INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or “target” for pages apparently lacking from the document photographed is “Missing Page(s)”. If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of “sectioning” the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University Microfilms International
300 N. Zeeb Road
Ann Arbor, MI 48106
Bamieh, Bassam Abbas

GENERAL MOMENT INVARIANTS AND THEIR APPLICATION TO THREE-DIMENSIONAL OBJECT RECOGNITION FROM A SINGLE IMAGE

Rice University

University Microfilms International 300 N. Zeeb Road, Ann Arbor, MI 48106

M.S. 1986
 PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark √.

1. Glossy photographs or pages 
2. Colored illustrations, paper or print 
3. Photographs with dark background √
4. Illustrations are poor copy 
5. Pages with black marks, not original copy 
6. Print shows through as there is text on both sides of page 
7. Indistinct, broken or small print on several pages 
8. Print exceeds margin requirements 
9. Tightly bound copy with print lost in spine 
10. Computer printout pages with indistinct print 
11. Page(s) ________ lacking when material received, and not available from school or author.
12. Page(s) ________ seem to be missing in numbering only as text follows.
13. Two pages numbered ________. Text follows.
14. Curling and wrinkled pages 
15. Dissertation contains pages with print at a slant, filmed as received 
16. Other 

University Microfilms International
RICE UNIVERSITY

GENERAL MOMENT INVARIANTS AND THEIR APPLICATION TO
3D OBJECT RECOGNITION FROM A SINGLE IMAGE

by

BASSAM ABBAS BAMIEH

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

MASTER OF SCIENCE

APPROVED, THESIS COMMITTEE:

R. J. P. De Figueiredo, Professor of
Electrical and Computer Engineering
and of Mathematical Sciences,
Director

J. B. Pearson, Professor of Electrical
and Computer Engineering

P. Varma, Assistant Professor of
Electrical and Computer Engineering

Houston, Texas
May, 1986
ABSTRACT

A method for the recognition and localization of 3D objects given a single image is developed and presented. The method is a library matching technique, that is, given an image, it is compared to a stored library of known world objects. Different notions are used to obtain a suitable description of 3D objects and their 2D images, these include moment invariants and subgraph isomorphism. The recognition problem is formulated and solved for rigid planar patches (RPP's) and polyhedra. The description of an RPP is in terms of its moment invariants, and thus recognition is a matter of comparing feature vectors made up of these invariants. The description of a polyhedron is in terms of an attributed graph of which the nodes represent the individual faces (RPP's) of the polyhedron. In this case recognition becomes a problem of subgraph isomorphism.

The critical part of these representations is the use of general moment invariants, which allow the classification of image functions modulo affine transformations of the image plane. A development of moment invariants using tensor notation is included.
Acknowledgements

I wish to acknowledge my advisor Dr. DeFiguiredo for his help in the duration of this research. Also Dr. Pearson and Dr. Varman of the Electrical and Computer Engineering department for being on my thesis committee. I am grateful to NASA and the National Science Foundation for supporting this research.

Most of all, I would like to thank my parents who made it all possible.
Table of Contents

Chapter 1. Introduction .................................................................................. 1

Chapter 2. Tensors, Moments, and General Invariants .......................... 4

2.1. Tensors and Moments ........................................................................ 4

2.2. General Invariants and their Construction ...................................... 9

Chapter 3. Application to 3D Object Recognition .................................. 16

3.1. Recognition and Localization of a Rigid Planar Patch (RPP) ........ 17

3.2. Recognition and Localization of Polyhedral Objects by the Subgraph isomorphism Algorithm ......................................................... 25

3.3. Experimental Results .......................................................................... 32

Chapter 4. Discussion and Conclusions .................................................... 40

Appendix A. Calculation of the Moments of a Polygon ......................... 42
CHAPTER 1

Introduction

By general moment invariants is meant, moment invariants which are invariant under affine transformations (linear transformations and translations) of the 2D plane. The term general is used because, as far as the author is aware, aside from the original paper of Hu [1], only invariants to orthogonal transformations have been mentioned in the literature [2]. Hu[1] notes the existence of general invariants but only uses the rotational invariants. Moment invariants have been used as features for 2D object recognition, where the object could be observed at any orientation in the plane. The approach presented here extends the use of moment invariants to the recognition of 3D objects from a single image, without the use of observed 3D data. It should be noted that such an approach is quite different from that of Sadjadi and Hall [9] who use 3D moment invariants for which 3D data is required. It is also different from that of Dudani, et al. [2] who used silhouettes of objects at many different viewing angles in the library for recognition purposes. Lozano-Perez and Grimson [7] have solved a similar recognition problem for planar shapes and polyhedra. But they use 3D data, and because the data is sparse their method is independent of the connectivity of the faces of an object, which is not the case in the approach presented here. Cyganski and Orr[6] used a tensor method to obtain the parameters of the affine transformations relating images of RPP objects under 3D rotations. Their method is used here for calculating the 2D transformation parameters after recognition. The literature on
shape recognition is vast, only authors that have used invariants for 3D object recognition are mentioned.

The method outlined in the present paper is for classification of a 3D object from its projection on an imaging plane. The library of possible objects contains 3D descriptions of these objects which are then "compared" to the 2D image of the observed object. This approach is facilitated by the following fact: Assume that images are formed by parallel projection; then, if a Rigid Planar Patch (RPP) object is undergoing rigid body motion in 3D, its image undergoes affine transformations in the plane (see fig. 1). And we utilize this fact by resorting to general moment invariants on the object image. This makes full use of the geometry inherent in the problem. This approach is extended for polyhedra by considering each face as an RPP object, and the surface of a polyhedron as an attributed graph with faces as nodes. The edges in this graph represent the connectivity of the faces. It is clear that a projection (an image) of such an object can be represented by a subgraph of the original full graph. Here, we assume that from an image of a polyhedron one can extract the wire-frame by edge detection (which is a reasonable assumption under most common lighting conditions). After the wire-frame is extracted, the attributed graph is extracted in the same manner as for 3D objects. The attributes in both cases are the general moment invariants. With this formulation the recognition algorithm becomes a subgraph isomorphism problem. It will be shown that as a consequence of the recognition process, the attitude and location of the recognized object are determined.

With suitable modifications, the concepts presented here can be applied to 3D objects more general than polyhedra, provided that these objects have a well defined
set of edges. The aim of this thesis is twofold: to present a consistent development of general moment invariants for a theoretical background, and to illustrate their applications in the recognition of RPP and polyhedral objects. Moment invariants are a special case of algebraic invariants. In chapter 2, the part of this theory which applies to moment invariants is briefly presented. This is done using a tensor formulation, because it clearly illustrates the transformation properties of moments. For this purpose, section 2.1 includes a brief introduction to the needed concepts from tensor theory.

In chapter 3, the 3D recognition problem is formulated as mentioned, and solved for the case of RPP's and polyhedra. Simulation results are also included.
CHAPTER 2

Tensors, Moments, and General Invariants

2.1. Tensors and Moments

Invariants are used in situations where classification of objects is to be done irrespective of a set of transformations. In our case, we want to classify objects irrespective of affine transformations of the plane. Such a situation arises when imaging a planar 3D object with arbitrary orientation as will be illustrated in chapter 3. Since tensors arise naturally when dealing with such transformations, we will be resorting to them shortly. At this point we should note that when central moments are used (as will be explained later in this thesis), the translation components of the affine transformation are canceled and only the linear part is left. Since we will be using central moments, we will consider in the sequel only linear transformations (with the understanding that affine transformations reduce to linear ones by using central moments). Also, we will refer to "general invariants" simply as "invariants".

Let the dimension of the underlying space be $n$ (in our case we consider the plane, and hence $n=2$), and let the coordinates in that space be $x^1, \ldots, x^n$, where a superscript denotes a particular variable not a power of it. Consider linear transformations such as:

$$
\hat{x}^i = \sum_{j=1}^{n} \Omega_j^i x^j, \quad i = 1, \ldots, n, \quad \Delta = |\Omega_j^i| \neq 0,
$$

(1)
where $Q^i_j$ are the elements of the transformation matrix, and $\Delta$ their determinant. It will be very useful to adopt the Einstein summation convention in which the transformation can be rewritten as

$$\hat{x}^i = Q^i_j x^j,$$

where it is understood that the tensors are to be summed over repeated indices. A **contravariant vector** $x^i$ is what is generally known as a vector and transforms in accordance with eq.(2). A generalization of the contravariant vector is the **contravariant tensor**, denoted by $A$, of order $k$ which is a collection of $n^k$ components each of which is denoted by $A^{i_1...i_k}$ where every index ranges from 1 to $n$, and the components transform in accordance with:

$$\hat{A}^{\alpha_1...\alpha_k} = Q_{i_1}^{\alpha_1} \cdots Q_{i_k}^{\alpha_k} A^{i_1...i_k}.$$  

(3)

In this context, each component is understood to be a function of the underlying space coordinates $x^i$. Note that a contravariant vector is a contravariant tensor of order 1. Let $P^i_j$ denote the elements of the inverse transformation to $Q^i_j$. Then, a **covariant tensor** $B$ of order $s$ is a collection of components which transform as:

$$\hat{B}_{\alpha_1...\alpha_s} = P^{i_1}_{\alpha_1} \cdots P^{i_s}_{\alpha_s} B_{i_1...i_s}.$$  

(4)

In other words, they transform in an opposite manner to contravariant tensors. A **mixed tensor** $C$ of covariance $s$ and contravariance $r$ is one which transforms as:

$$\hat{C}_{\beta_1...\beta_r}^{\alpha_1...\alpha_s} = Q_{\beta_1}^{i_1} \cdots Q_{\beta_r}^{i_r} P^{\alpha_1}_{i_1} \cdots P^{\alpha_s}_{i_s} C_{i_1...i_r}^{\alpha_1...\alpha_s}.$$  

(5)
There are several operations which can be performed on tensors. Tensors of the same covariance and contravariance can be added by simply adding the corresponding components and a tensor of equal covariance and contravariance is obtained. Tensors of any order can be multiplied. If $A$ and $B$ are two tensors of covariances $sp$ and contravariances $rk$ respectively, their product $C$ is of covariance $s+p$ and $r+k$. The components of $C$ are obtained by

$$C_{\alpha_1 \cdots \alpha_p}^{\beta_1 \cdots \beta_k} = A_{\beta_1 \cdots \beta_k}^{\alpha_1 \cdots \alpha_p} B_{\gamma_1 \cdots \gamma_k}^{\mu_1 \cdots \mu_p}. \quad (6)$$

Two operations which are important for constructing invariants are contraction and alternation. Both are operations performed on a single tensor. As a matter of notation, the number of superscripts and subscripts will indicate the contravariance and covariance of the tensor respectively. A contraction of a mixed tensor $A_{\mu \nu \rho \sigma}$ to form a tensor of lower covariance and contravariance is done by equating some of its superscripts and subscripts and summing over them. For example if we contract $A$ with respect to the indices $kn$ and $po$ we get:

$$B_{\nu \rho}^{ij} = A_{\lambda \mu \rho \nu}^{ijp}. \quad (7)$$

where repeated indices are summed over by the convention. Note that one can only sum over repeated indices for which one is a superscript and the other a subscript. Thus, only mixed tensors can be contracted.

One consequence of the tensor definition is that an invariant is simply a tensor of zeroth order, since it is unaffected by transformations. Thus, if a mixed tensor has equal covariance and contravariance order, by total contraction (with respect to
all indices) it becomes a tensor of zero\textsuperscript{th} order, i.e. an invariant (possibly zero).

There are certain special types of tensors which we will use, one such type being the symmetric tensor. A totally symmetric tensor is a tensor where the components are identical for any permutation of the indices, for example \( A^{123\ldots k} = A^{213\ldots k} \) and so on. Another type is the unit 2-vector \( e_{ij} \) which is a covariant tensor of order 2 and defined by:

\[
\begin{align*}
e_{ij} &= 0 \quad \text{for } i = j \\
e_{12} &= 1 \\
e_{21} &= -1.
\end{align*}
\] (8)

Instead of defining the operation of alternation for general tensors, we only define it for the special case we use. We consider only contravariant tensors for which \( n=2 \). A total alternation of a contravariant tensor \( A \) is a contraction of the tensor after multiplying it by \( e_{ij} \) as follows:

\[
B^{klmn} = e_{ij} A^{klijmn}.
\] (9)

This operation is useful in forming invariants as will be shown in section 2.2.

We now define moments and show they are a special class of tensors. The standard definition of moments is

\[
m_{pq} = \iint (x^1)^p (x^2)^q f(x^1, x^2) \, dx^1 dx^2,
\] (10)

where \( x^1, x^2 \) are the coordinates in the plane, and \( f(x^1, x^2) \) is the image function which is of bounded variation and vanishes outside of a bounded set. The sequence of numbers \( m_{pq} \) provides a unique representation of such a function, i.e. all moments
of two functions \( f_1 \) and \( f_2 \) are equal if, and only if, \( f_1 = f_2 \) almost everywhere. The number \( p+q \) is called the order of the moment \( m_{pq} \). The center of geometry of a function is defined as the point with coordinates \( \bar{x}^1 = \frac{m_{10}}{m_{00}} \) and \( \bar{x}^2 = \frac{m_{01}}{m_{00}} \). The central moments are then defined as:

\[
m_{pq} = \iiint (x^1 - \bar{x}^1)^p (x^2 - \bar{x}^2)^q f(x^1, x^2) \, dx^1 \, dx^2.
\] (11)

These moments have the property that they are invariant to translation. Thus, by using central moments an affine transformation is reduced to its linear part. From this point on, we will consider only central moments unless explicitly mentioned. In other words, we will assume that the center of coordinates is at the center of geometry.

When the underlying space (the plane) is linearly transformed, the moments of a given order get transformed as a tensor. The following definition of moment tensors clarifies this transformation property

\[
M^{ijk\ldots} = \iiint x^i x^j x^k \ldots f(x^1, x^2) \, dx^1 \, dx^2,
\] (12)

where \( ijk \ldots \) take the values of 1 or 2. To check that \( M \) is actually a contravariant tensor, consider a transformation of space:

\[
\bar{x}^i = Q^i_j x^j \quad \text{or} \quad x^i = P^i_j \bar{x}^j.
\] (13)

The new moments are:
\[ \hat{M}^{ijk} = \int z^i z^j z^k \cdots f(P_{\alpha}^1 \alpha, P_{\beta}^2 \beta) \, dx^1 dx^2 \]
\[ = \int Q_{\alpha}^i Q_{\beta}^j Q_{\gamma}^k \cdots f(x^1, x^2) \frac{1}{\Delta} \, dx^1 dx^2 \]
\[ \text{or} \]
\[ \hat{M}^{ijk} = Q_{\alpha}^i Q_{\beta}^j Q_{\gamma}^k \cdots M^{\alpha\beta\gamma} \frac{1}{\Delta}, \]

where \( \Delta \) is the Jacobian \( |Q_j^i| \). This factor is accounted for by noting how the zero\(^{th} \) order moment \( \mu \) transforms:

\[ \hat{\mu} = \mu \frac{1}{\Delta}. \]

Thus if we consider normalized moments \( m^{ijk} = M^{ijk}/\mu \) eq.(14) becomes

\[ \hat{m}^{ijk} = Q_{\alpha}^i Q_{\beta}^j Q_{\gamma}^k \cdots m^{\alpha\beta\gamma}, \]

which shows that \( m \) is indeed a tensor. Furthermore, its contravariant order is equal to its order as a moment (i.e. \( p+q \)). At this point we should compare the standard and tensor notation for moments. For the second order moments \( m^{11} = m_{20}, m^{22} = m_{02} \) and \( m^{12} = m^{21} = m_{11} \). The moment tensor is a totally symmetric tensor, and thus for order \( n \), it has \( n+1 \) distinct components. The first four moment tensors are:

- \( 0^{th} \) order \( \mu \)
- \( 1^{st} \) order \( (m_1, m_2) \)
- \( 2^{nd} \) order \( (m^{11}, m^{12}, m^{22}) \)
- \( 3^{rd} \) order \( (m^{111}, m^{112}, m^{122}, m^{222}) \).

### 2.2. General Invariants and their Construction

As seen in the previous section, the moment tensors \( m^i, m^{ij} \), group the moments in accordance with the way they transform. The goal now is to form invariants from
these tensors. In this paper we will only outline the main theorems of algebraic invariant theory and how it relates to this special case. The theory is involved, and the proofs of these theorems are nontrivial. Any attempt to investigate it more closely would take us far afield. For further detail, we refer the interested reader to Gurevich [10].

As noted earlier an invariant is a tensor of zero\textsuperscript{th} order. Thus, the way moment invariants are formed is by contracting the moment tensors down to the zero\textsuperscript{th} order. We start with some facts which apply to any tensor. Let $A$ be any tensor of any mixed order with components $a_1, ..., a_k$. Denote by $a'_1, ..., a'_k$ the components after a transformation of this tensor.

**Definition:** A relative invariant of the tensor $A$ is a function of its components $J(a_1, ..., a_k)$ such that:

$$J(a'_1, ..., a'_k) = g(Q^i_j) J(a_1, ..., a_k),$$

(18)

where $g(Q^i_j)$ is any function of the parameters of the underlying transformation. An absolute invariant is one for which $g(Q^i_j) = 1$.

**Definition:** An algebraic invariant of the tensor $A$ is an invariant which is an algebraic function of the tensor components.

In the sequel we restrict our attention to algebraic invariants only, because they exist in very special forms as illustrated by the next theorem, but we first need a definition.

**Definition:** An integral rational homogeneous function is a rational function with integer coefficients, and is homogeneous in every term.
Theorem 1: Every algebraic invariant of a tensor $A$ is an integral rational homogeneous function of the coefficients. Furthermore, the invariant transforms as

$$J(a_1',...a_k') = \Delta^w J(a_1,...a_k),$$

where $\Delta$ is the determinant of the transformation and $w$ is always an integer which is called the weight of the invariant.

So far we have not constructed invariants. We do that now.

Definition: The order of an invariant is the sum of the powers in any of its terms.

Since invariants are homogeneous functions, the order is well defined. The next theorem specifies how invariants are to be constructed from a tensor.

Theorem 2: Every algebraic invariant of weight $w$ and order $d$ of the tensor $A$ is a linear combination of terms each of which has been constructed from $A$ by $d-1$ tensor multiplications and $|w|$ total alternations.

Thus, we need only be concerned with those essential terms, since all others are determined by them. Theorem 2 is very useful because it gives an algorithm for constructing invariants. As an example, we show here the construction of an invariant from the second order moment tensor $m^{ij}$. This invariant is of order 2 and weight 2. Thus it must be constructed by 2 total alternations of the product $m^{ij}m^{kl}$ such as:

$$J_1 = \varepsilon_{ik}\varepsilon_{jp} m^{ij} m^{kp},$$

which after performing the summation comes out to

$$= 2 \left( m^{11}m^{22} - (m^{12})^2 \right),$$

(20)
which in standard notation is written as:

\[ m_{20} m_{02} - m_{11}^2. \]  \hspace{1cm} (21)

Of course the factor of two in eq.(20) is irrelevant since it is invariant also. Note that other choices of the alternation might yield a zero (for example \( \epsilon_i \epsilon_k r_m^i m^j m^k \)), which is an invariant, but vacuously so.

All algebraic invariants of the moment tensors (i.e. moment invariants) can be constructed in the same manner. The question to be posed now is: For a given tensor what is the "minimal" set of invariants that determine all other invariants? the notion of minimality is formalized in the following definition and theorem:

**Definition:** For a given tensor \( A \), a complete system of invariants \( J_1, J_2, \ldots \) is such that (i) every invariant of \( A \) is an integral rational function of \( J_1, J_2, \ldots \) and (ii) none of the invariants \( J_1, J_2, \ldots \) is an integral rational function of the remaining invariants.

It is a fact that[10] if given two tensors \( A \) and \( B \), and their complete systems of invariants are exactly equal, then there exists a linear transformation which transforms \( A \) into \( B \). In this case we say that the tensors \( A \) and \( B \) are equivalent modulo linear transformations.

On finding these complete systems of invariants, we have:

**Theorem 3 (Hilbert):** A complete system of invariants of any tensor consists of a finite number of invariants.

We can find all of these invariants by the process of Theorem 2, but the difficulty lies in determining which set of invariants forms a complete system. It can
be shown that the complete system for the second order moment tensor consists of one invariant only, and similarly the third order moment tensor has one invariant. The complete systems of the fourth and fifth order moments consist of two and three invariants respectively. These complete systems are shown in Table 1 for the 2\textsuperscript{nd} through 4\textsuperscript{th} order moment tensors. Formulas for higher order moments can be extracted from Salmon [5]. It should be noted that the $J_i$'s are not absolute invariants. To get an absolute invariant $I_i$ from an invariant $J_i$ we normalize using the determinant of the transformation which can be obtained from the zeroth order moment:

$$I_i = J_i \mu^w,$$

(22)

where $w$ is the weight of the invariant.

<table>
<thead>
<tr>
<th>Order of moments</th>
<th>complete system</th>
<th>weight of invariant</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$J_1 = m_{02}m_{20} + m_{11}$</td>
<td>-2</td>
</tr>
<tr>
<td>3</td>
<td>$J_2 = (m_{03}m_{30} - m_{21}m_{12})^2 - 4(m_{03}m_{12} - m_{21}^2)(m_{21}m_{30} - m_{12}^2)$</td>
<td>-6</td>
</tr>
<tr>
<td>4</td>
<td>$J_3 = m_{40}m_{04} - 4m_{31}m_{13} + 3 m_{22}^2$</td>
<td>-4</td>
</tr>
<tr>
<td></td>
<td>$J_4 = m_{40}m_{22}m_{04} - 2 m_{31}m_{22}m_{13} - m_{40}m_{13}^2 - m_{04}m_{31}^2 - m_{22}^3$</td>
<td>-6</td>
</tr>
</tbody>
</table>
Our ultimate aim in using invariants is to determine which image functions are equivalent modulo linear transformations of their domain. Since an infinite sequence of moments uniquely defines its image function, and since moments are tensors, then two image functions \( f \) and \( g \) are equivalent if and only if their corresponding sets of tensors \( (M^1_f, M^2_f, \ldots) \) and \( (M^1_g, M^2_g, \ldots) \) are equivalent modulo linear transformations tensor-by-tensor.

In any practical situation we cannot obtain the infinite set of tensors and thus the infinite set of invariants needed for complete classification. We thus use the following approximations:

(i) Note that the moments of an image function \( f \) are its expansion coefficients with respect to the basis \( 1, x, y, x^2, xy, y^2, \ldots \). Thus we consider two image functions equal if their moments are equal up to the fourth order moments. As shown later in the simulations this approximation turns out to be sufficient for our purpose.

(ii) From (i) we then conclude that two image functions are equivalent modulo linear transformations if their moment tensors up to fourth order tensors are equivalent under linear transformations. This second equivalence is tested by the complete sets of invariants of table 4.

Thus in summary we have the following fact:

**Fact:** If the moment expansions up to fourth order are considered a sufficient approximation of image functions, then two image functions are equivalent modulo linear transformations of their domains if and only if their invariants of table 4 are all equal.

Since digitized pictures are used in practice, we do not demand that the invariants be exactly equal. If we consider the invariants of one image function as a point
in $\mathbb{R}^n$ (called a feature vector), we then demand for equivalence that the feature vectors be within an $\varepsilon$ neighborhood of each other, where $\varepsilon$ is a prespecified threshold level.
CHAPTER 3

Application to 3D Object Recognition

The recognition problem can be stated precisely in the following setup. By a 3D object it is meant a full description of the surface in 3-space which bounds the object. Assuming this is given, we call this the 3D representation of the object. Let such representations be denoted by \( O_k \). By a 2D image, what is meant is a 2D image intensity function \( U: \mathbb{R}^2 \rightarrow \mathbb{R} \). \( U \) is assumed to be continuous and to have compact support. The reader should note that continuity is not a restrictive assumption since any real image (in continuous space) can be approximated with arbitrary accuracy by continuous functions.

As in the physical situation an image is formed by the projection of a 3D object on a 2D plane (the camera). Let this projection be denoted by \( P \). Thus in a particular position of the object \( U_a = P(O) \), if the object changes position by a rotation \( T \) then we observe another image \( U_b = P(T(O)) \). This situation is illustrated in fig. 1.

Definition: We say that an image \( U \) is generated by an object \( O \) if \( U = P(T(O)) \) for some 3D rotation and translation \( T \). (i.e. if \( U \) is the projected image of \( O \) in some position).

In the general recognition problem we have a reference set of 3D objects (called a Library) \( L = \{ O_1, \ldots, O_n \} \), and given an observed image we want to know if it is one of these objects, more precisely:
Statement of the general recognition problem: Given an observed image $U_o$, is there an $O_t \in L$ such that $U_o$ is generated by $O_t$.

Of course there could be more than one library object able to generate the image at different viewing angles. This is an inherent ambiguity, due to the many-to-one nature of the projection process.

We start with RPP objects. In section 3.1 the recognition problem is solved for RPP objects using moment invariants and the geometry of projection. The localization problem is also considered. Section 3.2 shows the use of this technique with attributed graph matching to solve the problem for polyhedra and to obtain the recognition algorithm.

3.1. Recognition and Localization of a Rigid Planar Patch (RPP)

In this section only, the word 'object' means an RPP. We first study the effects of the movement of an object in 3-space on its image. As in fig.1, Let $T$ be the transformation in 3D space between the reference and observed objects, and represent it as the most general rigid body motion. Under parallel projection, $T$ will induce a transformation $Q$ in the image plane (see fig. 1). We will use capital letters for 3D space coordinates and small letters to denote image plane coordinates. The most general 3D rigid body motion is a combination of a rotation about an arbitrary axis and a translation [4]. It is represented as:

$$
\begin{bmatrix}
X' \\
Y' \\
Z'
\end{bmatrix} = R \begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} + \begin{bmatrix}
\Delta X \\
\Delta Y \\
\Delta Z
\end{bmatrix},
$$

where
the rotation matrix, is an orthogonal $3 \times 3$ matrix which has three degrees of freedom.

Thus the transformation $T$ has a total of six degrees of freedom.

Fig. 1: Projection on the image plane.

Without loss of generality, consider the camera's frame of reference as the axes for 3-space, and align the image plane with the X,Y axes. Denote the image axes by $x,y$. Then from (1) we have:

\[
x' = r_1x + r_2y + r_3Z + \Delta x \\
y' = r_4x + r_5y + r_6Z + \Delta y,
\]

and we have a representation of $Q$ under parallel projection. However, the $Z$ terms
in eqs.(23) present a problem, because $Q$ must be expressed as a function of the image coordinates only. At this point, we have to make an assumption about the structure of the surface of an object. In other words, we need to represent the $Z$ terms in eq.(23) as $Z = f(x,y)$ (depending on whether the object surfaces are flat, spherical, cylindrical etc.). In this paper, we consider only objects which are, or can be meaningfully approximated as, polyhedra. Under this assumption,

$$Z = ax + by + c,$$

(24)

where $a, b, c$ are known parameters representing a face (RPP) of the library object. It is reasonable to assume that we we know $a, b, c$ because we should have a 3D description of the library object. Substituting (24) into (23) we arrive at:

$$x' = (r_1 + r_2 a) x + (r_2 + r_3 b) y + (r_3 c + \Delta x)$$

$$y' = (r_4 + r_6 a) x + (r_5 + r_6 b) y + (r_6 c + \Delta y).$$

(25)

Eq.(25) is a representation of $Q$ which is a general affine transformation in the plane.

Now for the recognition problem of RPP's. We need to determine, given an observed image $U_o$ whether it is generated by some library object $O_i$, i.e. if there is a position in 3-space for which the projection of $O_i$ is exactly $U_o$. Let us fix a standard non-degenerate position (i.e. the projection is not a straight line) of $O_i$, and call the projection in this position $U_i$. Now clearly $U_o$ is generated by $O_i$ if and only if there exists a linear transformation $Q$ in 2D such that $U_o = Q(U_i)$. Since this last condition exactly means the equivalence of $U_o$ and $U_i$ under linear transformations (in the sense of section 2.2), the recognition problem of RPP's is reduced to the following:
Recognition of RPP's: Given a non-degenerate reference image $U_i$ of $O_i$, then $U_o$ is generated by $O_i$ if and only if $U_i$ and $U_o$ are equivalent modulo linear transformations. In the sense of approximation given at the end of section 2.2, this is true if their feature vectors are within $\varepsilon$ neighborhood of each other.

Classes of equivalent objects under linear transformations are for example: all parallelograms (including squares and rectangles), and all triangles are also equivalent.

We recall that invariance to displacement in eq.(24) is accounted for by using central moments. Since under linear transformations in the plane the center of geometry of a figure is mapped onto the center of geometry of the resulting figure, we only have to consider linear transformations, not affine ones. Figure 2 shows a plot of the first two invariants $I_1$ and $I_2$ for two geometric figures considered as binary images, a parallelogram and a triangle. The values plotted are those of invariants of projections of these figures from many viewing angles. For this paper we use the nearest neighbor rule for classifying objects (the $\varepsilon$ neighborhood notion mentioned earlier). For the rest of the discussion two figures having the same feature vectors means that they belong to the same class.

A note should be made here on the actual calculation of moments. To extract the shape of a 2D figure, we use a binary image of it. For polygons, a better way of computing the moment integrals than by discrete sums is by calculating a line integral around its boundary via Green's theorem. The result is a formula which converts the coordinates of the vertices of a polygon to its moments of any order. This
method and its derivation are presented in the Appendix.

Fig. 2 Plot of $I_1$ vs. $I_2$ for a square and a triangle.
The recognition and localization process is in two parts, first determining the identity of the RPP object using invariants as features, then estimating the attitude and location of a recognized object relative to a library standard. After determining that two RPP objects are actually views of the same object in two different orientations (where one view is the library standard and the other is the observed), we can find the 3D transformation (22) between the two positions.

Consider once again the induced transformation $Q$ written as:

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} = \begin{bmatrix}
  Q_1^1 & Q_1^2 \\
  Q_2^1 & Q_2^2
\end{bmatrix} \begin{bmatrix}
  x \\
  y
\end{bmatrix} + \begin{bmatrix}
  \Delta x \\
  \Delta y
\end{bmatrix},
\]

with

\[
\begin{align*}
Q_1^1 & = r_1 + r_3 a & Q_1^2 & = r_4 + r_6 a \\
Q_2^1 & = r_2 + r_3 b & Q_2^2 & = r_5 + r_6 b \\
\Delta x & = r_3 c + \Delta X & \Delta y & = r_6 c + \Delta Y.
\end{align*}
\]

Assume for the moment that we can find all $Q_i^j$ and $\Delta x, \Delta y$ from the moments of the images. Now from the $Q_i^j$ parameters we need to find the $r_k$'s. This is easily done if we consider how the identification procedure is carried out. $Q$ is the 2D transformation between the observed RPP and an RPP in the library. If we assume the RPP in the library lies in the $(X,Y)$ plane of the object space (which is reasonable since this is a matter of standardizing the library), then in eq.(24), $Z=0$, i.e. $a=0$, $b=0$, $c=0$, and eqs.(27) are reduced to:

\[
\begin{align*}
Q_1^1 & = r_1, & Q_2^1 & = r_2, & \Delta x & = \Delta X, & Q_1^2 & = r_4, \\
Q_2^2 & = r_5, & \Delta y & = \Delta Y.
\end{align*}
\]
Since R is an orthogonal matrix, it is specified by 3 numbers. Four of the \( r_k \)'s have been obtained thus far. The five remaining \( r_k \)'s are obtained from five equations (see e.g. Ganapathy [11]) that express the fact that the sum of the squares of any row or column equals 1. If one desires the directional cosines of the rotation axis and the rotation angle, those are given by the following formulas (from Tsai and Huang [4]):

\[
\sin \theta = \frac{d}{2} , \quad n_1 = \frac{(r_8-r_6)}{d} , \quad n_2 = \frac{(r_3-r_7)}{d} , \quad n_3 = \frac{(r_4-r_2)}{d} \\
\cos \theta = \frac{d^2 r_1-(r_3-r_6)^2}{d^2-(r_8-r_6)^2}, \text{ where } d^2 = (r_8-r_6)^2+(r_3-r_7)^2+(r_4-r_2)^2 ,
\]

where \( n_1, n_2, n_3 \) are the directional cosines of the rotation axes, and \( \theta \) is the angle of rotation about that axis. It should be noted that \( \Delta z \) cannot be determined since it is lost in the projection process. Therefore by localization, we mean finding the X and Y components of the displacement. However, if one assumes perspective projection for image formation, the distance to the object can be calculated from the area of the RPP. This distance estimation is, however, most accurate when the object is close to the camera, which is the case in which parallel projection is not a good approximation.

We now consider the determination of the linear 2D transformation parameters. The translation components \( \Delta X, \Delta Y \) can be determined from the uncentered first order moments:

\[
\Delta X = Q_{13} = \frac{m_{10}}{m_{00}} ; \quad \Delta Y = Q_{23} = \frac{m_{01}}{m_{00}} .
\]

These coordinates can be used to obtain the central moments. The other linear
transformation parameters can be obtained by the method of tensor analysis, as was shown by Cyganski and Orr [6]. We summarize this method here for reference. As we have seen from chapter 2, moments of a particular order are tensors of contravariant order equal to their order as moments. If from the image function one can obtain first order contravariant tensors (i.e. vectors), then one can solve linear equations for the transformation parameters. The first order moment is such a tensor, but if we use central moments they are identically zero. What is done then is to contract higher order moments tensors to form two first order contravariant tensors. The following two tensors (formed by contraction of the 2\(^{nd}\), 3\(^{rd}\) and 4\(^{th}\) order moment tensors) are unit rank tensors:

\[
\begin{align*}
    t^m & = m_{ij} \varepsilon_{ikl} m_{kln} / \mu^4 \\
    v^m & = m_{ij} \varepsilon_{jik} \varepsilon_{kln} m_{nopq} / \mu^7 ,
\end{align*}
\]

where \( m_{ij} \) and \( m_{nopq} \) are the 2\(^{nd}\), 3\(^{rd}\) and 4\(^{th}\) moment tensors respectively, and \( \varepsilon_{ik} \) is the 2-vector defined previously. If \( t^m, v^m \) denote the observed RPP tensors and \( T^m, V^m \) the library RPP respectively, we have a set of four linear equations in the transformation parameters:

\[
\begin{align*}
    t^1 & = Q^1_1 T^1 + Q^1_2 T^2 \\
    t^2 & = Q^2_1 T^1 + Q^2_2 T^2 \\
    v^1 & = Q^1_1 V^1 + Q^1_2 V^2 \\
    v^2 & = Q^2_1 V^1 + Q^2_2 V^2 .
\end{align*}
\]

In this way, the \( Q^j_1 \)'s can be found from the moments of the observed and reference RPP's, and consequently the rigid body motion between the two RPP's is determined as discussed earlier. One note to be made here is that this technique does not work if
the RPP in any view has an axis of reflection symmetry. It can be shown that if the
RPP has an axis of reflection symmetry, the tensors mentioned above are all zero,
thus the equations are indeterminate. This includes such RPPs as all regular polygons,
parallelograms and triangles.

In summary, we have shown that the projection of an RPP transforms in an
affine manner if the RPP undergoes rigid motion in 3D. This fact allows for using
general moment invariants to classify RPP's from their images. We have also shown
how the orientation of the RPP (relative to a library standard) is determined after
recognition.

3.2. Recognition and Localization of Polyhedral Objects by the Subgraph iso-
morphism Algorithm

Based on the preceding considerations, we present a description of a given rigid
object $O$ in terms of the 3D-surface $S(O)$ defining its boundary. In this description,
the surface $S(O)$ is represented by an attributed graph.

Let $W(O)$ denote the "skeleton" or "wireframe" of $S(O)$, consisting of the set of
all edges on $S(O)$ (For illustration, please refer to fig. 3).
Assume also that the surface $S(O)$ consists of a set rigid planar patches (RPP’s) or "faces" $F_1,...,F_n$, the boundary of each face consisting of a contour (polygon) belonging to the wireframe $W(O)$. Our representation of the object $O$ is in terms of an attributed graph $G$ (which wraps around $S(O)$), the nodes $N_i$, $i=1,...,n$, of $G$ representing the faces $F_i$, $i=1,...,n$, and the edges $e_{ij}$, $i,j=1,...,n$ of $G$, the edges between two adjoining faces $F_i$ and $F_j$. With each node of the graph, we associate a feature (attribute) vector consisting of a set of moment invariants of the face that it represents, and
with each edge $e_{ij}$ of the graph a scalar feature (attribute) whose value is the angle between the normals to the corresponding two adjoining faces $F_i$ and $F_j$. Thus the attributed graph $G$ constitutes an invariant description of the 3D object $O$, which is invariant to the object's position and orientation. It must be noted, however, that the part of the surface $S(O)$ which is under view (unoccluded), constitutes a subgraph of $G$ as is clear from Fig. 3.

For the sake of illustration, we present in Fig. 4 a sketch of a more complex object and show the values of three moment invariants of one of the faces in two different orientations.

![Fig. 4 Illustration of a complex object in two different orientations](image-url)
Returning to our discussion of the recognition problem, if the object is viewed from any angle, and under suitable lighting conditions (the lighting conditions are such that no two adjacent faces have the same intensity. With one light source and a lambertian reflection model, this is always true.) we can extract a 2D wire-frame by edge detection. The wire-frame represents a subcollection of the faces of the original 3D object. The connectivity of the visible faces in the wire-frame is the same as that in the object that generated it. The invariants of the faces are also the same, but the angles cannot be readily determined. Thus this description is also invariant under projection (meaning of course, the part of it that is visible).

If we denote by $O$ the description of the object (not the object itself), and consider the projection $P$ as operating on the set of graphs, then it is clear that $P(O)$ is simply some subgraph of $O$ with all the edge attributes removed. This is the central fact that we use here. Let $G_o$ be the graph constructed from the observed wire-frame, and $O_f$ be the graph of a 3D model object. It is clear from the discussion above that: the wire-frame could have been generated from the model if and only if $G_o$ is a subgraph of $O_f$. With this fact, the 3D recognition problem becomes a subgraph isomorphism problem.

The recognition problem for Polyhedra: An image $U_o$ is generated from a library object $O_f$ if and only if $G_o$ is a subgraph of $O_f$.

Let us consider the available information in the graphs. The 3D model graph has all the information possible (edges, attributes, and angles), the wire-frame graph has only the connectivity and the node attributes (invariants). The angles between the
faces of a 2D wire-frame are not available. However, consider that a subgraph isomorphism algorithm applied to this problem yielded that: in the wire-frame, faces $\phi_1$ and $\phi_2$ correspond to faces $F_1$ and $F_2$ in the model, respectively. We can check this assertion by the method of attitude determination of RPP's of the previous section. If the $\phi_i$'s are considered as transformed versions of their respective $F_i$'s, then we can find the 3D transformation between them, i.e. the 3D surface normals of the $\phi_i$'s relative to those of the $F_i$'s. If the correspondence assertion is correct, then the angle between the surface normals of $\phi_1, \phi_2$ and that of $F_1,F_2$ must be equal, if this is not the case, then the original assertion must be false. This technique provides an additional constraint on the subgraph isomorphism algorithm. It is equivalent in some sense to assigning attributes on the edges of the wire-frame, but the important notion is that these attributes are not static, they change depending on the node correspondences.

One way of doing the subgraph isomorphism with the constraints that we have so far, is to use one of the standard algorithms for attributed graphs [8]. The result is a number of possible subgraph isomorphisms, which can then be checked individually with the angles constraint as outlined above. An alternative approach is the following. Since there are many local conditions constraining the matching of every node (invariants, angles, and edges to nodes with the same invariants), then it would be better to do the matching as a local operation, with the condition, of course, that in the end the result be a global matching. Suppose that we hypothesize that node $W_i$ in the wire-frame corresponds to node $N_j$ in the model. If there are nodes $W_{m_1}, \ldots, W_{m_k}$ in the wire-frame are adjacent to $W_j$, then there should be at least k nodes $N_{n_1}, \ldots, O_{n_k}$.
which are adjacent to \( N_j \) and all must match in some order. Suppose for notational convenience that these adjacent nodes are matched in the order \( W_{m_i} \rightarrow N_{n_s} \), this is called a matching configuration. The following constraints are necessary for any isomorphism:

(i) \( W_{m_i} \) must have the same feature vector as \( N_{n_s} \), for all \( s=1,...,k \).

(2) The angle between the faces corresponding to \( W_{m_i} \) and \( W_i \) must equal the angle between \( N_{n_s} \) and \( N_j \) for all \( s=1,...,k \).

(3) If any two \( W_{m_i} \)'s are connected, then the angle between them should equal that between their matching nodes.

**Definition**: A matching configuration which satisfies the above three conditions is called an admissible matching configuration at nodes \( W_i \) and \( N_j \). Clearly, if we have a subgraph isomorphism we have an admissible matching configuration at all corresponding nodes.

We present below, in the language-C-like syntax (in Fig. 5), a recursive algorithm which establishes admissible matching configurations at every node, and by propagating (or growing, thus its name) through the graph it makes sure all such configurations at different nodes are consistent with each other.

The edges that have been checked successfully are marked to prevent the algorithm from running in circles indefinitely. The driver algorithm would arbitrarily pick a node \( W_i \) in the wire-frame; then look for a node \( N_j \) in the model with the same feature vector. If found, these nodes are passed to the function "grow" until a successful
grow($W_i \cdot N_j$)

{
compile all admissible matching configurations of nodes with
unmarked edges around $W_i$ and $N_j$;

For every configuration do {

mark all edges of the configuration;
if ( grow($W_m \cdot N_n$) = SUCCESSFUL for all s )
then return(SUCCESSFUL);
else unmark edges of this configuration;

}

return(FAILED); [if there are no successful configurations]
}

Fig. 5 The recursive part of the subgraph isomorphism algorithm.

return. This is done for all the nodes $N_j$ with the same feature vectors as $W_i$ until a
successful return. If there are no successful returns we conclude that the wire-frame
is not a subgraph of the model.

To determine the attitude and location, we pick a wire-frame face and find its
attitude and location relative to its corresponding model face. As noted in the previ-
ous subsection, there may be more than one set of attitude parameters because of the
nonuniqueness of eq.(29). We reduce the number of possibilities by computing the
attitude of more faces in the wire-frame until a unique common set of parameters is
found.

3.3. Experimental Results

We have run simulations of the algorithm for several synthetic objects one of which is shown here. We also show here an experiment involving a real image from a camera. Fig. 6 shows a 3D object and explains its graph while Fig. 7 shows the particular view of this object used in the simulation.

![Diagram](image.png)

Fig. 6 The representation of an object by a graph.
Fig. 7 Observed wire-frame.

The reader will note that this object is highly symmetrical, thus there is some inherent ambiguity in the matching (i.e. we cannot determine if it is upside down, or if it is rotated by any multiple of a quarter full turn about its major axis). Since this object has only two geometrically distinct types of faces, namely squares and triangles, there are only two possible values of the feature vector of any node. Figs. 8 and 9 show the graphs of the observed object and of the model wire-frames in the position of Fig. 7. The part of the model’s graph enclosed in the solid line is the subgraph matched to the wire-frame using the recursive algorithm.
Fig. 8 The graph of the observed wire-frame.
Fig. 9 The graph of the model with possible matchings.

This correspondence is not unique, and the dotted lines show some (not all) of the other possible matchings, depending on the default starting point of the algorithm. The run time of the algorithm and the number of calls to the function "grow" depends heavily on the labeling of the faces and the default order in which configurations are set up. For example with a particular labeling for the example in Fig. 6 (which has 12 faces), the function "grow" was called once for every face. Since all calls were successful the recursion unwound, with a total of 12 calls to grow. In terms of performance, the execution time is taken up mostly in setting up the graph of the wireframe, calculating the moments and tensors. But as mentioned earlier, the run time of
the graph matching depends on the node labeling. The example just described was run on a VAX-750 in about 2 seconds.

For real images there are extra complications involving the extraction of the wireframe from the images. We have not addressed in this paper the problem of wireframe extraction. Many of the techniques for edge detection can be used for that purpose. We next discuss an example of a real image, other examples are also in [14]. Fig. 10 shows a camera image of a polyhedron called OCTBOX (a cylinder like object with 8 rectangular sides and octagonal top and bottom). The extracted wireframe of the polyhedron is also shown. After Wiener filtering to remove the noise in the image, the graph of the wireframe was extracted and the moments of each face computed. This was matched against a library containing a CUBE, an OCTBALL (the object of Fig. 6) and an OCTBOX. It failed to match both the CUBE and the OCTBALL, but it matched the OCTBOX.
Fig. 10 Camera image of a polyhedron.
Fig. 11 Wireframe of O1 and matching results.

Most of the run time of this example was taken up extracting the wireframe and filtering the noise. For the sake of clarity we explain in detail how the matching was done by an algorithm in one specific instance. The observed object in fig. 10 is named O1, this object matched to the library object OCTBOX as mentioned earlier. Fig. 11 shows the wireframe of O1 and the result of the matching. Face 0 of OCTBOX is the top octagonal face and faces 1 and 2 are two adjacent rectangular faces. The algorithm first picks the face A in O1 and tries to match it to face #0 in OCTBOX, this doesn’t yield a match because the moment invariants of the faces are different (A is a square while 0 is an octagon). Next face A is matched to face 1, then
the growing yields the correspondence $B \rightarrow 2$, which is possible for this view, thus the growing does not backtrack and the C face is matched to the 0 face (both are octagons), and we have a complete match.
CHAPTER 4

Discussion and Conclusions

It is useful to consider the consequences and limitations of this approach. The key point in the description is segmenting an object into individual faces with their relational structure described by a graph. This is done to account for possible occlusion. The two key ideas in the recognition process are: recognizing the "shape" of individual faces, and recognizing the "relational structure" among the faces. The algorithm works for all polyhedra even with occlusion. Because of the use of moment invariants as face attributes, the algorithm is fast for objects with faces of many shapes but somewhat slower for objects with many symmetries. But the experimental results shown here (which were of highly symmetrical objects) has shown that the algorithm converges very quickly. It is difficult to get an expression for the complexity of the algorithm since the expression depends on the structure of the observed and model objects.

We can relax the original condition that the object be a polyhedron. To be able to recognize the relational structure, the object has to have well defined edges suitable for extraction from an image under normal lighting conditions. To recognize the "shape" of a face, the concept of invariance is fundamental. The induced transformation is always linear in the image plane if and only if the boundary curve describing the edge of a face lies on a flat plane in 3D, although the actual surface of the face can be curved. These two observations allow for relaxing the conditions on the
permissible 3D object. Another observation is that if one has a technique for detecting, from the image, which surfaces are flat and which are curved (for example by looking at the grey level changes), then one can delete the curved surfaces from consideration, and the extracted graph would still be a subgraph of its generating object. This, undoubtedly, introduces additional ambiguity, but the algorithm would still work.
APPENDIX A

Calculation of the Moments of a Polygon

In this appendix we present a formula for calculating the moments of a polygon from the coordinates of its vertices. Consider a polygon as a binary image in the plane, with value 1 inside and 0 outside. Thus the moments of a polygon are computed by evaluating the following integral:

\[ m_{pq} = \iint_A x^p y^q f(x,y) \, dxdy , \]  
(A-1)

where the function \( f(x,y) \) is 1 inside the polygon and 0 outside, and the superscripts denote "power" of the variables. The integral can then be rewritten as:

\[ m_{pq} = \iint_A x^p y^q \, dxdy , \]  
(A-2)

where \( A \) is the polygon.

Since a binary polygonal image is completely determined by the coordinates of its vertices, then intuitively, any function of the polygon (such as moments) can be written as a function of the vertices. We find this second function using Green's theorem, by converting the integral of eq.(A-2) to a line integral around the boundary of the polygon.

Green's theorem states that if a function \( g(x,y) \) is integrable over a simply connected set \( A \), and can be decomposed as the sum of the derivatives of two functions, i.e. \( g(x,y) = \frac{\partial N(x,y)}{\partial x} - \frac{\partial M(x,y)}{\partial y} \), then,
\[
\oint_A \left[ M(x,y) \, dx + N(x,y) \, dy \right],
\]

(A-3)

where \( C \) is the closed curve that bounds the simply connected set \( A \). Suppose \( C \) is made up of several curves \( c_1, \ldots, c_m \) with the parametrizations \( c_i: [0,1] \to \mathbb{R}^2 \), with \( c_i(t) = (X_i(t), Y_i(t)) \). Then we can rewrite eq. (A-3) as:

\[
\oint_A f(x,y) \, dx \, dy = \sum_{i=1}^{m} \left[ M(X_i(t), Y_i(t)) \frac{dX}{dt} + N(X_i(t), Y_i(t)) \frac{dy}{dt} \right] dt,
\]

(A-4)

which is the form in which we will use it. The individual curves \( c_i \) in our case are the straight line segments connecting successive vertices. If the vertices of the polygon are denoted by \( x_i, y_i, i = 1, \ldots, n \) in counterclockwise order, then the curves \( c_i = (X_i, Y_i) \) have the following parametrizations:

\[
\begin{align*}
X_i &= t(x_{i+1} - x_i) + x_i, \\
Y_i &= t(y_{i+1} - y_i) + y_i.
\end{align*}
\]

(A-5)

We also have

\[
\frac{dX_i(t)}{dt} = (x_{i+1} - x_i), \quad \frac{dY_i(t)}{dt} = (y_{i+1} - y_i).
\]

(A-6)

In our case the function \( g(x,y) \) is the moment kernel \( x^p y^q \). To find the functions \( N \) and \( M \) we set

\[
\frac{\partial N}{\partial x} - \frac{\partial M}{\partial y} = x^p \, y^q.
\]

(A-7)

So simply let

\[
N = \frac{x^{p+1}}{p+1} \, y^q, \quad M = 0.
\]

(A-8)
If we substitute all of these quantities with the curve parametrizations in eq. (A-4), we get:

$$m_{pq} = \sum_{i=1}^{\infty} \frac{1}{p+1} \int_{0}^{1} [t(x_2-x_1)+x_1]^{p+1} [t(y_2-y_1)+y_1]^{q} (y_2-y_1) \, dt. \tag{A-9}$$

After changing the variables and many lines of algebraic manipulations we get:

$$m_{pq} = \sum_{i=1}^{\infty} \left[ \frac{a_i}{p+1} \sum_{k=0}^{kn} \left\{ \frac{q}{k} \alpha_i^k (y_i-a_i x_i)^{q-k} \frac{x_i^{p+k+2} - x_{i+1}^{p+k+2}}{p+k+2} \right\} \right]. \tag{A-10}$$

where $a_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$ is the slope of the $i^{th}$ line counted counterclockwise. And the quantity inside the square brackets is the definite integral of the quantity in square brackets in eq. (A-4). These quantities are the contribution of each line segment to the total closed line integral. As a matter of notation and for this formula to be written concisely, we denote by $x_{n+1} = x_1$ and $y_{n+1} = y_1$. If the $i^{th}$ line is vertical, and thus its slope $a_i$ is not defined, we have to obtain its term in another way. We do that by going back to the $i^{th}$ term in eq. (A-9) and setting $x_i-1-x_i = 0$. This leads to the $i^{th}$ line contribution to the line integral, denoted by $D_i$, which is:

$$D_i = x_i^{p+1} \left( \frac{y_{i+1}^{p+1} - y_i^{p+1}}{(p+1)(q+1)} \right). \tag{A-11}$$

Thus in summary, the moments of arbitrary indices $pq$ are computed from the vertices coordinates by eq. (A-10). If any of the lines is vertical we replace its corresponding term in eq. (A-10) by eq. (A-11).
Bibliography


