.

In the figure 3.7, the geometric meaning of the pole-polar relationship is illustrated. The points of tangency of line from the pole $x$ to the conic are also the intersection points of the polar $l$ with the conic $C$ [24, page. 58]. Thus, if the pole lies outside the conic, the polar intersect conic in two points (figure 3.7a). If we move the point $x$ to the conic $C$, the line $l$ then tangent to the conic $C$ at the point $x$ (figure 3.7b). That is, the tangent line is the special case of the pole-polar. Finally, if the pole lies inside the conic, the polar has no real intersection points with the conic [11]. As an application, the definition of the center of conic sections is also defined in terms of the pole-polar:
Figure 3.7: The pole-polar relationship. (a) Pole lies outside the conic. (b) Pole lies on the conic. (c) Pole lies inside the conic.
Definition 2 (Center of Conic). Given a conic $C$, the pole of the $l_\infty$ with respect to $C$ is called the center of $C$.

3.3 2D Projective transformation

Two simple geometric transformations of an object are the translation and rotation. Changing the viewpoint while looking at the object can be also seen as geometric transformation. As a result, the homogeneous coordinates of points are change with respect to the coordinate system and such changing can be modeled by the class of transformation i.e. projective or affine transformations [44, page. 142].

3.3.1 Representation of transformation

In general, projective transformation is the mapping from $m$-dimensional projective space $\mathbb{P}^m$ to $n$-dimensional projective space $\mathbb{P}^n$. However, we will concentrate on the mapping from $\mathbb{P}^2$ onto itself, co-called homography, which can be represented by matrix multiplication:

Definition 3 (Projective transformation (Homography)). A projective transformation from $\mathbb{P}^2$ onto itself is a linear transformation of real three-dimensional vectors represented by a non-singular $3 \times 3$ matrix:

$$
\begin{pmatrix}
  x'_1 \\
  x'_2 \\
  x'_3 
\end{pmatrix} =
\begin{bmatrix}
  h_{11} & h_{12} & h_{13} \\
  h_{21} & h_{22} & h_{23} \\
  h_{31} & h_{32} & h_{33}
\end{bmatrix}
\begin{pmatrix}
  x_1 \\
  x_2 \\
  x_3
\end{pmatrix},
$$

or more compactly, $\mathbf{x'} = H\mathbf{x}$.

Representation of homography in the form of a matrix has two main benefits. Namely, basic transformations such as translation rotation can be easily expressed in a matrix framework using linear algebra and the cascade of transformations can
be collapsed into one transformation by matrix multiplication [55, page. 69]. As a result, intermediate coordinate systems are not required.

Since the collinearity of a set of points is preserved under the homography transformation, the projective transformation from plane to itself is also called *collineation* of $\mathbb{P}^2$ [24, page 32]. Another name for projective transformation *direct linear transformation* (DLT) due to the linear relation between points [44, page 142].

Note that the homography matrix $H$ is homogeneous because its scaling is transferred to the scaling of $x'$ which does not changed the transformed point according to the equivalence relation defined in projective space [44, page. 142]. As a result, the transformation matrix $H$ represent the same transformation as $\lambda H$. Since $H$ is $3 \times 3$ matrix, where the degree of freedom is 8.

### 3.3.2 Transformation of lines and conics

The transformation from a line $l$ to another line $l'$ needs to preserve the collinearity, such that, if a point $x$ lies on a line $l$, then the transformed point $x'$ must lie on the transformed line $l'$ [24, page. 36].

To derive the transformation of line, we have

$$x^T l = 0 = (H^{-1} x')^T l = (x')^T (H^{-T} l).$$

Since the collinearity must be preserved, $x^T l = x'^T l'$. As a result, we can conclude that:

**Theorem 14** (Transformation of lines). A line $l$ is transformed to $l' = H^{-T} l$ by the point transformation $x' = H x$.

The transformation of a conic $C$ according to the point transformation $x' = H x$ is given by:
\[ x^T C x = (H^{-1} x')^T C H^{-1} x' \]
\[ = x'^T H^{-T} C H^{-1} x' , \]
which is a quadratic form \( x'^T C' x' \) with \( C' = H^{-T} C H^{-1} \). As a consequence, we state the transformation rule of a conic:

**Theorem 15** (Transformation of a conic). A conic \( C \) is transformed to \( C' = H^{-T} C H^{-1} \) by the point transformation \( x' = H x \).

For a dual conic, its transformation rule can be derived in a similar manner, and is given by:

**Theorem 16** (Transformation of a dual conic). A dual conic \( C^* \) is transformed to \( C^{*'} = H C^* H^T \) by the point transformation \( x' = H x \).

### 3.4 Classes of transformations

In previous chapter, we mentioned about the classes of geometric transformations that are Euclidean, similarity, affine and projective transformation. Recall that Euclidean \( \subset \) similarity \( \subset \) affine \( \subset \) projective transformation. In other words, Euclidean, similarity and affine transformations are the specialization or are the subset of the projective transformations. In terms of group theory, the projective transformation forms a group called *projective linear group*. Since other three geometric transformations are subsets of the projective transformation, these transformations are *subgroup* of the projective group [24, page. 37].

We will start from the Euclidean transformation and then similarity, affine and projective transformations respectively. Although the geometric transformations transformation can be categorized by the structure of their transformation matrices, alternatively, we can characterize the geometric transformations in terms of elements or
quantities that are preserved or invariant [24, page 38] and only important invariants will be presented in this chapter.

3.4.1 Class 1: Euclidean transformations

Euclidean transformation is the simplest transformation on the plane $\mathbb{R}^2$. It consists of only rotation and translation and represented as

$$
\begin{pmatrix}
x'_1 \\
x'_2 \\
1
\end{pmatrix} =
\begin{bmatrix}
\cos(\theta) & -\sin(\theta) & t_x \\
\sin(\theta) & \cos(\theta) & t_y \\
0 & 0 & 1
\end{bmatrix}
\begin{pmatrix}
x \\
y \\
1
\end{pmatrix}.
$$

It can be expressed more compactly as

$$
x' = H_E x =
\begin{bmatrix}
R \\
0^T \\
1
\end{bmatrix},
$$

where $R$ is a $2 \times 2$ orthogonal rotation matrix ($R^T R = I$), $t$ a translation vector, $0^T$ a null vector [24, page. 38]. The three degrees of freedom of this transformation consists of one from rotation angle and two from translation.

In Euclidean transformation, the distance between two points, the angle between two lines, and area are invariant. Moreover, parallelism is preserved.

3.4.2 Class 2: Similarity transformations

The similarity transformation is constructed by combing Euclidean transformation and an isotropic scaling together. According to [24, 39], it has matrix representation

$$
\begin{pmatrix}
x'_1 \\
x'_2 \\
1
\end{pmatrix} =
\begin{bmatrix}
s \cos(\theta) & -s \sin(\theta) & t_x \\
s \sin(\theta) & s \cos(\theta) & t_y \\
0 & 0 & 1
\end{bmatrix}
\begin{pmatrix}
x \\
y \\
1
\end{pmatrix},
$$

which can be represented more compactly as

$$
x' = H_S x =
\begin{bmatrix}
s R \\
0^T \\
1
\end{bmatrix},
$$
where the scalar \( s \) is the isotropic scaling. Note that the shape of an object is preserved under similarity transformation. The four degrees of freedom of this transformation consist of one from scaling and three from Euclidean transformation and these parameters can be computed by using two point correspondences.

In terms of invariance, only parallelism and the angle between two lines are preserved under the similarity transformation. Even though the length between two points and area are changed, the ratio between two lengths or two areas are invariant because the isotropy cancels out.

### 3.4.3 Class 3: Affine transformations

The affine transformation is an Euclidean transformation composed with a non-isotropic scaling. According to [24, page. 39], it has the matrix representation

\[
\begin{pmatrix}
{x'_1} \\
{x'_2} \\
1
\end{pmatrix} = \begin{bmatrix}
a_{11} & a_{12} & t_x \\
a_{21} & a_{22} & t_y \\
0 & 0 & 1
\end{bmatrix} \begin{pmatrix}
x \\
y \\
1
\end{pmatrix},
\]

which can be written more concisely in block form as

\[
x' = H_Ax = \begin{bmatrix}
A & t \\
0^T & 1
\end{bmatrix},
\]

where \( A \) is a 2 \( \times \) 2 non-singular matrix. Note that, affine transformation preserves the vanishing points and ideal line. This fact can be proved by transforming any point \((x, y, 0)^T\) by affine transformations. The results from the transformation are \((\lambda x, \lambda y, 0)^T\) which are points at infinity. The 6 degrees of freedom in affine transformation composed of one from rotation angle, two from translation, two from non-isotropic scaling and shearing.

Regarding invariance, the similarity is invariant of length ratios and angles between two lines are not preserved. Three important invariant are parallelism, ratio of areas and ratio of length of parallel line segments.
3.4.4 Class 4: Projective transformations

Projective transformation can be seen as a general non-singular linear transformation and it is the generalized version of all other transformations [24, page. 41]. We have already seen the representation of projective transformation from definition 3. According to [24, page. 41], it can be expressed more compactly in block form

\[ x' = H_p x = \begin{bmatrix} A & t \\ v^T & u \end{bmatrix}, \]

where the vector \( v = (v_1, v_2)^T \). Since the projective transformation is homogeneous, only the ratio of elements in the matrix representing the transformation is significant. As a result, the transformation consists of eight parameters.

In terms of invariants, parallelism, distance between two points and ratio of area are not preserved. The cross ratio which is the most simplest projective invariant is preserved. In next chapter, we will discuss about cross ratio and other invariants again in more detail.

3.5 Epipolar geometry and the fundamental matrix

The structure of stereo vision can be represented by epipolar geometry. That is, the basic constraint between two photographs is given in term of epipolar geometry. Its significant application in stereo vision is to answer corresponding geometry problems or stereo matching [24, page. 239]. That is to find the point correspondence between two scenes. To represent the epipolar geometry algebraically, a \( 3 \times 3 \) matrix called \( \text{fundamental matrix} \) \( F \) is used. One of the applications of fundamental matrix is projective reconstruction which is beyond the scope of this thesis.
3.5.1 Epipolar geometry

Epipolar geometry has been long known in photogrammetry that the coordinate of the projection of point and the two camera optical centers form a triangle (e.g [70, page. 605] and [47, page. 31]). Therefore, it can be written as an algebraic constraint for image coordinates of two stereo images [42, page. 109].

Let us consider a scenario that we take the images of a point $\mathbf{X}$ in 3D from two different views and its images are $\mathbf{x}$ in one image and $\mathbf{x}'$ in another image. There are two problems regarding this situation. The first problem is how to generate the relation between two image points $\mathbf{x}$ and $\mathbf{x}'$ and the second problem is how to constrain the corresponding point $\mathbf{x}'$ if we know only $\mathbf{x}$ [24, page. 239].

![Diagram](image)

(a)

Figure 3.8: Point correspondence problem.

To answer the first problem, let’s see figure 3.8a. The image points $\mathbf{x}$ and $\mathbf{x}'$, object point $\mathbf{X}$, and camera center form a plane $\pi$. In other words, if we project the image points $\mathbf{x}$ and $\mathbf{x}'$ back to 3D space, the back-projected rays intersect at $\mathbf{X}$, and
the rays lie in $\pi$. This fact is very significant for correspondence searching [24, page. 239].

For the second problem, we want the corresponding point of the point $x$ on the second image. Using entire area of the image as solution space may not be an intuitive idea. The epipolar geometry plays an importance role to constraint the solution space from 2D to 1D and this constraint called epipolar constraint or search space approximation [65, p. 236]. To make such constraint, lets determine the plane $\pi$ formed by the baseline and the projection ray casted from $x$, see figure 3.8.b. Since we already knew from above that the correspondence point $x'$ lie in the plane $\pi$, thus the point $x'$ must lie on the line of intersection between $\pi$ and image plane which is the line $l'$ [24, page. 240]. This leads to a fact that the line of intersection is the search space for the point corresponding to $x$.

Figure 3.9: The illustration of terminology in epipolar geometry
To give an example and introduce some terms in epipolar geometry, let's see figure 3.9a. The camera baseline is a line joining two camera centers and intersects each image planes at the epipole $e$ and $e'$ and it is on epipolar plane. Epipolar lines on one image plane can be defined by intersecting the image plane with epipolar plane. By moving the object point $X$ in 3D space, the epipolar plane will rotate about the baseline, as shown in figure 3.9b. and all of the rotating epipolar plane is called epipolar pencil. Furthermore, all epipolar generated from rotating epipolar plane about the base line intersect at the epipole.

### 3.5.2 Fundamental matrix $F$

According to [24, page. 242], the geometric derivation of fundamental matrix can be decompose into two steps, point transfer via plane and constructing the epipolar line. Refer to figure 3.10. We firstly transform the point $x$ to the second image by using the ray through the first camera center corresponding to the point $x$. This ray will intersect a plane $\pi$, which does not contain both of the two camera centers, at point $X$. Then, the point $X$ is projected back to the second image by using the ray through the second camera center and this ray intersects the second image plane at a point $x'$. We call this process transfer via the plane $\pi$ and can be represented by a 2D homography $H_\pi$ [24, page 242]. That is, $x' = H_\pi x$.

Next, since epipolar line $l'$ passes through $x'$ and epipole $e'$, we have $l' = e' \times x' = [e']_x x'$. Because $x' = H_\pi x$, we have

$$l' = [e']_x H_\pi x = Fx$$

where $F = [e']_x H_\pi$ is the fundamental matrix. Since, $x'$ lies on $l'$. That is, $x'^T \cdot l' = 0$. We have
Figure 3.10: The derivation of the fundamental matrix $F$.

\[ x'^T F x = 0, \]

which is very useful to check whether two point $x$ and $x'$ are correspondent.

Note, the existence of $F$ does not require any plane. The plane is used here just to define a point map from one image to another [24, page 243].
CHAPTER 4

PROJECTIVE INVARIANCE

Geometry is the science of correct reasoning on incorrect figures.

George Polya

Invariances are entities which are preserved under a class of geometric transformations. For example, point and straight line are the simplest projective invariances because images of points and straight lines are still points and straight lines. In Euclidean transformation, the distance between two points and angle between two straight line are invariant. To define the term invariance, let $s$ be a set of features and $T$ be a geometric transformation which belongs to a group of transformation, i.e. affine transformation. Function $I$ is called invariant if and only if $I(s) = I(Ts)$ for any $T$ in the transformation groups [36]. The function $I$ is called projective invariance if $T$ is a projective transformation.

In computer vision and photogrammetry, object recognition problems and reconstruction of three-dimensional structure from un-calibrated cameras can benefit from invariances. To illustrate the usefulness of invariants, let us use an example from [36]. Assuming that two image of an object are taken from two different viewpoint,
let $s_A$ and $s_B$ be two sets of features extracted from the first and second images respectively i.e. points or contours, and the transformation $T$ is the mapping from $s_A$ to $s_B$; that is, $s_B = Ts_A$. A function $I$ is invariant to the transformation $T$ if and only if $I(s_A) = I(Ts_A) = I(s_B)$. As a result, the feature matching problem becomes recovering the relation between the invariants $I(s_A)$ and $I(s_B)$ which is independent from the group of transformation which transformation $T$ belongs to [36].

In this chapter, we introduce some basic projective invariances of points, straight lines and curves. We will begin with cross ratio which is the simplest projective invariant for points and straight lines and then we present the $p^2$ invariance which is invariant to both projective and permutation of point or line labeling. We will end this chapter by introducing joint conic invariance which plays an importance role in proposed object recognition method presented in next chapter.

4.1 Invariance of points and straight lines

In the first chapter, we have seen that many geometric entities such as length, angle between two line and parallelism are not preserved under projective transformation. The simplest invariant entity in projective space is cross ratio which is the invariant feature of the configuration of points or line. We will only present the cross ratio of points because the cross ratio of lines can be easily constructed by interchanging between points and lines due to the duality principle. Cross ratio has been used in many applications such as image retrieval [62], classification and recognition [77, 80, 58], object detection [80] and visual control [89].

Although cross ratio is invariant to the projective transformation, it is very sensitive to point labeling. The points in both image must be labeled in the same order
before the cross ratio is evaluated. In other words, they must be associated prior to computing the invariance. Unfortunately, this information is rarely available and leads to labeling or permutation problems. A function that is not sensitive to the labeling problem is called \( p^2 \) (projective/permutation) invariance because it is invariant to projective transformation and point permutation or labeling \([45]\). Since, the \( p^2 \)-invariance work without knowing the correspondences of points, it can be efficiently used in many applications in which the point correspondence information is rarely available such as hand geometry biometric \([94]\), virtual reality \([38]\), image registration \([96, 74]\), and object tracking \([84]\).

Cross ratio in one dimension will be introduced first as an example of cross ratio and then the cross ratio in \( n \) dimension will be presented as the generalized version. Next, the \( p^2 \) invariance will be shown from the one dimensional case and then the \( n \) dimensional case.

### 4.1.1 Cross ratio in one dimension

A 1D projective space is defined as a line and the projective transformation between two 1D projective space can be represented as the central projection as shown in figure 4.1.

The cross ratio is defined with respect to figure 4.1. First, let the points \( x_1, x_2, x_3 \) and \( x_4 \) on the same line have (Euclidean) line coordinate \( t_1, t_2, t_3, \) and \( t_4 \) respectively. Thus, the cross ratio is define as:

\[
\text{Cross}(x_1, x_2, x_3, x_4) = \frac{(x_1 - x_3)(x_2 - x_4)}{(x_3 - x_2)(x_4 - x_1)}.
\]  

One can see that the cross ratio defined in equation 4.1 is the ratio of the length of the segment between two points. To illustrate how the cross ratio is invariant under
Figure 4.1: Cross ratio in one dimensional projective space. The right figure illustrate the perspective projection shown in the left image as a general projective transformation on between two lines.

projective transformation, let us incorporate the use of homogeneous represent into the definition of the cross ratio. We have

\[
\text{Cross}(x_1, x_2, x_3, x_4) = \frac{(x_1 - x_3)(x_2 - x_4)}{(x_3 - x_2)(x_4 - x_1)}
\]

\[
= \frac{\begin{vmatrix} x_1 & x_3 \\ 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_3 & x_2 \\ 1 & 1 \end{vmatrix}} \cdot \frac{\begin{vmatrix} x_2 & x_4 \\ 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_4 & x_1 \\ 1 & 1 \end{vmatrix}}
\]

\[
= \frac{|x_1 x_3| |x_2 x_4|}{|x_3 x_2| |x_4 x_1|},
\]

where

\[
|x_i x_j| = \det \left( \begin{bmatrix} x_{i1} & x_{j1} \\ x_{i2} & x_{j2} \end{bmatrix} \right).
\]

As a result, we can define the definition of the cross ratio more formally by using the homogeneous coordinate:

**Definition 4 (Cross ratio in one dimensional projective space).** Let \( x_1, x_2, x_3 \) and \( x_4 \) be four non-zero vectors in \( \mathbb{R}^2 \). The cross ratio is defined as

\[
\]

49
\[ \text{Cross}(x_1, x_2, x_3, x_4) = \begin{vmatrix} x_1 & x_2 & x_3 & x_4 \\ x_3 & x_2 & x_4 & x_1 \end{vmatrix}. \]

Next, in order to show the invariance of the cross ratio to the projective transformation, it is first necessary to show how the coordinate of points on a line can be transformed. Because a point \( x \) on a line can be represented in terms of homogeneous coordinate by a vector in \( \mathbb{R}^2 \), it transformed point is \( x' = H_{2 \times 2} x \). Thus, we have:

\[ |x'_i x'_j| = |Hx_i Hx_j| = |H||x_i x_j|. \]

Applying this fact to the definition of cross ratio, we get

\[ \text{Cross}(x'_1, x'_2, x'_3, x'_4) = \begin{vmatrix} x'_1 & x'_3 & x'_2 & x'_4 \\ x'_3 & x'_2 & x'_4 & x'_1 \end{vmatrix} = \frac{|Hx_1 Hx_3||Hx_2 Hx_4|}{|Hx_3 Hx_2||Hx_4 Hx_1|} \]
\[ = \frac{|H||x_1 x_3||H||x_2 x_4|}{|H||x_3 x_2||H||x_4 x_1|} = \frac{|x_1 x_3||x_2 x_4|}{|x_3 x_2||x_4 x_1|} = \text{Cross}(x_1, x_2, x_3, x_4). \]

As a result, we can show how the cross ratio is invariant under projective transformation:

**Theorem 17.** Let \( H \) be a homography matrix in \( \mathbb{P}^1 \) and let points \( x_1, x_2, x_3, x_4 \) and there transformed points \( x'_1, x'_2, x'_3, x'_4 \) be in \( \mathbb{P}^1 \), we then have

\[ \text{Cross}(x'_1, x'_2, x'_3, x'_4) = \text{Cross}(x_1, x_2, x_3, x_4). \]

Note that not only the finite points but also the ideal point can be used to compute the cross ratio.

### 4.1.2 Cross ratio in n dimension

In order to define cross ratio in \( n \) dimensional projective geometry, a configuration of \( n + 3 \) points \( x_i, i = 1, 2, \ldots, n + 3 \), is required. In \( n \) dimensional projective
space, a point \( x_i \) can be defined in terms of homogeneous coordinate by the \( n \)-tuples \((x_1, x_2, \ldots, x_{n+1})\). Similar to the definition of cross ratio in one dimensional projective space, the notion:

\[
\begin{vmatrix}
  x_1^{(1)} & \cdots & x_1^{(n+1)} \\
  x_2^{(1)} & \cdots & x_2^{(n+1)} \\
  \vdots & \ddots & \vdots \\
  x_{n+1}^{(1)} & \cdots & x_{n+1}^{(n+1)} 
\end{vmatrix},
\]

is the determinant of the homogeneous coordinates of \( n + 1 \) points. Note that these \( n + 1 \) points form a parallelepiped in \( n \)-dimensional projective space and its volume is proportional to the determinant defined in equation 4.1.2. According to [45], the definition of the cross ratio in \( n \)-dimensional projective space is defined as

**Definition 5** (Cross ratio in \( n \) dimensional projective space). *Given* \( n + 3 \) points \( x_1, x_2, \ldots, x_{n+3} \) in \( \mathbb{P}^n \). A cross ratio is defined by

\[
\text{Cross}_n(x_1, \ldots, x_{n+3}) = \frac{|x_1 \cdots x_{n-1} x_n x_{n+2}| |x_1 \cdots x_{n-1} x_{n+1} x_{n+3}|}{|x_1 \cdots x_{n-1} x_n x_{n+3}| |x_1 \cdots x_{n-1} x_{n+1} x_{n+2}|},
\]

According to definition 5, we can also prove that the cross ratio in \( n \) dimensional projective space is invariant under projective transformation by using the same way as we used in one dimensional projective space.

To illustrate the use of cross ratio in \( n \) dimensional projective space, let us use two dimensional projective space as an example. In two dimensional projective space, we need the configuration of 5 points in the projective plane to compute the cross ratio as shown in figure 4.2. A point \( x_i \) in projective plane is described in the homogeneous coordinates by the triplets \((x_i, y_i, z_i)\) and transformed by any nonsingular \( 3 \times 3 \) matrix defining a planar homography of the point configuration. Thus, the cross ratio in two
Figure 4.2: Cross ratio in two dimensional projective space. The configuration of five points in projective plane is used to compute the cross ratio. Any nonsingular $3 \times 3$ matrix $H$ defines a planar projective transformation of the configuration. Although the configuration of five points is changed according to the projective transformation, the cross ratio is preserved.

Dimensional projective space is given by:

$$\text{Cross}_2(x_1, x_2, x_3, x_4, x_5) = \frac{|x_1 x_2 x_4||x_1 x_3 x_5|}{|x_1 x_3 x_4||x_1 x_2 x_5|}. \quad (4.2)$$

$$= \begin{vmatrix} x_1 & x_2 & x_4 & x_1 & x_3 & x_5 \\ y_1 & y_2 & y_4 & y_1 & y_3 & y_5 \\ z_1 & z_2 & z_4 & z_1 & z_3 & z_5 \\ x_1 & x_3 & x_4 & x_1 & x_2 & x_5 \\ y_1 & y_3 & y_4 & y_1 & y_2 & y_5 \\ z_1 & z_3 & z_4 & z_1 & z_2 & z_5 \end{vmatrix}. \quad (4.3)$$

where $|x_i x_j x_k|$ is the area of the triangle of which $x_i$, $x_j$ and $x_k$ are the vertex.

4.1.3 $p^2$ invariance in one dimension

$p^2$ invariance is a magnitude that remain unchanged under projective transformation and also not sensitive to the point labeling. Such invariance can be defined by combining the cross ratio and a function that is invariant under the permutation of points [36]. We will first consider the behavior of the cross ratio in one dimensional
projective space under the permutation of points and then we will discuss about the permutation invariant function.

In one dimensional projective space, we need the configuration of four points $x_1$, $x_2$, $x_3$ and $x_4$ to compute the cross ratio. Since we have 4 points on a line, they can be ordered in $4!$ (or 24) different ways i.e. $x_3x_2x_1x_4$, $x_2x_3x_1x_4$, $x_4x_3x_2x_1$ etc. Problem arises when we do not know the correspondence of points between two images. That is, we do not know the correct point labeling on another image as shown in figure 4.3.

Figure 4.3: Permutation problem in one dimensional projective space. After the points on a line are transformed, if we do not know the correct correspondence of points before and after transformation, the cross ratio value will differ from what it should be.

Let us start from the permutation effect on the cross ratio. According to [71, p. 15], the $4!$ different orders of points $x_1$, $x_2$, $x_3$, $x_4$ give us six possible values of cross ratio. In other words, let $\lambda$ be the cross ratio of an order of the points $x_1$, $x_2$, $x_3$, $x_4$. Then, all possible permutations of the points provide the different values of the cross
ratio which are in the set
\[ \lambda, \frac{1}{\lambda}, 1 - \lambda, \frac{\lambda - 1}{\lambda}, \frac{\lambda}{\lambda - 1}. \] (4.4)

These six expressions are distinct functions of \( \lambda \). However, some of the values of these functions may be the same if \( \lambda \)s are assigned with a suitable numeral. This situation can occur if the configuration of four points \( x_1, x_2, x_3 \) and \( x_4 \) are related in some ways [69, p. 47]. If one of them is noted by \( \lambda_1 = \lambda \), the five other values, then, are
\[ \lambda_2 = \frac{1}{\lambda}, \lambda_3 = 1 - \lambda, \lambda_4 = \frac{1}{\lambda - 1}, \lambda_5 = \frac{\lambda}{\lambda - 1}, \lambda_6 = \frac{\lambda - 1}{\lambda}. \]

Because we need a function that is invariant to the permutation of indices, it is adequate to use any arbitrary symmetric functions of the \( \lambda_i, i = 1, \ldots, 6 \). That is,
\[ J(\lambda_1) = J(\lambda_2) = J(\lambda_3) = J(\lambda_4) = J(\lambda_5) = J(\lambda_6), \]
where \( J \) is a permutation invariant function. Note that these functions are invariant from both projective transformation and permutation because this function is symmetric to the cross ratio. The simplest two \( p^2 \) invariant functions are
\[ J_0(\lambda) = \sum_{i=1}^{6} \lambda_i = 3 \] and \[ J_0(\lambda) = \prod_{i=1}^{6} \lambda_i = 1. \] However, these two functions are trivial and are of no interest in an application because they result in constant values [36].

According to [36, 45], four different \( p^2 \) invariant function were introduced. That is,
\[ J_1(\lambda) = \frac{\lambda^6 - 3\lambda^5 + 3\lambda^4 - \lambda^3 + 3\lambda^2 - 3\lambda + 1}{\lambda^2(\lambda - 1)^2} \]
\[ J_2(\lambda) = \frac{2\lambda^6 - 6\lambda^5 + 9\lambda^4 - 8\lambda^3 + 9\lambda^2 - 6\lambda + 2}{\lambda^2(\lambda - 1)^2} \]
\[ J_3(\lambda) = 3 \]
\[ J_4(\lambda) = -3. \]
Figure 4.4: The invariants $J_1(\lambda)$ (solid line) and $J_2(\lambda)$ (dashed line).

Obviously, the function $J_1$ and $J_2$ are non-trivial functions and can be used in applications. The discontinuities appear at $\lambda = 0$ and $1$ because some of the points are the same point. In particular, we can define other $p^2$-invariances as any symmetric functions of the six cross ratios $\lambda_i$, e.g. the sum of square or the sum of pairwise products [45]. However these functions are linear combinations of the four invariants $J_1(\lambda), \ldots, J_4(\lambda)$ [46]. In [36], some examples of the $p^2$ invariances are

\[
J_5(\lambda) = \frac{(\lambda^2 - \lambda + 1)^3}{\lambda^2(\lambda - 1)^2} = J_1(\lambda) + J_2(\lambda)
\]

\[
J_6(\lambda) = \sum_{i=1}^{6} \sum_{j=1, j \neq i}^{6} \lambda_i \lambda_j = J_1(\lambda) + J_4(\lambda)
\]

\[
J_7(\lambda) = \sum_{i=1}^{6} \lambda_i^2 = -J_1(\lambda) + J_2(\lambda) - \frac{1}{2} J_3(\lambda).
\]

In figure 4.4 and 4.5, the graph of the functions $J_1(\lambda), \ldots, J_7(\lambda)$ are shown. We can see that all function are not bounded. Bounded $p^2$-invariances can be constructed by exploiting the fact that $J_1$ and $J_2$ do not have real root (their graphs do not intersect
Figure 4.5: The invariants $J_5(\lambda)$ (dash-dot line), $J_6(\lambda)$ (solid line) and $J_7(\lambda)$ (dash line).

Figure 4.6: The bounded invariants $J_6(\lambda)$. 

56
x-axis). It can be shown that the ratios of $J_1(\lambda)$ and $J_2(\lambda)$ is a bounded function. In figure 4.6, the dependence of

$$J_8(\lambda) = \frac{J_2(\lambda)}{J_1(\lambda)}$$

on $\lambda$ is shown. As seen in the figure 4.6, this function is bounded between 2 and 2.8.

In conclusion, the $p^2$-invariant function of the four points in one dimension projective space is independent from how the points is labeled. That is, when the cross ratio are evaluated by using a permutation invariant function, the same value is obtained independently from how the points are labeled. In other words, the six possible cross ratios correspond to the root of the equation $J(\lambda) = C$, where $J(\lambda)$ are any nontrivial $p^2$-invariant function.

4.1.4 $p^2$ invariance in higher dimensions

The cross ratio in $n$-dimensional projective space is defined by the configuration of $n + 3$ points. Thus, those $n + 3$ points can be labeled by $(n + 3)!$ different ways. Similar to the case of one dimension, the cross ratio in $n$-dimensional projective space is also sensitive to the permutation of points.

Although, the value of cross ratio can change due to different labeling, the permutation of the first $n - 1$ points, $x_1, \ldots, x_{n-1}$ does not affect the value of cross ratio in $n$-dimensional projective space [45]. In other words, only the permutation of the last four points is important and yields one of the six possible cross ratio values shown in equation 4.4.

In order to illustrate a practical example of the $p^2$ invariance in higher dimension, let consider two dimensional projective space, $\mathbb{P}^2$. Thus, we require five points $x_i, i = 1, \ldots, 5$, such that no three of them are on the same line. The cross ratio in two
dimensions is

$$\lambda_1 = \text{Cross}_2(x_1, x_2, x_3, x_4, x_5) = \frac{|x_1 \ x_2 \ x_4||x_1 \ x_3 \ x_5|}{|x_1 \ x_3 \ x_4||x_1 \ x_2 \ x_5|}$$  \hspace{1cm} (4.5)$$

Since the value of the cross ratio in equation 4.5 can be changed according to the different labeling of the last four points, the expression of the second independent cross ratio of the five coplanar points configuration can be also derived by interchanging the indices 1 and 2 [45]. As a consequence, the second cross ratio is:

$$\lambda_2 = \text{Cross}_2(x_2, x_1, x_3, x_4, x_5) = \frac{|x_2 \ x_1 \ x_4||x_2 \ x_3 \ x_5|}{|x_2 \ x_3 \ x_4||x_2 \ x_1 \ x_5|}.$$  \hspace{1cm} (4.6)$$

As stated in [36, 45], the \(p^2\)-invariance of 2D projective space is a 5D vector \(J\). Its computed components are

$$J^{(1)} = J(\lambda_1), \quad J^{(2)} = J(\lambda_2), \quad J^{(3)} = J\left(\frac{\lambda_1}{\lambda_2}\right),$$

$$J^{(4)} = J\left(\frac{\lambda_2 - 1}{\lambda_1 - 1}\right), \quad J^{(5)} = J\left(\frac{\lambda_1(\lambda_2 - 1)}{\lambda_2(\lambda_1 - 1)}\right),$$

where \(J\) is any \(p^2\) invariant function, and the ascending order of \(J^{(i)}\) is the sought \(p^2\)-invariance.

### 4.2 Invariances of smooth (non-algebraic) curve

Similar to the invariance of points, the invariances of algebraic curve such as conic, cubic and quartic curves are well studied in classical theory of invariance and such in variance are global pertaining to the shape as a whole [10]. In contrast to the invariances of point sets and algebraic curves, the invariances of smooth (non-algebraic) curves is still an on going research in photogrammetry, computer vision and mathematics [54, 57]. The main approach is the use of derivatives, so called differential invariances. These invariances are local to points on a shape.
In order to find the differential invariance of a shape, the coordinates of the shape points are explicitly parameterized by functions of some local parameters \( t \):

\[
x = x(t).
\]

The shape descriptors are then derived from the derivatives of \( x \). To give an example of this differential approach, let us use Wilczynski’s method shown in [86] and write the transformation as

\[
x' = \lambda(x)Tx,
\]

where the factor \( \lambda(x) \) is the differentiable function of point \( x \). To find invariance, we first find invariance to the transformation \( T \), \( \lambda \) and the curve parameter \( t \) respectively.

Let \( x(t) \) be a plane curve. Invariances to \( T \) are the roots of the linear algebraic system of equations

\[
x''' + 3p_1x'' + 3p_2x' + p_3x = 0
\]

with the three unknown parameters \( p_1, p_2 \) and \( p_3 \) at each point \( t \). These parameters are invariant to only the transformation \( T \) except the factor \( \lambda(x(t)) \) and curve parameter \( t \). Invariances for \( \lambda \) are the functions of \( p_1 \):

\[
P_2 = p_2 - p_1^2 - p_1' \\
P_3 = p_3 - 3p_1p_2 + 2p_1^3 - p_1''
\]

However, these are not invariant under change of the parameter \( t \). Thus, the full invariances are

\[
\Theta_3 = P_3 - \frac{3}{2}P_2' \\
\Theta_8 = 6\Theta_3\Theta_3'' - 7(\Theta_3')^2 - 27P_2\Theta_3^2.
\]
One can see that this invariant require seventh order derivative. Obviously, it is numerically infeasible with the presence of image noise. This is one of the disadvantage of differential approach [86].

Not only the requirement of higher order derivative, but also the reparametrization can create a problem to this approach because the curve parametrization affects the values of differentials [54]. In [20], it is shown that reparametrization can be seen as a further linear transformation. Note that reparametrization has no effect in algebraic object.

Since the differential invariant is local to points, correspondences between points of the original and transformed curves are required. Unfortunately, the correspondence information between two curves is rarely available.

High order derivative, points correspondence and reparametrization give us the impractically of using differential invariances directly. Many methods have been derived to avoid the problem we usually encounter in the differential invariances, for example distinguished points and representation by algebraic curve methods.

Distinguished points

Points that can be extracted before and after a projection are called distinguished points, such as the point of inflections. The distinguished points act as markers for helping us to uniquely locate the same point after projection [54]. A method used to extract the distinguished points can be derived by using some properties preserved under projective transformations. For example, two points on a curve that are preserved under projective transformations can be extracted from a bi-tangent to the curve. That is, the bi-tangent to a projected curve provides the projection of
the original tangent points [54]. The idea of bi-tangency is based on the fact that collinearity of two points and tangency are preserved under projective transformation. As a result, we can detect any two points that are on the same line in projective space as shown in figure 4.7.

![Diagram of bi-tangency and point of inflection]

Figure 4.7: Preservation of bi-tangency and point of inflection.

Detecting other distinguished points is also possible after some distinguished points are extracted points from a contour. If a ray casting from a distinguished point touches at some distinct points, then this point of tangency become a new distinguished point [54]. As another example, intersecting the curve by the straight line joining two inflection points provides distinguished points. This construction is depicted in figure 4.8.

Distinguished points are effectively used as descriptors for a curve and either form algebraic invariances, cross ratios or \( p^2 \) invariance, of these distinguished points set. They are also used to determine the transformation between two images by using the images of the curves as a clue.
Figure 4.8: Detection of other distinguished points. Two examples are shown here. Left: Ray casting approach. Right: Connection of two inflection points.

**Representation by algebraic curves**

Since the invariances for algebraic curves are well established, relating them to smooth curves is an efficient way to discover the invariances of smooth curves. That is, the algebraic invariances of these algebraic curves is used to portray the non-algebraic curve. One of the examples of this method is the use of implicit polynomials in model-based vision [10]. In this approach, an implicit polynomial

\[ f(x, y) = \sum_{i,j \geq 0, i+j \leq n} a_{ij} x^i y^j = 0 \]

of degree \( n \) is assume to represent a shape. For example, ellipses, circles, pairs of straight lines, hyperbolas are implicit polynomial of degree \( n \).

The invariance of curves can be computed from implicit polynomials used to represent the curve by employing some classical invariance theory of the implicit polynomial i.e. Gröbner bases, binary forms etc. [10]. A remark can be made that since the whole non-algebraic curve is represented by an algebraic curve, the obtained invariance is global. However, we can also get local invariance from non-algebraic curve at a point \( \mathbf{x} \) by using osculating curve [86]. That is, we fit a small part around the point
x with algebraic curves and then the invariance at the point x can be computed by using differential invariance.

4.3 Joint conic invariants

In this section, we will introduce the joint invariants of two conics which have an important role in the proposed object recognition method that will be presented in the next chapter. Note that, only the case of two coplanar conics is discussed here. Let \( C_1 \) and \( C_2 \) be the images in view 1 of two non-degenerate conics in the plane \( \Pi \) as shown in figure 4.9. The conics \( C_1 \) and \( C_2 \) are mapped to conics \( C'_1 \) and \( C'_2 \) in view 2. Invariant from the projective transformation, the joint conic will form entities that will be the same when they are computed individually from both images. These invariants can be interpreted in terms of the pole-polar relation and are shown to derive from an eigenvalue problem for a matrix defined from the conic coefficient matrices [52].

Before going to the derivation of the joint conic invariant, recall that a general conic curve in the plane is defined by the quadratic form:

\[
ax^2 + bxy + cy^2 + dxz + eyz + fz^2 = 0 = \mathbf{x}^T \mathbf{C} \mathbf{x} = 0,
\]

where \( x \) is the vector of homogeneous coordinate of a point in the plane and \( \mathbf{C} \) is a \( 3 \times 3 \) symmetric matrix of conic coefficients, that is,

\[
\mathbf{C} = \begin{bmatrix}
  a & b/2 & d/2 \\
  b/2 & c & e/2 \\
  d/2 & e/2 & f
\end{bmatrix}.
\]

Under a homography \( H \), a point conic \( \mathbf{C} \) is mapped to another conic \( \mathbf{C}' \) as

\[
\mathbf{C}' = H^{-T} \mathbf{C} H^{-1}.
\]
Figure 4.9: Images of two coplanar conics on plane \( \Pi \). There images on view 1 and 2 are related by the homography \( H_{3\times3} \) induced by the plane \( \Pi \). The joint conic invariants derived from images of the two conic in either view 1 or 2 are independent from the homography.

Since the homogeneous representation of a conic is \( x^TCx = 0 \), Without losing of generality, we can derive the pair of conics invariances by deriving the invariances of the pair of the quadratic forms which is the generalized version of the pair of conics invariances.

### 4.3.1 The pair of quadratic forms invariance

Let a pair of 3-variable quadratic forms be denoted by \( 3 \times 3 \) symmetric matrices \( A = (a_{ij})_{i,j \in \{1,2,3\}} \) and \( B = (b_{ij})_{i,j \in \{1,2,3\}} \). They can be transformed by the following transformation:

\[
A' = H^{-T}AH^{-1}, \quad B' = H^{-T}BH^{-1},
\]

where the determinant of \( H \) is equal to 1. In order to derive the invariant of joint conic, let's consider the linear combination of \( A \) and \( B \) written as

\[
S = \lambda A + \mu B,
\]
and this form transforms into

\[ S' = \lambda k' + \mu b' \]

According to [69, page 193], the invariance of the pair of quadratic forms can be induced from the determinants of \( S \) or \( S' \). It follows that

\[ |S'| = |H^{-1}|^2 |S| = |S|, \]

since \( |H^{-1}| = 1 \). As a result, \( |\lambda A + \mu B| \) is invariant to the transformation \( H \):

\[ |\lambda A + \mu B| = \lambda^3 I_1 + \lambda^2 \mu I_2 + \lambda \mu^2 I_3 + \mu^3 I_4, \quad (4.7) \]

with

\[ I_1 = \frac{1}{3}(a_{11}A_{11} + a_{22}A_{22} + a_{33}A_{33} + 2a_{23}A_{23} + 2a_{13}A_{13} + 2a_{12}A_{12}) = |A| \quad (4.8) \]

\[ I_2 = b_{11}A_{11} + b_{22}A_{22} + b_{33}A_{33} + 2b_{23}A_{23} + 2b_{13}A_{13} + 2b_{12}A_{12} \quad (4.9) \]

\[ I_3 = a_{11}B_{11} + a_{22}B_{22} + a_{33}B_{33} + 2a_{23}B_{23} + 2a_{13}B_{13} + 2a_{12}B_{12} \quad (4.10) \]

\[ I_4 = \frac{1}{3}(b_{11}B_{11} + b_{22}B_{22} + b_{33}B_{33} + 2b_{23}B_{23} + 2b_{13}B_{13} + 2b_{12}B_{12}) = |B|, \quad (4.11) \]

where \( A_{ij} \) and \( B_{ij} \) are respectively the cofactors of \( a_{ij} \) and \( b_{ij} \) [69].

As a consequence, the invariance of the determinant in equation 4.7 makes four coefficients \( I_1, I_2, I_3 \) and \( I_4 \) invariant to the transformation \( H \). However, the \( I_i \) become \( |H^{-1}|^2 I_i \) and they are no longer invariant to the transformation, if \( |H| \neq 1 \). To get rid of the effect from the transformation, one can use the ratios \( I_i/I_j \) as the pair of quadratic form invariances [22]. Namely,

**Definition 6 (The pair of quadratic forms invariance).** Let \( x^T A x \) and \( x^T B x \) be two quadratic forms. Their transformation invariances can be express as \( I_i/I_j \), where \( I_i \) and \( I_j \) are the coefficients of \( \lambda^m \mu^n \), in the polynomial \( |\lambda A + \mu B| \).
4.3.2 The derivation of the pair of conics invariance

The algebraic result just obtained above is now ready to interpreted geometrically.
As mentioned in the beginning of the section, we will use the concept of pole-polar to
derive the invariant of the joint conics. According to the pole-polar relationship,
a point (pole) in the plane can be mapped onto a line (polar) with respect to a conic
C. That is,

\[ I = Cx, \]

where a point \( x \) (pole) is mapped onto a line \( I \) (polar) by the conic \( C \).

First, let \( p_1 \) be a point in the plane. The polar of \( p_1 \) with respect to conic \( C_1 \) is
\( I_{polar} = C_1 p_1 \) as shown in figure 4.10. Then, the pole of \( I_{polar} \) with respect to conic \( C_2 \)
is \( p_2 = C_2^{-1} I_{polar} \). There are two possibilities for the location of the point \( p_2 \). That
is, the point \( p_2 \) lies either outside or inside the conic \( C_2 \) [11], as shown in figure 4.10.
Thus, we have a mapping from the point \( p_2 \) to the point \( p_1 \) induced by the two conics.

\[ p_1 = (C_1^{-1} C_2)p_2 \tag{4.12} \]

To get the invariances of the joint conics, the points \( p_1 \) and \( p_2 \) are demanded to be
the same point [22]. According to the definition of projective space, the points \( p_1 \)
and \( p_2 \) are the same if \( p_1 = \lambda p_2 \). Thus, the equation (4.12) becomes the eigenvalue
problem on a matrix. Namely,

\[ \lambda p = (C_1^{-1} C_2)p \]

As a result, a common pole of the two conics is one of the three eigenvectors of the
matrix \( C_1^{-1} C_2 \). The geometric interpretation of the configuration is shown in figure
4.11. Constructed from tangency, collinearity and incidences which are all projective
Figure 4.10: A point $p_1$ in the plane is the pole of polar $l_{polar}$ with respect to the conic $C_1$. Then, the pole $p_2$ of the line $l_{polar}$ with respect to the conic $C_2$ lies either inside or outside the conic $C_2$. (a) Pole $p_2$ lies outside the conic $C_2$. (b) Pole $p_2$ lies outside the conic $C_2$.

Figure 4.11: The geometric configuration corresponding to the eigenvalue problem, i.e. the configuration when $p_1 = \lambda p_2$. Using the eigenvalue of the $C_1^{-1}C_2$ as the poles, the obtained polar is common for both conics. (a) The pole lines outside of both conics. (b) The pole lies outside one of the conics.
invariances, this special configuration is projective invariant in relation to the two given conics [52].

Since the matrix $C_1^{-1}C_2$ has three eigenvectors corresponding to the poles of the conics $C_1$ and $C_2$, let $P_1$, $P_2$ and $P_3$ be respectively these poles. These three poles form a triangle called self-polar triangle [69, page 147], as shown in figure 4.12. Its definition is that each vertex of the triangle is the pole of the opposite side of the triangle [53].

We can use self-polar triangle as the triangle of reference for the conics. That is, we map the pole $P_1$, $P_2$ and $P_3$ to $(0, 0, 1)^T$, $(0, 1, 0)^T$ and $(1, 0, 0)^T$ respectively and the conics are also transformed according to the mapping of the poles. Thus, the conic coefficient matrix is diagonal when the self-polar triangle is used as reference [52] .
Namely,

\[
C_1 = \begin{bmatrix}
  a_1 & 0 & 0 \\
  0 & c_1 & 0 \\
  0 & 0 & f_1 \\
\end{bmatrix},
\]

\[
C_2 = \begin{bmatrix}
  a_2 & 0 & 0 \\
  0 & c_2 & 0 \\
  0 & 0 & f_2 \\
\end{bmatrix},
\]

\[
C_1^{-1}C_2 = \begin{bmatrix}
  a_2/a_1 & 0 & 0 \\
  0 & c_2/c_1 & 0 \\
  0 & 0 & f_2/f_1 \\
\end{bmatrix},
\]

\[
C_2^{-1}C_1 = \begin{bmatrix}
  a_1/a_2 & 0 & 0 \\
  0 & c_1/c_2 & 0 \\
  0 & 0 & f_1/f_2 \\
\end{bmatrix},
\]

Obviously, the eigenvalues of the \(C_1^{-1}C_2\) are simply its diagonal elements, \(\lambda_1 = a_2/a_1\), \(\lambda_2 = c_2/c_1\) and \(\lambda_3 = f_2/f_1\) and the eigenvalues of the matrix \(C_2^{-1}C_1\) are the reciprocal of the eigenvalues of the \(C_1^{-1}C_2\), \(1/\lambda_i\). Furthermore, \(|C_1| = a_1c_1f_1\) and \(|C_2| = a_2c_2f_2\).

According to equations 4.9 - 4.11, the terms \(I_1, I_2, I_3\) and \(I_4\) calculated from the reduced equations of the conics are

\[
I_1 = a_1c_1f_1 \quad (4.13)
\]

\[
I_2 = a_2c_1f_1 + c_2a_1f_1 + f_2a_1c_1 \quad (4.14)
\]

\[
I_3 = a_1c_2f_2 + c_1a_2f_2 + f_1a_2c_2 \quad (4.15)
\]

\[
I_4 = a_2c_2f_2. \quad (4.16)
\]
Since the invariances of the quadratic form are the ratios of these numbers, the invariances of $C^{-1}_1C_2$ or $C^{-1}_2C_1$ are

\[
\frac{I_2}{I_1} = \frac{a_2c_1f_1 + c_2a_1f_1 + f_2a_1c_1}{a_1c_1f_1} = \frac{a_2}{a_1} + \frac{c_2}{c_1} + \frac{f_2}{f_1} = \lambda_1 + \lambda_2 + \lambda_3,
\]

\[
\frac{I_3}{I_4} = \frac{a_1c_2f_2 + c_1a_2f_2 + f_1a_2c_2}{a_2c_2f_2} = \frac{a_1}{a_2} + \frac{c_1}{c_2} + \frac{f_1}{f_2} = \frac{1}{\lambda_1} + \frac{1}{\lambda_2} + \frac{1}{\lambda_3},
\]

\[
\frac{I_1}{I_4} = \frac{a_1c_1f_1}{a_2c_2f_2}.
\]

Clearly:

\[
\text{Trace}(C^{-1}_1C_2) = \frac{I_2}{I_1},
\]

\[
\text{Trace}(C^{-1}_2C_1) = \frac{I_3}{I_4},
\]

\[
\frac{|C_1|}{|C_2|} = \frac{I_1}{I_4}.
\]

However, Trace($C^{-1}_1C_2$) and Trace($C^{-1}_2C_1$) can change their magnitude due to the scaling of the conic matrices. As a result, the normalization of the conic coefficient matrices is required to make Trace($C^{-1}_1C_2$) and Trace($C^{-1}_2C_1$) invariant from the scaling. That is,

\[
|C_1| = |C_2| = 1.
\]

As a consequence, we can conclude that

**Theorem 18.** Given two coplanar non-generate conics $C_1$ and $C_2$ which are normalized so that $|C_1| = |C_2| = 1$. Two projective invariants associated with
these two conics are

\[ I_5 = \text{Trace}(C_1^{-1}C_2), \]
\[ I_6 = \text{Trace}(C_2^{-1}C_1) \]

Alternatively, we can also prove the invariance of \( I_5 \) and \( I_6 \) to the projective transformation by showing that \( \text{Trace}(C_1^{-1}C_2) = \text{Trace}(C_1'^{-1}C_2') \) and \( \text{Trace}(C_2'^{-1}C_1') = \text{Trace}(C_2'^{-1}C_1') \). Let consider the matrix multiplication \( C_1'^{-1}C_2' \). We have

\[
C_1'^{-1}C_2' = (H^{-T}C_1H^{-1})^{-1}H^{-T}C_2H^{-1} \]
\[
= HC_1'^{-1}H^{-T}C_2H^{-1} \]
\[
= HC_1'^{-1}C_2H^{-1}.
\]

Using the fact that \( \text{Trace}(ABC) = \text{Trace}(BCA) \), we have

\[
\text{Trace}(C_1'^{-1}C_2') = \text{Trace}(HC_1'^{-1}C_2H^{-1}) \]
\[
= \text{Trace}(C_1'^{-1}C_2H^{-1}H) \]
\[
= \text{Trace}(C_1'^{-1}C_2).
\]

Let us go to the case of conics form up to a scale factor. That is, we relax the condition \( |C_1| = |C_2| = 1 \) and the invariances of the pair of conics can be derived by exploiting the terms \( I_i, i = 1, \ldots, 4 \) in equations 4.14 - 4.16. Let the conics \( C_1 \) and \( C_2 \) are scaled by \( \alpha \) and \( \beta \) respectively. In this situation, we have

\[
I_1(\alpha C_1, \beta C_2) = \alpha^2 I_1(C_1, C_2),
\]
\[
I_2(\alpha C_1, \beta C_2) = \alpha^2 \beta I_2(C_1, C_2),
\]
\[
I_3(\alpha C_1, \beta C_2) = \alpha \beta^2 I_3(C_1, C_2),
\]
\[
I_4(\alpha C_1, \beta C_2) = \beta^3 I_4(C_1, C_2).
\]
Monomials $I_1^{n_1} I_2^{n_2} I_3^{n_3} I_4^{n_4}$ are true invariants if and only if

$$\forall (\alpha, \beta) \in \mathbb{R}^2 \quad \lambda^{3n_1} I_1^{n_1} \cdot \lambda^{2n_2} \mu^{n_2} I_2^{n_2} \cdot \lambda^{n_3} \mu^{2n_3} I_3^{n_3} \cdot \mu^{3n_4} I_4^{n_4} = I_1^{n_1} I_2^{n_2} I_3^{n_3} I_4^{n_4}.$$  

It can be implied that:

$$3n_1 + 2n_2 + n_3 = 0,$$

$$n_2 + 2n_3 + 3n_4 = 0.$$  

Refer to [22], two independent solutions with $(n_1, n_2, n_3, n_4) = (1, -2, 1, 0)$ and $(n_1, n_2, n_3, n_4) = (0, 1, -2, 1)$ are chosen. Thus, they provide two invariance:

$$I_7 = \frac{I_1 I_3}{I_2^3},$$

$$I_8 = \frac{I_2 I_4}{I_3^2}.$$  

With the help of equations 4.17 - 4.17, we can represent the invariances $I_7$ and $I_8$ in terms of trace and determinant of matrix. We have

$$I_7 = \frac{I_1 I_3}{I_2^3} = \left( \frac{I_3}{I_4} \right) \left( \frac{I_1}{I_2} \right)^2 \left( \frac{I_4}{I_1} \right)$$

$$= \frac{\text{Trace}(C_2^{-1} C_1)}{\text{Trace}^2(C_1^{-1} C_2)} \frac{|C_2|}{|C_1|}$$

$$I_8 = \frac{I_2 I_4}{I_3^2} = \left( \frac{I_2}{I_1} \right) \left( \frac{I_4}{I_3} \right)^2 \left( \frac{I_1}{I_4} \right)$$

$$= \frac{\text{Trace}(C_1^{-1} C_2)}{\text{Trace}^2(C_2^{-1} C_1)} \frac{|C_1|}{|C_2|}.$$  

As a result, we have a more general version of theorem 19. That is,
\textbf{Theorem 19.} Given two coplanar non-generate conics $C_1$ and $C_2$ defined up to scale. Two projective invariants associated with these two conics are

$$I_7 = \frac{\text{Trace}(C_2^{-1}C_1) |C_2|}{\text{Trace}^2(C_1^{-1}C_2) |C_1|},$$

$$I_8 = \frac{\text{Trace}(C_1^{-1}C_2) |C_1|}{\text{Trace}^2(C_2^{-1}C_1) |C_2|}.$$
CHAPTER 5

VIEW IN Variant OBJECT RECOGNITION

Let no one say that I have said nothing new... the arrangement of the subject is new. When we play tennis, we both play with the same ball, but one of us plays it better.

Blaise Pascal

Object recognition has been ongoing research in the field of computer vision for over five decades, and has numerous applications such as biometric, image and video retrieval, surveillance and last but not least medical diagnostics. Given an object extracted from its surrounding, the goal of object recognition is to assign a unique label to the object. The object recognition is usually obstructed by changes in the object appearances. For example, translation, rotation and scaling of the objects when they are viewed from different viewpoints. To overcome these problems, our challenge is to derive an object representation and recognition scheme invariant to the variation in the object shapes when it is viewed from different viewpoints. Our work proposes a object representation invariant to projective transformation and recognition method for “planar” object. Particularly, the object shape is represented by its apparent contour considered as a close planar curve.
The remaining of this chapter is organized as following. We will first present our proposed shape representation method. After that, how to used the proposed shape representation to extract the projective invariant feature from contour will be discussed. We will end this shape by discussing about the shape matching method.

5.1 Shape representation

The first step of the general framework in shape recognition is the shape representation. Moreover, the shape representation is also the crucial step in the shape recognition. In our approach, the contour-based shape representation is employed. That is, we define shape's boundary as a planar contour. Motivated by the used of algebraic curves, many researchers have applied straight line, quadratic or cubic curve and spline etc. to represent shapes boundary such that it is applicable for object recognition application. Among many algebraic curves, piecewise straight line segment approximation of the object boundary seem to be the simplest way to model the shapes of objects. Aside from the straight line approximation approach, the use of piecewise conics appear to be the next simplest curve to model the shape boundaries. In our approach, we adopt the assumption that the boundary of a shape can be approximately represented by piecewise conics according to the following advantages:

1. Conics are projectively invariant. Namely, the mapping of a conic under the projective transformation is still conic.

2. Conics are the quadratic curves. That is, we have to deal only with second order mathematical expression in the shape representation. This leads to representation simplicity.
3. Conics are fundamental image feature. the boundaries of many man-made and natural objects can be well defined by conics. In addition, various curves can be approximated by conics.

In the sequel of an object detection method, our shape representation method starts with edge detection applied to the object silhouette by applying the Canny edge detector. One may also use other edge detectors such as Laplacian of Gaussian (LoG) or Difference of Gaussian (DoG) as alternative edge detectors. In figure 5.1 (left), we shown the silhouette image of an object shape. The example of detected edge from Canny edge detector is shown in figure 1.1 (right).

![Figure 5.1: Edge detection. Left: Silhouette image of a shape (French curve) before applying edge detection. Right: Detected edge after applying Canny edge detector to the silhouette image on the left.](image)

The contour obtained from edge detector can be represented by $N$ piecewise conics, a set of $N$ piecewise conics $\{C_i\}$, where $i = 1, \ldots, N$, is therefore used to represent of the contour. To generate the set of $N$ conics from the contour, we first evenly sample $N$ points from contour i.e. the arc length between two sample points is constant, as demonstrated in 5.2. After $N$ points were sampled from the contour, we fit a conic to each boundary point by using the point and its immediate neighbors to estimate
the parameters of piecewise conics. For instance, refer to figure 5.3, the parameters of conic $C_i$ is computed from the conic fitting with the $i^{th}$ point and its immediate neighborhood.

Such conic parameters are referred to the non-parametric (algebraic) conic equation:

$$ax^2 + bxy + cy^2 + dx + ey + fz = 0.$$  \hspace{1cm} (5.1)
To estimate the conic parameters, least-square method is often be applied for the estimation. The principle of least square estimation is to minimize the square sum of error-of-fit and such error can be categorized into two main errors, algebraic and geometric error. The geometric error is defined as the orthogonal distance from the given point to the fitted conic. A conic is expressed in terms of the implicit equation, as shown in equation 5.1, and the algebraic error is then defined as the deviation from the expected value at each point i.e. the expected value is zero for the case of equation 5.1.

To gain the numerical performance and statistical behavior of conic fitting, point normalization is an important step before going to conic fitting [8]. In our approach, we utilize two normalization method, non-isotropic and isotropic scalings. That is, applying only isotropic scaling to the sampled points decrease the similarity between two object classes; on the other hand, non-isotropic scaling increase the similarity of objects in the same class. The effects of these two scaling methods will be demonstrate in experimental result chapter (chapter 6). In our approach, we first apply the non-isotropic scaling to the set of sampled points and then apply isotropic scaling to the scaled points. An example of applying two scalings is shown in figure 5.4.

The scaling can be done as following. In the case of non-isotropic, a scaling matrix $K$ is computed from the Choleski factorization of matrix $\sum_{i=1}^{N} u_i u_i^\top$ where $u_i = (x_i, y_i, 1)^\top$ for $i = 1, \ldots, N$. That is, $\sum_{i=1}^{N} u_i u_i^\top = KK^\top$ where $K$ is upper triangular. As a result, a point $u_i$ after applying non-isotropic scaling is $\hat{u}_i = K^{-1}u_i$. Next, the isotropic scaling transform a set of points $(\hat{x}_i, \hat{y}_i)$ to a new set of points such that the centroid of the new set of points is at the origin and their average distance from the origin is $\sqrt{2}$. 

78
Figure 5.4: The example of two scaling methods. The isotropic scaling preserved the object shape. While non-isotropic scaling changes the object shape and make the sampled points dense at high curvature regions.

After the sampled point was scaled, to fit conic to shape boundary, we use the least square method to estimate the conic parameters by minimizing algebraic error produced from equation 5.1. Before applying least square to estimate the parameters of conic in equation 5.1, we have to write the equation 5.1 to be linear in the unknown parameters:

\[
\begin{bmatrix} x_i^2 & x_i y_i & y_i^2 & x_i & y_i & 1 \end{bmatrix} h = A_i h = 0,
\]

(5.2)

where \( h = [a \ b \ c \ d \ e \ f]^T \). Since the number of parameters in conic equation is five, if more than five points are given, the set of equation \( Ah = 0 \) derived from equation 5.2 is over-determined. To obtain the estimated parameter vector \( h \), singular value decomposition technique (SVD) is applied. Specifically, if \( A = UDV^T \) with \( D \) diagonal with positive diagonal entries, arranged in the descending order down the diagonal, the estimated parameter vector \( h \) is then the last column of \( V \). One may note that
any other conic fitting method can also be employed without sacrificing the proposed shape representation method. In figure 5.5, we illustrate an example from fitting conics to the contour.

In conclusion, the concept of our shape representation is based on the use of a set of conic to represent the object boundary. Such conics are approximated from the object boundary. We hypothesize that a specific object has a unique set of conics approximated from the object’s boundary. However, some shape can break our hypothesis. An example is shown in figure 5.6. Object A and B in figure 5.6 are visually dissimilar. Particulary, parts of these two objects that make these two object visually dissimilar are the parts that are approximated by the conic shown in dash line. Although these two objects are not visually similar, sets of conics used
Figure 5.6: An example of shape that can make an ambiguity in our approach.

to represent these two shapes are the same especially the conic shown in dash line. This therefore make our approach match these two shape incorrectly. However, this situation rarely occurs.

5.2 Projective invariant feature

Our proposed object recognition approach is restricted to the case that the deformation of a planar shape can be described by projective transformation when the planar shape is captured from different viewing angle. To recognize such object, features invariant to projective transformation must be extracted from the object shape. Having defined the invariance of a pair of conics in chapter 3, let us demonstrate how we can use the invariance of two coplanar conics to define the invariance descriptor of a planar shape.
Recall from chapter 4, the invariances of the two coplanar conics are

\[
I_1 = \frac{\text{trace}(C_2^{-1}C_1)}{\text{trace}^2(C_2^{-1}C_1)} \frac{|C_2|}{|C_1|}
\]

\[
I_2 = \frac{\text{trace}(C_1^{-1}C_2)}{\text{trace}^2(C_2^{-1}C_1)} \frac{|C_1|}{|C_2|}
\]

(5.3)

where \( C_1 \) and \( C_2 \) are two coplanar conics. Note that these two conics is obtained from the contour fitting and the invariant measure computed from equation 5.3 remain the same even though these two conics are transformed due to the transformation of the object. The advantage of choosing this invariant measure among other invariances of the pair of coplanar conics is related to the normalization with respect to the determinants of \( C_1 \) and \( C_2 \). Using only one measure from equation 5.3 for recognition purpose cannot be suitable because we cannot label these two conics correctly i.e. which one is \( C_1 \) or \( C_2 \) when the object shape is deformed. That is, the incorrect conic labeling yields the incorrect measure. To fix such problem, we have chosen the combination of both \( I_1 \) and \( I_2 \), \( I = \{I_1, I_2\} \) as a measure invariant to projective transformation and we use it to compute histogram discussed in detail in next paragraph.

![Figure 5.7: Two conics \( C_i \) and \( C_j \) approximated from the contour boundary define an invariance.](image)

Figure 5.7: Two conics \( C_i \) and \( C_j \) approximated from the contour boundary define an invariance.
We have had a set of $N$ coplanar conics \( \{C_i\} \), where \( i = 1, \ldots, N \). By using all possible pair of conics, a set of invariant measures evaluated from equation 5.3 will be obtained. That is, for a particular conic \( C_i \), we use every other conic \( C_j \), where \( i \neq j \), and compute the invariant measure given in equation 5.3. Hence, to match two shapes, we define a feature vector based on distribution of two possible invariance measure induced from all possible conic pairs composing the object shape rather than individually matching conic pairs obtained from two shapes. In other words, we include all two possible invariant measure while computing histogram and this make the histogram dense. One of the advantage of using histogram as the feature vector is based on the insensitivity to the perturbation of conic parameter i.e. \( C_i + \Delta_i \) where \( \Delta_i \) is the perturbation to the conic parameters. This situation can occur when noise exists on contour. Such perturbation on the conic parameters will perturb the invariant measure induced from a conic pair. However, with appropriate bin size, such perturbation will be absorbed. Moreover, the characteristics of this histogram are as follows:

1. Density - the histogram is dense because it is generated from the invariant measure of all possible conic pairs. For example, if we sample 300 points from a contour, the number of all possible pairs of conic is \( C_2^{300} \). With this characteristic, this histogram is not sensitive to noise and offers an accurate similarity measure between two histograms.

2. Projective invariance - the histogram provides projective invariant information of a shape because the histogram construct from the invariance of conic pairs approximated from the shape boundary.
The projective invariance characteristic of this histogram is the most favorable property for matching shape under projective transformation. As a result, the histogram of invariant value computed from equation 5.3 offer us a projectively invariant non-parametric description of the shape and can be used as the feature for recognition purposes.

Due to the shape of the conic sections the measure given in equation 5.3 can produce negative and position values with large magnitude. This is due to observing conics, which are close to being degenerate, such as a line. To overcome this problem, Let $D$ be the feature vector defined based on histogram of the invariant measure. The distribution is then half-wave rectified into two non-negative groups, which relates to negative and positive values of $I_3$ [13]. Each group is considered a separate distribution, $D^+$ (positive invariant group) and $|D^-|$ (negative invariant group) such that $D = D^+ - |D^-|$. The distribution variables are then log-scaled to produce the object representation pair: $\log D^+$ and $\log |D^-|$. That is, the log-scaled histogram is used as the invariant feature of the object. The example of these two distributions are demonstrate in figure 5.8.

5.3 Shape matching

In the previous section, we define the invariant feature of a shape by using two histograms. In other words, these two histograms are considered as the signature of the shape. Therefore, matching between two shape can be done by using metric or similarity (dissimilarity) measure to determine the distance or similarity (dissimilarity) between their histograms. As a consequence, two shapes that have the closest histogram will be considered as the same shape. In this section, we will discuss some
commonly used histogram dissimilarity measures and we will show the matching performance of those measure in the next section, experimental result.

Typically, a histogram \( \{h_i\} \) is defined as a function that map a \( d \)-dimensional integer vector \( i \) to a nonnegative real. Particularly, the vector \( i \) represent bin of the histogram and the associated real is a measure of the mass of the distribution that fall into the corresponding bin. For our derived histogram, \( i \) is one dimensional vector, the set of possible invariance value defined in equation 5.3 is split into \( M \) interval and \( h_i \) is the number of invariance that is in the interval indexed by \( i \) (a scalar in this case). There are several metric proposed for measure the dissimilarity between two histograms \( H = h_i \) and \( K = k_i \), where \( i = 1, \ldots, M \). In our approach, we employ the Minkowski-form distance to measure the similarity between two histogram. This distance is state as following:

\[
d_{L_r}(H, K) = \left( \sum_{i=1}^{M} |h_i - k_i|^r \right)^{1/r}.
\]
This distance is frequently referred to as the norm between two vectors. The most commonly used distance are $L_1$, $L_2$ and $L_\infty$. In particular, the $L_1$ and $L_2$ are sometime called Manhattan or Taxicab and Euclidean distance respectively in many literature.

Since the proposed invariance feature of a shape use a pair of histograms $D^-$ and $D^+$, the matching score between two shape is then defined as:

$$Sim = d_{L_r}(\log D^+_1, \log D^+_2) + d_{L_r}(\log |D^-_1|, \log |D^-_2|)$$  \hspace{1cm} (5.4)$$

where the histograms $D_1$ and $D_2$ belong to the first and second shape respectively. Lower similarity score yields a good match.
CHAPTER 6

EXPERIMENTAL RESULTS

An expert is a man who has made all the mistakes which can be made in a very narrow field.

Niels Bohr

This chapter demonstrates the performance of the proposed method for object recognition under projective transformation. According to the proposed object recognition method, two scalings methods, non-isotropic and isotropic scalings, are employed to scale points extracted from an object’s boundary. In this chapter, we also compare the implementation of our approach by using only one scaling method to demonstrate the benefits from using a cascade of scaling methods.

6.1 Data set

For tests conducted here, we used a a home generated data set consisted of 51 silhouette images. This image data set was generated by first collecting 17 different planar objects. To demonstrate the performance of the proposed method in terms of distinguishing objects in the same class, some of the 17 specific objects are in the same class. For example, images of two butterflies were used. These set of
collected images were color, gray scaled or black and with images. However, these images were them processed to be silhouette images and stored them in BMP format. To make the complete image data set, we applied two projective transformations, which were randomly generated, to the silhouette images of 17 specific planar objects. Although, the projective transformations were randomly generated, we used only a set of projective transformations that did not project parts of the objects to infinity.

Before recognizing scaled shape, the images were filtered with a Gaussian filter to reduce noise on the objects boundaries. After that, we extracted the boundaries of objects by employing Canny edge detector to the images. We then sampled points from the resulting objects boundaries. For all of the experiments demonstrated here, we sampled 300 points from objects boundaries.

Table 6.1: Test images

<table>
<thead>
<tr>
<th>Object</th>
<th>Original</th>
<th>Transformed version 1</th>
<th>Transformed version 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>French curve 1</td>
<td><img src="1" alt="Image" /></td>
<td><img src="2" alt="Image" /></td>
<td><img src="3" alt="Image" /></td>
</tr>
<tr>
<td>French curve 2</td>
<td><img src="4" alt="Image" /></td>
<td><img src="5" alt="Image" /></td>
<td><img src="6" alt="Image" /></td>
</tr>
<tr>
<td>Butterfly 1</td>
<td><img src="7" alt="Image" /></td>
<td><img src="8" alt="Image" /></td>
<td><img src="9" alt="Image" /></td>
</tr>
<tr>
<td>Object</td>
<td>Original</td>
<td>Transformed version 1</td>
<td>Transformed version 2</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>-----------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Butterfly 2</td>
<td><img src="10" alt="Image" /></td>
<td><img src="11" alt="Image" /></td>
<td><img src="12" alt="Image" /></td>
</tr>
<tr>
<td>Star</td>
<td><img src="13" alt="Image" /></td>
<td><img src="14" alt="Image" /></td>
<td><img src="15" alt="Image" /></td>
</tr>
<tr>
<td>Mouse</td>
<td><img src="16" alt="Image" /></td>
<td><img src="17" alt="Image" /></td>
<td><img src="18" alt="Image" /></td>
</tr>
<tr>
<td>Cap</td>
<td><img src="19" alt="Image" /></td>
<td><img src="20" alt="Image" /></td>
<td><img src="21" alt="Image" /></td>
</tr>
<tr>
<td>Rabbit</td>
<td><img src="22" alt="Image" /></td>
<td><img src="23" alt="Image" /></td>
<td><img src="24" alt="Image" /></td>
</tr>
<tr>
<td>Object</td>
<td>Original</td>
<td>Transformed version 1</td>
<td>Transformed version 2</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------</td>
<td>-----------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Cat 1</td>
<td><img src="25" alt="Image" /></td>
<td><img src="26" alt="Image" /></td>
<td><img src="27" alt="Image" /></td>
</tr>
<tr>
<td>Cat 2</td>
<td><img src="28" alt="Image" /></td>
<td><img src="29" alt="Image" /></td>
<td><img src="30" alt="Image" /></td>
</tr>
<tr>
<td>Dog 1</td>
<td><img src="31" alt="Image" /></td>
<td><img src="32" alt="Image" /></td>
<td><img src="33" alt="Image" /></td>
</tr>
<tr>
<td>Dog 2</td>
<td><img src="34" alt="Image" /></td>
<td><img src="35" alt="Image" /></td>
<td><img src="36" alt="Image" /></td>
</tr>
<tr>
<td>Big water jar</td>
<td><img src="37" alt="Image" /></td>
<td><img src="38" alt="Image" /></td>
<td><img src="39" alt="Image" /></td>
</tr>
</tbody>
</table>
6.2 Use of only non-isotropic or isotropic scaling

In this section, we illustrate the benefit of using non-isotropic and isotropic scaling in our object recognition approach as mentioned in chapter 5. In this experiment, the interval of histogram is between -100 and 100 and bin width is 0.3. We employed the $L_2$ norm to match two histograms. The matching score between objects are illustrated in terms of confusion matrix as shown in figure 6.1. Each elements of the confusion matrix correspond to the matching between two objects. Namely, the value in row $i$ and column $j$ of the confusion matrix is the matching score between objects $i$ and $j$. Note that low similarity value indicates a good match. The similarity value is represented in terms of color tone where dark blue indicates low similarity value (good match) while dark red indicates high similarity value (bad match).
Comparing the similarity values in off-diagonal area of the confusion matrix shown in figure 6.1.a and 6.1.b, one can see that the similarity values in the off-diagonal area of the confusion matrix generated from using only isotropic scaling seem to increase from those of non-isotropic scaling, while the similarity value in diagonal area of the confusion matrix from using only non-isotropic scaling seem to reduce. It can be concluded that applying only isotropic scaling to the sampled points decrease the similarity between two object classes; on the other hand, non-isotropic scaling increase the similarity of objects in the same class. The result from this experiment suggests us to use a cascade of both scaling methods.
Figure 6.1: Matching result from using only isotropic or non-isotropic scaling.
6.3 Use of both scalings in the proposed method

In the previous section, we show that non-isotropic scaling increase the similarity in an object class, while isotropic scaling increase dissimilarity between object classes. In this section, we demonstrate the result from using both of these two scaling methods. In this experiment, the bin width of a histogram is 0.3 and the interval of the histogram is between -100 and 100.

Table 6.3 shows the matching result after performing an the experiment by using both non-isotropic and isotropic scalings. If an object matches with objects in the same class, we shown it in bold face number; while incorrect match is shown in italic number. The first matched object is an object having lowest similar value when do matching with an considered object. The performance of the system can be evaluated by computing the true positive (correct matching) and false negative (incorrect matching) values shown in table 6.4.

We can also demonstrate the performance of the object recognition system by using the clusters of matched object. The clusters of matched objects can be formed by grouping objects having the best matching. For example, from table 6.3, the object number 1 matches with object number 3 and object number 3 matches with object number2; the object numbers 1, 2 and 3 are therefore in the same cluster. The result from clustering is presented in table 6.5. Note that object in the same class should be in the same cluster. To indicate an object begin in a wrong cluster, we enclose it in a box i.e. the shape of a dog in cluster number 2.
<table>
<thead>
<tr>
<th>Shape</th>
<th>1st matched</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>15</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>18</td>
<td>16</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>21</td>
<td>20</td>
</tr>
<tr>
<td>22</td>
<td>44</td>
</tr>
<tr>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>24</td>
<td>23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shape</th>
<th>1st matched</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>27</td>
</tr>
<tr>
<td>26</td>
<td>12</td>
</tr>
<tr>
<td>27</td>
<td>25</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>29</td>
<td>28</td>
</tr>
<tr>
<td>30</td>
<td>28</td>
</tr>
<tr>
<td>31</td>
<td>33</td>
</tr>
<tr>
<td>32</td>
<td>43</td>
</tr>
<tr>
<td>33</td>
<td>31</td>
</tr>
<tr>
<td>34</td>
<td>10</td>
</tr>
<tr>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>36</td>
<td>35</td>
</tr>
<tr>
<td>37</td>
<td>39</td>
</tr>
<tr>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>39</td>
<td>38</td>
</tr>
<tr>
<td>40</td>
<td>34</td>
</tr>
<tr>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>42</td>
<td>41</td>
</tr>
<tr>
<td>43</td>
<td>45</td>
</tr>
<tr>
<td>44</td>
<td>42</td>
</tr>
<tr>
<td>45</td>
<td>43</td>
</tr>
</tbody>
</table>

Table 6.3: Matching result. The bold face number indicate correct matching, while italic number indicate false matching.
Table 6.4: Performance of the proposed recognition method

<table>
<thead>
<tr>
<th>True positive (TP)</th>
<th>False negative (FN)</th>
<th>Hit rate (TP/(TP + FN)) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>9</td>
<td>80%</td>
</tr>
</tbody>
</table>

Since clustering is a binary classification model, the most commonly used model performance metric is accuracy defined by the following formula:

\[ \text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \]

where TP, TN, FP and FN are the True Positive, True Negative, False Positive and False Negative respectively. True positive is defined as assigning an object to its correct class while true negative is defined as not assigning the object to its incorrect class. Next, not assigning an object to its correct class is called false positive, while assigning the object to its incorrect class is called false negative. To illustrate how to compute each terms in each clusters, let us the cluster number 4 which is the butterfly shape cluster as an example. Since we assigned two butterfly shapes in the same cluster and did not assign one butterfly shape in this class, true positive is then two and false positive is one. Because a cat shape is in this cluster, false negative is then one. Finally, true negative is forty one since we did not assign other forty shape that is not the butterfly shape to this cluster. The result from computing these four values for each cluster is shown in table 6.6. The average accuracy for clustering the data set by using the proposed method is about 98%. The performance for the proposed object recognition method can be also assessed using the measure of True
Positive Rate (TPR) or hit rate defined as TP/(TP+FN) and the average TPR for the proposed method on this date set is about 89%.

Table 6.5: Cluster result

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Matched shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster</td>
<td>Matched shape</td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
</tr>
<tr>
<td>3</td>
<td><img src="image1" alt="Matched shape" /></td>
</tr>
<tr>
<td>4</td>
<td><img src="image2" alt="Matched shape" /></td>
</tr>
<tr>
<td>5</td>
<td><img src="image3" alt="Matched shape" /></td>
</tr>
<tr>
<td>6</td>
<td><img src="image4" alt="Matched shape" /></td>
</tr>
<tr>
<td>Cluster</td>
<td>Matched shape</td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
</tr>
<tr>
<td>7</td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td>8</td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td>9</td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td>10</td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td>11</td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td>12</td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td>Cluster</td>
<td>Matched shape</td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
</tr>
<tr>
<td>13</td>
<td><img src="image1" alt="Illustrations" /></td>
</tr>
<tr>
<td>14</td>
<td><img src="image2" alt="Illustrations" /></td>
</tr>
<tr>
<td>15</td>
<td><img src="image3" alt="Illustrations" /></td>
</tr>
</tbody>
</table>

As one can observe from table 6.5, some cluster contain incorrect matching result. This means that the shape features (histograms in the proposed method) of two mismatched shapes are similar although these two shape are not visually similar. Note that our proposed object recognition method uses a set of conics composing an object boundary to compute the invariant shape feature. If two shapes have some parts such
Table 6.6: Clustering performance

<table>
<thead>
<tr>
<th>Cluster</th>
<th>True Positive</th>
<th>False Positive</th>
<th>False Negative</th>
<th>True Negative</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>39</td>
<td>0.93</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>41</td>
<td>0.96</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>42</td>
<td>0.98</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>42</td>
<td>0.98</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>42</td>
<td>0.98</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>42</td>
<td>0.98</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>40</td>
<td>0.93</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>41</td>
<td>0.96</td>
</tr>
<tr>
<td>Average</td>
<td>2.53</td>
<td>0.47</td>
<td>0.47</td>
<td>41.53</td>
<td>0.98</td>
</tr>
</tbody>
</table>
that conics composing these parts are projectively equivalent, the invariant shape features can be similar. For example, let's consider shape number 35 and 49 in cluster 14. We first approximately align these two shapes by using only scaling and rotation. Actually, we can use any transformation i.e. projective to align two shape. As shown in figure 6.2, one can observe that some parts of these objects are very similar and some conic composing these two shapes are the same. As a result, conics composing these similar parts make the ambiguity in shape representation which affect directly the shape invariant feature.

Figure 6.2: Shape alignment
6.4 Experiment with biological vision

To compare the matching result from the proposed method with biological vision, we also performed two experiments with biological vision by using the same data set as we used to test the proposed method. Note that images used for biological vision experiment are the boundaries of the object in the data set since we use only the boundaries of objects in the proposed method. Moreover, all examinees who did the experiment given permission to show the result from their test. In each experiments, examinees were asked with different questions to group the images of objects. Moreover, only 3 minutes were given for grouping those images and those examinees did not know the exact amount of objects and number of images for each objects. Note that each examinee participated in only one experiment.

6.4.1 The first experiment on biological vision

In the first experiment, five examinees were asked with a question “group images of the same object”. Three examinees asked questions during the time they were doing the experiment. Such question was only about the meaning of the object i.e. “does these two dogs are considered as the same object?”, see figure 6.3, and advices were given depended on the questions i.e. “both two dogs are different”.

Results from the experiment showed that the examinees who asked questions about the meaning of object during the experiment grouped the images correctly. However, two examinee cannot group the object number 40 in its correct cluster. That is, they thought that the object in image number 40 is not the same object as in image number 41 and 42. For other two examinees who did not ask any question
Figure 6.3: Some question arose during the first experiment. The images shown here are for the question “Are these two dogs considered as the same object?”

during the experiment, they group images in semantical way. Namely, they grouped the images of butterflies or dogs together, as shown in table 6.7.

As a result, we can infer from this experiment that examinees try to group the image in semantical way such as some examinees who grouped the images of butterflies and dogs together, or in terms of geometric i.e. using object’s boundary for grouping object. They chose a way to group the images based on the instruction or constraint they got. The use of geometric information for grouping the object is similar to the proposed object recognition method using the object’s boundary.

6.4.2 The second experiment on biological vision

The second experiment was performed with 8 examinees and the procedure of this experiment was as following:

1. Each examinee was introduced with the situation of viewing a unique object from different angle. Specifically, the images of an object can be changed due to different viewing angles.

2. The examinees were then asked with a question “group the image which you think that they are the images of the same object”.

104
Table 6.7: Grouping result from the examinees who did not ask any equation during the experiment.

<table>
<thead>
<tr>
<th>Object</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Butterfly</td>
<td><img src="image" alt="Butterfly Images" /></td>
</tr>
<tr>
<td>Dog</td>
<td><img src="image" alt="Dog Images" /></td>
</tr>
</tbody>
</table>

The explanation about the object viewing given before the examinees did the experiment acted as a constraint for the examinees such that they must group the images of a unique object and also told the examinees that they must used only geometric information (object's boundaries) for grouping the images. Moreover, it made the examinees more understand on the task that they had to perform since there was no examinees ask for further information about the experiment when they were doing the experiment.

The result from this experiment showed that three examinees can group the images correctly; while other five examinees cannot group some images correctly. Such incorrect grouping are shown in table 6.8 and 6.9. It is interesting that the proposed
method can group the object number 40, 41 and 42 together in the correct group; while the biological vision considered the object number 40 not in the same group with object number 41 and 42 both in the first and second experiments. This situation normally happens when the transform versions of the object’s image is very different from the original one i.e. very oblique image of a planar object. The proposed object method overcomes this problem since it use a shape feature that is invariant from viewing in different angle, while biological vision may not do so.

Table 6.8: First incorrect image grouping

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Matched shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>![Shape 1]</td>
</tr>
<tr>
<td>2</td>
<td>![Shape 2]</td>
</tr>
</tbody>
</table>
Table 6.9: Second incorrect image grouping

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Matched shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>![Diagram of shape 1]</td>
</tr>
<tr>
<td>2</td>
<td>![Diagram of shape 2]</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


